Document 1

Australian Government
Department of Employment and Workplace Relations

Student and Training Activity Reporting System (STARS)

Solution Design

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

Notice:

Where DXC has proposed scopes of services that are additional to the scope of services that were contemplated by the parties in as at the commencement of the contract, DXC has identified this scope in the Solution Design Document (see Appendix H) and the Project Management Plan. This additional scope would only be provided subject to a change variation being agreed by the parties; (1) detailing the additional services; and (2) agreeing reasonable additional charges in respect of the additional services.

The schedule for the program has assumed that the additional services would be agreed and delivered. If agreement is not reached on the additional scope, the schedule would need to be reworked to account for the reduced activities

DXC is preparing an implementation Scope Options Paper which will incorporated into the Project Management Plan.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 2 of 1295

1 Document Management

1.1 Document Control

Version	Change Description	Date
D.0.1	Draft for comment by STARS embedded team. Please note the Technical Architecture section is yet to be completed; the content in this section will reflect the detailed Azure resources to be configured to support the solution services.	
D.0.2	Added Technical Architecture section. Updated Section 8. Included feedback from first round of comments.	01/09/2022
D.0.3	Added clarifications following reviewer feedback.	16/9/2022
V.1.0	First release version for acceptance.	29/09/2022

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This is a living document and will be updated as required.

1.2 Document Author

Role	Name
Lead Solution Architect	s 47F(1) s 47F(1) s 47F(1)

1.3 Document Reviewers

Role	Name
	s 47F(1)
	s 47F(1)
	s 47F(1)

	s 47F(1)
	s 47F(1)
	Microsoft
	s 47F(1)
	s 47F(1)
	NCVER
	Oobe
	s 47F(1)
	SA
	s 47F(1)
	VDS Delivery
	VET Data Design

1.4 Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

1.5 Related Documents

Reference	Document
A	STARS Solution Requirements Document – V.1.0.docx
B	Attachment 1 – Collaboration Diagrams – V.1.0.vsdx
C	Attachment 2 – BPMN Diagrams – V.1.0.vsdx
D	Attachment 3 – API Design – V.1.0.docx Please note that the API Design document is a living document that will be updated as detailed API design and development work is undertaken. The draft version D.0.1 currently contains global design decisions and principles, with several example use cases.
E	STARS Integration Register – V.1.0.xlsx
F	STARS Security Traceability Matrix – V.1.0.xlsx
G	STARS Threat Risk Assessment – V.1.0.docx
H	Attachment 4 – Data Packet ERD – V.1.0.vsdx

Reference	Document
I	Attachment 5 – Technical Architecture – V.1.0.vsdx
J	Attachment 6 – VET-IS ERD.zip
K	Delivery and Support Plan – V.1.0.docx

1.6 Acronyms*

Term/Acronym	Expanded Form
ACSC	Australian Cyber Security Centre
AD	Active Directory
ADF	Azure Data Factory
ADLS	Azure Data Lake (Gen2) Storage
AI	Artificial Intelligence
API	Application Programming Interface
APIM	Application Programming Interface Manager
ASQA	Australian Skills Quality Authority
AVETMISS	Australian Vocational Education and Training Management Information Statistical Standard
BAU	Business as usual
BI	Business Intelligence
CARMS	Classification and Reference Management System
CMS	Contract Master Schedule
DESE	Department of Education, Skills and Employment
DevOps	Development Operations
DEWR	Department of Employment and Workplace Relations
DIU	Data Integration Units
ETL	Extract, Transform and Load
Iaas	Infrastructure as a Service
IAM	Identity and Access Management
ICT	Information and Communication Technology
IRAP	Information Security Registered Assessors Program
ISM	Information Security Manual

Term/Acronym	Expanded Form
ITIL	Information Technology Infrastructure Library
IVETS	Integrated Vocational Education and Training System (NSW Dept of Education)
JSON	JavaScript Object Notation
JWT	JSON Web Token
LAW	Log Analytics Workspace
ML	Machine Learning
NCVER	The National Centre for Vocational Education Research
NCI	National Careers Institute
NESM	New Employment Services Model
NFR	Non-Functional Requirements
NSA	National Skills Agreement
NSC	National Skills Commission
NVA	Network Virtual Appliance
NVETR	National Vocational Education and Training Regulator
OIDC	OpenID Connect
PaaS	Platform as a Service
PIA	Privacy Impact Assessment
PoC	Proof of Concept
PMP	Project Management Plan
PR	Production Environment
PRISM	Provider Registration and International Student Registration System
PSPF	Protective Security Policy Framework
RAAIDD	Risk Actions, Assumptions, Issues, Decisions, and Dependencies
RBAC	Role Based Access Control
REST	Representational State Transfer
RG	Resource Group
RTO	Registered Training Organisation
SaaS	Software as a Service
SAML	Security Assertion Markup Language
SDK	Software Development Kit

Term/Acronym	Expanded Form
SDLC	Software Development Life Cycle
SEE	Skills for Education and Employment
SHIR	Share-Hosted Integration Runtime
SKU	Stock Keeping Unit
SMS	Student Management System
SSO	Single Sign On
SSON	Skills Senior Officials' Network
STA	State Training Authority
TGA	Training.gov.au (National VET directory website)
TLS	Transport Layer Security
TRA	Trades Recognition Australia
TVA	Total VET Activity
UAT	User Acceptance Testing
USI	Unique Student Identifier
VDP	Unique Student Identifier
VDS	VET Data Streamlining
VET	Vocational Education and Training
VIS	VET Information Standard
VM	Virtual Machine
VNet	Virtual Network
VSL	VET Student Loans
VVCT	VSL, VET Compliance and TRA
WCAG	Web Content Accessibility Guidelines

Contents

1 Document Management 3

1.1 Document Control 3

1.2 Document Author 3

1.3 Document Reviewers 3

1.4 Document Approval 4

1.5 Related Documents 4

1.6 Acronyms 5*

2 Introduction 16

2.1 Purpose and Content 16

3 VDS Project 17

3.1 Background and Overview 17

3.2 Project Objectives 17

3.3 Scope and Exclusions 19

3.3.1 Out of Scope 19

4 References 20

5 Solution Architecture and Design Approach 21

5.1 Enterprise Architecture 21

6 STARS Solution Overview 24

6.1 Solution Approach 24

6.2 STARS Capabilities 24

6.3 Assumptions 25

7 Solution Architecture & Design Principles 27

7.1 Overview and Purpose 27

7.2 Out of Scope 27

7.3 Document References 27

7.4 Purpose and Relationship 27

7.5 DEWR Digital Delivery Principles 28

7.6 Platform Design Principles 29

7.6.1 Whole of Australian Government (WoAG) Compliance 29

7.6.2 Design for Flexibility and Future Evolution (FFE) 29

7.6.3 Performance and Efficiency 30

7.6.4 Availability & Resilience 32

7.6.5 Design for Operations 35

7.7 STARS Software Design Principles 38
7.7.1 Streamlined data collection 38
7.7.2 Build connections 38
7.7.3 Common data language 38
7.7.4 Clarify authority 38

7.8 STARS Technical Design Principles 38
7.8.1 Design for Scalability 38
7.8.2 Scalable configuration of Azure resources 39
7.8.3 Availability 40
7.8.4 Logging and Monitoring 42
7.8.5 STARS Networking 43

8 Information Architecture and Design 45

8.1 Overview and Purpose 45

8.2 Information Model – VET Information Standard (R1) 45
8.2.1 VET-IS Conceptual Model 45
8.2.2 VET-IS Logical Schema 46
8.2.3 VET-IS Data Packet Schema 47

8.3 Data Quality Model 48

8.4 Data Governance Approach 50

8.5 Master Data Management Approach 52
8.5.1 Master Data Solution Functionalities 52
8.5.2 STARS MDM Approach 54

8.6 Aggregated Validations (a.k.a. Output Editing) 55

8.7 High Level Conceptual Data Stores 55

8.8 Event Sourcing Model and Event Data Stores 57

8.9 Reporting and Analytics Approach 58
8.9.1 Reporting and Analytics Database 58
8.9.2 Reporting 59
8.9.3 Reporting and Analytics ETL 60

8.10 Data Integration Approach 60

8.11 Data Extract Approach 61

8.12 Highspeed Bulk Data Share Approach 61

8.13 Data Administration 62
8.13.1 System Management 62
8.13.2 Audit 62
8.13.3 Data Management 62

8.14 Physical Design 62
8.14.1 Assumptions 62
8.14.2 Known Issues 63
8.14.3 Data Schemas 63
8.14.4 Data Infrastructure 63
8.14.5 Service Data Stores 64

8.14.6 Master Database 69
8.14.7 Reporting and Analytics Databases 70

9 Business Architecture and Design 71

9.1 Overview and Purpose 71

9.2 Business Context – As Is 71
9.2.1 Context Diagram - As Is – High Level 72
9.2.2 As Is: Funded 73
9.2.3 As Is: Fee-for-service 74

9.3 Business Context – To Be 75
9.3.1 STARS Business Context Diagram 76
9.3.2 System Context and Information Flows – To Be 78

9.4 Submission Pipelines 80
9.4.1 Funded Training Activity Pipeline 81
9.4.2 Fee for Service Training Activity Pipeline 83

9.5 Business Processes – To Be 85
9.5.1 Notes 85
9.5.2 High Level Process – Submission through to Mastering – RTO:STARS:STA Scenario 86
9.5.3 High Level Process – Submission through to Mastering – RTO:STA:STARS Scenario 91

9.6 Workflow and Rules Management Approach 95
9.6.1 Workflows 95
9.6.2 Rules Engine 97

10 Software Architecture and Design 98

10.1 Overview and Purpose 98

10.2 Solution Capabilities 99
10.2.1 STARS Capabilities 99
10.2.2 STARS Capability Services 102

10.3 Solution Services 109
10.3.1 Service Interfaces 109
10.3.2 Services Overview 111
10.3.3 Solution Services Context Diagram 118

10.4 Rules Engine 119
10.4.1 Rules Engines Overview 119
10.4.2 Rules Engine Terminology and Usage 119
10.4.3 Managing the evolution of STARS validation rules 120
10.4.4 Efficiency 121
10.4.5 Auditing & traceability 121
10.4.6 Rules Configuration 121
10.4.7 Errors and Notifications 121
10.4.8 Rules Engine Service 121

10.5 Service Design 122
10.5.1 Core Information Services 122
10.5.2 Submission Pipeline Services 131
10.5.3 Decision Services 145

10.5.4 STA Services 150
10.5.5 Data Consumption Services 151
10.5.6 Support Services 153
10.5.7 Identity and Access Management Services 158

11 Technical Architecture and Design 159

11.1 Solution Services 160
11.1.1 Master Data Service 161
11.1.2 Reference Data Service 162
11.1.3 Provider Service 165
11.1.4 Notification Service 167
11.1.5 Submission Service 170
11.1.6 Staging Service 173
11.1.7 Release Service 176
11.1.8 Mastering Service 179
11.1.9 Reporting and Analytics Provisioning Service 182
11.1.10 Decision Services 186

11.2 Azure Data Estate 188
11.2.1 Azure Data Estate Overview 188
11.2.2 Components Overview 190

11.3 CI/CD Pipeline Design 192

12 Security Architecture and Design 195

12.1 Overview and Purpose 195

12.2 Security Design Principles 195

12.3 Security Boundary 196
12.3.1 System Scope 198
12.3.2 Physical Security 198
12.3.3 Identity Management 198
12.3.4 Network Perimeter 199
12.3.5 Networking 200
12.3.6 PaaS Services and Applications 200
12.3.7 Encryption Key Management 201
12.3.8 Data 201
12.3.9 User Roles and Access to Data 201
12.3.10 Secured Administration and Development 203

12.4 Security Functions 203
12.4.1 Govern 203
12.4.2 Protect 205
12.4.3 Detect 205
12.4.4 Respond 206

13 Software Development Kit (SDK) 207

13.1 Distribution of versioned SDK artefacts 207

14 Software Development Tools and Practices 208

Appendix A. VET-IS Logical Model Entity Relationship Diagrams 209

Appendix B. VET-IS Data Packet Logical Entity Relationship Diagram 212

Appendix C. Stakeholder Overview 213

Appendix D. Glossary 215

Appendix E. Enterprise Data Model (An Example of Data Governance) 217

Appendix F. BPMN Notation 218

Appendix G. Requirements Traceability Matrix 219

Appendix H. Solution Software Components – Changed Scope 260

Software Component Assumptions 260

Metrics and Complexity Ratings 261

Software Component Assumptions Tables 263

Work Order Solution Feature Scope (Original Scope) 263

STARS Solution Features (New Scope) 267

Tables

Table 1: VDS Project Objectives 17
Table 2: Enterprise Architecture Domains and Artifacts 22
Table 3: Document References 27
Table 4: Flexibility & Future Evolution Architecture & Design Principles 30
Table 5: Performance and Efficiency Architecture & Design Principles 30
Table 6: Availability & Resilience Architecture & Design Principles 33
Table 7: Management & DevOps Architecture & Design Principles 36
Table 8: STARS Conceptual Data Stores 55
Table 9: STARS Data Schemas 63
Table 10: STARS Azure Data PaaS Scenarios 63
Table 11: STARS Data Stores 64
Table 12: STARS Capabilities 99
Table 13: STARS Services 102
Table 14: Stakeholder Users of the STARS System 213

Figures

Figure 1: Current state (B:17) 17
Figure 2: Future state (B:18) 17
Figure 3 STARS System and Data Governance 19
Figure 4: Enterprise Architecture Domains and Relations 22
Figure 5: STARS Main Capability Services 25
Figure 6: VDS Technical Principles Hierarchy and Relationship Map 28
Figure 7: DEWR Digital Delivery Principles 29
Figure 8: Azure Monitor 37
Figure 9: STARS Networking (I:1) 44
Figure 10: VET-IS Conceptual Model 46
Figure 11: VET-IS Logical Model 47
Figure 12: VET-IS Data Packet Schema 48
Figure 13: Data classification and personas example 49
Figure 14: Example data personas 50
Figure 15: Data Governance is a People, Process and Technology Approach 50
Figure 16: MDM functionalities 53
Figure 17: Master Data Management Overview 53
Figure 18: Event Sourcing Example 58
Figure 19: VET Collections 72
Figure 20: Business Context Diagram – As Is (B:5) 73
Figure 21: As Is: Funded context (B:6) 74
Figure 22: As-Is: Fee for service context (B:7) 75
Figure 23: STARS Context Diagram – External Actors 77
Figure 24: STARS Context & Information Flows – RTO:STARS:STA Scenario (B:9) 78
Figure 25: STARS Context & Information Flows – RTO:STA:STARS Scenario (B:10) 79

Figure 26: STARS Submission Pipeline – Funded Activity (B:3) 82
Figure 27: STARS Submission Pipeline – Fee for Service Activity (B:3) 84
Figure 28: High Level Process – Submission through to Mastering, Reporting and Analytics – RTO:STARS:STA Scenario (C:3) 86
Figure 29: High Level Process – Submission through to Mastering, Reporting and Analytics – RTO:STA:STARS Scenario (C:4) 91
Figure 30: Simplified Logic App workflow illustrating hand off to Rules Engine 96
Figure 31: STARS High Level System Capabilities & Services (B:11) 108
Figure 32: STARS Solution Services Context Diagram (B:12) 118
Figure 33: Core Information Services (B:12) 122
Figure 34: Master Data Service (B:12) 123
Figure 35: Reference Data Service (B:12) 124
Figure 36: Submission Service (B:12) 126
Figure 37: Provider Service (B:12) 129
Figure 38: Submission Pipeline Services (B:12) 131
Figure 39: Funded Pipeline (B:3) 132
Figure 40: FFS Pipeline (B:3) 132
Figure 41: BPMN Diagrams 132
Figure 42: Staging Service (B:12) 134
Figure 43: Release Service (B:12) 137
Figure 44: Mastering Service (B:12) 140
Figure 45: Reporting and Analytics Provisioning Service(s) (B:12) 143
Figure 46: Decision Services (B:12) 145
Figure 47: Rules Service (B:12) 146
Figure 48: Validation Service (B:12) 147
Figure 49: STA Service (B:12) 150
Figure 50: STA Gateway Service (B:12) 150
Figure 51: Data Consumption Services (B:12) 151
Figure 52: Downstream Consumer Reporting & Analytics Services (B:12) 152
Figure 53: USI Transcript Service 152
Figure 54: STA Bulk Data Integration Service 152
Figure 55: Downstream Consumer ETL Pipeline Services (B:12) 152
Figure 56: Support Services (B:12) 153
Figure 57: Notifications Service (B:12) 153
Figure 58: Event Service (B:12) 155
Figure 59: Azure Event Grid 156
Figure 60: Logging Service (B:12) 156
Figure 61: Azure Monitor 157
Figure 62: Audit Service (B:12) 157
Figure 63: Identity and Access Management Services (B:12) 158
Figure 64 STARS Azure Landing Zone Architecture (I:14) 159
Figure 65: Master Data Service (I:2) 161
Figure 66: Reference Data Service (I:3) 163
Figure 67: Provider Service (I:4) 165
Figure 68: Notification Service (I:5) 168

Figure 69: Submission Service (I:6) 171
Figure 70: Staging Service (I:7) 174
Figure 71: Release Service (I:8) 177
Figure 72: Mastering Service (I:9) 180
Figure 73: Reporting and Analytics Provisioning Service (I:10) 183
Figure 74: Decision Services (I:11) 186
Figure 75: STARS Azure Data Estate (I:12) 189
Figure 76 CI/CD Process (I:16) 193
Figure 77 Build Agent Architecture (I:15) 194
Figure 78: STARS Workload Production Security Boundary Architecture 196
Figure 77: Overview of STARS tenancy and subscriptions 197
Figure 78: VDS Cloud Internet Gateway High Level Architecture 199
Figure 79 Perimeta for Azure Network Overview 200
Figure 80: STARS Security Function Architecture 203
Figure 81: VET Stakeholder Map 213
Figure 82: Data governance example 217
Figure 83: BPMN Notation 218

2 Introduction

2.1 Purpose and Content

This document forms the solution design for the Student and Training Activity Reporting System (STARS), incorporating:

- Project Background and Overview
- Project Objectives
- Approach and Solution Overview
- Solution Architecture and Design Principles
- Information Architecture and Design
- Business Architecture and Design
- Software Architecture and Design
- Technical Architecture and Design
- Security Architecture and Design
- Development Tools and Practices

In addition to documenting the solution design for the solution, this document forms a source document for VDS Project planning, estimations and a baseline for future change requests. DXC notes that changes to solution design documented here will potentially affect any estimations or proposed future work.

The STARS Solution Design Document is the main design artefact for STARS. The Solution Design addresses the following items detailed in the STARS Solution Requirements Document:

- VDS Project Objectives
- Solution Context
- Solution Information Architecture
- Solution Business Architectures
- Solution Capabilities
- Business Requirements
- Functional Requirements
- Non-Functional Requirements

A traceability matrix to Functional Requirements can be found in Appendix G: Requirements Traceability Matrix

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 16 of 1295

3 VDS Project

3.1 Background and Overview

The Vocational Education and Training (VET) ecosystem is complex with nearly 4 million VET students within Australia and offshore, supported by more than 4,000 VET providers delivering a variety of diverse courses from engineering to hospitality and community services.

The importance of timely and accurate data is critical for deciding the future of vocational education within Australia. The current process to collect and report data is experiencing significant lag with delays up to 21 months from the time of training to when reporting is available. This delays decisions, resulting in a VET environment that is not in step with the needs of VET providers and students – both current and prospective.

The VET Data Streamlining ICT System Build and Deployment Project (the project) seeks to uplift the VET sector in Australia through the implementation of a comprehensive, reliable and near real-time data management repository. This repository will serve as a single source of truth to provide VET regulators, governments, VET stakeholders and data consumers with comprehensive information – required to make timely and proactive decisions regarding the education and training requirements of Australians.

The VDS Program will transform current VET data collection, validation, storage and reporting processes by implementing a streamlined data submission pathway to a central repository. The new solution will improve the overall transparency of the VET ecosystem, enhance decision making, support policy development and improve consumer engagement and trust.

The solution being proposed and developed through DXC will be called STARS – Student Training and Activity Reporting System.

Figure 1: Current state (B:17)

Figure 2: Future state (B:18)

3.2 Project Objectives

The VDS Project aims to achieve the following objectives.

1. Improve the VET information standard to meet the needs of the VET sector.
2. Modernise the VET data ICT system to increase its flexibility, responsiveness, access and availability.
3. Achieve a range of benefits, including reduced administration and compliance costs associated with training activity administration, submission and access for training provider, governments and regulators, once the program is implemented.

The VDS Project, through the achievement of the above objectives, seeks to deliver the following outcomes:

Table 1: VDS Project Objectives

No.	Timescale	Outcome
1	Short-term	The VET information standard contains elements that meet user need.
2	Short-term	Users trust the STARS system to deliver timeliness, accuracy, integrity, security, privacy and access.
3	Short-term	Increased ease of reporting leads to greater RTO submission compliance.
4	Medium-term	RTOs face a lower regulatory burden when meeting their reporting obligations.
5	Medium-term	Quality, up-to-date and relevant data leads to improved government decision making and policy outcomes.
6	Medium-term	Quality, up-to-date and relevant data enables consumers to make more informed decision about training options.
7	Medium-term	Students can access up-to-date records, providing greater VET system integrity.
8	Medium-term	Up-to-date and relevant data lead to better targeted compliance activities and corresponding system performance.
9	Medium-term	VET funding allocation based on accurate market information leads to less waste and improved market outcomes.
10	Medium-term	Performance information in the revised VET information standard enables management against KPIs.
11	Long-term	Capacity to compare data between jurisdictions and programs leads to improved policy through competition.
12	Long-term	VET information standard can adapt to meet changing stakeholder needs.
13	Long-term	Greater market competition due to better market information.

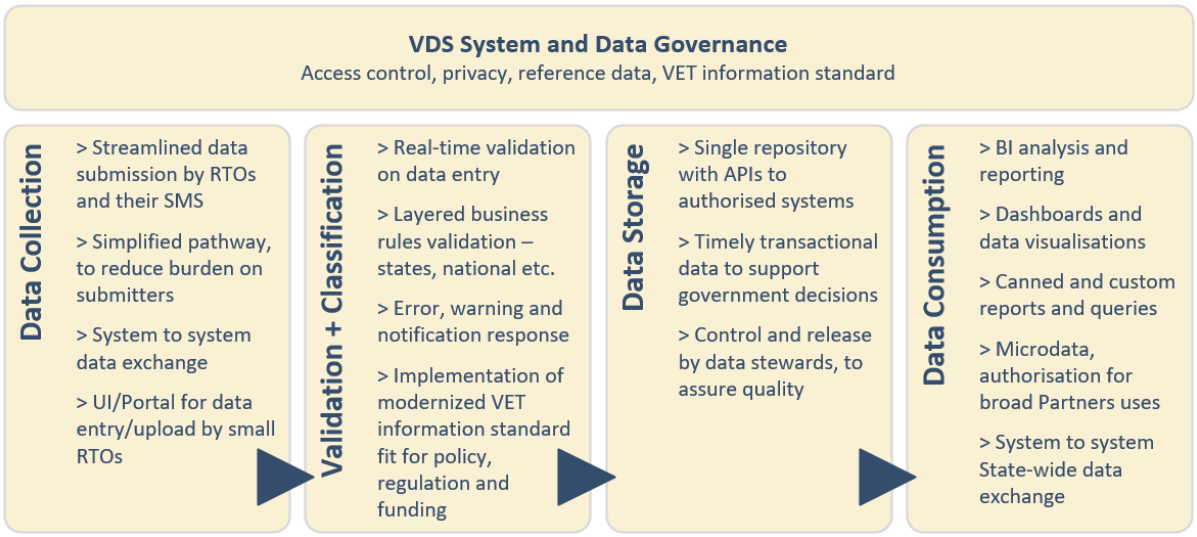


Figure 3 STARS System and Data Governance

3.3 Scope and Exclusions

The STARS solution scope has been identified within the Business Requirements and Solution Requirements as submitted under Work Package 1, documented in the STARS Solution Requirements Document.

This Solution Design covers STARS Release 1 only. At various points in this document, it has been noted that no requirements have yet been provided for certain features. It is expected that in many cases, missing requirements will be developed during the life of the project, which will result in additional design and development work. This will be subject to the agreed project scope change management process.

3.3.1 Out of Scope

The following requirements and activities are out of scope for this solution design:

- Modifications required for partners' systems (SMS providers, state and territory systems and NCVER's systems). These will be conducted by those parties.
- Redevelopment of the current NCVER-managed portals and systems associated with submission, validation and processing of AVETMISS-based data.
- Provision of cloud infrastructure which is to be provided as DEWR material (i.e., shared services outside of the Landing Zone subscriptions).
- Provision of a native mobile app (NB: Progressive Web Application (PWA) technology will instead be used for all public user interfaces)
- Integration of other data collected by NCVER, for example, through the student outcomes survey,
- The survey of employer use and views and the VDS student experience survey.
- Capabilities and functionality not specifically identified within the STARS Solution Requirements Document for Release 1.

4 References

See 1.5 Related Documents

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 20 of 1295

5 Solution Architecture and Design Approach

This section describes the approach taken to developing the Solution Architecture and Design.

Broadly, the approach is to analyse the new Business and Information Models, and Capabilities required for STARS using Enterprise Architecture principles, described below.

The models and capabilities will be designed and delivered through a combination of modern data management services, workflow services, and capability services, following Service Oriented Architecture (SOA) principles[1].

Each of the required capabilities are described in the STARS Solution Requirements Document as a collection of capability services. These capabilities are described in section 6 STARS Solution Overview.

STARS will utilise Service Oriented Architecture (SOA) principles for design and implementation, coupled with Master Data Management (MDM) principles for population of the Master Database. The solution will be hosted in Microsoft Azure and will conform to the Microsoft Well-Architected Framework, utilising recognised Azure Reference Architectures.

Many architectural and design principles have been established for the solution, described in section 7 'Solution Architecture & Design Principles'.

Unless otherwise noted, these principles carry across all the architecture and design sections below. For example, all logging will be implemented following the principles in section 7.8.4 'Logging and Monitoring'; in this instance the detailed service descriptions will not explicitly describe logging as it is already described by the principles.

5.1 Enterprise Architecture

The document has been developed to align with Enterprise Architecture using a framework that incorporates a process of identifying, analysing and documenting requirements (in the STARS Solution Requirements Document) and later (in the Solution Design) producing design artifacts across Business Architecture, Information Architecture, Software Architecture, Technical Architecture and Security Architecture.

The relationships between these architecture elements are as follows:

[1] See https://en.wikipedia.org/wiki/Service-oriented_architecture for a definition of SOA.

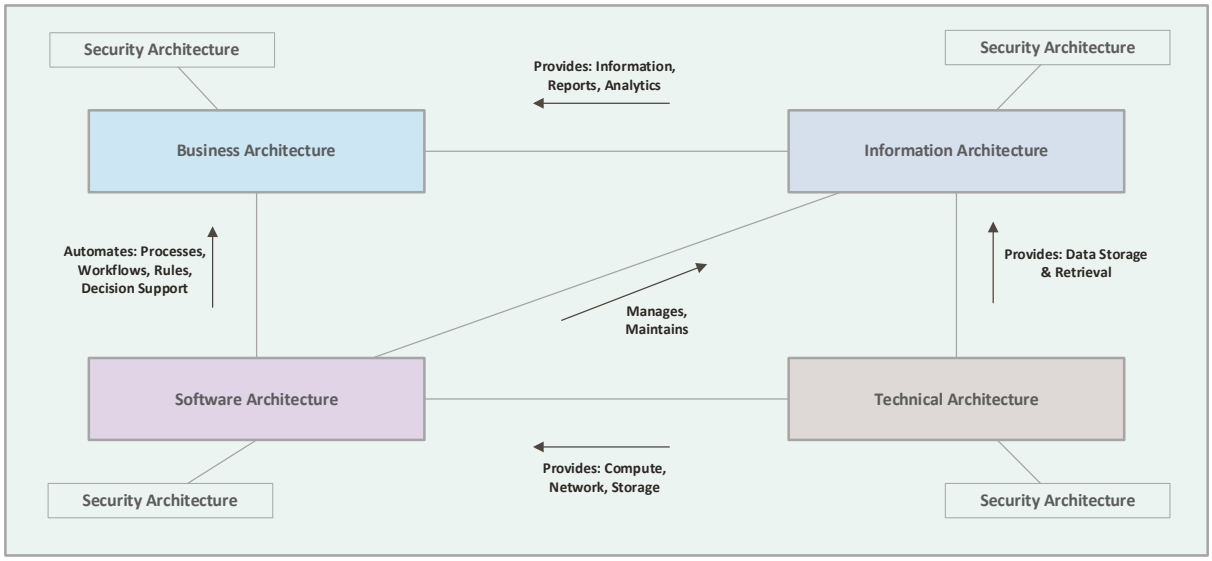


Figure 4: Enterprise Architecture Domains and Relations

Table 2: Enterprise Architecture Domains and Artifacts

Architecture Domain	Artifacts (High Level – NB: examples; not all apply)
Business Architecture	• Context Diagrams • Business Process Descriptions • BPMN Diagrams • Workflow Descriptions • Workflow Activity Diagrams • Business Rules • User Experience (UX) Journey Maps • Data Governance
Information Architecture	• Conceptual Models • Logical Models • Physical Models • Analytics Dimensional Models • Reports • Data Quality Rules • Technical Data Management
Software Architecture	• Reference models • Service Designs • Workflow Designs • User Experience (UX) Designs • User Interface Designs • API Designs
Technical Architecture	Technology Stack Specific, but in the context of STARS being implemented in Azure, the Technical Architecture is broadly based on the Azure Well-Architected Framework and Microsoft recommend Azure Landing Zone configurations.

Architecture Domain	Artifacts (High Level – NB: examples; not all apply)
	The Technical Architecture covers many elements, but includes designs and specifications for items such as (not an exhaustive list): • Azure Landing Zone (adapted from Microsoft best-practices) • Azure AD Tenant Configuration • Management Group and Subscription Configuration • Resource Group Configuration • Secrets Management (e.g., Key Vault Configurations) • VNET Configurations • Network Security Groups • Database Configurations • Storage Account Configurations • Function App Configurations

6 STARS Solution Overview

6.1 Solution Approach

STARS will utilise Service Oriented Architecture (SOA) principles for design and implementation, coupled with Master Data Management (MDM) principles for population of the Master Database. The solution will be hosted in Microsoft Azure and will conform to the Microsoft Well-Architected Framework[2], utilising recognised Azure Reference Architectures[3].

The architecture and design principles are described in detail in section 7 'Solution Architecture & Design Principles'.

Of note is that the entire STARS solution will be implemented within the Microsoft Azure cloud platform and will heavily utilise Platform as a Service (PaaS) offerings. Accordingly, the Solution Design is oriented around utilising Azure features and resources that support the design and implementation of loosely coupled services. These include, for example, having services implemented in separate Resource Groups, using features such as Azure Functions, Azure SQL, and service-specific Azure App configurations.

Loose coupling will be achieved using Message- and Event-Driven approaches for communication between services, using features such as Azure Service Bus and Azure Event Grid. Services that present HTTPS/REST/JSON APIs will have their URLs controlled through Azure Application Gateway. This will enable a consistent URL scheme that also facilitates loose coupling.

6.2 STARS Capabilities

The STARS Solution Requirement Document identifies several key solution capabilities required to satisfy the new STARS Business and Information Models described in this document, such as anytime data submission and the new VET Information Standard, and associated requirements.

STARS will deliver the required capabilities through loosely coupled services, adhering to Service Oriented Architecture (SOA) principles[4]:

[2] https://docs.microsoft.com/en-us/azure/architecture/framework/

[3] https://docs.microsoft.com/en-us/azure/architecture/

[4] See https://en.wikipedia.org/wiki/Service-oriented_architecture for a definition of SOA.

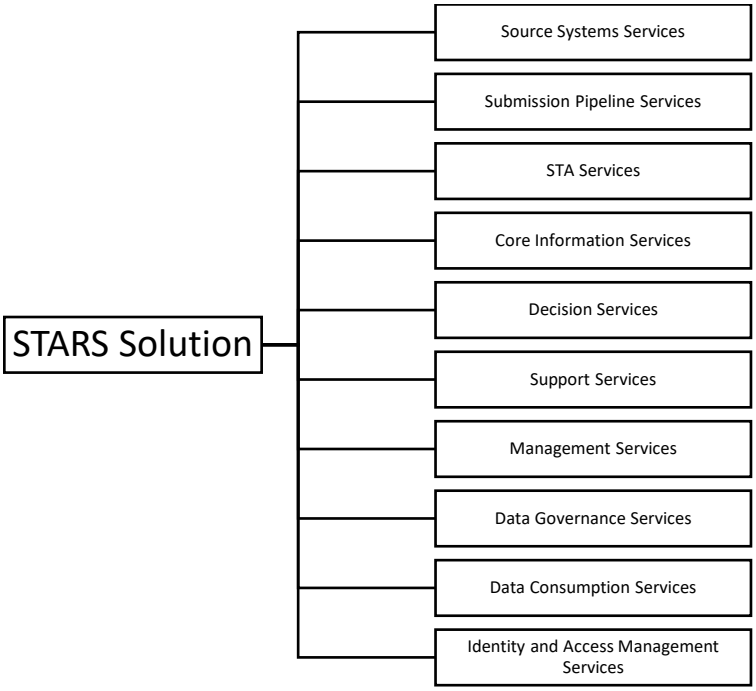


Figure 5: STARS Main Capability Services

Main STARS capability delivery services:

- Source System Services: Facilities for RTOS and STAs to submit VET data to STARS either electronically through modern API technology or through a web-interface
- Submission Pipeline Services: An automated data processing pipeline underpinned by modern workflow, data quality and business rules management technology. The pipeline is responsible for validating data, passing it on to STAs for review and release, storing data in a Master Database and finally for populating downstream reporting and analytics services.
- STA Services: Automated mechanisms for exchanging VET data with STAs.
- Core Information Services: cloud-based PaaS services to manage key VET data sets.
- Decision Services: VET data validation, data quality and business rules management services.
- Support Services: services to support the operation of STARS, such as notification services.
- Management Services: management portals to support and oversee the operation of STARS by system operators.
- Data Governance Services: support of STARS data governance requirements.
- Data Consumption Services: services to provide downstream VET data consumers with data, reports & analytics. These services will also provide data for NCVER systems.
- Identity & Access Management Services: services to support many RTO users and SMSs to gain access to STARS.

6.3 Assumptions

1. STARS will collect data in VET-IS format only.
2. If source data is identified to be invalid, the errors will be rectified via a new version of the submission from the training provider (submitted data will not be modified by anyone other than the original provider, or ASQA on their behalf if the original provider is no longer available).

3. The Commonwealth/National reporting view will not have access to STA-specific data in the master database.
4. STARS will receive a feed of curated reference data from NCVER's CARMS system in Release 1.
5. Any STA reference data required for STA validation rules within STARS will be provided through integrations.
6. Data Governance (see section 8.4): It is assumed that Jurisdictions have people assigned and onboarded – as part of a Data Governance Framework – and those on-boarded in these roles have the technical capability of how to perform the required tasks, through a Data Literacy Model.
7. Data Governance (see section 8.4): Jurisdictions meet on a scheduled basis (fortnightly, monthly) – and the communication lines are open in a horizontal fashion so that cross functional issues can be discussed in an open manner rather than a siloed approach.
8. Metadata Management (see section 8.5.2.2): Data classifications pre-exist and will be made available during Detailed and Analysis and Design consultations
9. Metadata Management (see section 8.5.2.2): It is also assumed that all the required functionalities of Microsoft Purview will be handled through an existing Data Governance Framework at DEWR which has pre-allocated Data Governance Personas (Owner, Steward, Custodian - on-boarded through a data literacy framework) that can ongoingly manage and maintain the data classifications, business glossaries and data catalogues created through Microsoft Purview.

7 Solution Architecture & Design Principles

7.1 Overview and Purpose

The STARS Solution Architecture and Design Principles presents a structured set of technical guidelines and guardrails that will be used to ensure the STARS Solution aligns with the following well-architected solution pillars[5]:

- Reliable
- Secure
- Cost-effective
- Achieves Operational Excellence
- Performance Efficiency

The solution architecture principles captured below are based on proven practices developed from and informed by the Department of Employment and Workplace Relations (DEWR) Digital Delivery Principles, learnings from the VDS ePoC Technical Solution conducted in 2021, as well as Microsoft Azure Application Architecture best-practices.

7.2 Out of Scope

In-depth secure-by-design principles are not covered within this document. Developers should refer to advice and guidelines published by the Australian Cyber Security Centre (ACSC).

7.3 Document References

Table 3 below provides the complete lists of artefacts which have been used to inform the development of the architecture and design principles.

Table 3: Document References

Document	Reference / Link
Microsoft Cloud Adoption Framework	
DESE Digital Delivery Principles	
DESE Architecture Operating Model	
ePoC Technical Solution Architecture	

7.4 Purpose and Relationship

Using this defined set of Azure-based solution architecture principles, it is envisioned they will be used to set the technical boundaries and guidelines for the design and deployment of all technical components associated with the STARS technical solution.

[5] The principles are drawn from the Microsoft Azure Well Architected Framework, a set of guiding tenets and principles for building effective cloud-based solutions. Please see https://docs.microsoft.com/en-us/azure/architecture/framework/ for details.

Figure 6 below illustrates the relationships that this principles document has with other technical artefacts, and how it will be used to both guide and assure the development of future technical artefacts such as the STARS Detailed Design and the STARS Solution Architecture Document.

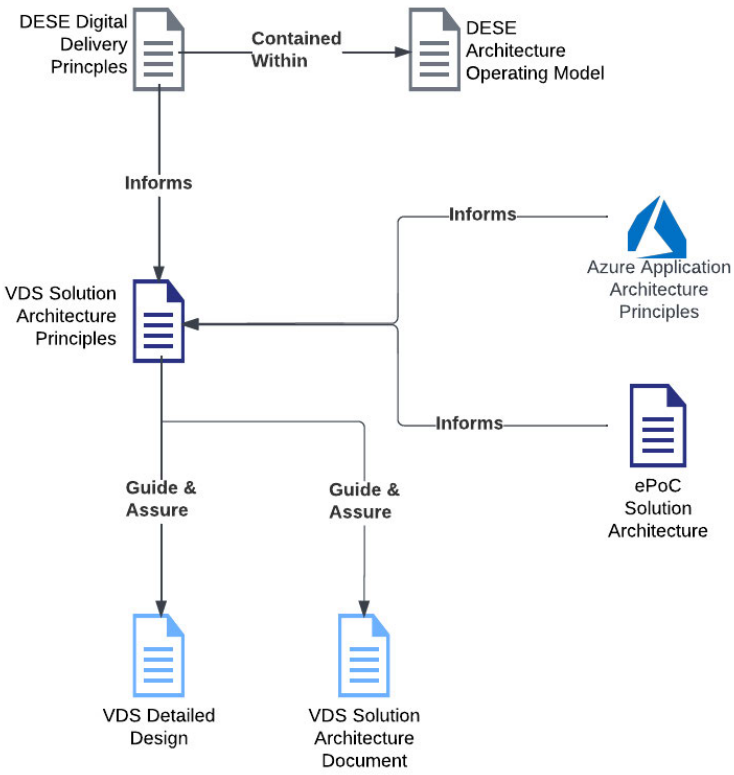


Figure 6: VDS Technical Principles Hierarchy and Relationship Map

7.5 DEWR Digital Delivery Principles

DEWR's Digital Delivery Principles help guide the build of systems in alignment with DEWR's strategic vision, so that they are simple, helpful, trusted, connected, transparent and efficient. In the design and development of the STARS technical solution, these principles must be considered and aligned too where possible.

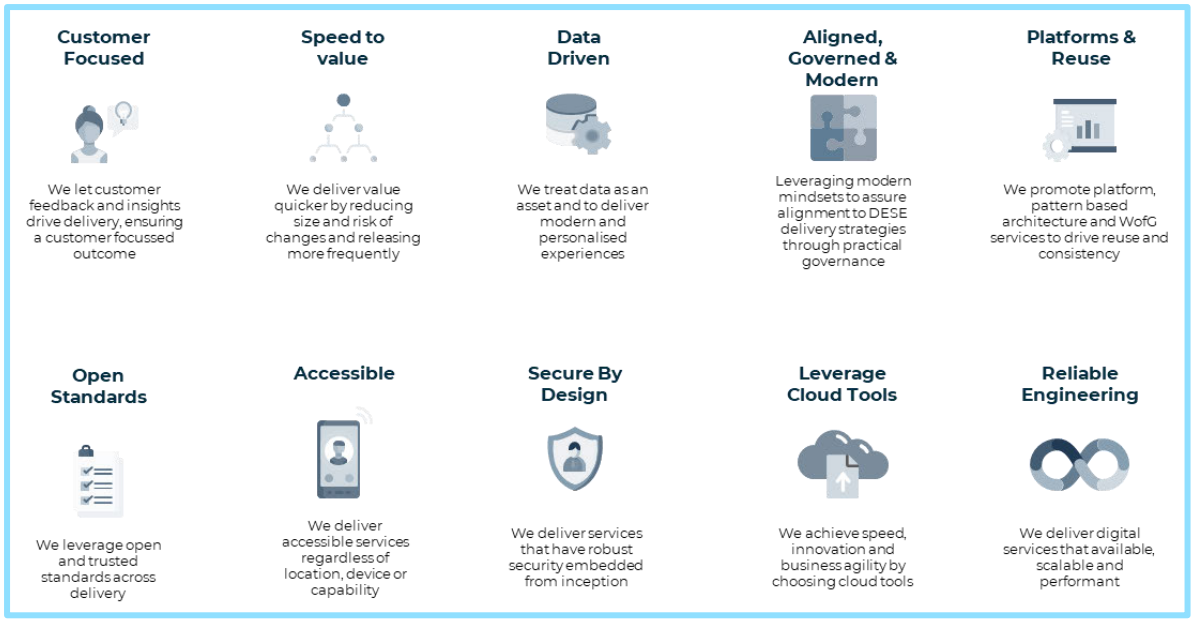


Figure 7: DEWR Digital Delivery Principles

7.6 Platform Design Principles

In addition to the principles outlined within Figure 7 above, all design and development activities must also adhere to the STARS technical architecture principles as described in this section. These principles are categorised and sit within five STARS technical architecture pillars.

- Whole of Australian Government Compliance (WoAG)
- Flexibility and Evolution
- Performance and Efficiency
- Availability & Resilience
- Management & DevOps

The pillars and their associated technical principles are detailed below.

7.6.1 Whole of Australian Government (WoAG) Compliance

The delivered solution must be built to comply with the following regulatory requirements and standards:

- Australian Government Protective Security Policy Framework (PSPF)
- Australian Government Information Security Manual (ISM)
- Australian Privacy Principles
- Open Web Application Security Project guides
- Australian Government Open API Standards
- Web Content Accessibility Guidelines 2.0 (WCAG)
- DTA Digital Service Standard

7.6.2 Design for Flexibility and Future Evolution (FFE)

For the STARS technical solution to be successful it must be able to be flexibly and easily evolved to incorporate changes in business requirements, apply bug fixes, and add new features over time. To achieve excellence in this pillar, designs should look to deploy the following principles.

Table 4: Flexibility & Future Evolution Architecture & Design Principles

Principle #	Principle	Rationale
FFE01	Stateless Design	When a user (e.g., STA or RTO) session must be routed to the same instance to function correctly, the application is said to be sticky, or require session affinity. The microservices deployed within the STARS solution should be implemented as stateless services and should not require session affinity, unless stated and approved otherwise.
FFE02	Workload decomposition	Different services provided by the STARS solution may have different scaling requirements. For instance, while the validation of provider information may be an intermittent operation, the validation of training participation may be heavily used by clients of the system. Separating these business functions into different microservices allows the portions of the system to scale as required.
FFE03	Utilise Asynchronous Messaging	Asynchronous messaging is a way to decouple the message producer (e.g., STA) from the consumer (e.g., STARS). The producer does not depend on the consumer responding to the message or taking any particular action. With a pub/sub architecture, the producer may not even know who is consuming the message. New services can easily consume the messages without any modifications to the producer or existing consumer services.

7.6.3 Performance and Efficiency

The cloud allows for elastic scaling of resources. As load ramps up and down the STARS Solution can be configured to scale accordingly. To take advantage of the scaling capabilities of the Azure cloud, several design principles should be followed.

Table 5: Performance and Efficiency Architecture & Design Principles

Principle #	Principle	Rationale
PE01	Offload resource intensive tasks	Sophisticated data processing requirements may be better handled by offloading into a batch type process utilising separate resources from those providing the client facing API's.
PE02	Design for horizontal auto-scaling	Most resources in the Azure cloud can be auto scaled depending on a variety of factors, including known schedules or performance metrics, such as CPU load. Use of autoscaling will require management and monitoring of associated usage costs. PaaS services should be utilised where appropriate to support automatic operation and management of scaling features.

PE03	Utilise Managed Services (i.e., PaaS and SaaS) where possible	Where possible utilise PaaS services over IaaS.
PE04	Select the best data-store for the job	Pick the storage technology that is best fit for the data and how it will be used, e.g, Blobs for unstructured and semi-structured data, SQL for structured, Cosmos for globally available NoSQL. Data storage accounts should also be appropriately tiered based on the use case i.e., Hot, Cool and Archive. Timelines for data passing from stages of Hot to Cool to Archive have yet to formalised.

The following sections describe specific scalability information for Azure PaaS services in scope.

7.6.3.1 Azure Logic Apps

Azure provides a number of options for deploying Logic Apps that support private network communication and scalability.

An Integration Service Environment can be used with 'Consumption' to provide high-scale workloads, with customer-controlled scalability.

'Standard' Logic Apps can be deployed through single-tenant Azure Logic Apps, where hosting plans are specified to manage scalability. They could also be deployed to an App Service Environment where additional scaling capacity and flexibility is required.

The best option is yet to be decided.

7.6.3.2 Azure Functions

The Azure Functions Premium plan provides appropriate dynamic scaling capabilities with minimal 'warm-up' time etc.

See: Azure Function Scalability

7.6.3.3 Azure SQL Database

Azure SQL Database supports dynamically adding more resources to single database, with elastic pools with minimal downtime. Elastic pools enable us to define maximum resource limit per group of databases in the pool. It supports to easily change resources (CPU power, memory, IO throughput, and storage) allocated to any databases or elastic pools. Azure SQL Database also supports to scale-down the resources when they are not needed to lower the cost.

7.6.3.4 Azure Cosmos DB

Azure Cosmos DB supports auto scale provisioned throughput. Auto scale provisioned throughput in Azure Cosmos DB allows us to scale the throughput (RU/s) of a database or container automatically

and instantly. The throughput is scaled based on the usage, without impacting the availability, latency, throughput, or performance of the workload. Auto scale provisioned throughput is well suited for mission-critical workloads that have variable or unpredictable traffic patterns and require SLAs on high performance and scale.

7.6.3.5 Azure Service Bus

Microsoft provides a number of services to facility scalability in Azure Service Bus. Azure Auto-Scale can monitor the messages in queue (or CPU load) and start or stop instances to maintain targets. Partitioned Queues (and topics) may also be used to increase throughput, if required.

7.6.3.6 Azure Event Grid

Event Grid is a highly scalable, serverless event broker and is designed to handle massively high volumes of events with optimal performance.

7.6.3.7 Azure API Management

In the Production environment the Premium tier service will be used. This allows configuration of multiple units across multiple regions and provides the option for autoscaling through Azure Monitor auto scale.

7.6.3.8 Azure Databricks

Azure Databricks is capable of processing big data on autoscaling infrastructure. With Auto-Scaling, Databricks clusters spin-up and scale for processing massive amount of data when needed and spin down when not in use.

7.6.3.9 Azure Data Lake Gen2

ADLS can store virtually any size of data, and any number of files. There is no limitation on the number of block and data nodes.

7.6.3.10 Azure Data Factory

Azure Data Factory will be using Azure Integration Runtime to copy data from Azure Cosmos db. Azure integration runtime provides a fully managed, serverless compute in Azure. Azure Integration runtime is elastic managed infrastructure to handle data at scale with configurable DIUs (Data Integration Units) per run.

7.6.3.11 Azure Synapse SQL Serverless

Azure Synapse SQL Serverless is pay-per-query, not always-on compute, ideal for logical data warehouse with quick basic discovery and exploration. Being serverless, scaling is done automatically to handle query resource requirements.

7.6.4 Availability & Resilience

A resilient application routes around failure. Identify the critical paths in your application. In the design of the STARS technical solution, there should always be several considerations:

- Is there redundancy throughout the application stack?
- When a component fails does a mechanism exist to fail over to something else?

To ensure the STARS solution is both highly available and resilient to failure, the following technical architecture principle need to be considered:

Table 6: Availability & Resilience Architecture & Design Principles

Principle #	Principle	Description
AR01	Solution should be self-healing across all layers	For integrations to external services, failed operations should be retried, as it is not uncommon for a transient failure to occur when calling an external service, or the external service is running under heavy load. Should the failure persist for an external service, a circuit breaker pattern can be implemented to stop our system from hammering and waiting for responses from a failing service. Fail over can be used when one instance can't be reached, and traffic is routed to an alternate instance.
AR02	Use load-levelling to support failover and distribution of load	Load levelling can be used to queue work asynchronously. It is anticipated that this mechanism would not apply to real time (synchronous) API calls but can be applied should there be a requirement for non-real time (asynchronous) processing of large sets of data.
AR03	Solution should be redundant and highly available across two regions (region-pair)	Based on the business requirements, the STARS solution must be available in the event a region-wide data centre outage occurs in the primary region (Australia East). The application must be designed to perform URL redirects / rewrites so that it can auto-failover to the secondary region (Australia Southeast)
AR05	Graceful degradation	This is the principle that should certain components of the system fail that the system can continue with somewhat reduced functionality.
AR06	Throttling	Some clients may monopolise available throughput and measures can be put in place to throttle the client for a period.

The STARS Azure platform services will be built with High Availability as per below:

7.6.4.1 Azure Cosmos Database

For higher availability, Azure Cosmos DB in STARS will be configured to replicate data across two regions "Australia East" and "Australia Southeast".

Azure Cosmos DB account in STARS will be configured with a single write region in "Australia East" and a read region in "Australia Southeast" with Service-Managed failover enabled. Enabling this

Service-Managed failover option will allow Azure Cosmos DB to fail over to the second region with no user action should a region become unavailable.

7.6.4.2 Azure SQL Database

For business continuity and higher availability of Azure SQL Databases (in Elastic pool), the SQL databases will be configured in the auto-failover group across two regions "Australia East" (Primary region) and "Australia Southeast" (Secondary region).

The Primary database will be configured in "Australia East" and the secondary database in "Australia Southeast" in the auto-failover group, this feature will manage the database replication and automatically recover databases in a secondary region after an unplanned event that results in full or partial loss of the SQL Database availability in the primary region.

7.6.4.3 Azure Data Lake Gen2

Data Lake is not a core element of STARS but is proposed as a storage service for raw data that may in future need to be analysed. If required, DXC recommends the GZRS option:

- Locally redundant storage (LRS) – This is the least expensive replication option where data is replicated three times in the same data centre in the primary region. LRS provides 99.999999999% availability.
- Zone-redundant storage (ZRS) – This is recommended for high availability and performance; with ZRS data is replicated across three Availability zones in the primary region. However, ZRS by itself may not protect your data against a regional disaster where multiple zones are permanently affected.
- Geo-Zone redundant storage (GZRS) – This is recommended for high availability and disaster recovery, this uses ZRS in the primary region and geo-replicates your data to a secondary region.

Other PaaS Azure services availability is 99.99% and this will be sufficient for the STAR solution to provide high availability to the department.

7.6.4.4 Azure Service Bus

Azure Service Bus based messaging will form a key element of the STARS solution.

Azure Service Bus Premium will be configured, at least in production environments within Australia East. DXC recommends availability zones to reduce the risk of service outages at region level. DXC recommends that Service Bus Geo-disaster recovery is appropriately configured to cater for extended region-level failures.

Non-disaster recovery-based approaches to handling major Service Bus outages will be explored during the build stages; these will revolve around, for example, automatic compensating transactions and then retransmission of submissions through the STARS pipeline. An enabler for this is the STARS event stores and correlation IDs; the latter will enable the system to track progress of submissions through the former, allowing for identification of where data needs to be 'restarted/resubmitted.'

7.6.4.5 Azure Event Grid

Azure Event Grid based event processing will form a useful element of the STARS solution, though not as critical as Service Bus.

Azure Event Grid Premium will be configured, at least in production environments within Australia East. DXC recommends availability zones to reduce the risk of service outages at region level. Event Grid region-level outages will not be critical to the operation of STARS[6]. Microsoft does have guidance on replication strategies for Event Grid across regions, but DXC does not at this point believe they are worth implementing.

7.6.4.6 Azure Functions

Azure Functions will be heavily used across STARS. Functions will be configured in both Australia East and Southeast regions in an active/passive formation, with the active functions within Australia East (along with all other 'active' services.) Failover between functions will be automatically enabled through either Azure App Gateway or Azure APIM.

Premium Plans will be configured, at least in production environments.

7.6.4.7 Resilience

The components of the STARS will be designed to detect, respond and alert/log failures in several ways.

For integrations to external services, failed operations will be retried, as it is not uncommon for a transient failure to occur when calling an external service, or the external service is running under heavy load. Should the failure persist for an external service, circuit breaker patterns will be implemented to stop the system from hammering and waiting for responses from a failing service, e.g., constant requests for reference data the TGA web service is down. Communication paths will also be considered when configuring retries to avoid compounded retry paths that could result in exponential volumes of requests to downstream components.

Load levelling can be used to queue work asynchronously. It is anticipated that this mechanism would not apply to real time (synchronous) API calls but can be applied should there be a requirement for non-real time (asynchronous) processing.

Fail over can be used when one instance can't be reached, and traffic is routed to an alternate instance.

Throttling – appropriate rate limits and the like can be applied to public-facing APIs to ensure individual external users do not pose a threat to the health of the overall system.

7.6.5 Design for Operations

Designing for operations is still a critical part of operating the STARS technical solution. Important functions of the operations team (initially DEWR, which will become NCVER's responsibility in the future) include:

- Continuous Integration / Continuous Deployment Processes (CI/CD)
- Monitoring Escalation
- Incident response
- Security auditing

[6] More of a nuisance level… Event Grid will be used to transmit 'informational', non-critical messages around STARS.

Table 7: Management & DevOps Architecture & Design Principles

Principle #	Principle	Description
MD01	Instrumentation across all layers	Instrumentation is the capture of raw data within the system and within the Azure environment can be enabled through a variety of mechanisms depending on the component, including: Application logs, for instance telemetry data pulled into Azure Application Insights Metrics collected by Azure Monitoring Agents (relevant for VMs) Service fabric diagnostics – as per https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-diagnostics-overview Azure AD Logs
MD02	Logging at all levels	All generated logs must be centrally collected and be exposed to approved logging and Security Information and Event management (SIEM) solutions. In principle, the following should be considered in relation to the analysis and management of logging in the STARS solution: Log data across all environments should be appropriately tagged (According to environment and function) and stored within the central Log Analytics Workspace (LAW). Log events at service boundaries with a common correlation ID so that requests can be tracked across the application Use structured logging techniques to automate the consumption and logging of data for instance use of the Azure resources' metrics and Log Analytics. This enables queries to be run against the logs. All logs (both internal and externally generated logs) should be fed to the centralised SIEM for Threat Detection and Analysis. All logs are to be retained for up to 7 years (LAW is currently configured in Preview-Mode to retain logs for up to this period in archive storage)
MD04	Deploy Components Independently	When the STARS Delivery team can deploy a single service independently of other services in the application, updates can happen more quickly and safely. Bug fixes and new features can

		be rolled out at a more regular cadence. The application and release process must be updated to support independent updates. Segregation of Deployment Pipelines and Jobs for key components and effective management of Repos and Branching in Azure DevOps should be used to provide this level of component independence and separation.

7.6.5.1 STARS Logging and Monitoring

The following section describes how these principles will be enacted through in the STARS Azure PaaS platform:

- STARS will utilise standard Azure Monitor-based functionality throughout
- Each Azure service will be configured to log events at an appropriate level.
- Application code will be configured to use Application Insights and log all system and audit events using structured logging[7].
- Azure Log Analytics will be utilised to store log data from within Azure Monitor and be used for post-hoc analysis and reporting of log and audit events.

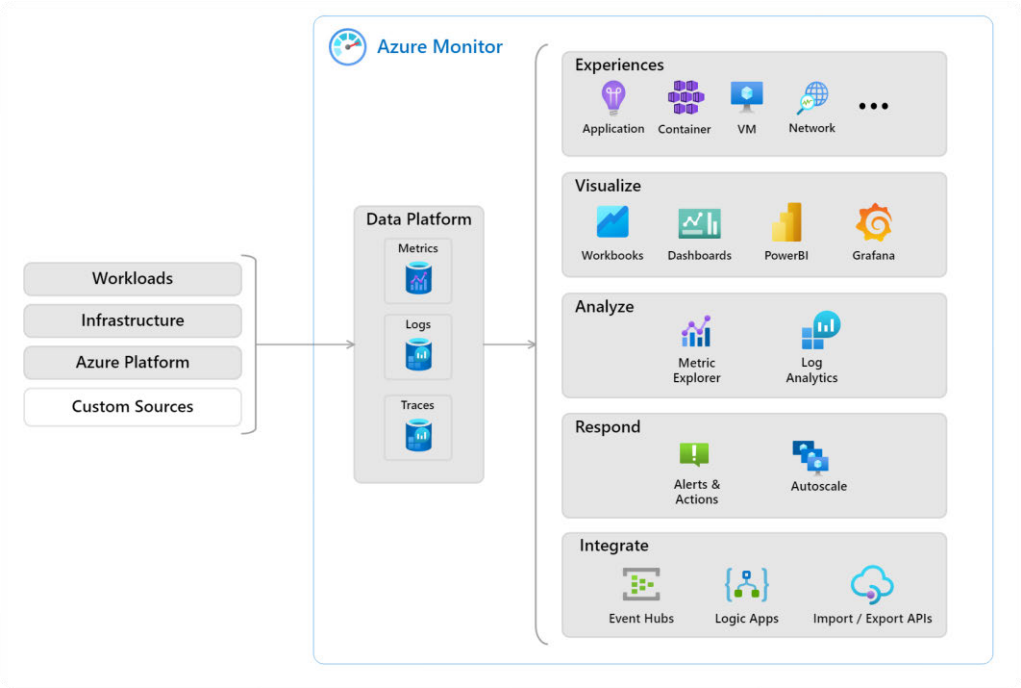


Figure 8: Azure Monitor

[7] See https://www.bounteous.com/insights/2021/05/04/structured-logging-microsofts-azure-application-insights/ and https://docs.microsoft.com/en-us/azure/architecture/example-scenario/logging/unified-logging

7.6.5.2 STARS Networking

STARS will rely on the oobe-designed and configured Azure Landing Zone-based infrastructure. DXC will work jointly with oobe and DEWR cloud architects to design a fit-for-purpose networking infrastructure that facilitates both public and private access to STARS by RTOs, NCVER, STAs and other stakeholders.

Of note are the likely need for private endpoints to allow NCVER direct access to, for example, Azure SQL or Cosmos DB. Public access, e.g., via browser apps, to the STARS APIs will likely be through a combination of Azure Application Gateway and Azure APIM – configured to allow for dual-region failover.

The following diagram shows, at a high level, some of the considerations and Azure alternatives for public/private pathways into STARS services.

7.7 STARS Software Design Principles

The following common design principles have been identified at the program level, with the goal that programs and systems operate as a complementary ecosystem that improves end-user experience and leads to better policy/program decisions.

7.7.1 Streamlined data collection

Reduce the number of data processes and systems that VET operators interact with to reduce reporting burden.

7.7.2 Build connections

Invest in interoperability between systems so that (with permission) data can be shared rather than re-collected.

7.7.3 Common data language

Simplify terms and move to alignment of data models and definitions across the VET sector and with schools and higher education where possible.

7.7.4 Clarify authority

Update data use frameworks to support integration and sharing of data, whilst ensuring a best practice approach to data security and the use of personal information.

7.8 STARS Technical Design Principles

Scalability will be designed into all aspects of the STARS. This includes making the most of Azure platform features for each of the system components, together with thoughtful design and architecture to maximise system scalability.

7.8.1 Design for Scalability

The following concepts and approaches will ensure the design of the solution is flexible and scalable.

Avoid single point of failure – deployment configurations will ensure each component of the solution has redundancy build in.

Scale horizontally, not vertically – Each request is able to spin up a new instance of our Function Apps, meaning processing limits of each instance will not be reached.

Push work as far away from the core as possible – using an Angular front-end provides rich client capabilities that allow certain workloads to be applied in the client's browser, rather than through a central processing capability.

API first – A key part of the solution design is the ability to provide a common set of services that will facilitate multiple processing interfaces. These common services, deployed through scalable Azure resources, will ensure the same experience is able to be achieved through any of the channels provided.

Provide as fresh as needed data – certain data within the system will be updated infrequently. The STARS will be designed to allow caching of data that matches its refresh period.

Strive for statelessness – State information will be kept in as few places as possible, and within components that are made for it (i.e., distributed data stores etc). This approach ensures maximum flexibility for scaling the solution and ensures a consistent experience across multiple interface channels.

7.8.2 Scalable configuration of Azure resources

The STARS solution will utilise Azure PaaS services extensively. These services all have elastic-scaling and high-availability configurations, described below:

7.8.2.1 Serverless code execution: Azure Functions

The Azure Functions Premium plan provides appropriate dynamic scaling capabilities with minimal 'warm-up' time etc.

See: Azure Function Scalability

7.8.2.2 Relational data: Azure SQL Database

Azure SQL Database supports dynamically adding more resources to single database, with elastic pools with minimal downtime. Elastic pools enable us to define maximum resource limit per group of databases in the pool. It supports to easily change resources (CPU power, memory, IO throughput, and storage) allocated to any databases or elastic pools. Azure SQL Database also supports to scale-down the resources when they are not needed to lower the cost.

7.8.2.3 Non-relational data: Azure Cosmos DB

Azure Cosmos DB supports auto scale provisioned throughput. Auto scale provisioned throughput in Azure Cosmos DB allows us to scale the throughput (RU/s) of a database or container automatically and instantly. The throughput is scaled based on the usage, without impacting the availability, latency, throughput, or performance of the workload. Auto scale provisioned throughput is well suited for mission-critical workloads that have variable or unpredictable traffic patterns and require SLAs on high performance and scale.

7.8.2.4 Azure API Management

In the Production environment the Premium tier service will be used. This allows configuration of multiple units across multiple regions and provides the option for autoscaling through Azure Monitor auto scale.

7.8.2.5 Azure Databricks

Azure Databricks is capable of processing big data on autoscaling infrastructure. With Auto-Scaling, Databricks clusters spin-up and scale for processing massive amount of data when needed and spin down when not in use.

7.8.2.6 Azure Data Lake Gen2

ADLS can store virtually any size of data, and any number of files. There is no limitation on the number of block and data nodes.

7.8.2.7 Azure Data Factory

Azure Data Factory will be using Azure Integration Runtime to copy data from Azure Cosmos db. Azure integration runtime provides a fully managed, serverless compute in Azure. Azure Integration runtime is elastic managed infrastructure to handle data at scale with configurable DIUs (Data Integration Units) per run.

7.8.2.8 Azure Synapse SQL Serverless

Azure Synapse SQL Serverless is pay-per-query, not always-on compute, ideal for logical data warehouse with quick basic discovery and exploration. Being serverless, scaling is done automatically to handle query resource requirements.

7.8.3 Availability

The Azure Cloud environment allows for redundancy within the system. Each of the Azure resources included in the STARS will be deployed with appropriate configurations to ensure high availability where required.

During all build stages, the underlying Azure infrastructure will be configured and tested repeatedly in various failure modes to ensure that HA is achieved.

The STARS Azure platform will be built with High Availability as per below:

7.8.3.1 Azure Cosmos Database

For higher availability, Azure Cosmos DB in STARS will be configured to replicate data across two regions "Australia East" and "Australia Southeast".

Azure Cosmos DB account in STARS will be configured with a single write region in "Australia East" and a read region in "Australia Southeast" with Service-Managed failover enabled. Enabling this Service-Managed failover option will allow Azure Cosmos DB to fail over to the second region with no user action should a region become unavailable.

7.8.3.2 Azure SQL Database

For business continuity and higher availability of Azure SQL Databases (in Elastic pool), the SQL databases will be configured in the auto-failover group across two regions "Australia East" (Primary region) and "Australia Southeast" (Secondary region).

The Primary database will be configured in "Australia East" and the secondary database in "Australia Southeast" in the auto-failover group, this feature will manage the database replication and automatically recover databases in a secondary region after an unplanned event that results in full or partial loss of the SQL Database availability in the primary region.

7.8.3.3 Azure Data Lake Gen2

Data Lake is not a core element of STARS but is proposed as a storage service for raw data that may in future need to be analysed. If required, DXC recommends the GZRS option:

- Locally redundant storage (LRS) – This is the least expensive replication option where data is replicated three times in the same data centre in the primary region. LRS provides 99.999999999% availability.
- Zone-redundant storage (ZRS) – This is recommended for high availability and performance; with ZRS data is replicated across three Availability zones in the primary region. However, ZRS by itself may not protect your data against a regional disaster where multiple zones are permanently affected.
- Geo-Zone redundant storage (GZRS) – This is recommended for high availability and disaster recovery, this uses ZRS in the primary region and geo-replicates your data to a secondary region.

Other PaaS Azure services availability is 99.99% and this will be sufficient for the STAR solution to provide high availability to the department.

7.8.3.4 Azure Service Bus

Azure Service Bus based messaging will form a key element of the STARS solution.

Azure Service Bus Premium will be configured, at least in production environments within Australia East. DXC recommends availability zones to reduce the risk of service outages at region level. DXC recommends that Service Bus Geo-disaster recovery is appropriately configured to cater for extended region-level failures.

Non-disaster recovery-based approaches to handling major Service Bus outages will be explored during the build stages; these will revolve around, for example, automatic compensating transactions and then retransmission of submissions through the STARS pipeline. An enabler for this is the STARS event stores and correlation IDs; the latter will enable the system to track progress of submissions through the former, allowing for identification of where data needs to be 'restarted/resubmitted.'

7.8.3.5 Azure Event Grid

Azure Event Grid based event processing will form a useful element of the STARS solution, though not as critical as Service Bus.

Azure Event Grid Premium will be configured, at least in production environments within Australia East. DXC recommends availability zones to reduce the risk of service outages at region level. Event

Grid region-level outages will not be critical to the operation of STARS[8]. Microsoft does have guidance on replication strategies for Event Grid across regions, but DXC does not at this point believe they are worth implementing.

7.8.3.6 Azure Functions

Azure Functions will be heavily used across STARS. Functions will be configured in both Australia East and Southeast regions in an active/passive formation, with the active functions within Australia East (along with all other 'active' services.) Failover between functions will be automatically enabled through either Azure App Gateway or Azure APIM.

Premium Plans will be configured, at least in production environments.

7.8.3.7 Resilience

The components of the STARS will be designed to detect, respond and alert/log failures in several ways.

For integrations to external services, failed operations will be retried, as it is not uncommon for a transient failure to occur when calling an external service, or the external service is running under heavy load. Should the failure persist for an external service, circuit breaker patterns will be implemented to stop the system from hammering and waiting for responses from a failing service, e.g., constant requests for reference data the TGA web service is down. Communication paths will also be considered when configuring retries to avoid compounded retry paths that could result in exponential volumes of requests to downstream components.

Load levelling can be used to queue work asynchronously. It is anticipated that this mechanism would not apply to real time (synchronous) API calls but can be applied should there be a requirement for non-real time (asynchronous) processing.

Fail over can be used when one instance can't be reached, and traffic is routed to an alternate instance.

Throttling – appropriate rate limits and the like can be applied to public-facing APIs to ensure individual external users do not pose a threat to the health of the overall system.

7.8.4 Logging and Monitoring

- STARS will utilise standard Azure Monitor-based functionality throughout:
- Each Azure service will be configured to log events at an appropriate level.
- Application code will be configured to use Application Insights and log all system and audit events using structured logging[9].
- Azure Log Analytics will be utilised to store log data from within Azure Monitor and be used for post-hoc analysis and reporting of log and audit events.

[8] More of a nuisance level… Event Grid will be used to transmit 'informational', non-critical messages around STARS.

[9] See https://www.bounteous.com/insights/2021/05/04/structured-logging-microsofts-azure-application-insights/ and https://docs.microsoft.com/en-us/azure/architecture/example-scenario/logging/unified-logging

- All logs will be ingested via the shared Log Analytics Workspace in Australia East and further analysed via the SIEM capability (Microsoft Sentinel).

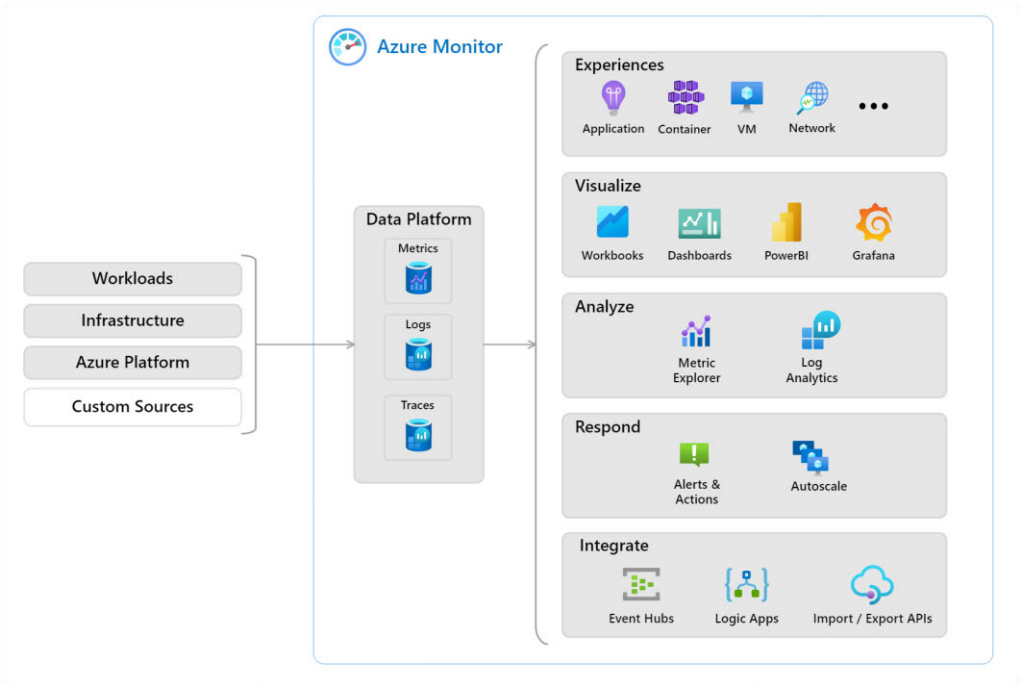


Figure 9: Azure Monitor

7.8.5 STARS Networking

STARS will rely on the oobe-designed and configured Azure Landing Zone-based infrastructure. DXC will work jointly with oobe and DEWR cloud architects to design a fit-for-purpose networking infrastructure that facilitates both public and private access to STARS by RTOs, NCVER, STAs and other stakeholders.

Of note are the likely need for private endpoints to allow NCVER direct access to, for example, Azure SQL or Cosmos DB. Public access, e.g., via browser apps, to the STARS APIs will likely be through a combination of Azure Application Gateway and Azure APIM – configured to allow for dual-region failover.

The following diagram shows, at a high level, some of the considerations and Azure alternatives for public/private pathways into STARS services, including Azure ExpressRoute or site-to-site VPN for secure communication with NCVER on-premises systems.

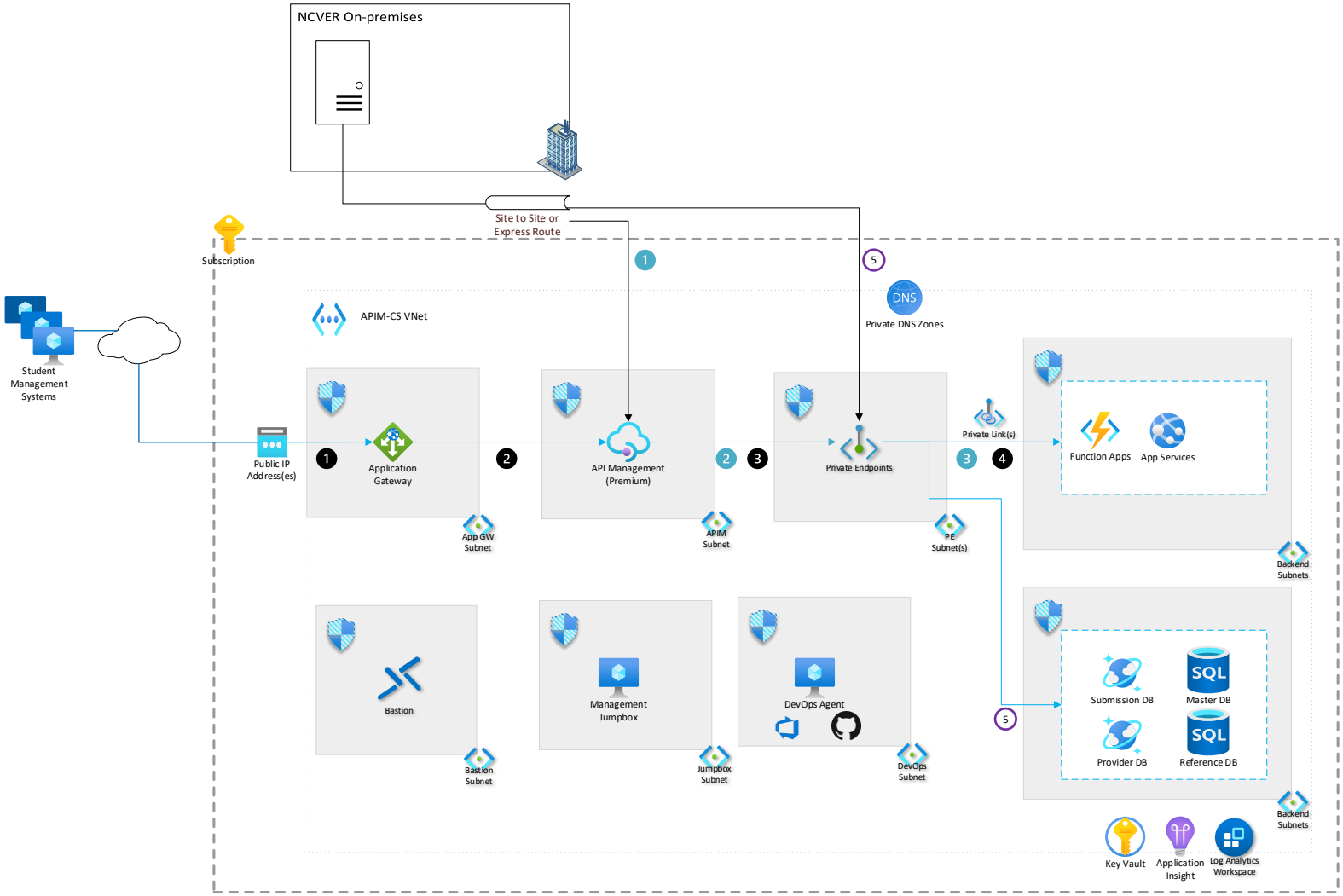


Figure 10: STARS Networking (I:1)

8 Information Architecture and Design

Please note that many detailed requirements are not yet provided surrounding Data Governance, MDM, external data exchange, reporting and analytics. Accordingly, the solution design proposes a set of flexible approaches and Azure-based technologies that are most likely able to support detailed requirements as they are created or evolved during the life of the project (subject to the project scope change management process), and beyond.

8.1 Overview and Purpose

This Information Architecture and Design section addresses the information management capabilities and requirements for the STARS solution.

It describes the following:

- The underlying Information Model – the VET Information Standard (R1)
- The associated logical schemas
- A data quality model that will be used to describe the roles and processes associated with various data stores
- The approach to Data Governance
- The approach to Master Data Management (MDM)
- The approach to Reporting and Analytics
- The approach to data exchange between STARS and external systems
- The Physical Design of key data stores and associated data infrastructure

8.2 Information Model – VET Information Standard (R1)

The new STARS system will utilise the new VET Information Standard (VET-IS), which replaces the outdated AVETMISS information standard with a modern fit-for-purpose and backwards-compatible information architecture that better fits the reporting and operational requirements for VET data.

Note that Release 1 of the VET-IS is targeted to the information required for the VET Provider Collection. As such it does not contain information about employers.

The new VET-IS can be considered through several perspectives:

8.2.1 VET-IS Conceptual Model

The VET-IS aims to achieve a higher level of flexibility and extensibility than AVETMISS. It does this by abstracting people, organisations, contracts, enrolments, subjects, etc., to a set of higher-level concepts:

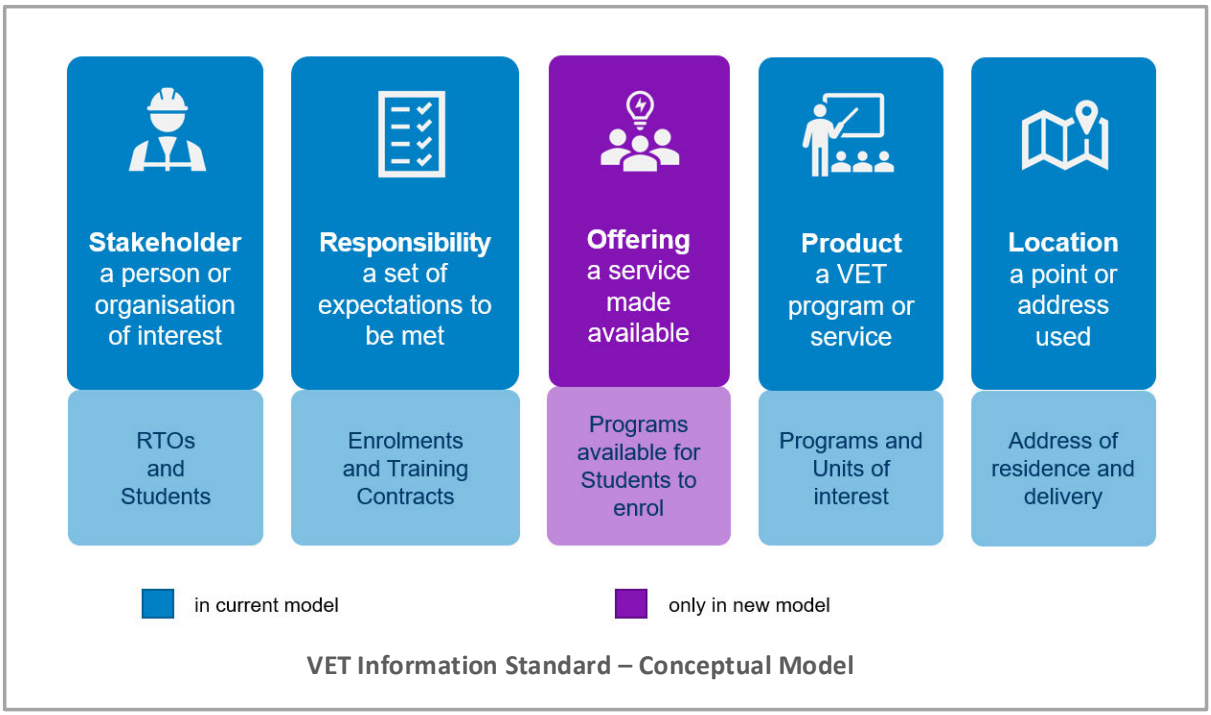


Figure 11: VET-IS Conceptual Model

- Stakeholder: representing people, organisations. NB: employers will be covered in future releases.
- Location: representing addresses, virtual spaces.
- Product: representing programs, subjects.
- Offering: representing a product offered by a training organisation.
- Responsibility: representing many things, including enrolments, training contracts, financial arrangements. Responsibilities are between Stakeholders and address obligations between them. For example, an enrolment is a responsibility between a training organisation and student.
- Event (representing a business event – currently not in R1)

The logical model (below) and physical model (described in the Solution Design) give effect and materialise these concepts within the STARS Master Database.

8.2.2 VET-IS Logical Schema

The intention of the VET-IS Logical Model is to flexibly represent the many types of conceptual entities in the VET sector, without having to model them in (very many) separate tables.

The VET-IS Logical Schema Release 1 is a normalised relational schema that models the following primary logic entities and their relationships:

- Stakeholder (representing people, organisations. NB: employers will be covered in future releases)
- Location (representing addresses, virtual spaces)
- Product (representing programs, subjects)
- Offering (representing a product offered by a training organisation)
- Responsibility (representing many things, including enrolments, training contracts, financial arrangements)
- Event (representing a business event, but not lifecycle events – currently not in R1)

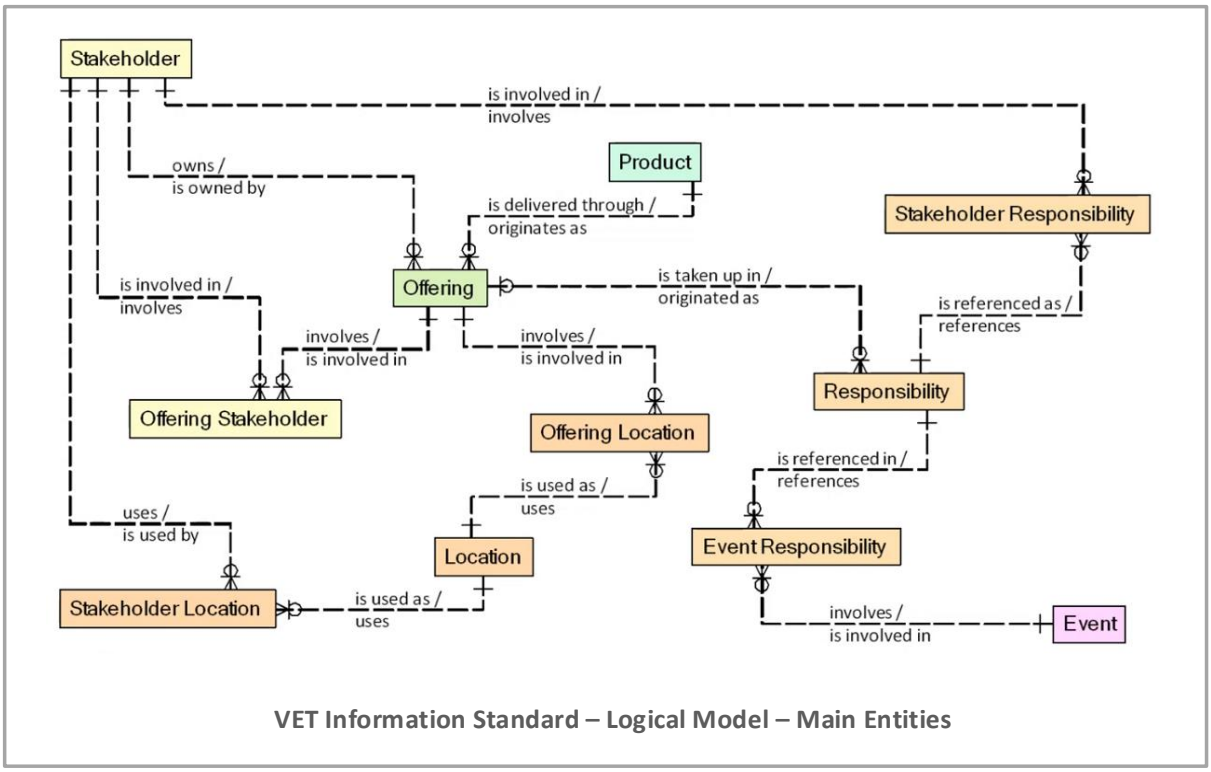


Figure 12: VET-IS Logical Model

Figure 12: VET-IS Logical Model presents a high-level view of the main entities in the logical model. Detailed logical schemas are presented in Appendix A: 'VET-IS Logical Model Entity Relationship Diagrams'.

8.2.3 VET-IS Data Packet Schema

The VET-IS Data Packet Schema forms both a logical information model and the basis for implementing data interchange protocols between STARS and external systems, including Student Management Systems (SMS). The Data Packet Model informs system-to-system data presentation formats and, for example, Application Programming Interfaces (API).

The model is a de-normalised projection of the VET-IS logical schema into 'Packets' of data elements that represent the common primary conceptual entities found in the VET sector, e.g., Students & Enrolments. Data Packets primary use, within the STARS context, are as Data Transfer Objects (DTO) between external systems (e.g., RTO SMS, STA systems) and STARS.

In this sense, the Data Packet Schema is a 'projection' of the Logical Schema into commonly recognised topics. For example, a Student data packet is represented as a Stakeholder within the Logical Schema, as is a Training Provider. An Enrolment data packet is represented as a Responsibility.

Data Packets will also likely be the 'public face' of the VET-IS in terms of describing, reporting and exchanging data with external stakeholders, such as RTOs, SMS providers and STAs.

The Data Packet types currently defined in VET-IS R1 are:

- Government Packet
- Training Organisation Packet

- Training Organisation Location Packet
- Program Packet
- Subject Packet
- Offering Packet
- Delivery Location Packet
- Student Packet
- Student Location Packet
- Program Enrolment Packet
- Subject Enrolment Packet

VET-IS Release 1 is targeted to specifying and shaping the information and data required for the VET Provider Collection and thus there is not currently support for the VET Apprentice and Trainee collection (including Employers).

The following diagram shows the high-level relationships between the Data Packet types. The detailed information model for the packets (e.g., data attributes & rules) as well as the actual data presentation format(s) and interchange protocols (e.g., JSON data and associated JSON schemas, web service APIs.) will be developed during Work Package 2.

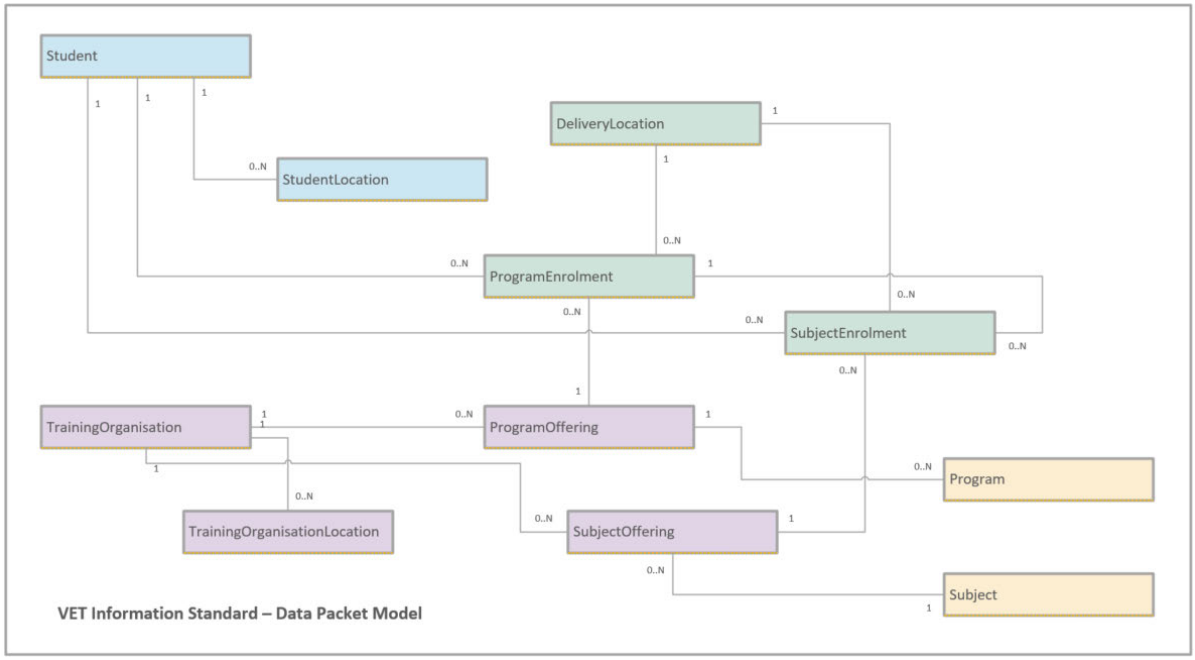


Figure 13: VET-IS Data Packet Schema

By way of terminology, VET-IS Data Packages are a collection of VET-IS Data Packets. Packages only contain packets that relate to each other and contain a complete set of information. E.g., a package containing an Enrolment packet must also contain the details of the student enrolled in a Student packet, along with Delivery Location, Subject, Offering and Training Organisation packets.

8.3 Data Quality Model

A focus of the STARS solution design is to build in many data quality control mechanisms, including for example, data quality rules, data quality management workflows, and the application of MDM processes.

The STARS Solution Design will reflect the following broad categorisations for data quality, as applied to information received, processed, and used throughout the system:

Bronze: the raw unvalidated VET data received by STARS from RTOs and STAs. The raw data is ingested in JSON or a defined bulk file format and stored in non-relational data storage. Bronze data will be to a limited extent queryable and reportable but is not considered a suitable for STA operational or national reporting needs as it is optimised for writes, not reads.

Silver: derived from VET bronze level data that has undergone validation using data quality and business rules but is still optimised for writes rather than reads. Silver data is considered suitable for STA operational needs (e.g., to support RTO payment cycles) and STA reporting. It is not considered suitable for national reporting needs. De-duplication and some data cleansing may occur. Flags, aggregates and other derived items are added as needed to support operational reporting (yet to be determined by STA, regulators and NCVER).

Gold: derived from VET silver level data that has further undergone MDM processes to, for example de-duplicate and merge data. It forms a reliable and quality source-of-truth at national level. Gold data is also stored in high-speed relational databases, but only in the VET-IS Logical Schema. Gold level student data includes enrolment information merged from multiple STAs, for example. Gold data is suitable for both national reporting needs and as a source for other external systems, including transcripts for USI. Gold data is to be stored in read-optimised data stores to facilitate reporting.

NCVER utilise other systems which include perturbation, and statistical inferences to estimate residential population (ERP) for downstream publications and outputs. 'Gold' data needs be tied to governance mechanisms throughout transition and treated as 'preliminary' until transition to STARS is complete. NCVER outputs should be classified as 'platinum' / 'final' until then

Based on the above layering of data into Gold, Silver and Bronze, the diagrams below present some *example* data personas and how they use these layers.

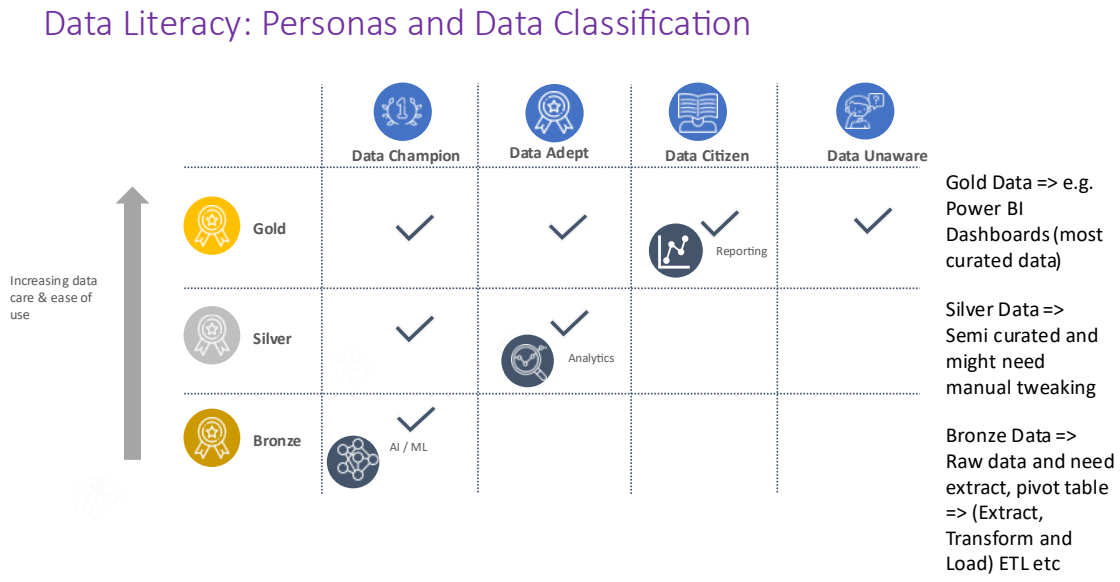


Figure 14: Data classification and personas example

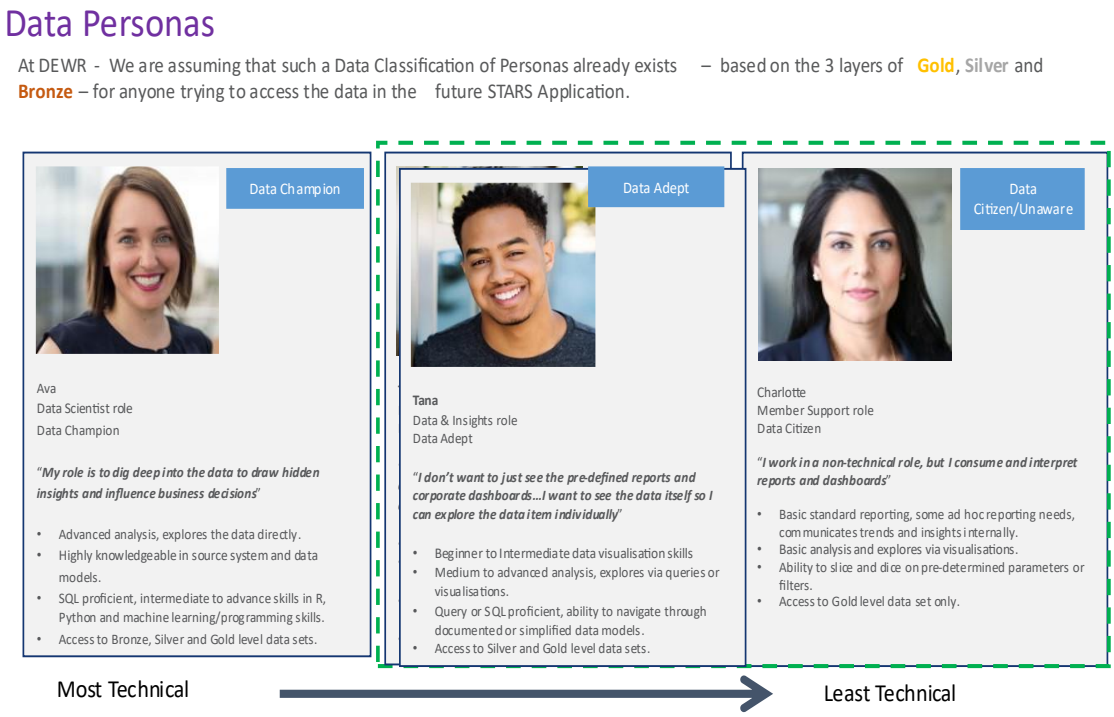


Figure 15: Example data personas

8.4 Data Governance Approach

Data Governance is the power to enhance the capability of a system through roles and responsibilities. It is a combination of People – Process & Technology working together in harmony to heal a system through proper governance and compliance and through empowerment of governance personas (Data: Owners, Stewards and Custodians).

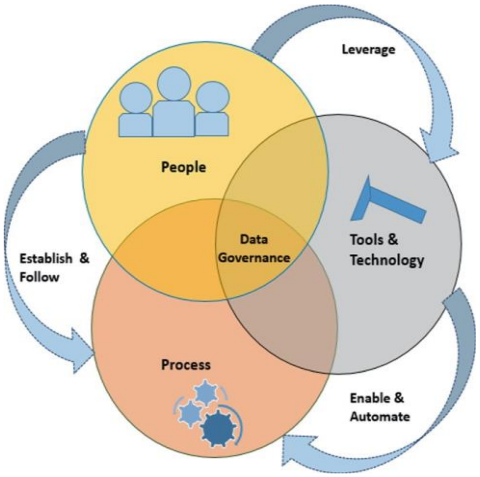


Figure 16: Data Governance is a People, Process and Technology Approach

The STARS solution improves data governance and empowers authorised data consumption via integration through Microsoft Purview. Microsoft Purview is an Azure service that provides core data governance capabilities including data catalogue, business glossary, data lineage and data classification features.

A Systems Operator will be able to configure compliance metrics in STARS on the behalf of the VET Regulator(s). DXC understands that compliance reporting will be most likely be performed using Power BI. As such, relevant Measures (a Power BI term) will likely need to be determined and implemented, either directly in the Master DB or in Power BI datasets.

The KPI Metrics for Data Governance – to be delivered through a Power BI dashboard(s) based on the data collected in MS Purview repository - will be workshopped with DEWR in the design stages of the project.

At any rate, DXC has proposed flexible approaches to discovery and design of the Master DB and any Analytics databases and reporting dashboards, described further below. DXC believes future Data Governance requirements will be covered by these approaches.

STARS will ensure that the appropriate data access controls are implemented:

- Jurisdictions are responsible for quality assurance and release of national data on training activity they fund/co-fund according to agreed timeframes.
- Jurisdiction specific data items may be held by that jurisdiction(s), unless otherwise required by legislation.
- Business/Program rules applicable to a jurisdiction, can be accessed by the jurisdiction.

STARS provides for these controls in several ways:

- All data submissions are associated with the relevant jurisdiction – this is included in the overall Master DB schema. The Submission Services (Service DB) will be structured to separate submissions by jurisdiction/provider.
- Noting that data submissions are associated with jurisdictions, the processing rules within the system will take this into account.
- The solution provides a Data Management and Release portal for jurisdictions to manage and control the release of their data. Again, the rules engine will be brought to bear here, as well as physical separation of the underlying data in the staging areas and logical separation (through partitioning) within the Master DB.
- DXC will need to clarify what or how jurisdictions view of rules is to look like; however, this is likely to be an embedded report within the Data Management/Release portals.

It is assumed that these Jurisdictions have already people assigned and onboarded – as part of a Data Governance Framework – and those on-boarded in these roles have the technical know - how of how to perform these tasks noted above, through a Data Literacy Model.

Also, the assumption is that these jurisdictions meet on a scheduled basis (fortnightly, monthly) – and the communication lines are open in a horizontal fashion so that cross functional issues can be discussed in an open manner rather than a siloed approach.

Please find attached an example of an Enterprise Data Model in Appendix D, which is an example of how a Data Governance model is formed through assignment of Subject Areas to Data Governance Personas.

8.5 Master Data Management Approach

Note: MDM detailed requirements have not yet been developed, nor has an analysis been undertaken of data quality issues requiring MDM intervention (e.g., level and nature of duplications or data merges). MDM designs, processes, mappings, etc., will be developed during the life of the project as requirements and data issues arise, subject to the scope change management process.

Accordingly, the Solution Design describes the approach, tools and general processes that will be applied to Master Data.

Master data represents data about the business entities that provide context for business transactions.[10] Master data isn't frequently altered, and though it's used to describe transactions, it isn't transactional in nature.

In the STARS system master data management (MDM) will be implemented using native Azure solutions. STARS will use Azure Data Factory, Microsoft Purview, Azure Logic App, and Azure SQL Database to implement master data solution based on the native Azure ecosystem.

8.5.1 Master Data Solution Functionalities

A typical master data solution loads data from the multiple data source systems. It can also clean, match, and integrate the data across the distributed data landscape and use it to create golden master data records stored in a central master database system. It has the capabilities to capture data, standardise, clean, de-duplicate, prepare comparison data, match, integrate, send email and work with business rules. Once master data has been created and centrally stored, it can be synchronized with all systems that use master data to ensure their consistency. Services can connect to the master database to check and verify data while submitting the new data.

The core functionalities of an MDM solution are described in the diagram below:

[10] https://en.wikipedia.org/wiki/Master_data

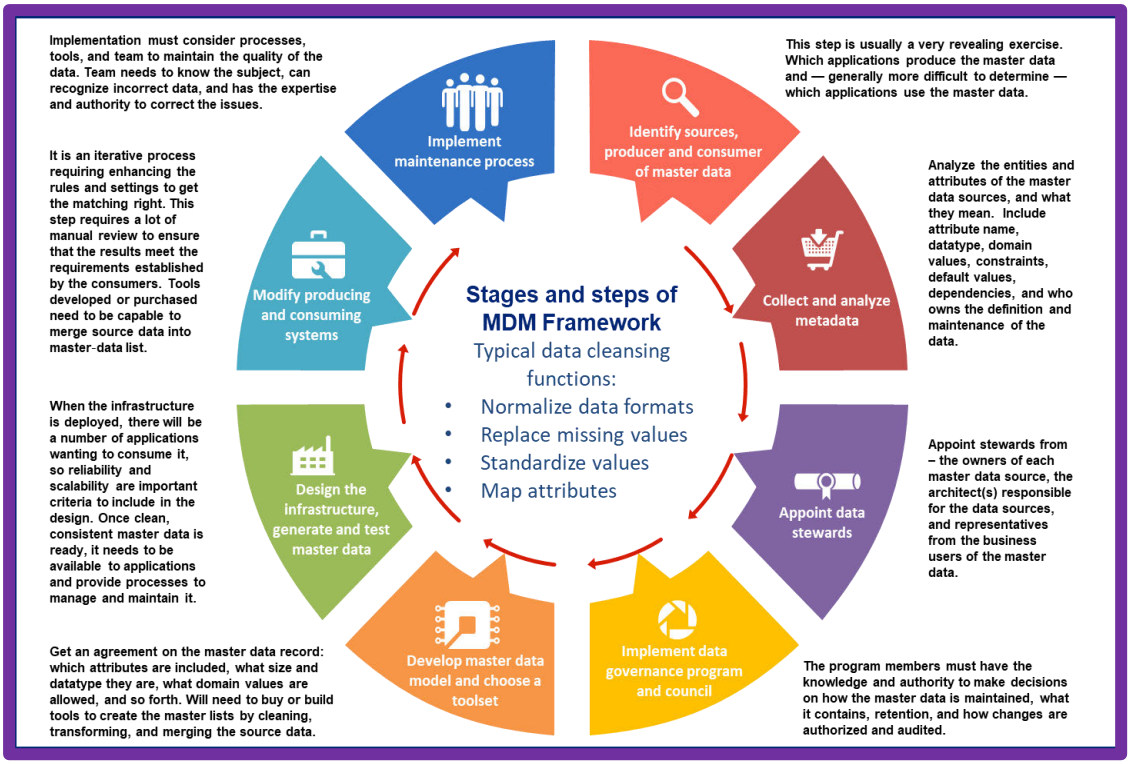


Figure 17: MDM functionalities

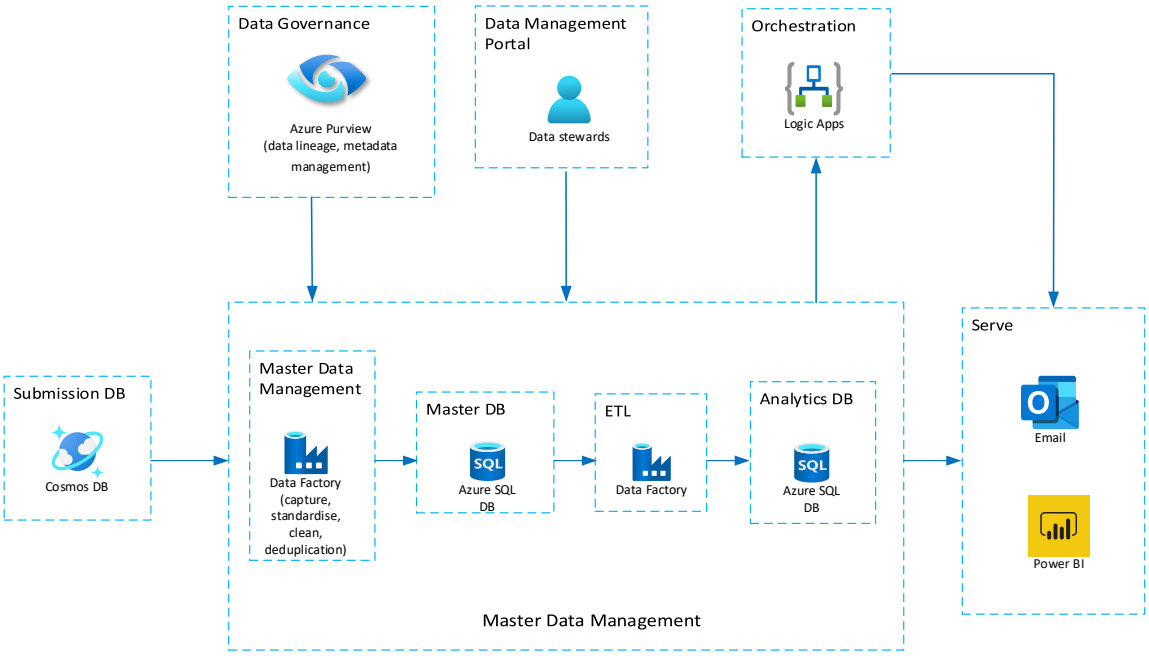


Figure 18: Master Data Management Overview

8.5.2 STARS MDM Approach

8.5.2.1 Capture, Standardise, Clean and De-duplication

Azure Data Factory is the cloud-based ETL and data integration service that allows for data-driven workflows for orchestrating data movement and transforming data at scale. STARS will use Azure Data Factory to create and schedule data-driven workflows that can ingest data from different data stores. Raw data will be organized into meaningful data stores using Azure Data Factory, including those based on Azure Blob Storage, Azure SQL Databases, Azure Cosmos DB (SQL API) etc.

Azure Data Factory will be used to capture the data, standardise, clean, de-duplicate, prepare comparison data, match, and integrate with other solutions.

STARS will include data quality transformations in an Azure Data Factory pipeline using scripts. Azure Data Factory will deduplicate rows where sufficient identifiers are available for an exact match (e.g., USIs, Training Contract Numbers).

Custom scripts are likely required by the logic to merge matched with appropriate survivorship. Data Factory will provide the functionalities of data capture, data standardise, clean data using transformation, de-duplicate data, match and integrate against business rules.

There are three types of activities that can be configured in the Azure Data Factory: Data movement activities, Data transformation activities, and Control flow activities. As part of Data transformation activities, Data Flow activities, Azure Function App activities, and Databricks Notebooks may be used in the Data Factory to capture, clean, and standardise the data. As part of Control flow activities, For Each, Get Metadata, Lookup Activity and Filter activities will be used to work with individual records from source data sources.

Azure Data Factory Mapping Data Flows has several capabilities that allow data cleansing by finding possible duplicates. Regular Expressions inside of ADF Data Flow may be used for pattern matching instead of exact string matching.

8.5.2.2 Data Lineage and Metadata Management

Microsoft Purview provides a unified data governance solution to help manage and govern cloud data. It provides the capability to create a holistic, up-to-date map of the data landscape with automated data discovery, sensitive data classification, and end-to-end data lineage. It enables data consumers to access valuable, trustworthy data management. Some of the functionalities it can provide are the data lineage, metadata management, data catalogue, data discovery, data classification, and business vocabulary.

A primary of any data catalogue is its ability to show the lineage between data products. Microsoft Purview supports data lineage capture from three Azure Data Factory activities:

- Copy data
- Data flow
- Execute SSIS package

Physical data names of discovered master data will be mapped to any common business vocabulary in Microsoft Purview. Microsoft Purview allows application of system or custom data classifications on file, table, or column assets. (Data classifications are like subject tags.) Microsoft Purview marks and identifies the content of specific data types found within the data estate during scanning.

Sensitivity labels may be used to identify the categories of classification types within the organizational data.

One of Microsoft Purview's platform features is its ability to show the lineage between datasets created by data processes. Systems like Data Factory, Data Share, and Power BI capture the lineage of data as it moves. Custom lineage reporting is available via REST API; this may be used in development of reporting dashboards (reporting requirements not yet provided.)

It should be noted that data classification created and tagged automatically through Microsoft purview – must be classified through a pre-approved classification process, e.g., PII (Personally Identifiable Data), Protected (Default), Confidential or Public), prior to the tool being used to perform the tagging. It is assumed that such data classifications pre-exist.

It is also assumed that all the above functionalities of Microsoft Purview – will be handled through an existing Data Governance Framework at DEWR – which has pre-allocated Data Governance Personas (Owner, Steward, Custodian - on-boarded through a data literacy framework) that can ongoingly manage and maintain the data classifications, business glossaries and data catalogues created through Microsoft Purview.

8.6 Aggregated Validations (a.k.a. Output Editing)

Aggregate validation requirements, including reporting, have not yet been provided.

DXC will work with data custodians/scientists to facilitate validation performed on aggregated data sets that have already been released into the master database. This may include further design and development of appropriate analytics databases and reports.

8.7 High Level Conceptual Data Stores

To support processes and services identified below in the Business Architecture and Software Architecture sections, several conceptual data stores have been identified. These data stores will provide storage and retrieval capacity for information held within STARS, including Data Packets, Data Packages and VET-IS logical entities of all types.

Table 8: STARS Conceptual Data Stores

Data Store	Purpose	Description
Provider	To store information about training providers. To store slowly changing information from training providers, e.g., delivery locations, offerings. To temporarily store rapidly changing information as it is received, in readiness for preparing submissions, e.g.,	The Provider store will be utilised by the Provider Services. This includes information managed by micro-RTOs in the Provider Portal.

	Data Packets that are being aggregated into Packages, such as full Enrolment information.	
Submission	To store all received submissions in the form of Data Packets and Packages. To store meta data relating to submissions. To store para-data relating to submissions.	The Submission store will retain all received information indefinitely, to support reporting and analytics relating to submissions.
Staging	To store information about submissions that have passed STARS-hosted data quality and business rules (including national and hosted STA rules), while they are being assessed and then released by STAs.	Whilst the Submission store can hold both packets and packages, the Staging store is purely package oriented. This is because STAs will require the complete context of information – not just individual packets.
Master – STA	The Master STA partitions will be used for STA operational and reporting needs. Each STA will have their own partition.	The STA partitions will contain 'silver' level data (passed data quality and business rules), to retain and support STA operational-level information and reporting. This includes both national and state-specific data elements specified in the VET-IS logical schema. Note that the STA partitions are likely to have a level of duplicate or inconsistent data across partitions (e.g., Students with same USI in different states, but with slightly different personal details, such as personal names.)
Master - National	The Master National partition will be used for national reporting and as a source of truth for national VET data, including student transcripts.	The National partition will contain 'gold' level data (passed data quality and business rules, de-duplicated, enriched with national classification data), held in VET-IS logical schema format, minus any STA operational-level information. When packages are released by STAs they converted into VET-IS schema format, de-duplicated and mastered using MDM processes. E.g., Students are converted into Stakeholders, Enrolments into Responsibilities. Data will

		be matched and merged, so that, for example, the same Student with enrolment records in different states will have their data merged into a single entity. It is likely that downstream reporting will require projections (views) that transform VET-IS schema data into VET-IS data packet entities, e.g., Stakeholders as Students, Responsibilities as Enrolments.
Reporting & Analytics	Projections of Master data in formats suitable for aggregated validations, and reporting and analysis by VET data consumers.	Reporting and analytics requirements are yet to be provided. Aggregated validation requirements are yet to be provided.
Reference	Storage and retrieval of key reference data.	Reference data supports both operational and reporting requirements of the solution.

8.8 Event Sourcing Model and Event Data Stores

STARS will require a flexible approach to recording, managing, responding to, and reporting business events. This is underlined by the requirement for STARS to manage potentially many different STA specific workflows and business rules that will grow and evolve over time. It will not be sufficient, say, to simply record various statuses against a submission – this will quickly become unwieldly as the number of possible statuses and associated lifecycles grow.

Instead, STARS will utilise the Event Sourcing Pattern[11]. Instead of storing just the current state of the data in domain entities, such as submissions, STARS will use a set of append-only Event Stores to record the full series of events associated with or actions taken on submissions. This will simplify evolving workflows by avoiding the need to constantly update the VET-IS information model to reflect changing processes.

When a workflow needs to decide on what to do with, say, a submission, it simply just runs an aggregate query on the events associated with the submission. The 'sum' of the events effectively plots where a submission is within its lifecycle in the pipeline. If a new workflow arises that needs extra processing steps, it is simply a matter recording new events associated with the new processing steps.

The Event Stores also simplify auditing and data lineage requirements.

[11] See https://docs.microsoft.com/en-us/azure/architecture/patterns/event-sourcing

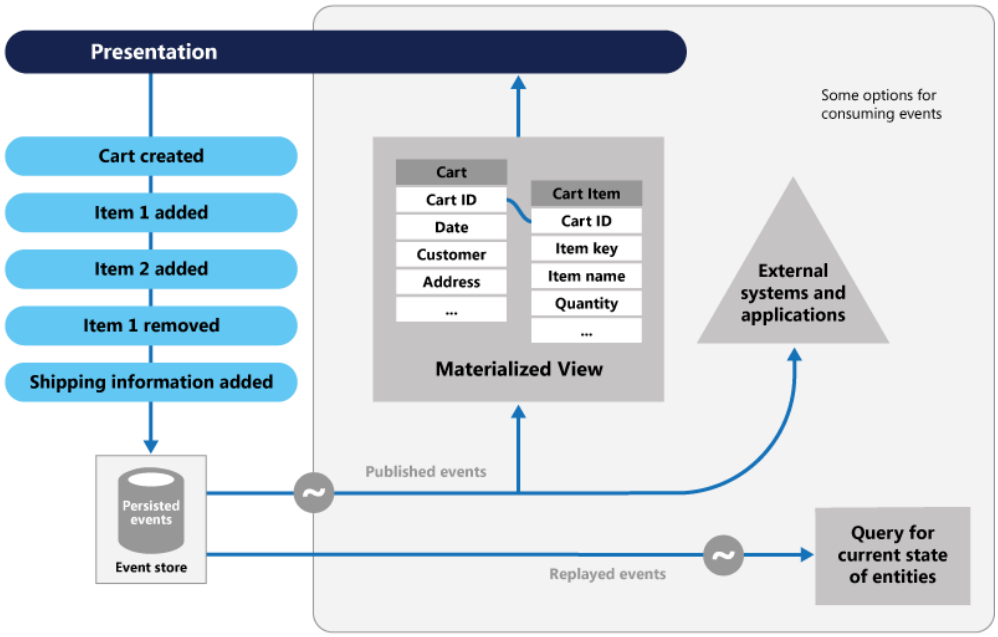


Figure 19: Event Sourcing Example

Many solution services described in this design incorporate Event Data Stores. These will all record the full lifecycle events associated with, say, submissions.

Although each service has its own 'logical' Event Store, the STARS implementation will likely combine the various Event Stores into a single global Event Store – this will allow for aggregate queries on the lifecycle of submissions throughout the entire submission pipeline.

8.9 Reporting and Analytics Approach

Note: Analytics and Reporting detailed requirements have not yet been provided. Analytics and Reporting databases, designs, processes, dashboards, etc., will likely be developed during the life of the project as requirements are developed, subject to the project scope change management process.

Accordingly, the Solution Design describes the approach, tools and general processes that will be applied to Analytics and Reporting.

8.9.1 Reporting and Analytics Database

An analytics and reporting database (or likely multiple databases) may be designed and implemented to offload data reporting and visualisation activities.

In this case a dedicated reporting and analytics database will be continuously populated from the core application databases. This will centralise reporting and analytics data and offload workloads from the core application databases, to ensure operational performance does not decrease as data consumption activities increase.

STARS will allow reporting on point in time data stored in STARS Master Database. The STARS will provide this through temporal tables within Azure SQL (see section 8.14.6). This is provided to retain the audit trail of changes within the master database.

While temporal tables are an important audit feature, they are not the best approach to providing as-at style reporting for visualising and analytics purposes.

STARS may include the automated and continuous population of an analytics database for the specific purpose of supporting current and as-at reporting requirements, as they are developed and provided. This would involve the conversion of normalised master database structures into dimensional models which are optimised for analytics workloads. This approach will ensure performance of reporting workloads is optimised and that the increase of reporting requirements does not impact core operational performance.

This transformation approach also allows for improved handling of as-at reporting requirements including:

- The star schema will be driven from the VET-IS Logical Schema
- The creation of date and time dimensions
- The creation of slowly changing dimensions (type-2 SCD's) that provide the point in time data selection capabilities
- Current data flags to conversely support right-now reporting
- Transformation of data into dimension/fact star schemas to optimise reporting workloads
- Implementation of organisation and reporting hierarchies to allow drill down and roll up features
- Creation of key KPI's and metrics to ensure consistency in reporting definitions across all reporting including self-service.

This approach will allow comparison of data between different time periods including:

- Comparison of data in aggregate over different date periods (e.g., reporting comparison of compliance for one year against another)
- Comparison of changes to data records over different date periods (e.g., how has subject enrolments changed over time).

8.9.2 Reporting

DXC is proposing to deliver the department a core set of Power BI reports that deliver key reporting and analytical requirements as part of the solution (noting that reporting requirements are not yet provided.)

As reporting and analytics requirements will continue to rise over time, subject to the project scope change management process, DXC may also deliver a suitable data platform to allow the department to continue to deliver new data reporting capabilities by leveraging the reporting database models delivered by DXC, which may be designed to support self-service reporting.

The STARS solution incorporates both Azure SQL temporal tables for point-in-time reporting, and dimensional (time-series) analytics databases for use by Power BI (and other tools).

The solution proposes an independent database for BI and Analytics. This is provided to ensure data is restructured into a format to simplify the development of reporting and analytics features, and to ensure the core application performance is not impacted as the reporting and analytics workloads increase.

Reports, dashboards, and queries may be designed to be customisable so STARS users can use pre-set reports and dashboards or create a wide range of customisable reports, queries and dashboards

to analyse. Data can be interrogated in the solution or downloaded from the repository (subject to access rules).

STARS reporting will also provide Power BI restricted workspaces to STAs that will include Power BI Row-level security, roles, etc. Depending upon what Power BI license the department has, Analysis services (cubes) might be used in case Power BI Premium is not available.

It has been identified that data quality dashboards (i.e., validation results analysis) in the Provider Portal would be of value to support Providers to maintain high data quality. Validation reports and dashboards built in Power BI via Azure Synapse can also be integrated/embedded into the Provider portal if required.

For both the above scenarios, the final decision on the most suitable option for reporting is yet to be determined.

As proposed above, the solution provides an approach to delivery of a reporting and analytics components which will restructure the application data into a format designed to simplify the data model into a form that is suitable for user driven self-service reporting. Key capabilities of this self-service capability include:

- Availability of data catalogue, glossary, and data lineage information
- Simplified data models designed to support self-service reporting
- Data structures supporting Point-in-Time reporting and “as-at” analysis
- Integrated security ensuring only authorised data is made available to authorised users at all points on the data consumption solution.

8.9.3 Reporting and Analytics ETL

Data Factory ETL pipelines may be used to extract data from the Master Database and build star schema data marts for consumption by Power BI on a scheduled or event-driven basis.

8.10 Data Integration Approach

Note: STA data integration detailed requirements have not yet been provided. STA data integration pipelines, etc., will likely be developed during the life of the project as requirements are developed, subject to the project scope change management process.

Accordingly, the Solution Design describes the approach, tools and general processes that will be applied to STA Data Integration.

STARS will submit submission data to the associated STA for approval and state specific validations at the appropriate point in the data validation workflow. The STA will respond to STARS with a Pass\Fail notification (along with details of the errors, warnings and/or messages) indicating how the data submission should proceed through the data validation workflow.

External Reference Data that is being stored within STARS will be refreshed at regular intervals via a frequently scheduled Data Factory ETL pipeline.

Where external system synchronisation is required and has been approved by the external system this will be catered for using Data Factory ETL pipelines upon STARS internal data update.

8.11 Data Extract Approach

Note: Many data extract detailed requirements have not yet been provided. Data extraction pipelines, etc., will likely be developed during the life of the project as requirements are developed, subject to the project scope change management process.

Accordingly, the Solution Design describes the approach, tools and general processes that will be applied to STARS data extraction, along with current known extract requirements (e.g., AVETMISS 8.0 extracts).

STARS will allow data to be exported from all relevant data stores. The primary mechanism for data export will be Data Factory workflows, including exporting data from the Master DB for STA or NCVER operational and reporting needs.

Data Export Pipeline – allows an extract of data to data-consumers – for instance to provide detailed valid data to support contract management, payment and program assurance to partner systems.

STARS will allow a dataset to be exported in CSV or XML format (formats TBD). STARS will also provide the capability to extract AVETMISS 8.0 NAT files from Staged or Mastered data to support RTO, STA and ASQA requirements.

The DXC STARS provides this through Data Factory. As mentioned elsewhere, DXC is proposing a flexible approach to data ETL and analytics; we foresee this requirement would be covered under this approach.

Note that Data Factory is very flexible in the import and export data formats and will provide the department a suitable platform for current and future data extract and downstream integration needs.

Data is provided for ETL via a set of views with filter criteria based on Partner or Participant data access limitations.

The structure of views and content is identical across Partners or Participants except for SQL select criteria – standardization across ETL OR Power BI facilitation.

It is expected that Power BI Datasets and Reports for different STAs are built based on a group of views but are identical other than access to set of views for specific user group. There will also be capability to allow STAs to ingest external datasets for state-specific reporting, but details on this are yet to be determined, so the scope of this capability cannot be defined at this time.

8.12 Highspeed Bulk Data Share Approach

Note: High speed bulk data share detailed requirements have not yet been provided. Data extraction pipelines, Azure Data Share configurations, etc., will likely be developed during the life of the project as requirements are developed, subject to the project scope change management process.

Accordingly, the Solution Design describes the approach, tools and general processes that will be applied to STARS high speed bulk data sharing.

An option to for data sharing includes the use of the new Azure Data Share feature. This feature provides secure, automated data replication to authorised consumers (e.g., STAs) via the Azure Data

share capability. This could, for example, be used to replicate authorised data to other departments who also use Azure Data Share without the need to manually build data transfer pipelines.

Azure Data Share will enable STARS to:

- Share data with partners outside of STARS
- Keep track of who STARS has shared data with
- Choice of snapshot or in-place sharing
- How frequently data consumers are receiving updates to data

Data Share will allow partners to pull the latest version of STARS data as needed or allow them to automatically receive incremental changes to your data at any defined interval.

Azure Data Share enables partner data consumers to:

- View a description of the type of data being shared
- View terms of use for the data
- Accept or reject an Azure Data Share invitation
- Accept data shared
- Access data in place or trigger a full or incremental snapshot of shared data

8.13 Data Administration

8.13.1 System Management

Microsoft Purview provides role based and asset-based access permissions.

The data management portal will restrict data read\write permissions as defined by the client.

8.13.2 Audit

Any data modified in the Master database will have the system user responsible for the update and the time it was performed. All previously updated versions of data attribute values and audit information will be available since go-live of the STARS solution. The capture of audit information relating to data modified within the existing solution during any cut over transition period (i.e., staged go-live) is the responsibility of the existing solution.

8.13.3 Data Management

Microsoft Purview will provide the data discovery, categorisation and lineage functionality for the STARS solution. Data glossaries and dictionaries will be maintained in Purview with the ability to visualise data values using Power BI.

Data fixes that need to be applied to mastered data will be made through a data management portal.

External reference data will be updated by Data Factory ETL pipelines either when external data is updated or as per a regular schedule.

8.14 Physical Design

8.14.1 Assumptions

Refer to section 6.3 Assumptions

8.14.2 Known Issues

1. VET data captured in the existing solutions (i.e., NCVER systems) may not contain the sufficient information to be compatible with the new VET-IS or requirements for VDS and the STARS solution. This will likely affect transition requirements, including future data migrations.

8.14.3 Data Schemas

STARS will utilise several data schemas, oriented around the VET Information Standard and system operational requirements, such as validating data, recording events and presenting reference data.

Table 9: STARS Data Schemas

Schema	Type	Role
VET-IS Data Packet	JSON	Transmission to APIs Storage of JSON data in Cosmos DB
VET-IS Physical	Relational (SQL)	Storage of data in Master DB
STARS Event	JSON	Storage of STARS events in Cosmos DB
STARS Reference	Relational (SQL)	Storage of data in Reference DB
STARS Rules	JSON	Storage of rules configuration in Rules DB
STARS Validation	JSON	Storage of validation outcomes in Validation DB

8.14.4 Data Infrastructure

All data storage and retrieval will be performed entirely using appropriate Azure PaaS data services.

The STARS solution will utilise a mix of document-oriented, relational and analytics (OLAP) database architectures to meet the needs of data ingest, validation, master data and reporting.

Of note is that the database architecture and supporting database services are designed to facilitate and support growth for large volumes of VET data. Further, the databases are intended to fully support changes to the VET Information Standard as they arise.

The solution data stores will utilise the following Azure PaaS Services:

Table 10: STARS Azure Data PaaS Scenarios

Scenario	Azure Service
Document oriented (JSON) data, such as submissions (e.g., data packets & packages).	Cosmos DB
Relational data, such as Master Data (e.g., VET-IS Stakeholders, Responsibilities, Products).	Azure SQL
Reporting & Analytics (requirements to be provided).	Azure SQL Analysis Services, Azure Synapse or Power BI

8.14.5 Service Data Stores

STARS will maintain several data stores to support the operational and reporting requirements of the solution. These are a mix of document-oriented and relational databases, managed by STARS services:

Table 11: STARS Data Stores

Data Store	Services	Type	Schema	Azure Service	Role	Comments
Provider DB	Provider	Document (JSON)	VET-IS Data Packet (JSON)	Cosmos DB	Store and retrieve training provider information.	The Provider DB will be utilised by the Provider Services, forming the main repository for storing provider information. This includes information managed by micro-RTOs in the Provider Portal. It will store information about training providers, including: • slowly changing information e.g., provider details, delivery locations, offerings. • rapidly changing information as it is received, in readiness for preparing submissions, e.g., Data Packets that are being aggregated into Packages, such as full Enrolment information.

Submission DB	Submission	Document (JSON)	VET-IS Data Packet (JSON)	Cosmos DB	Store and retrieve submitted VET data. Store meta data relating to submissions. Store para-data relating to submissions.	The Submission DB will be utilised by the Submission Service, forming the main repository for storing submitted VET data. Other services will retrieve submission data from the Submission Service, e.g., to support the Staging Service to retrieve submission data for transmission to STAs. The Submission DB will retain all received information indefinitely, to support audit, lineage, correlation, reporting and analytics relating to submissions.
Submission Event DB	Submission	Document (JSON)	STARS Event	Cosmos DB	Store and retrieve Submission Service events.	The Submission Event DB will store all lifecycle and management events surrounding submissions.
Validation DB	Validation	Document (JSON)	Validation	Cosmos DB	Store and retrieve Validation Service events.	The Validation DB will store all lifecycle and management information surrounding validations. This will include the outcomes of submission validations.
Staging Event DB	Staging	Document (JSON)	STARS Event	Cosmos DB	Store and retrieve Staging Service events.	The Staging Event DB will be utilised by the Staging Service to store and retrieve all lifecycle and management events surrounding staging operations. It will, for example, store information about submissions that are being or have been assessed and validated by STAs.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

Release Event DB	Release	Document (JSON)	STARS Event	Cosmos DB	Store and retrieve Release Service events.	The Release Event DB will be utilised by the Release Service to store and retrieve all lifecycle and management events surrounding release operations. It will, for example, store information about submissions that are being or have been released by STAs.
Mastering Event DB	Mastering	Document (JSON)	STARS Event	Cosmos DB	Store and retrieve Mastering Service events.	The Mastering Event DB will be utilised by the Mastering Service to store all lifecycle and management events surrounding mastering operations.
R & A Provisioning Event DB	R & A Provisioning	Document (JSON)	STARS Event	Cosmos DB	Store and retrieve R & A Provisioning Service events.	The R & A Provisioning Event DB will store all lifecycle and management events surrounding provisioning of information into downstream reporting and analytics.
Master – STA	STA Services, Mastering	Relational	VET-IS Physical + STA Extensions	Azure SQL	Store and retrieve STA silver-level VET data.	The Master STA partitions will be used for STA operational and reporting needs. Each STA will have their own partition. The STA partitions will contain ‘silver’ level data (passed data quality and business rules), to retain and support STA operational-level information and reporting. This includes both national and state-specific data elements defined in the VET-IS logical schema.

						Note that the STA partitions are likely to have a level of duplicate or inconsistent data across partitions (e.g., Students with same USI in different states, but with slightly different personal details, such as personal names.) It is likely that STA operations and reporting will require projections (views) that transform VET-IS schema data into VET-IS data packet entities, e.g., Stakeholders as Students, Responsibilities as Enrolments.
Master – National	All	Relational	VET-IS Physical	Azure SQL	Store and retrieve national gold-level VET data.	The Master National partition will be used for national reporting and as a source of truth for national VET data, including student transcripts. The National partition will contain 'gold' level data (passed data quality and business rules, de-duplicated, enriched with national classification data), held in VET-IS logical schema format, minus any STA operational-level information. When packages are released by STAs they converted into VET-IS schema format, de-duplicated and mastered using MDM processes. E.g., Students are converted into Stakeholders, Enrolments into Responsibilities.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

						Data will be matched and merged, so that, for examples, the same Student with enrolment records in different states will have their data merged into a single entity. It is likely that downstream reporting will require projections (views) that transform VET-IS schema data into VET-IS data packet entities, e.g., Stakeholders as Students, Responsibilities as Enrolments.
Rules DB		Relational	STARS Rules	Azure SQL	Store and retrieve operational and configuration data for rules.	The Rules DB will be utilised by the Rules Services to manage the lifecycle and application of coded rules and rule sets. Rules include data quality and business rules.
Reference DB	Reference + All	Relational	STARS Reference	Azure SQL	Store and retrieve reference data.	The Reference DB will be managed by the Reference Services. It will support many operational and reporting services across the solution, e.g., providing reference category data for validation.
Validation DB	Validation	Relational	STARS Validation	Azure SQL	Store and retrieve validation data.	The Validation DB will be utilised by the Validation Service. It will support management and lifecycle events surrounding validation, including validation results.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

8.14.6 Master Database

Note: Master Database detailed requirements have not yet been provided. Detailed schemas, partitioning designs, etc., will be developed during the life of the project as requirements are developed, subject to the project scope change management process.

Accordingly, the Solution Design (currently) describes the approach, tools and general processes that will be applied to storing mastered data.

Please also note that detailed database and schema analysis, design and implementation work will be undertaken during Work Package 2.

The Master Database will be implemented using Azure SQL Server. This database will hold information in VET-IS Physical Schema format, including Stakeholders, Responsibilities, Products, Offerings and (post R1) Events.

Data will be held mainly in temporal tables, for audit and time-based analysis.

Temporal tables (SYSTEM_VERSIONING) will be used to retain history on applicable objects in the Azure SQL Master database. Changes to or audit on data will be tracked using this mechanism. Temporal tables are a programmability feature of Azure SQL Database that allows the tracking and analysis of the full history of changes in data, without the need for custom coding. Temporal tables keep data closely related to time context so that stored facts can be interpreted as valid only within the specific period. This property of temporal tables allows for efficient time-based analysis and getting insights from data evolution[12].

Master DB data will form all or part of the data to populate analytics database, e.g., Power BI or Azure SQL Analytics Services (other sources include the staging data in Cosmos DB).

Master data will be made available to stakeholder via data extracts, reports, etc. This data will have access controls down to the column level utilising Azure SQL security features.

Submission data will be stored in the relational Master database in alignment with the VET Information Standard (VET-IS) Physical Schema. Please refer to VET Information Standard (VET-IS) Logical Entity Relationship Diagrams in Appendix A and the VET-IS Physical Schema.

The Master Database is separated into two broad sets of partitions:

- STA. The STA partitions will contain ‘silver’ level data (passed data quality and business rules), to retain and support STA operational-level information and reporting. This includes both national and state-specific fields defined in the VET-IS physical schema. Each jurisdiction will have its own partition, with jurisdictions only able to access their own data.
- National. The National partition will contain ‘gold’ level data (passed data quality and business rules, de-duplicated, enriched with national classification data), held in VET-IS physical schema format, minus any STA operational-level information. The National partition will be used for national reporting and as a source of truth for national VET data, including student transcripts.

[12] For more information: https://docs.microsoft.com/enus/azure/azure-sql/temporal-tables

8.14.7 Reporting and Analytics Databases

Note: Analytics detailed requirements have not yet been provided. Detailed schemas and BI designs, etc., will be developed during the life of the project as requirements are developed, subject to the project scope change management process.

Accordingly, the Solution Design (currently) describes the approach, tools and general processes that will be applied to analytics.

Please also note that detailed database and schema analysis and design will be undertaken during Work Package 2.

Analytics databases and cubes, to be used for downstream reporting services, e.g., Power BI. Data from the master database and staging databases will be extracted on a continuous basis and stored in suitable analytics data stored, e.g., Power BI data models, Analysis Services cubes, etc.

9 Business Architecture and Design

Please note that many detailed requirements are not yet provided surrounding lower-level STARS processes and workflows. Accordingly, the solution design proposes a set of flexible approaches that are most likely able to support detailed requirements as they are created or evolved during the life of the project (subject to the project scope change management process), and beyond.

Please also note that detailed workflow and rules analysis, design and implementation work will commence during Work Package 2 and the continue through the life of the project. Please see commentary regarding workflow and rules management in section 9.6.

9.1 Overview and Purpose

This Business Architecture and Design section addresses the business context, main business processes, workflow capabilities and requirements for the STARS solution.

It describes the following:

- The current business context and information flows relating to submission and reporting of VET data
- The to-be business context and information flows
- The high-level submission pipelines that guide the development of major STARS business processes
- The high-level STARS business processes, covering identified scenarios for submission and processing of VET data
- The approach to flexibly automate yet to be specified business processes and workflows

9.2 Business Context – As Is

RTOs conduct both fee-for-service (FFS) and government funded training ('funded'). Funded training is where student fees are paid for by government(s) on behalf of students to RTOs. Funded training is conducted under a contract between an RTO and a STA and/or Federal Government, with the contracts covering many conditions on the training services provided. FFS training is not conducted under any contracts between RTOs, STAs or the Federal Government and is charged direct to students. In either instance, RTOs must comply to many regulatory requirements covering delivering training services in the VET sector, including reporting.

As part of VET training regulatory requirements, RTOs must collect and report on training activity throughout the course of a year, including both fee-for-service (FFS) and government funded training.

At intervals, ranging from monthly through to annually depending on circumstances, RTOs report training activity to either or both of NCVER and the relevant STA. Currently this is done via transmission of data files in AVETMISS format.

FFS training activity data is submitted by RTOs to NCVER directly. FFS data may be submitted to an RTOs local STA, depending on the STA's requirements, but activity reporting to NCVER is required by national and state training regulatory requirements.

Funded training activity is submitted by RTOs to their local STA. STAs then process and collate the training activity data locally, prior to submitting the collated data to NCVER. Please note that the

term collate in this context refers to gathering and collating data from multiple sources, not merging or summarising records.

NCVER collects both FFS and funded activity data from RTOs and STAs and transforms the data into several reporting data sets, known as “collections”. These include National VET Provider Collection; VET in Schools Collection; National Apprentice and Trainee Collection.

Several key reports/data sets are sourced from the VET Collections, including “Government-funded students and courses”, “Total VET students and courses”, “VET in Schools”.

The diagram below illustrates the relationship between submission, collection and reporting (see: https://www.ncver.edu.au/research-and-statistics/collections/students-and-courses-collection).

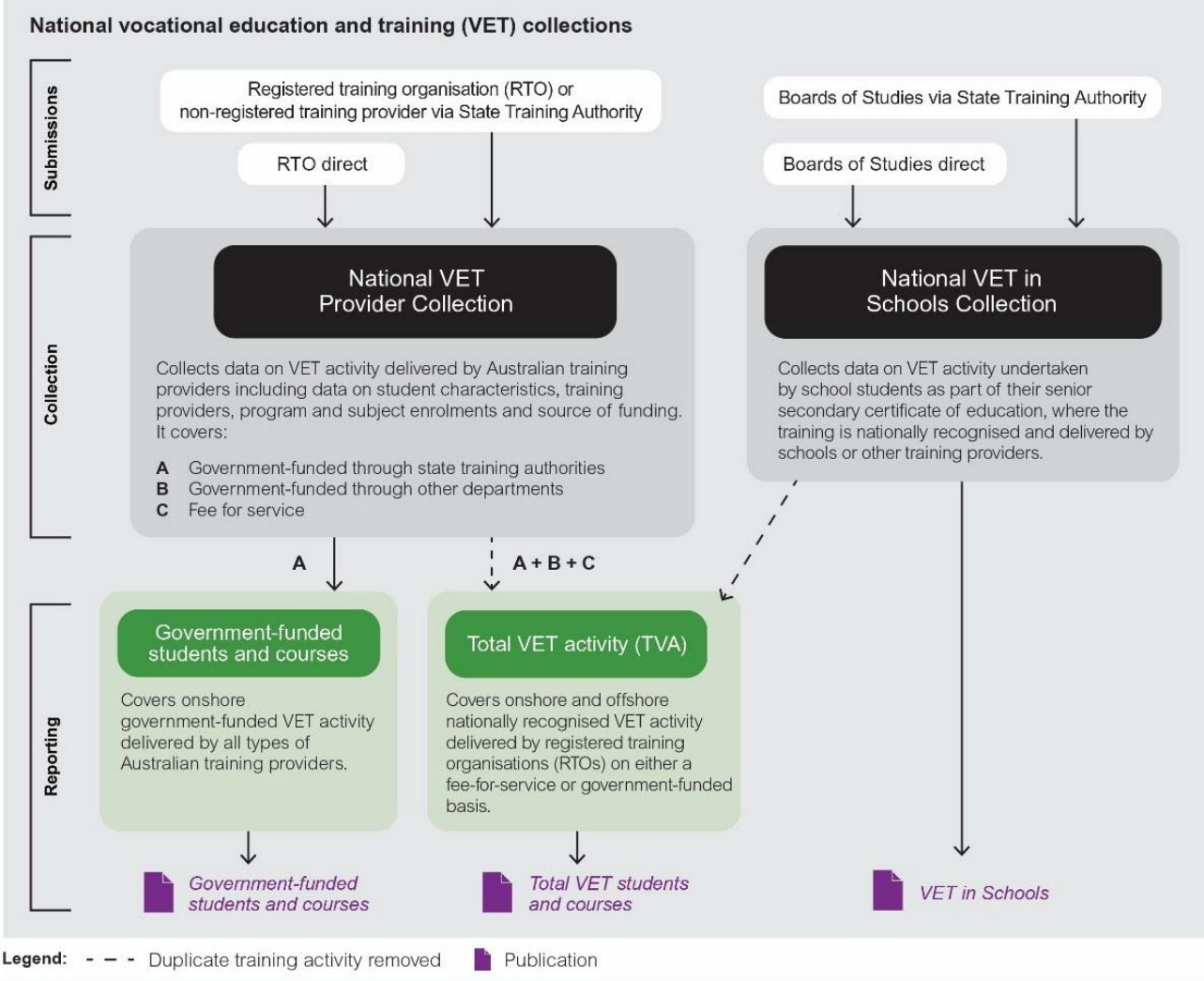


Figure 20: VET Collections

9.2.1 Context Diagram - As Is – High Level

The collaboration diagrams below broadly shows the interactions between RTOs, STAs and NCVER.

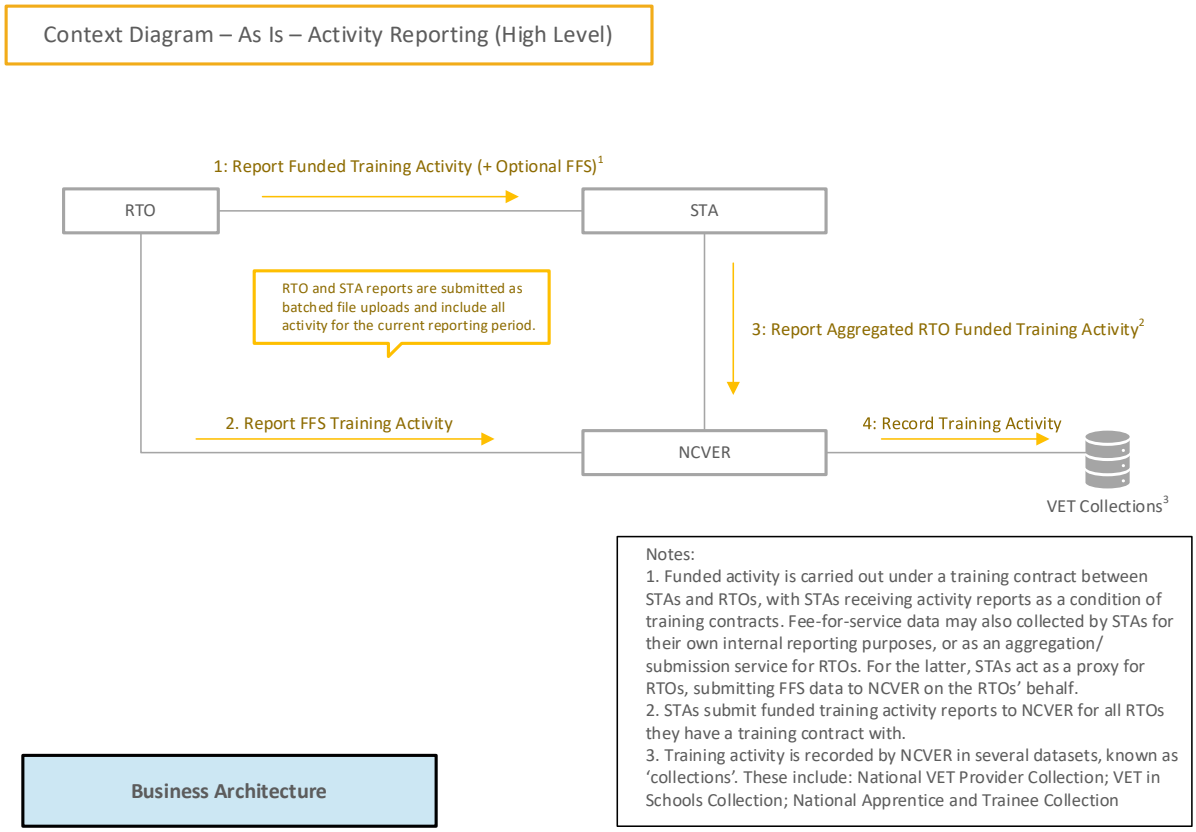


Figure 21: Business Context Diagram – As Is (B:5)

9.2.2 As Is: Funded

The following collaboration diagram shows the as-is funded flow in more detail. Activity reports for funded training activity are generated transmitted as AVETMISS files by RTOs (noting an exception for some states, where the STA provides a user interface to enter data for micro-RTOs with no SMS – in this case the STA generates files from entered data). These files are firstly validated by RTOs using the NCVER validation portal to ensure that there are no field or inter-file relationship errors, prior to transmission to the relevant STA. STAs have various mechanisms to receive these files, ranging from specially developed web portals through to simply attaching files to emails.

The STAs process the received files through their own systems, using the received training data for internal business reasons, such as to ensure compliance with existing contracts and to authorise payments to RTOs. STAs also collate training activity data from multiple RTOs into a single AVETMISS-formatted consolidated submission to NCVER.

The STA consolidated AVETMISS files are uploaded and validated using the NCVER validation portal to ensure that there are no field or inter-file relationship errors, prior to eventual 'finalisation' when the data is confirmed as being fully valid. Finalisation of a collection submission triggers the NCVER systems to import and process the submitted AVETMISS data into the various collection databases.

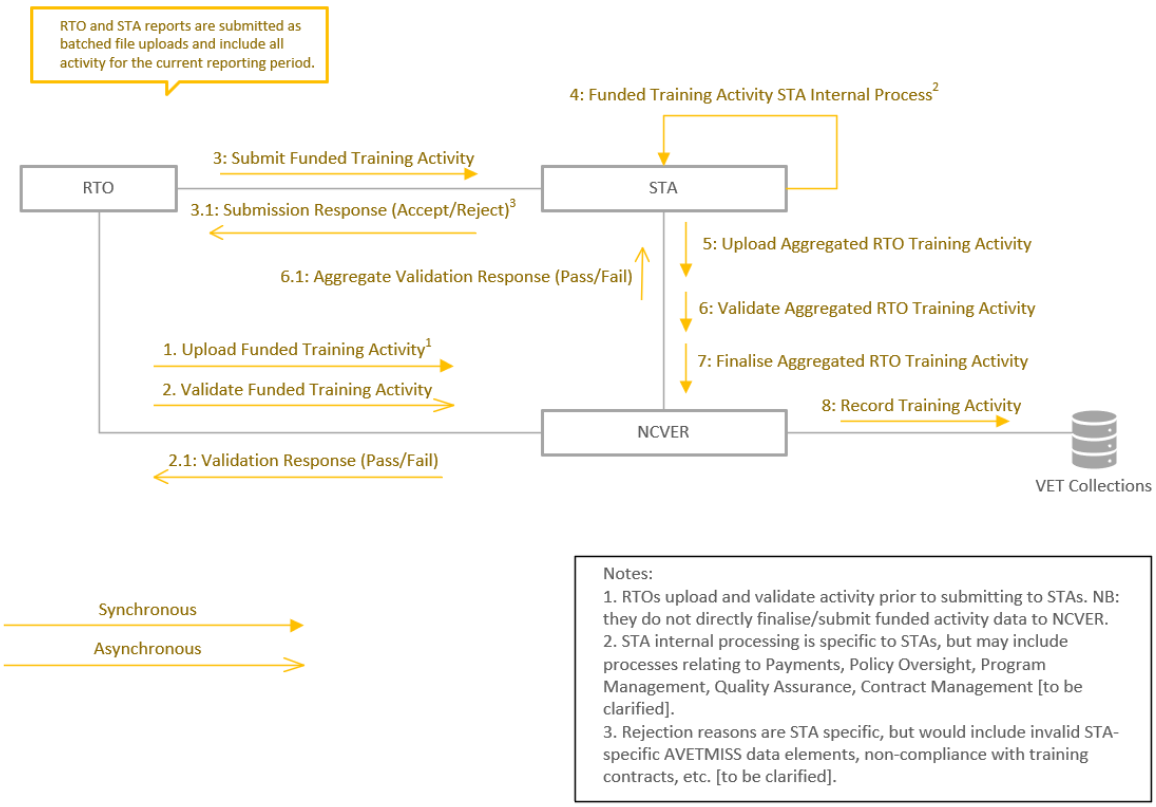


Figure 22: As Is: Funded context (B:6)

9.2.3 As Is: Fee-for-service

The following collaboration diagram shows the as-is FFS flow in more detail. Activity reports for FFS training activity are generated transmitted as AVETMISS files by RTOs. These files are firstly uploaded and validated by RTOs using the NCVER validation portal to ensure that there are no field or inter-file relationship errors, prior to eventual 'finalisation' when the data is confirmed as being fully valid. Finalisation of a collection submission triggers the NCVER systems to import and process the submission data into the various collection databases.

The STAs process the received files through their own systems, using the received training data for internal business reasons, such as to ensure compliance with existing contracts and to authorise payments to RTOs. STAs also collate training activity data from multiple RTOs into a single AVETMISS-formatted submission to NCVER.

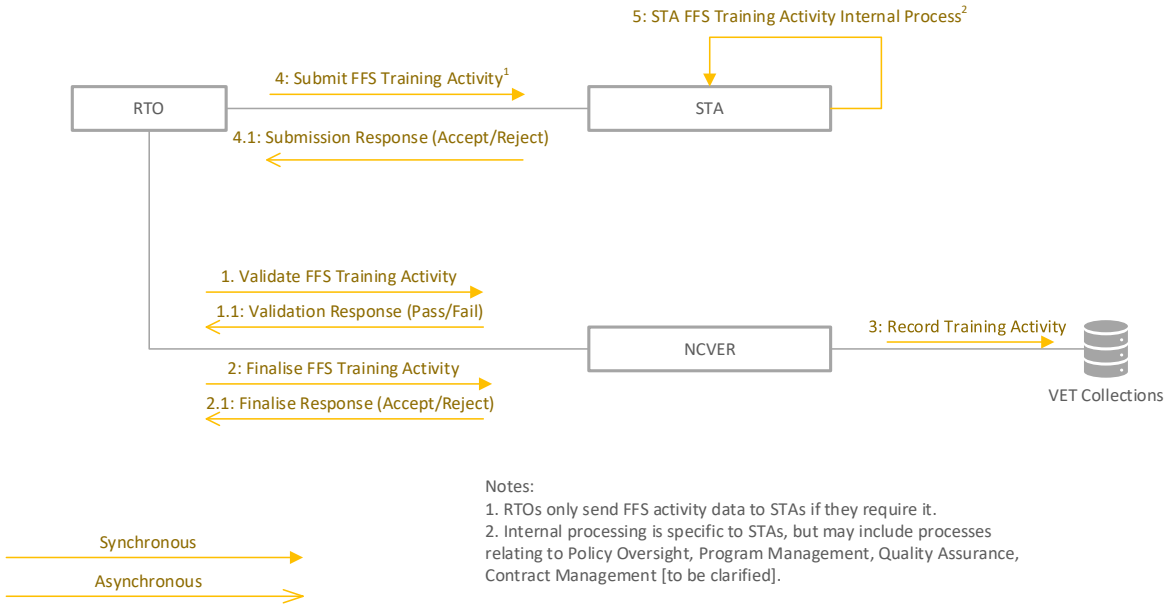


Figure 23: As-Is: Fee for service context (B:7)

9.3 Business Context – To Be

The intent of the STARS system is manyfold, including improving the quality and timeliness of training activity reporting, to provide a high-speed central 'clearing-house' for VET data and to provide an extensible, future-ready platform for future reporting and operational requirements of stakeholders involved with VET sector data.

The current as-is processes, systems and information standards are primarily oriented to support reporting requirements with some support for operational requirements. STARS will continue to support reporting requirements, including future requirements. It will also more fully support operational requirements, particularly for STAs and the Commonwealth, whilst reducing the burden on RTOs.

To this end, the first release of STARS is to simplify the current reporting pathways and associated business processes for RTOs and STAs, utilising the new VET Information Standard (VET-IS).

Rather than RTOs requiring managing and maintain multiple business processes for submitting data through multiple pathways to meet contractual, compliance and regulatory requirements, they will be able to submit all VET data through one pathway via STARS, including for both funded and fee for service activity. The aim to simplify and reduce the number of business processes.

STARS will also support a scenario where RTOs continue to submit VET data to STAs, with STAs then processing the data within their own systems, prior to then passing on batched RTO VET data to STARS.

VET data will be able to be submitted any time.

STARS will validate and route data through to STAs for their specific validation, compliance, contract management and regulatory requirements. Again, STARS will enable STAs to simplify and reduce the number of their business processes relating to management of VET data.

Finally, STARS will become a source of truth for VET data, facilitating VET data consumption and reporting, including for current and future NCVER VET collections.

In general, overall VET sector regulatory, compliance and contractual obligations will remain, (noting that RTOs may in future, for example, have contracts modified to allow for the use of the STARS mechanisms.)

9.3.1 STARS Business Context Diagram

The following 'to-be' diagram shows the broad interactions between STARS and external systems and stakeholders. Note that the diagram does not show all detail regarding, for example, data quality processes, the transmission and sequence of messages between systems and any escalation or error reporting pathways – these are covered elsewhere in this document and the Solution Design.

Training Organisations will submit VET data into STARS, through two main pathways, either through system-to-system interfaces, or via a to-be-developed web portal.

STA-specific VET data submitted by RTOs into STARS, such as funded activity, is routed through to STAs automatically via system-to-system interfaces. When the data has passed STA validation and quality checks, they release it for further processing into a national database – the Master Database.

The Master Database contain into two broad 'partitions':

- STA. The STA partition will contain 'silver' level data (passed data quality and business rules), to retain and support STA operational-level information and reporting. This includes both national and state-specific data elements defined in the VET-IS logical schema.
- National. The National partition will contain 'gold' level data (passed data quality and business rules, de-duplicated, enriched with national classification data), held in VET-IS logical schema format, minus any STA operational-level information. The National partition will be used for national reporting and as a source of truth for national VET data, including student transcripts.

STARS will require a constant feed of reference data, such as course information; this will be supplied through system-to-system interfaces with TGA (via NCVER) and others.

Since STARS also collects VET data for student training outcomes, it will also provide transcript information for the USI.

Finally, STARS will provide information to NCVER for population of the various VET collection data bases and downstream reporting and analytics requirements.

Note, the following diagram is not intended to be a comprehensive view of all integrations or external integrations but provides high level context for the STARS solution.

STARS Context Diagram

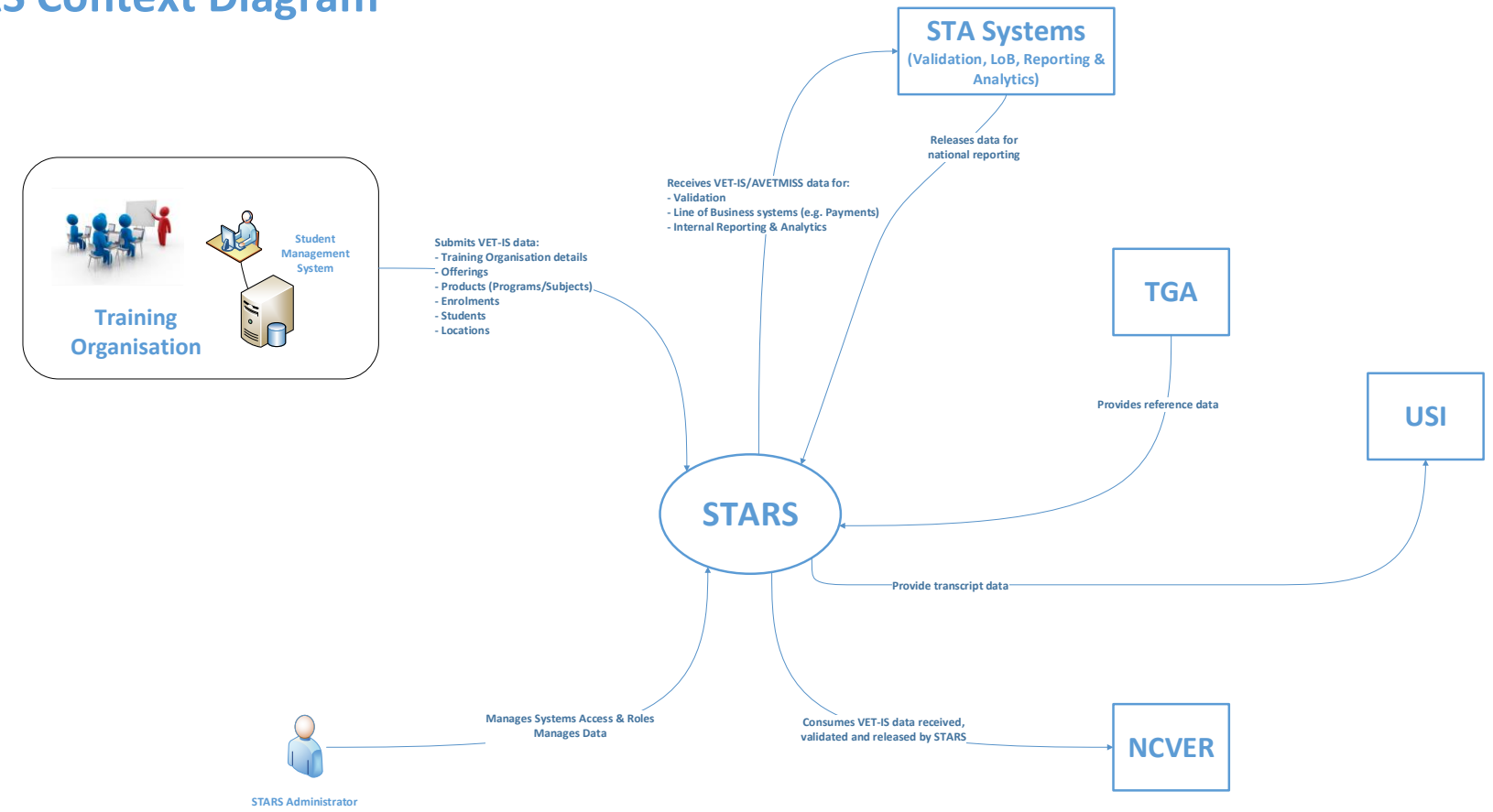


Figure 24: STARS Context Diagram – External Actors

9.3.2 System Context and Information Flows – To Be

This section presents high-level context diagrams and associated information flows (i.e., the arrows in the diagrams.)

The two main scenarios are show, i.e., RTOs submitting to STARS when hand off to STAs, and RTOs submitting to STAs with STAs submitting collated and batched RTO data to STARS.

Note that these context diagrams don't show specific communication mechanisms (i.e., direct entry, vs SMS SDK or API vs Data Share, etc) as they do not affect the business processes.

9.3.2.1 Context Diagrams

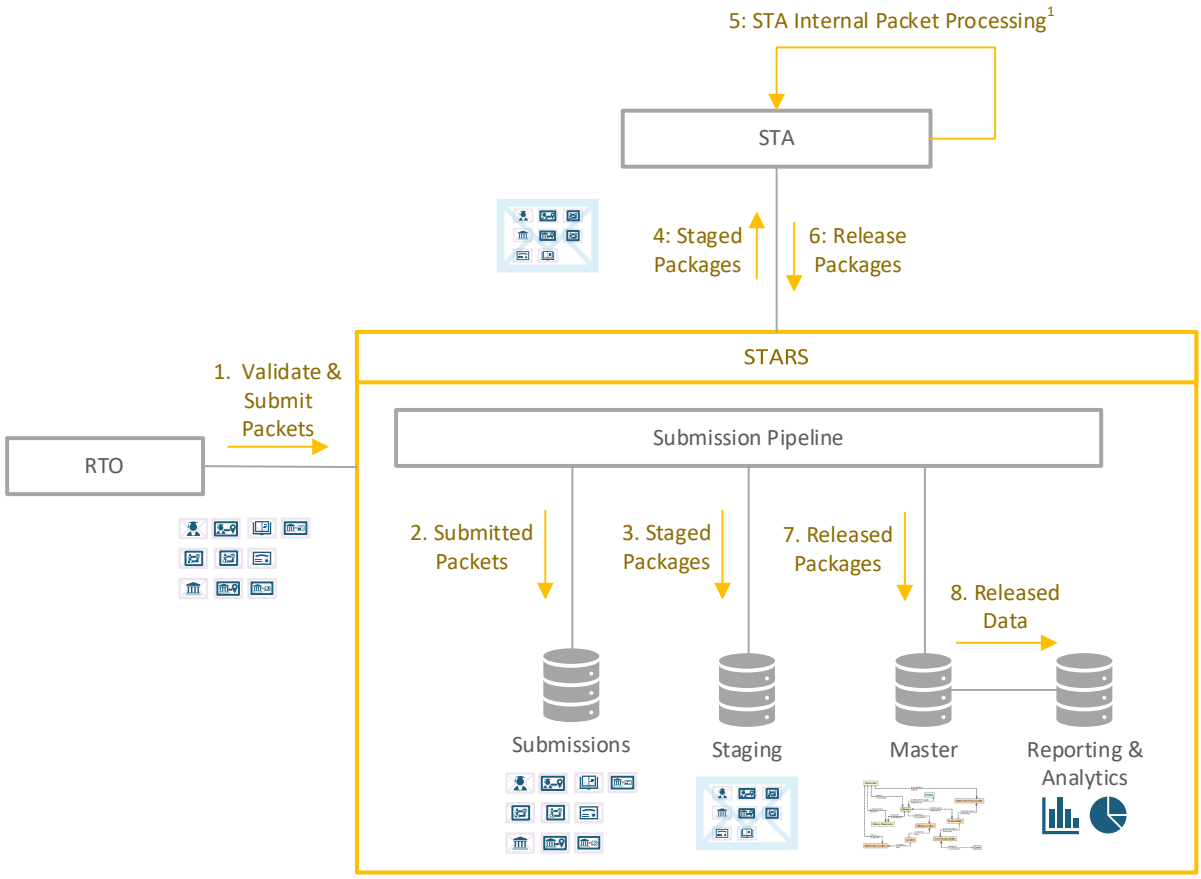


Figure 25: STARS Context & Information Flows – RTO:STARS:STA Scenario (B:9)

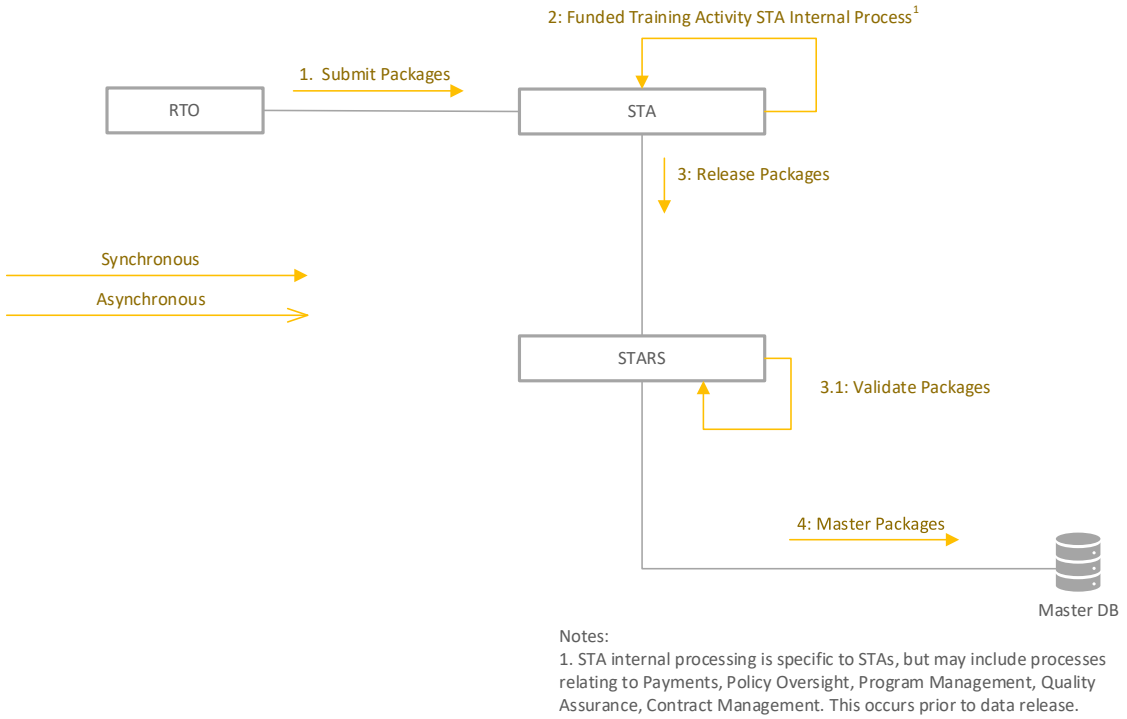


Figure 26: STARS Context & Information Flows – RTO:STA:STARS Scenario (B:10)

9.3.2.2 Information Flows

Continuous Submission:

In counter point to the as-is processes, RTO reports are no longer performed via batch uploads that include all activity and supporting contextual information for the current reporting period (e.g., student details along with RTO details, locations, etc.)

Instead, VET information is submitted directly to STARS as VET Information Standard Packets, at any time.

Alternatively, RTOs may submit VET data to STAs who will then process and submit the data to STARS.

Packet Submission and Validation:

Information Packets are submitted directly from RTO SMS or STA systems to STARS, using modern API technology. Packets may be submitted individually or as collections (Packages.)

Packets are validated automatically against data quality and business rules upon receipt by STARS.

There are of course dependencies between information in the various Packet types, as well as with information already held in STARS and STA systems. These dependencies are checked and enforced by the STARS rules as well. Validation results (e.g., pass/fail/warnings/info, with details) are returned to the RTOs automatically for RTO attention and correction within their SMSs

Submitted Packets in their original form, along with the details of who they were submitted by and when, are held indefinitely in the Submissions data store for traceability and reporting.

Package Staging (RTO:STARS:STA scenario only):

The Staging data store holds information about Packages for funded training while they are being validated and then released by STAs.

When required (e.g., for funded training) Packets are also automatically validated by STA systems using STA internal data quality and business rules. Validation results (e.g., warnings/pass/fail) are returned to STARS and then returned to source automatically. STAs may also utilise these Packets for internal business processes (e.g., to support RTO payment cycles.)

Package Release:

Packets not requiring STA validation nor STA release (e.g., FFS training) are released automatically to the Master Database. Otherwise, STAs review received VET data packages during staging and when satisfied, release the packages for mastering.

Released Packages, having passed all data quality and business rules, are transformed into database records conforming to the VET Information Standard data schema, in readiness for mastering.

Package Mastering:

The Master Database after validation stores all VET data, including history and lineage information. Data is held using the VET Information Standard data schema. The records are inserted into the Master Database, both into the STA (silver) and National (gold) partitions using MDM processes.

R1 of the VET Information Packets and Schema is presently defined only to capture information for the Total VET Activity collection (and associated Stakeholders). Future releases will include information for the other collections.

Notes:

1. STA internal processing is specific to STAs, but may include processes relating to Payments, Policy Oversight, Program Management, Quality Assurance, and Contract Management. This may occur prior to or after data release.
2. In the case of RTOs submitting via STAs, STAs may transform RTO data to remove any STA operational data or map STA-specific categorisation field values to national values. Data from multiple RTOs will be collated and batch submitted to STARS.

9.4 Submission Pipelines

STARS will implement business process pipelines that cater for two basic scenarios: submission of government funded/subsidised training activity data and submission of fee-for-service training activity.

Note that the following descriptions refer to data packets and packages (of data packets), referring to VET-IS Data Packets (e.g., Student, Training Organisation, Enrolment, Subject)[13].

[13] In this context VET-IS packets and packages are a logical construct, analogous to 'records' and 'files' in older ICT terminology. The difference is that packages must only contain packets that relate to each other and must contain a complete set of information. E.g., a package containing an Enrolment packet must also contain the details of the student enrolled in a Student packet, along with Delivery Location, Subject, Offering and Training Organisation packets.

9.4.1 Funded Training Activity Pipeline

RTOs are usually required to regularly report funded training activity data to State Training Authorities (STA) for which RTOs have a funded training contract. In turn, reported activity is collated and reported by STAs (currently to NCVER in AVETMISS format.) STAs use this data for internal compliance and business processes, including payment processing.

STARS will support this scenario conceptually through the following value chain pipeline:

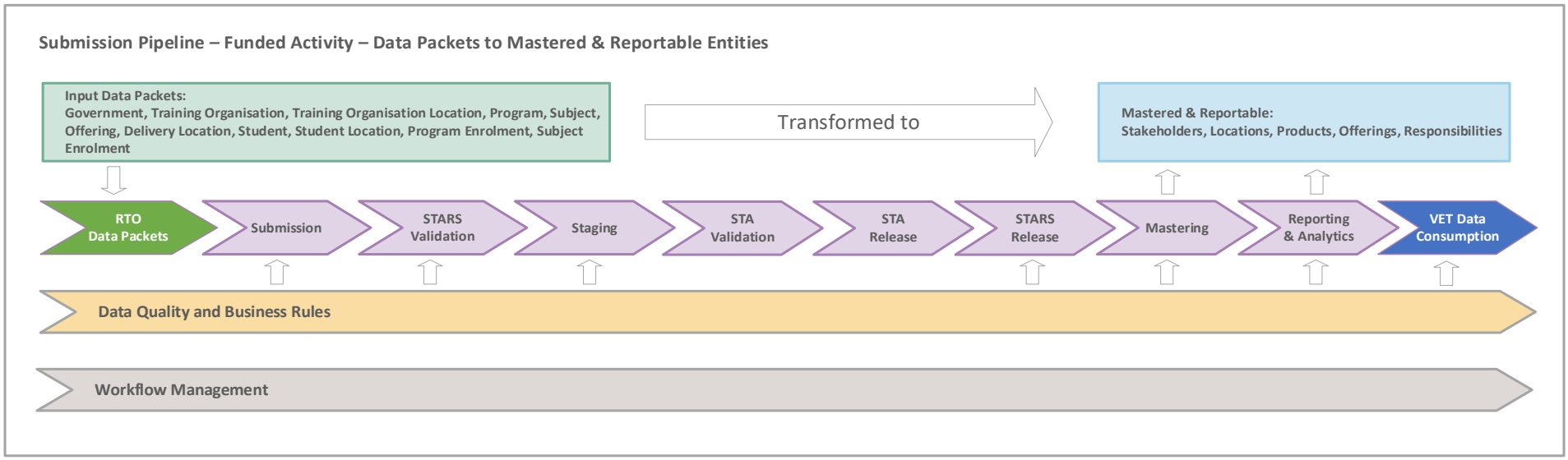


Figure 27: STARS Submission Pipeline – Funded Activity (B:3)

RTOs submit VET-IS data packets into STARS. Packets are validated against both data quality and business rules upon entry to the system. Packets are packaged appropriately (e.g., Enrolment packets along with the complete set of context packets), staged and then passed through to STAs for both validation against STA internal data quality and business rules and (assuming they are accepted) processing them in their internal Line-of-Business (LoB) systems. STAs will then release valid packages from the STARS staging area (possibly automatically with exception rules), from which packages are then mastered. Mastering involves a final set of data quality and business rule processing, de-duplication activities and creation of 'golden-records' representing the various entities in the system, e.g., student records.

Once mastered, data will be loaded into a variety of yet to be specified reporting and analytics databases.

9.4.2 Fee for Service Training Activity Pipeline

RTOs are usually required to report fee-for-service training activity for national regulatory and statistical purposes.

The Fee for Service Pipeline is like the Funded Pipeline; the difference is that there is no involvement with STA systems for either validation or STA internal business processes.

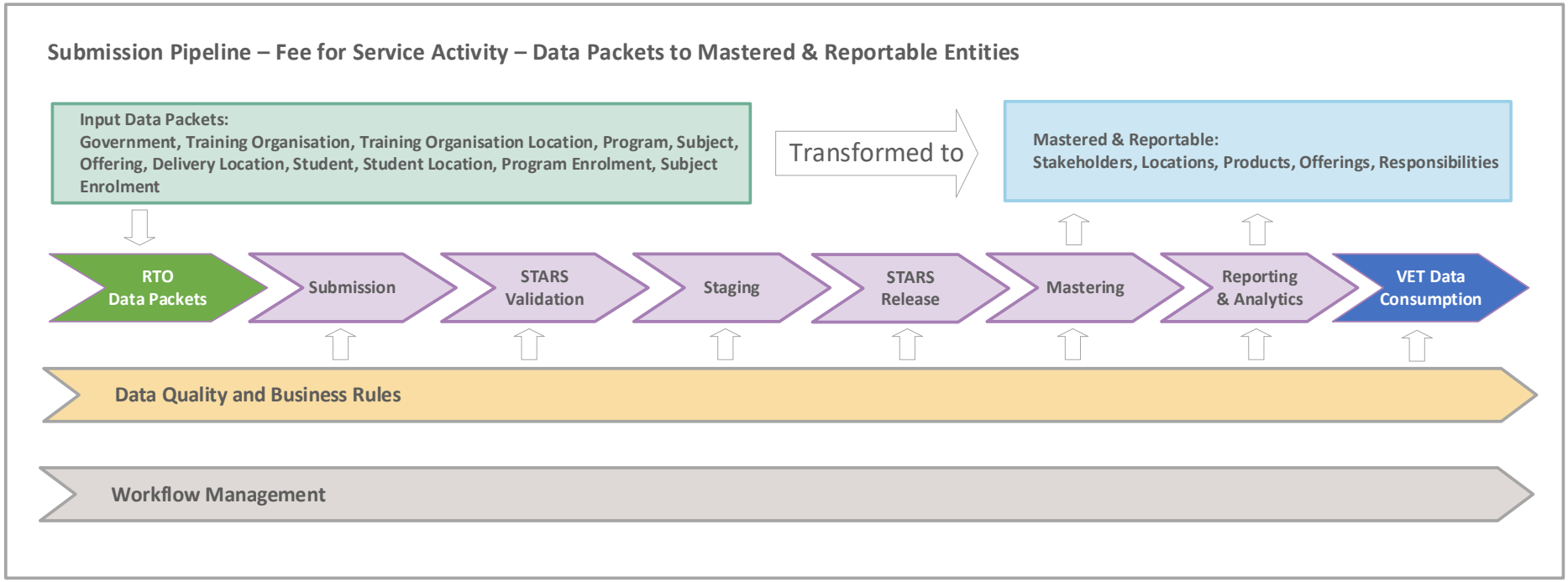


Figure 28: STARS Submission Pipeline – Fee for Service Activity (B:3)

Once validated, packages are staged, released and mastered automatically, with potential to halt or redirect packages at the staging and release gates by exception, utilising relevant data quality or business rules.

9.5 Business Processes – To Be

This section describes the main process to be followed by STARS, namely, the submission to mastering & reporting process underlying the submission pipelines described in section 9.4. As further described in section 10 'Software Architecture and Design', STARS will utilise a service-oriented architectural approach, utilising messaging and event-driven paradigms.

This will be reflected in the following to-be process descriptions, where the major conceptual services are represented in BPMN swim lanes, along with BPMN messages and events. Please refer to Appendix C 'BPMN Notation' for an overview of the symbols used below.

Please refer to the Visio document: Attachment 2 – BPMN Diagrams – D.0.1.vsdx to view the full size BPMN diagrams, shown below.

9.5.1 Notes

1. A general principle of STARS is that data issues are corrected at the source, i.e., RTOs. The process diagram in Figure 29: High Level Process – Submission through to Mastering, Reporting and Analytics utilises BPMN Link event notation ( ) to present a simplified presentation of the process flow when there are issues with submission data. In most instances, data issues encountered at any stage will result in process control reverting to the RTO swim lane, effected by the 'Rejected submission' BPMN link event. A key objective is to ensure as much validation occurs immediately upon submission as possible, reporting all validation results at once, and that any opportunity for later processes to raise additional validation errors is minimised. However, it may not be possible to undertake all checks immediately upon submission, so the design allows for rejection to occur at later stages.

2. STARS will provide mechanisms for data provenance and audit for all information received. An important element to support this is the concept of correlation. Correlation metadata will be recorded at every stage of the process, allowing STARS to trace and record the path of data throughout the entire system, including from which RTO and at what time any data was submitted. This is denoted in the BPMN diagram in tasks that include 'Correlate' in their names.

3. Most services and actors within the to-be process will communicate and coordinate through messages and events. This is reflected through message flows ( ) and message events ( ) in the BPMN diagrams below.

9.5.2 High Level Process – Submission through to Mastering – RTO:STARS:STA Scenario

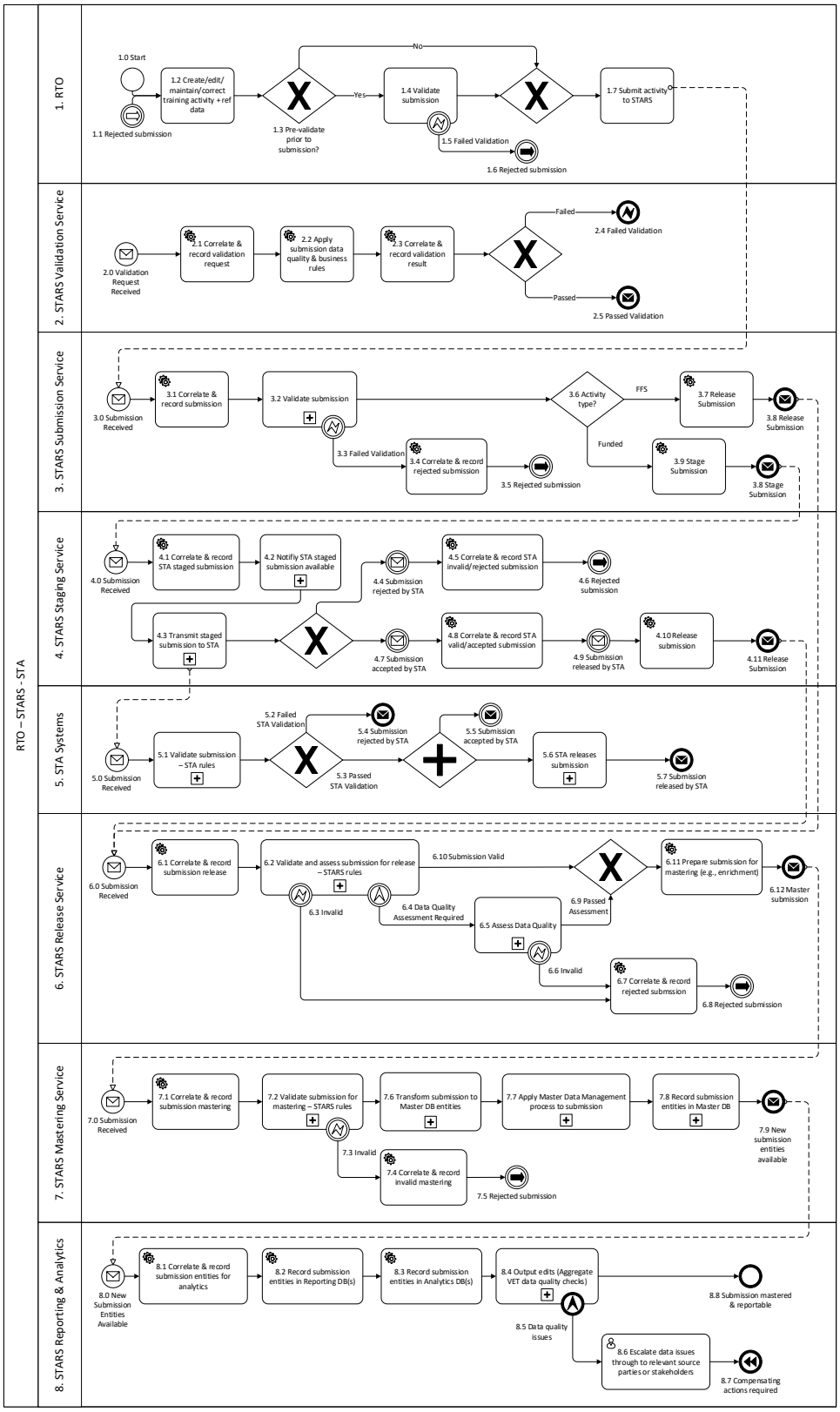


Figure 29: High Level Process – Submission through to Mastering, Reporting and Analytics – RTO:STARS:STA Scenario (C:3)

9.5.2.1 Process Summary

This process represents the scenario whereby RTOs submit all their activity data through STARS, with funded data being routed through to STAs as appropriate.

VET activity information, centred around Enrolments, is submitted directly to STARS by RTOs as VET Information Standard Packets, at any time. Submissions are validated, if necessary, handed on to STAs for validation and release and then mastered into the Master Database. Finally, mastered information is made available for reporting and analytics. The process within STARS is mostly effected via automated services, namely Submission, Validation, Staging, Release, Mastering and Reporting & Analytics.

9.5.2.2 Process description

Working from the top of Figure 29: High Level Process – Submission through to Mastering, Reporting and Analytics:

9.5.2.2.1 **RTO Operations and Submission**

RTOs (1.0) maintain information within their SMS (or the Provider Portal), updating student and enrolment information (1.2). At any time, either explicitly via a user action or via an automated/triggered processes within an SMS system, RTOs can validate information prior to formally submitting it to STARS (1.3, 1.4). If validation data quality or business rules aren't passed, RTOs will have the opportunity to correct information (1.5, 1.6, 1.2).

When information is ready, it is formally submitted to STARS (1.7) with no further involvement other than to correct data should issues be identified further along the process. This is reflected in the 'Rejected submission' link (1.1).

The information submitted to STARS at this stage consists of VET-IS packages containing complete information (as VET-IS packets) based on Enrolments, including the associated Training Organisation, the Student, the Delivery Location and the Enrolment details.

9.5.2.2.2 **Validation**

The Validation Service is utilised by both RTOs when pre-validating information prior to submission as well as other STARS services during the submission process. For brevity, the BPMN diagram shows the entry point to Validation as a 'Validation Request Received' message (2.0), noting that validation requests are received from multiple parties and services. Validation results are passed back to the requestor as either 'Failed Validation' (2.4) or 'Passed Validation' (2.5) messages (passed validation may include warnings and information messages).

The Validation Service records and correlates all requests (2.1) for later reporting and auditing. A submission is validated according to data quality and business rules applicable for the context in which the validation request is received (5.2); for example, a submission request from an RTO is validated against a set of National data quality rules, a request from the Release services is validated against release-level data quality and business rules.

In any instance, validation outcomes are correlated and recorded (2.3), with the results passed back to the requester, in the form of failed/passed validation messages (2.4, 2.5).

9.5.2.2.3 Submission

The Submission service has the role of receiving information from RTOs, correlating and recording the information for later reporting and audit, validating information and then passing information on to the Staging Service or Release Service. Invalid submissions are referred to the RTO for correction.

Submissions containing complete Enrolment information are received from RTOs (3.0). The submission request and the submission data are correlated and recorded in the Submission Data Store (3.1). Data quality and business rules are applied (3.2) via validation request to the Validation Service (STARS Validation Service above).

If the submission fails validation (3.3), it is rejected, with details of the rejected submission recorded in the Submission Data Store (3.4), with control passing back to the RTO for their attention (3.5).

After successful validation, a distinction is made between funded and fee-for-service (FFS) activity, based on defined rules (3.6). FFS data is released to mastering (3.7) via a message to the Release Service (3.8). Funded activity is forwarded on via a message to the Staging Service (3.8), for further attention by STAs.

9.5.2.2.4 Staging, STA Validation and STA Release

Funded submissions will often require validation and then formal release into the national data sets by STAs. The Staging Service manages this requirement.

Valid submissions, containing complete Enrolment information (4.0), are correlated and recorded in the Staging Data Store (4.1). STAs are notified that a new submission is available (4.2). At some interval later the submission is transmitted to the STA (4.3).

Upon receipt of submission data (5.0), validations utilising internal STA data quality and business rules are performed within STAs' own business processes and systems (5.1). STAs inform STARS if a submission has failed validation (5.2) through a message (5.4). When received (4.4) the details of the failed/rejected submission are correlated and recorded (4.5). The submission is then rejected with notification passing to the RTO for their attention and correction (4.6).

Submissions that have passed validation by STAs (5.3) are notified to STARS, again by a message (5.5). Upon receipt (4.7) STARS correlates and records that the STA has accepted the submission. STARS then waits for the STA to release the submission (which may be automated, depending on configured rules) (4.9).

At some time later, the STA releases the submission (5.6), notifying STARS via another message (5.7). Upon receipt (4.9) STARS initiates the release of the submission through a message to the Release Service (4.11).

9.5.2.2.5 Release

Valid submissions for FFS or for funded activity released by STAs are vetted and then prepared for release into national data sets (i.e., into the Master Data Store) by the Release Service. The service applies release-level data quality and business rules, with the capability of raising escalations for submission data that exceeds aggregate thresholds. The service may also enrich submissions with additional data required for reporting and analytics.

The Release Service receives submissions either directly from the Submission Service or the Staging Service (6.0). Submission release requests are correlated and recorded (6.1).

Submissions are validated and assessed against any relevant release-stage data quality and business rules (6.2). Invalid submissions (6.3) are correlated and recorded (6.7) prior to being rejected and referred to the RTO (6.8).

The process also checks submissions against any aggregate submission warning thresholds or for outliers from acceptable trends. If required, an escalation is raised (6.4) for STARS administrators to assess possible aggregate or trend errors (6.5). If the assessment rejects the submission (6.6) it also correlated and recorded (6.7) prior to being rejected and referred to the RTO (6.8).

Valid submissions (6.10) and submissions that have passed assessment (6.9) are prepared for mastering, including any relevant data enrichment (e.g., adding additional categorisation data for downstream analytics) (6.11). The Release Service then initiates mastering of the submission through a message to the Mastering Service (6.12).

9.5.2.2.6 Mastering

The Mastering Service transforms submissions containing information in VET-IS Data Packet Schema entities (e.g., Student, Enrolment, Subject) into VET-IS Logical Schema entities (e.g., Stakeholder, Responsibility, Product). When transformed, the information is passed through Master Data Management processes, for example, de-duplication, enrichment and preparation of 'golden' records.

Released submission requests (7.0) for mastering are correlated and recorded (7.1). The submission is further validated against any mastering-level data quality and business rules (7.2). Invalid submissions are recorded and correlated (7.4) prior to being rejected and referred to RTOs for attention (7.5).

Once validated, submissions are transformed from VET-IS Data Packets to VET-IS Logical Schema entities (7.6). The Master Data Management processes are applied to these entities (7.7), prior to recording them appropriately in the Master Database (7.8). Finally, the Mastering Service informs the Reporting & Analytics Service that new submission entities are available via a message (7.9).

9.5.2.2.7 Reporting, Analytics and Aggregate Data Quality Checks

The Reporting & Analytics Service(s) extracts, transforms and loads Master DB entities into relevant reporting and analytics databases, to support downstream data consumption requirements and aggregate data quality checks (a.k.a. output edits). At time of authoring, reporting and analytics requirements have not been provided. Accordingly, this section of the process description is generic and will be refined as STARS reporting, analytics and aggregate data quality check requirements are provided, analysed and documented.

New submission entity events (messages) (8.0) are recorded and correlated (8.1). As appropriate the entities are retrieved from Master DB and recorded in suitable reporting (8.2) and analytics (8.3) databases (requirements TBD). A set of aggregate data quality checks may be performed using these reporting and analytics databases (8.4) (requirements TBD).

Data quality issues may result in an escalation (8.5) to upstream sources, such as RTOs (8.6). In this instance there may be compensating actions required (8.7) – such as having RTOs resubmit their data.

Assuming there are no issues, the process ends with submissions recorded successfully in relevant reporting and analytics databases. At this point submissions are reportable through national data sets (8.8).

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 90 of 1295

9.5.3 High Level Process – Submission through to Mastering – RTO:STA:STARS Scenario

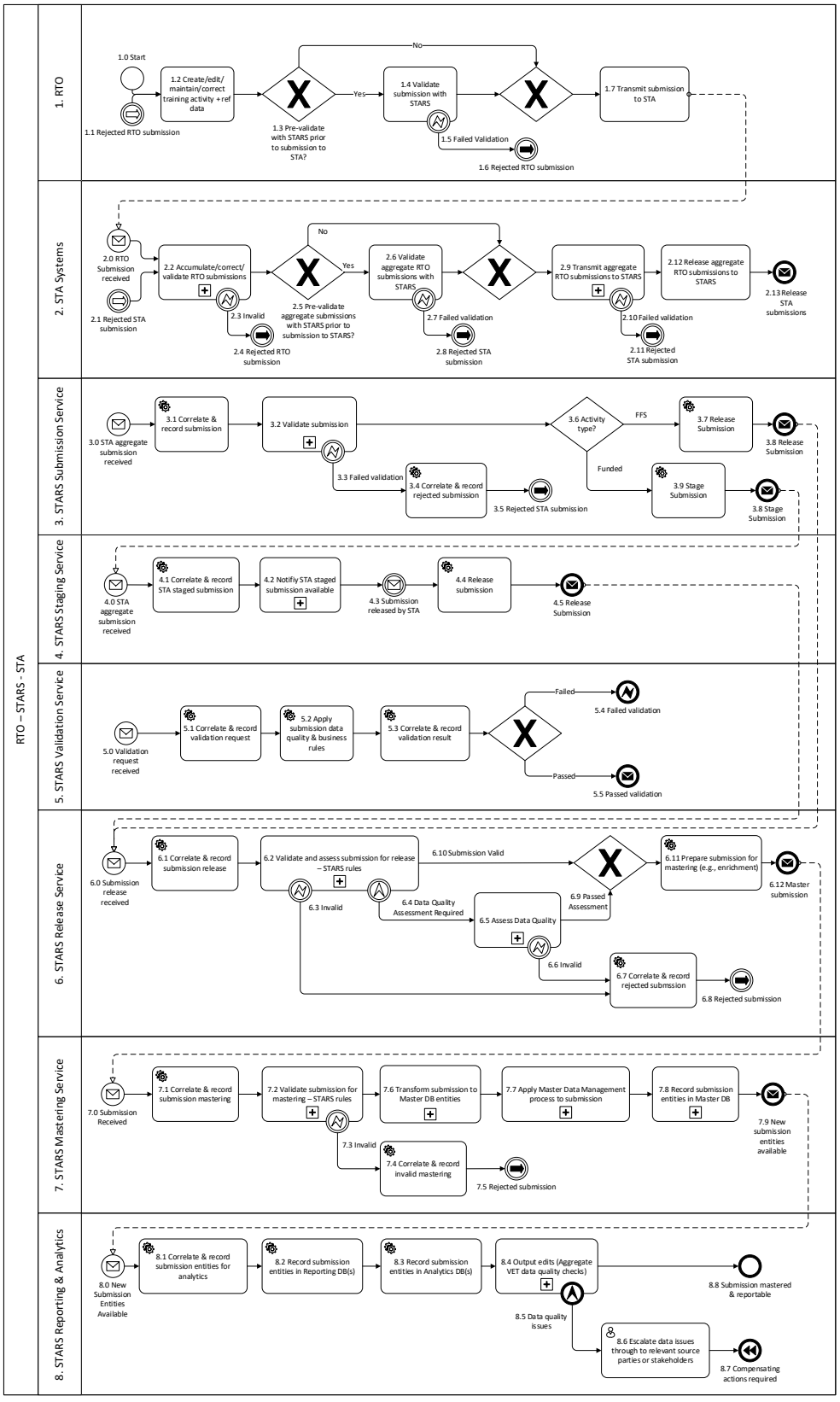


Figure 30: High Level Process – Submission through to Mastering, Reporting and Analytics – RTO:STA:STARS Scenario (C:4)

9.5.3.1 Process Summary

This process represents the scenario whereby RTOs submit activity data through STAs and thence to STARS.

VET activity information is submitted directly to STAs by RTOs in the data format and on the schedules agreed through their contractual arrangements. Submissions are validated and processed via STA systems, with any issues referred to RTOs.

STAs gather and collate data on behalf of RTOs. Data is submitted and then released to STARS in VET-IS Data Packet schema and formats. STARS validates STA submissions and then masters into the Master Database. Finally, mastered information is made available for reporting and analytics. The process within STARS is mostly effected via automated services, namely Submission, Validation, Staging, Release, Mastering and Reporting & Analytics.

9.5.3.2 Process description

Working from the top of Figure 30: High Level Process – Submission through to Mastering, Reporting and Analytics – RTO:STA:STARS Scenario:

9.5.3.2.1 **RTO Operations and Submission**

RTOs (1.0) maintain information within their SMS, for example, updating student and enrolment information (1.2). If required, RTOs may validate submissions using the STARS validation service prior to transmission to STAs (1.3, 1.4) with the opportunity to correct any errors (1.5, 1.6, 1.1). Data is submitted to STAs in the format and on the schedules agreed (1.7).

9.5.3.2.2 **STA Systems**

STAs receive data from RTOs (2.0), accumulating, validating and processing through their own internal systems and business processes (2.2). Invalid submissions are referred to the originating RTO (2.3, 2.4)).

If required, STAs may validate submissions using the STARS validation service prior to formal submission to STARS (2.5, 2.6) with the opportunity to correct any errors (2.6, 2.7, 2.8). When ready, STAs submit data collected from multiple RTOs to STARS (2.9). STARS will validate submissions on receipt, STAs are informed of any validation errors (2.10) for attention and correction (2.11, 2.1).

Valid STA submissions are then released by STAs (2.12) to STARS for use in national data sets via a release message (2.13).

9.5.3.2.3 **Validation**

The Validation Service may be utilised by both RTOs and STAs when pre-validating information prior to submission, as well as other STARS services during the submission process. For brevity, the BPMN diagram shows the entry point to Validation as a 'Validation Request Received' message (5.0), noting that validation requests are received from multiple parties and services. Validation results are passed back to the requestor as either 'Failed Validation' (5.4) or 'Passed Validation' (5.5) messages.

The Validation Service records and correlates all requests (5.1) for later reporting and auditing. A submission is validated according to data quality and business rules applicable for the context in which the validation request is received (5.2); for example, a submission request from an RTO is

validated against a set of National data quality rules, a request from the Release services is validated against release-level data quality and business rules.

In any instance, validation outcomes are correlated and recorded (5.3), with the results passed back to the requester, in the form of failed/passed validation messages (5.4, 5.5).

9.5.3.2.4 Submission

The Submission service has the role of receiving information from STAs, correlating and recording the information for later reporting and audit, validating information and then passing information on to the Staging Service or Release Service. Invalid submissions are referred to the STAs for correction.

Submissions containing complete Enrolment information are received from STAs (3.0). The submission request and the submission data are correlated and recorded in the Submission Data Store (3.1). Data quality and business rules are applied (3.2) via validation request to the Validation Service (see STARS Validation Service above).

If the submission fails validation (3.3), it is rejected, with details of the rejected submission recorded in the Submission Data Store (3.4), with control passing back to the STA for their attention (3.5). The result of validation of a submission (or part thereof) can include Errors, Warnings, and Messages. If no errors are present, the data will proceed in the pipeline, but there may still be Warnings and Messages returned for attention of the STA.

After successful validation, a distinction is made between funded and fee-for-service (FFS) activity, based on the existence/non-existence of contract information within the Enrolment packet (3.6). FFS data is released to mastering (3.7) via a message to the Release Service (3.8). Funded activity is forwarded on via a message to the Staging Service (3.8), for attention by STAs.

9.5.3.2.5 Staging and STA Release

Funded submissions will require validation and then formal release into the national data sets by STAs. The Staging Service manages this requirement. Note that as information is received directly from STAs, there is no need to validate them using STA internal systems – this will have been done prior to submission by the STAs. NOTE: STARS validation will have occurred as STARS is still responsible for data quality.

Valid submissions, containing complete Enrolment information (4.0), are correlated and recorded in the Staging Data Store (4.1). STAs are notified that a new submission is available (4.2) – this is to allow STAs to record a reference to the staged data.

At some time later, the STA releases the submission (2.12), notifying STARS via another message (2.13). Upon receipt (4.9) STARS initiates the release of the submission through a message to the Release Service (4.10, 4.11).

9.5.3.2.6 Release

Valid submissions for FFS or for funded activity released by STAs are vetted and then prepared for release into national data sets (i.e., into the Master Data Store) by the Release Service. The service applies release-level data quality and business rules, with the capability of raising escalations for submission data that exceeds aggregate thresholds. The service may also enrich submissions with additional data required for reporting and analytics.

The Release Service receives submissions either directly from the Submission Service or the Staging Service (6.0). Submission release requests are correlated and recorded (6.1).

Submissions are validated and assessed against any relevant release-stage data quality and business rules (6.2). Invalid submissions (6.3) are correlated and recorded (6.7) prior to being rejected and referred to the RTO (6.8).

The process also checks submissions against any aggregate submission warning thresholds or for outliers from acceptable trends. If required, an escalation is raised (6.4) for STARS administrators to assess possible aggregate or trend errors (6.5). If the assessment rejects the submission (6.6) it is also correlated and recorded (6.7) prior to being rejected and referred to the RTO (or STA if they are the source of the data) (6.8).

Valid submissions (6.10) and submissions that have passed assessment (6.9) are prepared for mastering, including any relevant data enrichment (e.g., adding additional categorisation data for downstream analytics) (6.11). The Release Service then initiates mastering of the submission through a message to the Mastering Service (6.12).

9.5.3.2.7 Mastering

The Mastering Service transforms submissions containing information in VET-IS Data Packet Schema entities (e.g., Student, Enrolment, Subject) into VET-IS Logical Schema entities (e.g., Stakeholder, Responsibility, Product). When transformed, the information is passed through Master Data Management processes, for example, de-duplication, enrichment and preparation of ‘golden’ records.

Released submission requests (7.0) for mastering are correlated and recorded (7.1). The submission is further validated against any mastering-level data quality and business rules (7.2). Invalid submissions are recorded and correlated (7.4) prior to being rejected and referred to RTOs or STAs for attention (7.5).

Once validated, submissions are transformed from VET-IS Data Packets to VET-IS Logical Schema entities (7.6). The Master Data Management processes are applied to these entities (7.7), prior to recording them appropriately in the Master Database (7.8). Finally, the Mastering Service informs the Reporting & Analytics Service that new submission entities are available via a message (7.9).

9.5.3.2.8 Reporting, Analytics and Aggregate Data Quality Checks

The Reporting & Analytics Service(s) extracts, transforms and loads Master DB entities into relevant reporting and analytics databases, to support downstream data consumption requirements and aggregate data quality checks (a.k.a. output edits). At time of authoring, reporting and analytics requirements have not been provided. Accordingly, this section of the process description is generic and will be refined as STARS reporting, analytics and aggregate data quality check requirements are provided, analysed and documented.

New submission entity events (messages) (8.0) are recorded and correlated (8.1). As appropriate the entities are retrieved from Master DB and recorded in suitable reporting (8.2) and analytics (8.3) databases (requirements TBD). A set of aggregate data quality checks may be performed using these reporting and analytics databases (8.4) (requirements TBD).

Data quality issues my result in an escalation (8.5) to STARS administrators for them to raise data issues with upstream sources, such as RTOs (8.6). In this instance there may be compensating actions required (8.7) – such as having RTOs resubmit their data.

Assuming there are no issues, the process ends with submissions recorded successfully in the Master Database and any relevant reporting and analytics databases. At this point submissions are reportable through national data sets (8.8).

9.6 Workflow and Rules Management Approach

9.6.1 Workflows

Implicit in many business and functional requirements in the STARS Solution Design Document is the ability for STARS to support different submission workflows and business rules that are particular to the needs of operational stakeholders, e.g., STAs and NCVER.

These specific workflows will be run within the context of the major business processes described within section 9.5 'Business Processes – To Be'. To place this in context, many of the tasks and collapsed sub-processes across many of the process swim lanes will be orchestrated and managed by automated workflows.

STARS will deliver this capability through a combination of a flexible workflow management solution based on Azure Logic Apps[14], coupled with a flexible Rules Engine.

In essence, workflows are responsible for orchestrating the flow of data through STARS, but they hand off evaluating the data (e.g., against data quality rules) to a rules engine that can evaluate many submissions against many rules at high speed.

As a very high-level and simplified example, a workflow within the Submission Service is used to manage a bulk upload by RTOs of VET data in a defined file format. At a (very) high level, the sequence of tasks the workflow runs might be:

1. The workflow 'picks up' the files, hands them off to a transformation function to convert the files to an internal VET-IS representation.
2. The workflow then passes on the VET-IS format data to the rules engine function to check the data against both national and STA rules.
3. The rules engine function evaluates the VET-IS format data quickly, returning the validation results in two categories, the data that passed validation and the data that failed.
4. The workflow receives back the validation results from the rules engine; it sends the passed data on to the Staging Service and the failed data back to the RTO (along with reasons.)

This shown by the following simple Logic App workflow, the step 'Validate VET-IS in Rules Engine' is the point where the workflow hands off the work of validating the data to the high-speed rules engine:

[14] See https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-overview for a good introduction to Logic Apps and their capabilities.

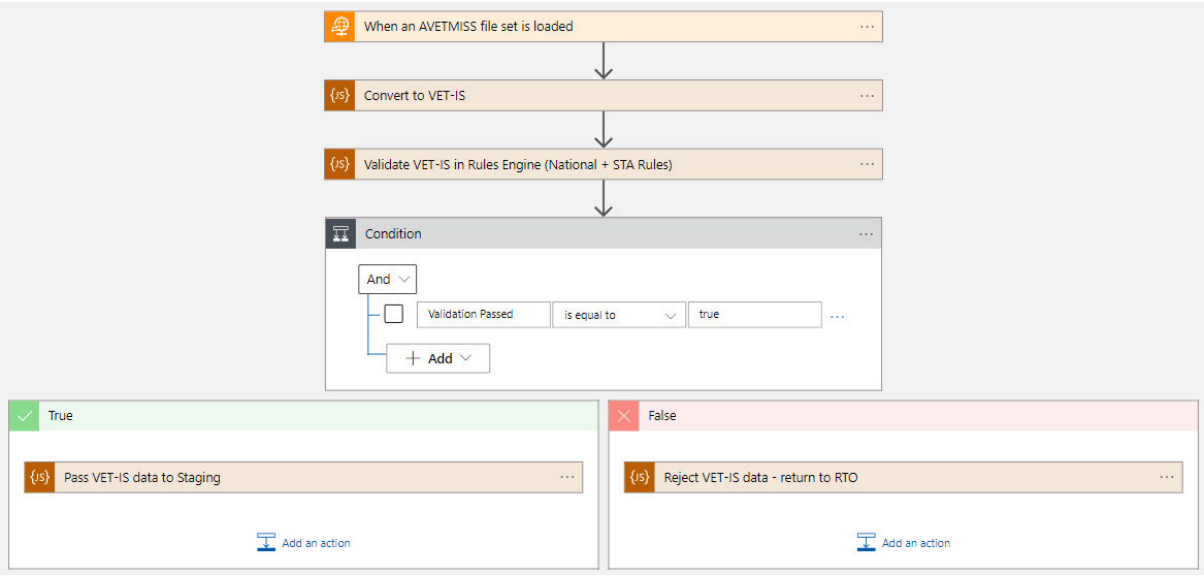


Figure 31: EXAMPLE Simplified Logic App workflow illustrating hand off to Rules Engine
(Please note: the actual workflows implemented in STARS will be much more elaborate.)

During the implementation of STARS, Logic App workflows will be designed and implemented to manage the various scenarios for processing submissions, based on the business rules and requirements of stakeholders.

It is important to note that Logic App workflows can link dynamically to other workflows; this will allow for, say, a high-level workflow managing the Submission Service to hand off control to separate, more specific STA workflows. This will enable STARS to cater for, say, an STA having a special funding program that requires additional data quality and business rules to be evaluated during validation.

At many points below in the Software Architecture and Design sections, services within the Submission Pipeline are controlled and orchestrated with Logic App workflows. This will enable STARS to achieve a great deal of flexibility in adapting to stakeholder processing requirements both during the in the initial development of STARS and when it is operational. This flexibility will also be facilitated by the loosely coupled design of the services, achieved by the underlying message- and event-driven communications architecture.

To give another simple example, building on the scenario above where an STA has established a new funding program. The STA wishes to receive an email notification whenever STARS receives VET data relating to the new program that does not meet some quota criteria, aside from any existing validation rules. In this case, the STA-specific Logic App workflows will be modified to check for the existence of the funding code within received submissions from the STA's RTOs. Applying the quota criteria (e.g., via a script embedded in the workflow) the workflow can detect if it needs to notify the STA. If the criteria is not met, then the workflow will automatically send an email with details of the relevant RTOs and submissions attached. This scenario, and others similar, will not require re-coding or re-factoring of the STARS solution; instead, it is a matter of deploying a new configuration of the relevant Logic Apps.

Logic Apps can connect natively to many Azure Services, including the various data services (e.g., Azure SQL). Custom connectors can also be developed, e.g., to facilitate automatic integration with

STA systems. DXC will work with STAs during the project to evaluate requirements and propose solutions for integrating workflows with STA business processes and systems. This work will commence during Work Package 2.

9.6.2 Rules Engine

The DXC team are currently evaluating appropriate rules engine technology based on the requirements for STARS, with particular emphasis on ensuring that the rules platform management infrastructure can cater for the complexity associated with the differing requirements of STAs and NCVER. The final choice will be made during Work Package 2, once work has completed to establish and analyse the numerous data quality and business rules for the STAs and national reporting, along with the attendant workflows. The rules engine of choice will be able to perform well in a range of scenarios, including a high volume of small validations, as well as validations across large datasets, with appropriate scaling options to facilitate this. An important factor in the decision will also be the manageability of the rules, where delays are minimised between identifying a required change, and enacting that change. It's noted that some changes will require recompilation and deployment of source code, but the chosen tool should minimise the requirement for this, in preference for simple scripting/configuration that can be updated at runtime.

The rules engine technologies are discussed in more detail below in section 10.4 'Rules Engine'.

10 Software Architecture and Design

Please note that many detailed requirements are not yet provided surrounding lower-level STARS capabilities, processes, workflows and rules. Accordingly, the solution design proposes a set of flexible approaches that are most likely able to support detailed requirements as they are created or evolved during the life of the project (subject to the project scope change management process), and beyond.

Please also note that analysis and design work for workflows and rules will commence during Work Package 2 and continue during the project.

10.1 Overview and Purpose

This Software Architecture and Design section addresses the major services that deliver the capabilities and workflows for the STARS solution.

It describes the following:

- The main capabilities that have been identified for delivery by STARS
- The service model developed to deliver the capabilities
- How the service model will automate the high-level STARS business processes
- The approach to flexibly automate and manage yet to be specified data quality and business rules
- The approach to flexibly automate yet to be specified business processes and workflows

STARS will utilise Service Oriented Architecture (SOA) principles for design and implementation, coupled with Master Data Management (MDM) principles for population of the Master Database. The solution will be hosted in Microsoft Azure and will conform to the Microsoft Well-Architected Framework[15], utilising recognised Azure Reference Architectures[16].

Service Technical Architecture and APIs

Please note that the solution service discussions and descriptions in this section are primarily focussed on the roles, capabilities and functionality provided by the services, and the interactions between the services.

The technical implementation details, including the configuration of Azure resources that implement or support the operation of the services, is describe in section 11 'Technical Architecture and Design'. This section describes how requests between services are transported and routed.

As mentioned above, STARS will follow SOA principles and will be implemented as (many) loosely coupled services. Loose coupling will be achieved using, for example, message- and event-driven architectures, primarily using Azure Service Bus and Azure Event Grid, respectively. Message and event payloads conforming to several standardised information JSON-based schemas.

Many services present Application Programming Interfaces (API), typically as HTTP/REST/JSON based endpoints. API designs are presented in a separate document, please refer to Attachment 3 - API

[15] https://docs.microsoft.com/en-us/azure/architecture/framework/

[16] https://docs.microsoft.com/en-us/azure/architecture/

Design D.0.1.docx[17]. API-based services (both internal and external) will be managed through Azure API Management. For external-facing services STARS will utilise Azure Application Gateway as an intermediary between the APIs and their clients; this will allow for a consistent URL scheme across the solution.

10.2 Solution Capabilities

The core function of STARS is to receive VET data submissions as VET-IS Data Packets, and then pass the data through a submission pipeline of validation, staging for STA processing/validation/release (funded data) and then mastering. Mastered submission data is available for downstream VET data and reporting & analytics consumers. Refer to sections 8.2 'Information Model – VET Information Standard (R1)'and 9.4 'Submission Pipelines' for diagrams and descriptions of the information and pipelines to be implemented in STARS.

This Solution Requirements Document identifies and describes required high-level **capabilities** required to deliver the new STARS Business and Information Models, along with the associated requirements, and the main submission processing pipelines.

The capabilities required are presented as collections of high-level services, in line with SOA principles. For example, the submission pipeline capability that incorporates submission, validation, staging, release and mastering capabilities is reflected as several high-level services. These include the Submission Service, Validation Service, Staging Service, Release Service and Mastering Service. Again, please note that these are not implementation-level services, but rather should be considered as a logical grouping of lower-level software services and workflows that are be described in more detail below.

10.2.1 STARS Capabilities

Table 12: STARS Capabilities

Capability	Description
Source Systems & Services	STARS will include capabilities to allow VET-IS data to be sourced from: RTOs Student Management Systems (SMS). STAs internal systems. A web-based portal for micro-RTOs. The functionality of the portal is limited to entering/uploading and maintaining the VET-IS Data Packet entities (i.e., it is not a replacement for a full SMS).
Submission Pipeline Services	STARS will include capabilities to: Receive submitted VET-IS Data Packets and Packages from source systems. Validate received Data Packets and Packages, returning results to source systems. Stage received Packages for validation and then release by STAs.

[17] Please note the API design document will evolve as the solution is implemented. At the time of authoring, the design document lays out the API principles and standards to be followed during implementation.

Capability	Description
	Release Packages along with any supporting enrichment. Master Packages into VET-IS Logical Entities using MDM practices. Provide mastered entities to downstream Reporting & Analytics (requirements to be determined.)
Decision Services	STARS will include capabilities to: Maintain and manage automated data quality and business rules. Apply data quality and business rules to data and within processes and workflows along the Submission Pipeline, including validation of received submissions.
Mastering Services	STARS will include capabilities to: Maintain the Master Database using MDM practices, including de-duplication/duplicate detection, 'golden' record creation.
STA Integration Services	STARS will include capabilities to: Transmit and receive information between STARS and STA systems. Provide where necessary transformations between VET-IS entities and STA-specific data formats, such as AVETMISS 8 (specific requirements to be determined.)
Downstream Data Consumption Services	STARS will include capabilities to: Present USI transcript information via a web service to the USI registry. Provide bulk transfers of mastered data from STARS to STAs (specific requirements to be determined.) Provide reporting & analytics data for downstream data consumers (specific requirements to be determined.) Provide automated ETL pipelines for downstream data consumption by, for example, NCVER systems (specific requirements to be determined.)
System Support Services	STARS will include capabilities to: Manage and provide Reference Data to support decision and enrichment capabilities for data received and processed by the Submission Pipeline. Manage and create Notifications to relevant parties for purposes such as alerting STAs about outstanding submission releases. Master data query services to allow various other services and processes to query for VET-IS Logical Schema entities and their VET-IS Data Packet projections (e.g., as Stakeholders or as Students). (Note: Master data is only created or updated by the Mastering Service.)

Capability	Description
Provider Support Services	STARS will include capabilities to: Allow micro-RTOs to maintain and submit VET-IS data without the use of an SMS. Allow RTOs to view the progress of their submissions. Allow RTOs to receive notifications from STARS. Allow RTOs to request exemptions for submissions that have not passed validation.
System Management Services	STARS will include management services and management user interface capabilities to: Manage the submission pipeline. Manage STARS organisation information & registration (e.g., RTOs). Manage STARS user information & registration (e.g., RTO staff).

10.2.2 STARS Capability Services

The STARS high level services and are grouped and categorised into the following capabilities:

Table 13: STARS Services

Capability	Service/Component	Description
Source Systems & Services	RTO SMS	RTO Student Management Systems manage their data internally, but (may) use the STARS SDK to transmit VET-IS data to the Submission service via the Submission API.
	Provider Portal	The Provider Portal provides the user interface for entry of both slowly changing date (Provider details, locations, products, offerings, students, etc.) and activity data (enrolments, subjects, programs). It communicates with the Provider Service. More details are provided below in Provider Support Services.
	Provider Service	The Provider Service stores and maintains activity-related data. When it receives a request to submit activity, it extracts relevant activity data and formats it into an activity submission request (i.e., JSON payload) for transmission to the Submission Service. It also provides information to help manage submissions, including submission status/progress, submission history, as well as operations to search and view previously submitted students etc. These services support functionality delivered through the Provider Portal or SMS SDK. More details are provided below in Provider Support Services.
	STA Batch RTO Submission	STAs can receive submissions from RTOs. These submissions are validated by STA internal processes and systems. RTO submissions are then submitted to STARS in batches.
Submission Pipeline Services	Submission Service	The Submission Service receives, records and validates all submissions along with the full history of submission events. It is the source of truth for all submissions (valid or not). Valid submissions are passed to the Staging service automatically.

Capability	Service/Component	Description
	Staging Service	The Staging Service maintains and stages validated submission data in readiness for STA validation and release along with the full history of staging events. NB: Staging rules include automatic release rules.
	Release Service	The Release Services manages and records released data submissions along with the full history of release events. It is the source of truth for what has been released by STAs.
	Mastering Service	The Master Service maintains and manages the Master Database, including MDM. It is the source of truth for all Stakeholders, Products, Offerings, Responsibilities, along with a full history of mastering events and data lineage. All master data will be traceable back through release, staging and submission.
	Reporting and Analytics Service(s)	The Reporting and Analytics service manages and maintains the various reporting and analytics databases. It includes ETL pipelines that load the databases. At the time of authoring, reporting and analytics requirements are not yet provided.
Decision Services	Rules Service	The Rules Service stores and then provides rules and rule sets for the main services within STARS along with the full history of rules management events. Rules are managed via the Rules Portal.
	Rules Portal	The Rules Portal manages rules and rule sets stored in the Rules Service.
	Validation Service	The Validation Service validates submissions based on applicable data quality and business rules obtained from the Rules Service. It records all validation requests and results along with the full history of validation events.
Mastering Services	Mastering Service	See Mastering Service description above under Submission Pipeline Services. In addition to forming part of the submission pipeline, the Mastering Service may also form a capability for any future data migration or upstream legacy system integrations.

Capability	Service/Component	Description
STA Integration Services	STA Integration Gateway Service	The integration gateway exchanges messages between STARS and projects (maps) VET-IS schema data to STA-specific schema data. It is the channel through which STA validation requests are made and through which STA submission release messages are received.
Downstream Data Consumption Services	STARS Downstream Consumer Reporting & Analytics Services	Specific Reporting and Analytics requirements are yet to be provided. In the absence of specific requirements, it is recognised that STARS will provide some form of reporting & analytics capability for downstream consumers, including STAs. It is expected that this will be in the form of specific data extracts for STA-specific mastered data, with each STA only having access to data sourced within their own jurisdiction.
	USI Transcript Service	Specific USI Transcript requirements are yet to be provided. In the absence of specific requirements, it is recognised that STARS will provide transcript data to the USI registry, sourced from training activity reported by RTOs. During the STARS transition period it is likely that STARS-hosted transcript data will be provided to NCVER, to be merged with NCVER-hosted transcript data and then provided to the USI registry via the NCVER USI service.
	STA Bulk Data Integration Service	Specific STA bulk data integration requirements are yet to be provided. In the absence of specific requirements, it is recognised that STARS will provide some form of bulk data transmission of training activity data to downstream STA system. STAs will only receive data relevant to their jurisdictions.
	STARS Master DB Downstream Consumer ETL Pipeline Services	Specific downstream consumer ETL requirements are yet to be provided. In the absence of specific requirements, it is recognised that STARS will provide some form of automatic ETL pipelines to provide mastered data regularly to other systems, including NCVER operational and reporting databases.

Capability	Service/Component	Description
System Support Services	Reference Data Service	The Reference Service manages and provides Reference Data for many business rules, the Rules Engines and workflows throughout STARS, as well as any data enrichment that may be needed during mastering.
	Reference Data Management Portal	The Reference Data Management Portal manages data held in the Reference Service.
	Notifications Service	The Notifications Service manages and distributes Notifications raised by any of the other capabilities/services. Notifications are used to alert or inform system users, e.g., to alert an STA that a submission release is overdue.
	Notifications Management Portal	The Notifications Management Portal will be used to manage and maintain notification messages sent by the Notifications Service.
	Master API Service	The Master API Service will allow various other services and processes to query (only) for VET-IS Logical Schema entities and their VET-IS Data Packet projections (e.g., as Stakeholders or as Students). (Note: Master data is only created or updated by the Mastering Service.)
Provider Support Services	Provider Service	The Provider Service maintains and presents the Stakeholders, Products, Locations, Offerings and Responsibilities associated with the training provider/RTO. It also will be able to submit training activity data into the Submission Pipeline on behalf of the training provider/RTO. These VET Information Standard Schema Entities will be stored as their associated VET Information Standard Data Packet types, i.e.: - Government Packet - Training Organisation Packet

Capability	Service/Component	Description
		- Training Organisation Location Packet - Program Packet - Subject Packet - Offering Packet - Delivery Location Packet - Student Packet - Student Location Packet - Program Enrolment Packet - Subject Enrolment Packet
	Provider Portal	The Provider Portal provides the user interface for entry of both slowly changing data (Provider details, locations, products, offerings, students, etc.) and activity data (enrolments, subjects, programs). The Portal will enable: Micro-RTOs to maintain and submit VET-IS data without the use of an SMS. RTOs to view the progress of their submissions. RTOs to receive notifications from STARS. RTOs to request exemptions for submissions that have not passed validation.
System Management Services	Pipeline Management UIs	The Submission Pipeline Management UIs provide oversight and control of submissions as they move throughout STARS.

Capability	Service/Component	Description
	Organisation & User Management UIs	The Organisation & User Management UIs provide STARS administrators the ability to register and maintain RTO details, as well as register and maintain RTO staff who will be utilising the Provider Portal.

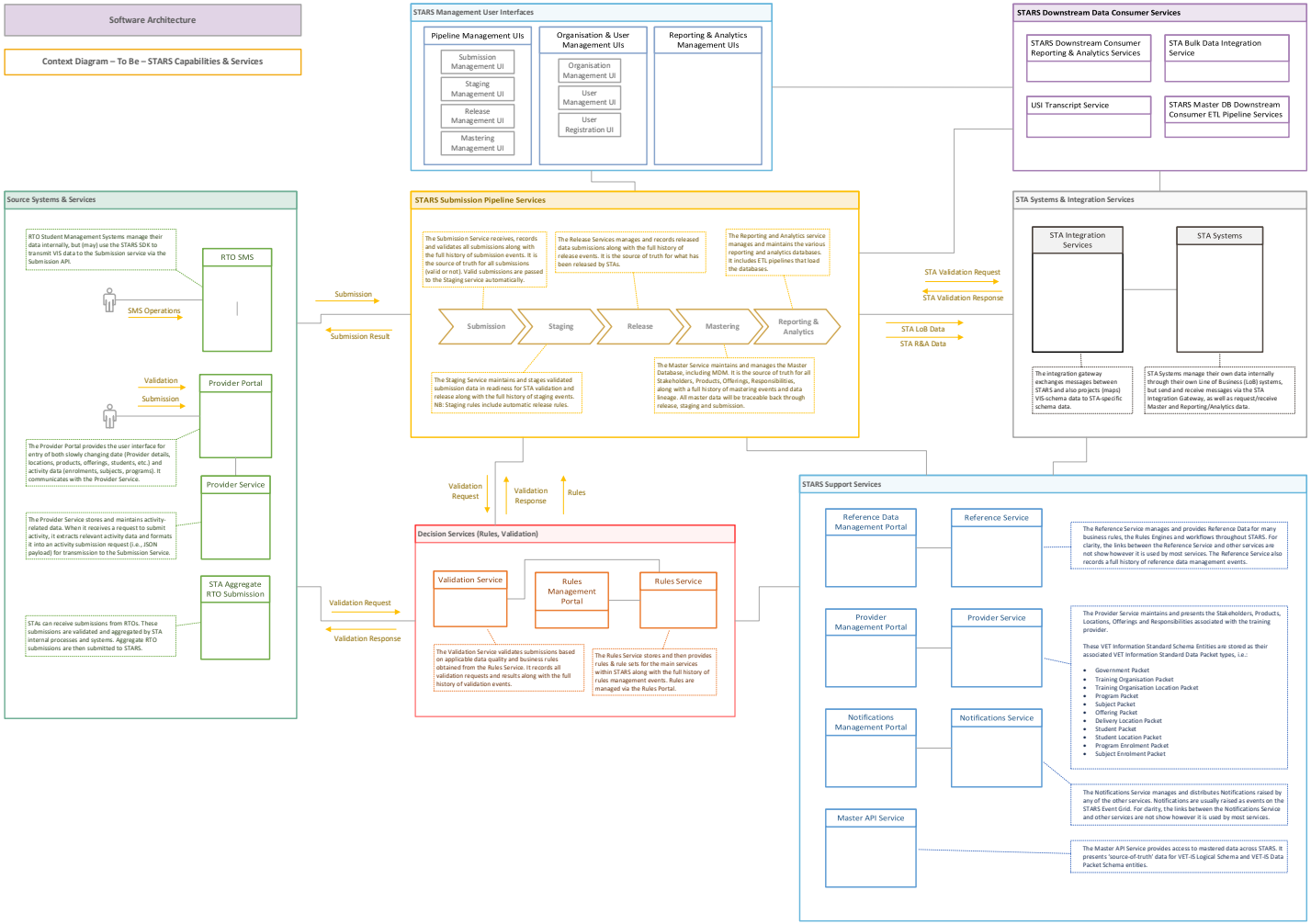


Figure 32: STARS High Level System Capabilities & Services (B:11)

10.3 Solution Services

As mentioned above, STARS will follow Service Oriented Architecture (SOA) principles and will be implemented as (many) loosely coupled services. Loose coupling will be achieved using message- and event-drive architectures, with message and event payloads conforming to several standardised information JSON-based schemas.

Please note that the solution services as described in this document are not 'microservices', i.e., a service that provides a limited function relating to specific domain (e.g., a 'student' microservice that only provides information on students.) The solution services are oriented around providing complete functionality for a solution capability, e.g., validating a submission or staging a submission for an STA.

Nor are the solution services strictly 'web services', however, many of them utilise HTTPS/REST/JSON as their communications transport.

The solution services will, in fact, be implemented as a collection of microservices that act in concert, most often orchestrated by a Logic App workflow. This will be described in detail in sections 10.5 'Service Design' and 11 'Technical Architecture and Design'.

10.3.1 Service Interfaces

All the STARS services will receive and respond to requests, messages and events through various interface protocols and mechanisms.

Interface	Description
HTTPS/REST/JSON	The services will listen for and respond to HTTP requests (GET/POST/PUT/PATCH/DELETE/HEAD). The service will follow standard REST standards for requests. Data will be received and sent in JSON format conforming to JSON Schema standards. Note that a consistent global STARS URL scheme will be implemented using Azure Application Gateway.
SQL	The service will receive SQL requests for relational data result sets. This has applicability for services that manage data in Azure SQL or Cosmos DB[18].
Message	The services will monitor (poll) message queues (in Azure Service Bus) for incoming information, forming the payload of the message. Typically, this interface is one-way, meaning that services receiving messages will not directly respond back to the service sending the message.
Event	The services will listen for events raised by other services (via Azure Event Grid). Events do not typically have a large payload, rather meta data that might point to, say, a

[18] Note that Cosmos DB based services will receive SQL queries wrapped in HTTP/JSON API calls. This is usually abstracted away by the calling service's client code libraries.

DXC TECHNOLOGY Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 109 of 1295

	submission that has failed validation. This interface is one-way, meaning there is no meaningful context where a service responds to another service that has raised an event.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 110 of 1295

10.3.2 Services Overview

This section provides an overview of the main services within STARS. The services are described in more detail in section 10.5 'Service Design' and in Attachment 1 – Collaboration Diagrams – D.0.1.vsdx. A high-level solution services context diagram is presented in section 10.3.3 'Solution Services Context Diagram'.

The STARS services are grouped around the major capabilities:

Service Group	Service	Interfaces	Comments
Capability: Core Information Services			The Core Information Services provide data storage, retrieval and management services for all main data stores and associated entities. These services are utilised by many services within STARS, including the Submission Pipeline Services, downstream Reporting & Analytics Services, and downstream Data Consumption Services.
	Master Data Service	HTTP/REST/JSON SQL	The Master Data Service is the core data repository for STARS. It manages and provides VET-IS data. It is the source of truth for all Stakeholders, Products, Offerings, Responsibilities, along with data lineage. All master data will be traceable back through release, staging and submission. The Master Data Service provides an API for querying data. This is utilised throughout the system. It is expected that the underlying Master Database will be integrated with other services, e.g., STA Bulk Data Share, using ETL Pipelines.
	Reference Data Service	HTTP/REST/JSON SQL	The Reference Service manages and provides Reference Data for many business rules, the Rules Engines and workflows throughout STARS. For clarity, the links between the Reference Service and other services are not show however it is used by most services. The Reference Service also records a full history of reference data management events.

Service Group	Service	Interfaces	Comments
	Submission Service	HTTP/REST/JSON Message	The Submission Service receives, records and validates all submissions along with the full history of submission events. It is the source of truth for all submissions (valid or not). Valid submissions/packets are notified to the Staging service automatically via messaging. Submissions are received either via the Provider Service (for RTO sourced data) or via the STA Gateway Service (for STA sourced RTO data.)
	Provider Service	HTTP/REST/JSON	The Provider Service also acts as the gateway between Training Providers and STARS. Amongst other responsibilities, it: • Provides the entry point for submissions from SMSs using the STARS public APIs • Stores activity-related data for micro-RTOs using the Provider Portal. The Provider Service maintains and presents RTO data and the Stakeholders, Products, Locations, Offerings and Responsibilities associated with the training provider. When it receives a request to submit activity via the Portal, it extracts relevant activity data from the Provider DB and formats it into an activity submission request (i.e., JSON payload) for transmission to the Submission Service via the message bus. These VET Information Standard Schema Entities are stored as their associated VET Information Standard Data Packet types, i.e.: • Government Packet • Training Organisation Packet • Training Organisation Location Packet • Program Packet • Subject Packet • Offering Packet

Service Group	Service	Interfaces	Comments
			• Delivery Location Packet • Student Packet • Student Location Packet • Program Enrolment Packet • Subject Enrolment Packet The Provider Service creates Surrogate Keys automatically for new data as it is received from SMSs via direct API calls or via manual entry in the Provider Management Portal. New and updated Packets are candidates for Validation and Submission; these processes are initiated by Provider Portal users manually or via API call for SMSs.
Capability: Submission Pipeline Services			The Submission Pipeline is at the heart of STARS. It is responsible for receiving, validating, staging, releasing and mastering submissions received both from RTOs and STAs.
	Submission Service		See Submission Service above, under Core Information Services. The Submission Service plays a core role within the pipeline where it receives raw information. It also provides submission information to other services both within the pipeline and elsewhere. The submission receives submissions via the message bus from both the Provider Service (e.g., from RTOs) and from the STA Batch Submission service.
	Staging Service	HTTP/REST/JSON Message	The Staging Service stages validated submission data in readiness for STA validation and release along with the full history of staging events. NB: Staging rules include automatic release rules.

Service Group	Service	Interfaces	Comments
	Release Service	HTTP/REST/JSON Message	The Release Services manages released data submissions along with the full history of release events. It is the source of truth for what has been released by STAs.
	Mastering Service	HTTP/REST/JSON Message	The Mastering Service manages the MDM processes for populating the Master Database. It receives released submissions and performs, e.g., duplicate detection, creation of golden records, etc.
	Reporting and Analytics Provisioning Service	HTTP/REST/JSON Message	The Reporting and Analytics Provisioning Service manages and maintains the flow of data into various reporting and analytics databases. It includes ETL pipelines that load the databases.
Capability: Decision Services			The Decision Services provide the means for Submission Pipeline Services, and their embedded workflows, to control the flow of information within the system. These include both submission validation services and rules engine services.
	Rules Service	HTTP/REST/JSON	The Rules Service stores and then provides rules & rule sets for the main services within STARS along with the full history of rules management events. Rules are managed via the Rules Portal.
	Validation Service	HTTP/REST/JSON	The Validation Service (pre)validates submissions based on applicable rules (sourced from the Rules Service). It records all validation requests and results along with the full history of validation events.
STA Services			STARS will provide services to facilitate the interchange of VET data with STA systems. Although detailed STA integration requirements are not yet

Service Group	Service	Interfaces	Comments
			provided, it is expected that STAs will require information to be translated from VET-IS to AVETMISS format. Additionally, although STARS will often receive and process VET data in real time (i.e., not batched), STA systems are likely to only be able to receive batched VET data for some time. It is also likely that STAs receiving batched data will need the data batches containing complete information for the current reporting period(s).
	STA Gateway Service	TBD	The integration gateway exchanges messages between STARS and projects (maps) VET-IS schema data to STA-specific schema data. It also provides facilities for STA submission of bulk RTO data.
Capability: Management Services			Management services provide system and user interface capabilities to manage the submission pipeline, RTO organisation information & registration, and user information & registration (e.g., RTO staff).
	Pipeline Management UIs		The Submission Pipeline Management UIs provide oversight and control of submissions as they move throughout STARS.
	Organisation and User Management UIs		The Organisation & User Management UIs provide STARS administrators the ability to register and maintain RTO details, as well as register and maintain RTO staff who will be utilising the Provider Portal.
Capability: Data Consumption Services			The Data Consumption Services collectively provide downstream VET data consumers the ability to report and analyse STARS data. They also provide the

Service Group	Service	Interfaces	Comments
			means for STAs to obtain bulk data, and for NCVER to receive data for existing reporting systems.
	STARS Downstream Consumer Reporting and Analytics Services		Specific Reporting and Analytics requirements are yet to be provided. In the absence of specific requirements, it is recognised that STARS will provide some form of reporting & analytics capability for downstream consumers, including STAs. It is expected that this will be in the form of specific data extracts for STA-specific mastered data, with each STA only having access to data sourced within their own jurisdiction.
	USI Transcript Service		Specific USI Transcript requirements are yet to be provided. In the absence of specific requirements, it is recognised that STARS will provide transcript data to the USI registry, sourced from training activity reported by RTOs. During the STARS transition period it is likely that STARS-hosted transcript data will be provided to NCVER, to be merged with NCVER-hosted transcript data, and then provided to the USI registry via the NCVER USI service.
	STA Bulk Data Integration Service		Specific STA bulk data integration requirements are yet to be provided. In the absence of specific requirements, it is recognised that STARS will provide some form of bulk data transmission of training activity data to downstream STA system. STAs will only receive data relevant to their jurisdictions.
	STARS Master DB Downstream Consumer ETL Pipeline Services		Specific downstream consumer ETL requirements are yet to be provided. In the absence of specific requirements, it is recognised that STARS will provide some form of automatic ETL pipelines to provide mastered data regularly to other systems, including NCVER operational and reporting databases.

Service Group	Service	Interfaces	Comments
Capability: Support Services			Support Services provide 'cross-cutting' system-wide services to support the operation of other services in STARS.
	Notifications Service	HTTP/REST/JSON Event	The Notifications Service manages and distributes Notifications raised by any of the other services. Notifications are usually raised as events on the STARS Event Grid. It is used by most services.
	Event Service		The Event Service (Azure Event Grid) manages and distributes system events raised by one service to all other subscribing services.
	Logging Service		The Logging Service (Azure Monitor, Log Analytics Workspace) manages and reports on logging for all services.
	Audit Service		The Audit Service (Azure Monitor, Log Analytics Workspace, Azure Security Center) manages and reports on audit events for all services, including items such as who accessed the system, updated records, etc.
Capability: Identity and Access Management Services			Identity and access management services provide both user and system service identity management and access controls for STARS services. These services will be delivered as configuration items, using Microsoft modern identity platform, Azure AD, and Azure RBAC. Please refer to the Security Architecture and Design chapter for more detail.
	Azure AD		Azure AD will manage users, organisations, groups.
	Managed Identities		Managed Identities will be created for all system services.
	Azure Roles and RBAC		Azure RBAC will provide granular access control for all principals to STARS services.

10.3.3 Solution Services Context Diagram

Figure 33: STARS Solution Services Context Diagram below shows all the solution services in context, organised via capabilities – please refer to Attachment 1 – Collaboration Diagrams – D.0.1.vsdx for a more detailed (and accessible) view. The individual services are diagrammed and described in more detail in section 10.5 ‘Service Design’.

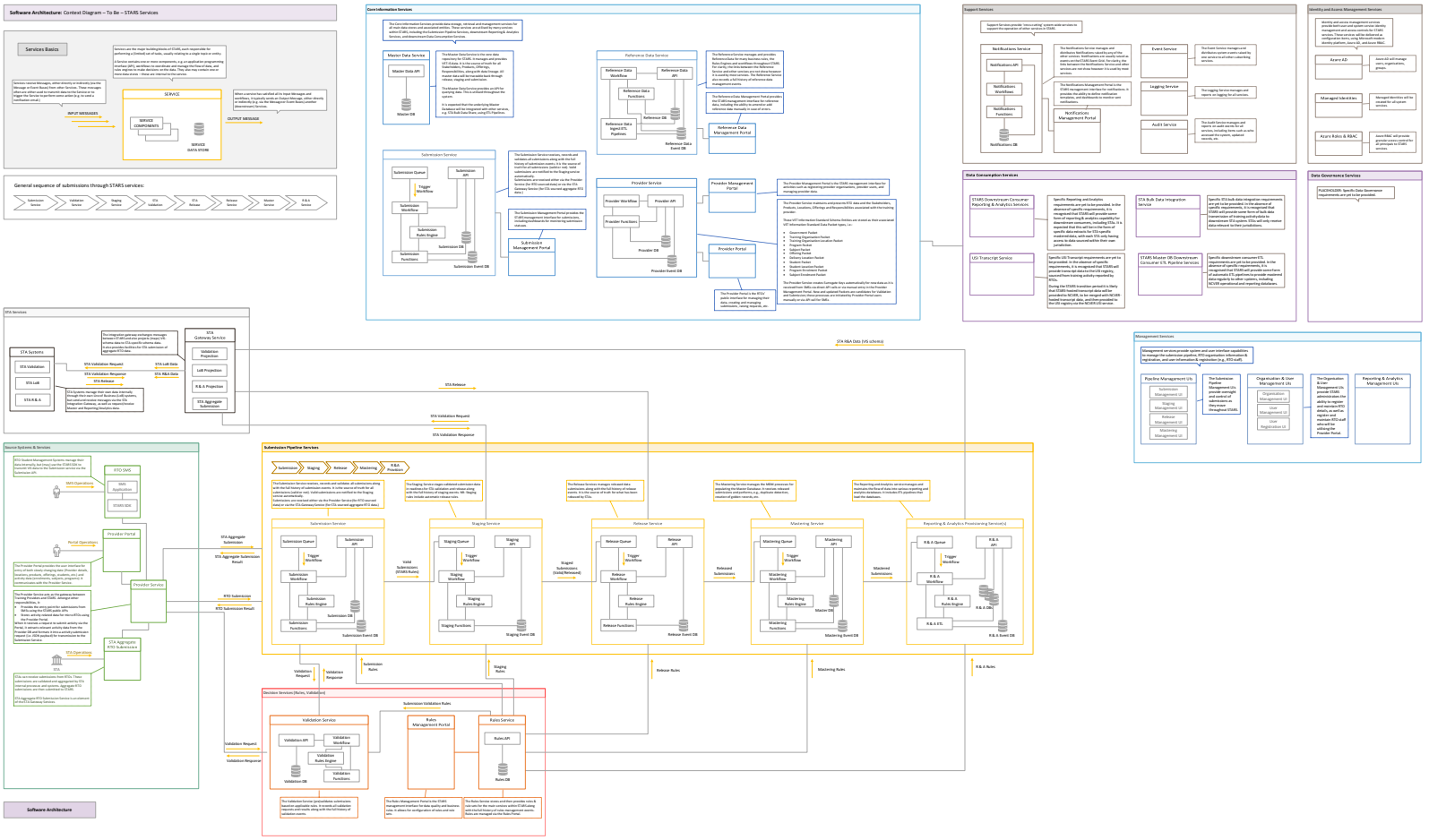


Figure 33: STARS Solution Services Context Diagram (B:12)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

10.4 Rules Engine

The DXC team are currently evaluating appropriate rules engine technology as informed by the requirements for STARS, with particular emphasis on ensuring that the rules platform management infrastructure can cater for the complexity associated with the differing requirements of STAs and NCVER. The final choice will be made during Work Package 2, once work has completed to establish and analyse the numerous data quality and business rules for the STAs and national reporting, along with the attendant workflows.

At the time of writing, DXC has established two candidate engines: Drools and NRules. NRules is a relatively light weight but powerful tool but has limited rules management ability. Drools is also a powerful tool with more extensive rules management abilities. Drools will require a greater level of administration and technical infrastructure within the Azure environment, but this will likely be offset by its ability to more effectively manage complexities associated with the differing requirements between STAs and NCVER.

Both candidates utilise optimised rule processing algorithms based on the Rete Algorithm[19] and are developed and maintained by experts in the field.

10.4.1 Rules Engines Overview

The STARS rules engine will utilise a widely used open-source rules engine (*Drools/NRules – see comments above*).

Please note that DXC is not developing the rules algorithms and engine as such but are using the open-source rules-engine tools to build a complete rules-processing framework for submission validation, etc.

The rules engine operates entirely in memory, at extremely high speed. It will operate directly on STARS data in memory, removing any integration requirements.

The STARS rules engine will be configuration driven, allowing for flexibility for enabling/disabling rules, setting validity periods, and even determining which data elements or jurisdiction-specific information the rules apply to. Rules themselves will be managed as 'Rule Sets', with different rule sets applied for different requirements, such as STA-specific validation rules.

10.4.2 Rules Engine Terminology and Usage

The rules engine will utilise optimised rule processing algorithms based on the Rete Algorithm[20] , developed and maintained by experts in the field. The Rete algorithm can efficiently process many rules against large volumes of data, but to do so can require a lot of computer memory – at least the size of all the data being evaluated plus any lookup/reference data required and any results.

From a usage perspective, the rule engine maintains and executes rules, assembled as rule sets. Each rule is made up of two sections, a 'match/when' clause, and an 'action/do' clause. When a condition is matched, the rule specifies an action to be performed.

---

[19] See https://en.wikipedia.org/wiki/Rete_algorithm
[20] See https://en.wikipedia.org/wiki/Rete_algorithm

DXC TECHNOLOGY
Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 119 of 1295

The rules engine maintains an area of memory known as 'working memory' to hold information while processing rules.

Data to be evaluated against the rules, known as 'facts' is loaded into the working memory. Any supporting information, such as reference lookup values (e.g., from the Reference Data Service), is also loaded into working memory – this ensures that the rules engine can operate at high speed without having to pause to access, e.g., external databases. Typically, items used for lookups are cached in memory by the rules engine upon start up.

When ready, the rules engine 'executes' the rule sets against the facts (data) within the working memory. The combination of working memory, facts, rule sets and the execution of rules and actions is known as a 'session'.

During the session, rules that match against facts in the working memory perform their associated actions. Even though it is possible for a rule action to perform some external task (e.g., update a record in a database), this is not recommended. Instead, it is better for rule actions to 'assert' new facts that represent some outcome of matching existing facts back into working memory. For example, an invalid submission is recorded, as 'invalid', back into working memory.

When all rules and actions have completed, the rule engine code then examines the results of the session. In the example above, any 'invalid submission' records are extracted from working memory and can be passed back to the requesting service.

10.4.3 Managing the evolution of STARS validation rules

Ongoing management of STARS rules several aspects, noting that the embedded engine uses rules written in either in code or a Domain Specific Language (DSL)[21] that is used to create pseudo-English scripts that define the rules. The latter will allow, for example, technical business analysts to define rules.

Code or DSL scripts for new or evolving rules of course can easily be created or adapted. Even though there might be many rules, the rules themselves usually only require a paragraph or two of code that require not much coding effort – more time is usually spent analysing and specifying the rules.

The rules code itself (and by implication the rules logic) is managed by a source code management system (Azure DevOps Repositories). The code and any changes and subsequent releases are all tracked and linked to requirements (i.e., User Stories/Product Backlog Items), allowing for full traceability and history of changes to rules, including approvals.

Coded or DSL-based rules can be rapidly developed (usually in a few hours) and automatically tested and deployed to UAT, staging and production environments – completely under the control of STARS Product Owners/SMEs/Administrators.

Note that rules will be applicable to certain scopes. This may include 1 or more versions of the VET-IS, or a particular STA, or a particular data element/entity. As such, they have the flexibility to manage the evolution of changes to the VET-IS or changing business requirements.

[21] See https://docs.jboss.org/drools/release/5.3.0.Final/drools-expert-docs/html/ch05.html for details of the Drools rules DSL.

DXC TECHNOLOGY Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 120 of 1295

10.4.4 Efficiency

The STARS rules engine service calls program functions directly to validate a record completely in memory, extremely fast. This makes it feasible to provide as close to real time validation for data submissions, at least for the first stages of the STARS data submission process.

10.4.5 Auditing & traceability

The rules engine provides mechanisms to log and audit execution of rules. This will enable recording of which rule failed and why. STARS will then log these events for diagnostics purposes as well as to provide detailed reasons for validation failure, e.g., for reporting back to RTOs.

10.4.6 Rules Configuration

Rules are dynamically loaded at runtime by the rules engine. The dynamic loading feature, along with dynamic rules parameters (e.g., applicable data elements, allowed values, dates applicable) allows for the rules engine to be configured via settings and parameters stored in a database or through the Rules Management UI.

Each of the rules will have a Description associated with them, that will provide a business-readable representation of the rules.

The Rules Management UI will also provide the ability to export the rules a user has ownership of. The structure of this export will be defined in detailed design.

10.4.7 Errors and Notifications

The rules engine has several facilities to monitor execution and troubleshoot issues with any rules. Typically, information surfaced by the monitoring functions are exposed through a logging framework.

The rules engine libraries will integrate with any of the Azure logging frameworks, allowing for detailed tracing through, say, Azure Monitor.

10.4.8 Rules Engine Service

Many of the solution service designs described in section 10.5 ‘Service Design’ include a service-specific ‘logical’ rules engine component. These will be affected as specific rules execution sessions that contain the solution service specific rules, within a single Rules Engine Service. Each solution service will be associated with their own HTTP/REST/JSON endpoint within the Rules Engine Service, bound to the relevant the execution sessions[22].

The Rules Engine Service will be hosted within a suitable Azure App Service environment[23], provisioned with enough compute resources to ensure that multiple sessions can execute concurrently, for the expected validation workloads.

---

[22] Routing of requests to the Rules Service (and other services) will be via Azure Application Gateway. This will enable STARS to be configured with multiple App Service or Azure Container-based resources to handle workloads, but all through a unified, consistent URL scheme. Please refer to the Technical Architecture section for details.

[23] If Drools is chosen, the compute environment may be one of the Azure container-based services, as this will likely be a lower cost/higher performance option.

DXC TECHNOLOGY Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 121 of 1295

10.5 Service Design

Please refer to Attachment 1 - Context Diagrams – D.0.1.vsdx

This Visio file contains detailed diagrams showing all the solution services in context.

This section provides a detailed description of each of the solution services to the level of, for example, microservices and data stores. In general, microservices will be implemented as Azure Functions.

Most solution services will be contained within their own Azure Resource Group, alongside supporting Azure infrastructure, such as Azure App Configurations, Key Vaults, Storage Accounts, etc. The solution service code, along with ARM templates for their associate Azure infrastructure will be managed in separate Azure DevOps Repositories and built and deployed in separate Azure DevOps Pipelines.

These will be described in more detail in section 11 ‘Technical Architecture and Design’.

10.5.1 Core Information Services

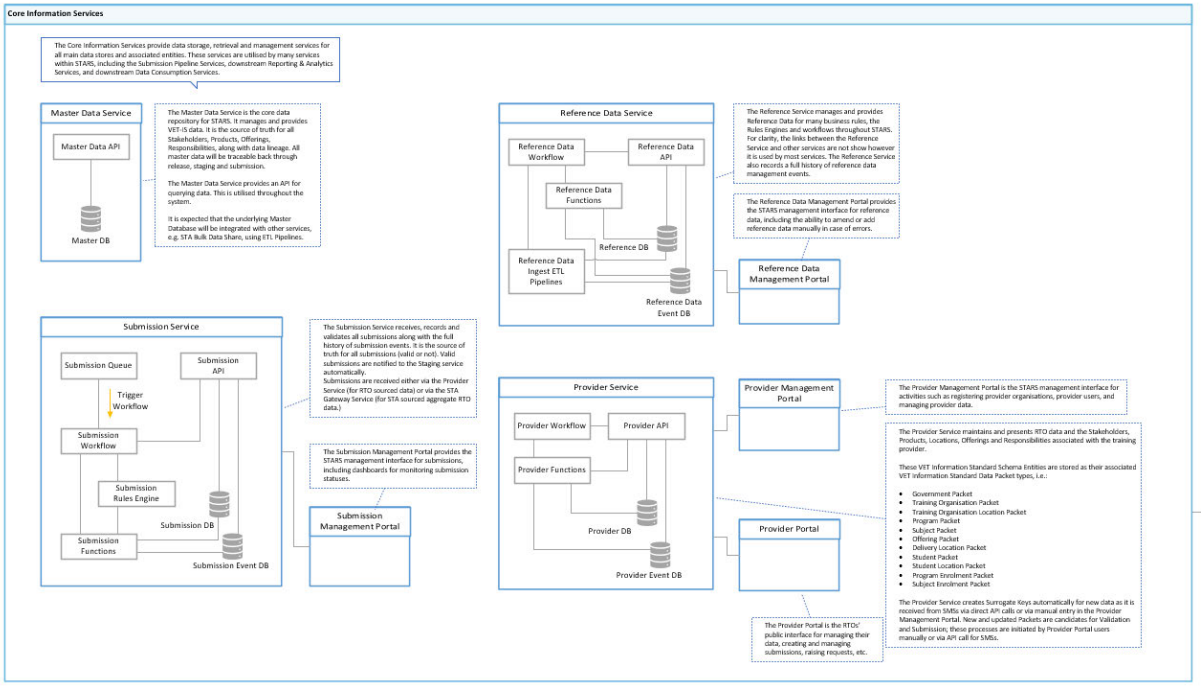


Figure 34: Core Information Services (B:12)

The Core Information Services provide data storage, retrieval and management services for all main data stores and associated entities. These services are utilised by many services within STARS, including the Submission Pipeline Services, downstream Reporting & Analytics Services, and downstream Data Consumption Services.

The Core Information Services include the Master Data Service, the Reference Data Service, the Submission Service, and the Provider Service.

The services are described further below.

10.5.1.1 Master Data Service

Refer to section 11.1.1 ‘Master Data Service’ for a technical description.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 122 of 1295

10.5.1.1.1 Service Description

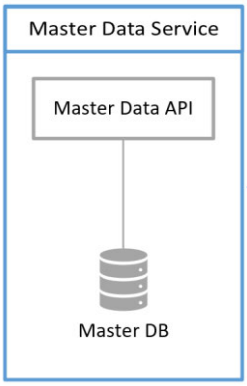


Figure 35: Master Data Service (B:12)

The Master Data Service is the core data repository for STARS. It manages and provides VET-IS data, held in VET-IS Logical Schema format within the Master Database, hosted in Azure SQL. It is the source of truth for all Stakeholders, Products, Offerings, Responsibilities, along with data lineage. All master data will be traceable back through release, staging and submission.

The Master Data Service provides an HTTP/REST/JSON API for querying data. This is utilised throughout the system to lookup VET-IS data as needed.

It is expected that the underlying Master Database will be integrated with other services, e.g., STA Bulk Data Share, using ETL Pipelines.

As well as providing HTTP/REST/JSON data through the API, the Master DB will be able to be queried directly using standard SQL Server Transact SQL.

It is highly likely that, for example, STAs (or most other STARS data consumers) are unable to utilise VET-IS Logical Schema entities (e.g., Responsibilities) but will prefer to work with, say, Students and Enrolments. In this instance the Master Data Service will also provide projections, or views, of VET-IS entities into the usual VET domain entities. For example, views will be created that present Stakeholders as Students, Responsibilities as Enrolments.

10.5.1.1.2 Service Components

10.5.1.1.2.1 Master Database

The Master Database will be implemented in Azure SQL.

As noted in section 8.14.6, the Master Database is separated into two broad partition sets:

- STA. The STA partitions will contain ‘silver’ level data (passed data quality and business rules), to retain and support STA operational-level information and reporting. This includes both national and state-specific data elements defined in the VET-IS logical schema. Each jurisdiction will have its own partition, with jurisdictions only able to access their own data.
- National. The National partition will contain ‘gold’ level data (passed data quality and business rules, de-duplicated, enriched with national classification data), held in VET-IS logical schema format, minus any STA operational-level information. The National partition will be used for national reporting and as a source of truth for national VET data, including student transcripts.

The partitioning scheme will utilise standard Azure SQL table partitioning functionality.

10.5.1.1.2.2 Master Data API

The Master Data API will be implemented as an Azure Function App, utilising HTTP endpoints. The API will be defined during Work Package 2, pursuant to the design and implementation of the Master DB and associated schemas and partitioning scheme.

The API will conform to the STARS Design Principles and the STARS API Design Document.

10.5.1.1.3 Service Interfaces

HTTP/REST/JSON: See Master Data API, below. Master data will be queryable through the API, using REST semantics. Refer to the STARS API documentation for more detail.

DXC TECHNOLOGY
Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 123 of 1295

SQL: See Master Database, above. The Master DB will be queryable via SQL DDL and DML statements, depending on the calling user/service's access.

10.5.1.1.4 Service Security

The service will conform to the Security Architecture. Access to the Master Data Service will be controlled by Azure RBAC, with all calling services identified and granted access through Managed Identities.

10.5.1.2 Reference Data Services

Refer to section 11.1.2 'Reference Data Service' for a technical description.

10.5.1.2.1 Service Description

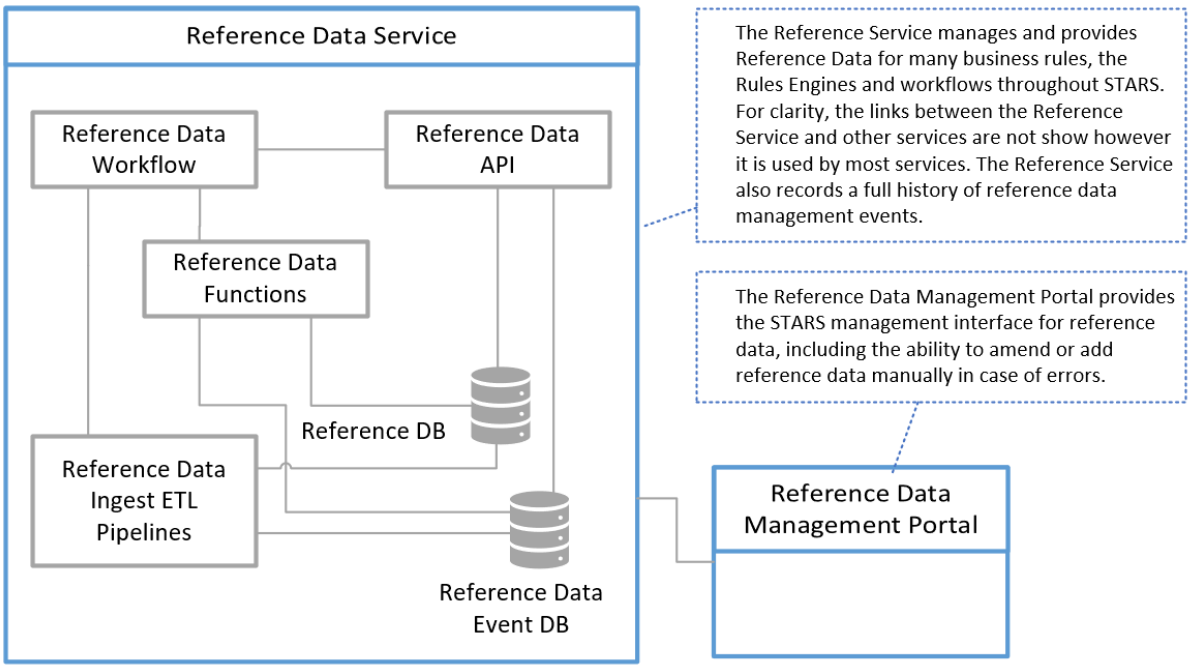


Figure 36: Reference Data Service (B:12)

The Reference Service has two primary roles:

1. Manage the workflows and processes involved with sourcing reference data from external sources, such as TGA.
2. Managing and providing reference data through a consistent schema, for use by other STARS solution services.

The Reference Service manages and provides reference data for many data quality and business rules, used by the Validation Service, Rules Engines and workflows throughout STARS. The Reference Service is used by most services to source reference or lookup data. The Reference Service also records a full history of reference data management events.

The Reference Data Management Portal provides the STARS management interface for reference data, including the ability to amend or add reference data manually in case of errors. This is important, as source systems are occasionally missing information or contain errors; the portal will allow STARS system operators to amend information so that STARS processing can continue.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 124 of 1295

10.5.1.2.2 Service Components

10.5.1.2.2.1 Reference Data Workflow

The Reference Service internal processes will be controlled by one or more Azure Logic App workflows. These workflows will manage, for example, the process for handling alerts raised by other services for invalid reference data. The workflows will also manage the processes for ingesting source data from external services, including handling exceptions.

10.5.1.2.2.2 Reference Data Functions

The Reference Service code-based functionality will be implemented as Azure Function Apps. These functions will support the operation of the service's workflows as well as functionality provided through the Reference Data Management Portal. For example, the portal will have functions to amend reference data, the actual code to implement this at the database level will be within the Function Apps.

10.5.1.2.2.3 Reference Data Ingest ETL Pipelines

The Reference Service will maintain several Azure Data Factory based data ingest ETL pipelines. These will be configured to connect to upstream reference data sources using the appropriate transport and protocols. The pipelines will extract, transform and then load reference data into the Reference DB, using a consistent schema.

10.5.1.2.2.4 Reference Data API

The Reference Data API will be implemented as an Azure Function App, utilising HTTP endpoints. The API will be defined during Work Package 2, pursuant to the design and implementation of the Reference DB and associated schemas.

Care will be taken to ensure reference data can be presented via the API to workflows and the rules engine(s) in a format to allow rapid processing of rules – please see the discussion in section 10.4 'Rules Engine'.

The API will conform to the STARS Design Principles and the STARS API Design Document.

10.5.1.2.2.5 Reference DB

The Reference Database will be implemented in Azure SQL. It will utilise a standardised schema to ensure consistency in the management and presentation of reference data. Care will be taken to ensure reference data schemas can be presented (via the Reference Data API) to workflows and the rules engine(s) in a format to allow rapid processing of rules – please see the discussion in section 10.4 'Rules Engine'.

10.5.1.2.2.6 Reference Data Event DB

All service events relating to the lifecycle and management of reference data will be recorded, in the STARS Event schema, in Cosmos DB.

10.5.1.2.3 Service Interfaces

HTTP/REST/JSON: See Reference Data API, above. Reference data will be queryable through the API, using REST semantics. Refer to the STARS API documentation for more detail.

DXC TECHNOLOGY
Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 125 of 1295

SQL: See Reference Database, above. The Reference DB will be queryable via SQL DDL and DML statements, depending on the calling user/service's access.

10.5.1.2.4 Service Security

The service will conform to the Security Architecture. Access to the Reference Data Service will be controlled by Azure RBAC, with all calling services identified and granted access through Managed Identities.

10.5.1.3Submission Service

Refer to section 11.1.5 'Submission Service' for a technical description.

10.5.1.3.1 Service Description

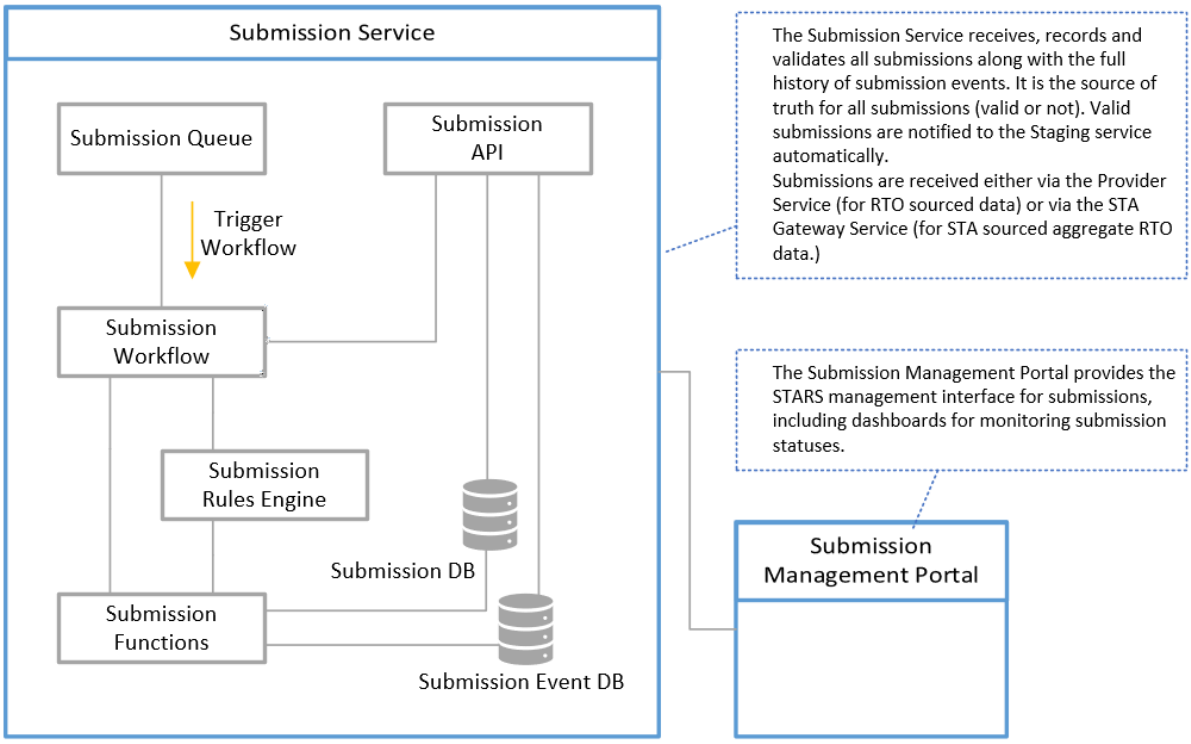


Figure 37: Submission Service (B:12)

The Submission Service has two primary roles:

1. Manage the workflows and processes involved with receipt, validation and hand-off of submissions to the Staging Service.
2. Managing and providing submission data through a consistent schema, for use by other STARS solution services.

The Submission Service receives, records and validates all submissions along with the full history of submission events. It is the source of truth for all submissions (valid or not). Valid submissions are notified to the Staging service automatically.

Submissions are received either via the Provider Service (for RTO sourced data) or via the STA Gateway Service (for STA sourced RTO data.) Submissions are received from these services as messages in the Submission Queue.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 126 of 1295

Submissions across the entire submission pipeline are held and processed in VET-IS JSON schema format. Services that feed-in submissions to the pipeline (via the Submission Service) are responsible for transforming VET data from the original format, into the VET-IS JSON schema.

The Submission Management Portal provides the STARS management interface for submissions, including dashboards for monitoring submission statuses.

10.5.1.3.2 Service Components

10.5.1.3.2.1 Submission Queue

The Submission Service does not receive submissions directly from RTOs or STAs; submissions are placed into the Submission Queue, in VET-IS JSON schema format, by the Provider or STA Batch Submission services.

The Submission Queue will be implemented within Azure Service Bus. It is likely that multiple queues will be utilised to support various STARS submission operational purposes (e.g., to support different VET-IS versions, or different STA business processes.) In this regard, the Submission Queue is a 'logical' construct that supports loose coupling between the Submission Service and any upstream services that are collecting submissions for the pipeline.

10.5.1.3.2.2 Submission Workflow

The Submission Service internal processes will be controlled by one or more Azure Logic App workflows.

The Submission Queue is monitored by the Submission Workflow Logic App(s). As submissions arrive in the queue, the workflow will record the submission in the Submission DB, prior to initiating the relevant validation workflows (see Validation Service for details.)

Valid submissions are handed-off to the Staging Service (funded) or Release Service (FFS) via messages to their respective input queues, and a 'success' message returned to the originating service.

Invalid submissions are returned to the originating service for attention by either the RTO or STA, along with detailed reasons for validation failure.

The details are scenario specific, but generally submission results are returned to the upstream service via 'call-back' URLs hosted within the upstream service, with the call-backs uniquely associated with each input submission.

10.5.1.3.2.3 Submission Functions

The Submission Service code-based functionality will be implemented as Azure Function Apps. These functions will support the operation of the service's workflows as well as functionality provided through the Submission Management Portal. For example, the portal will have functions to monitor submission data and results, the actual code to implement this at the database level will be within the Function Apps.

10.5.1.3.2.4 Submission Rules Engine

The Submission Rules Engine, and any associated rules, is intended to assist with the operation of the Submission Workflows. Please note that this does not include submission validation rules – these are

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 127 of 1295

handled by the Validation Service. The submission rules engine role will apply any business rules associated with processing of entire submissions (not the content of the submissions).

10.5.1.3.2.5 Submission API

The Submission API will be implemented as an Azure Function App, utilising HTTP endpoints.

The API will allow other services to flexibly query and extract submission data in JSON format, including information on the submissions themselves and all the submission lifecycle data, including submission status.

The API will be defined during Work Package 2, pursuant to the design and implementation of the Submission DB and associated VET-IS SQL and JSON schemas.

Care will be taken to ensure submission data can be presented via the API to workflows and the rules engine(s) in a format to allow rapid processing of rules – please see the discussion in section 10.4 'Rules Engine'.

The API will conform to the STARS Design Principles and the STARS API Design Document.

10.5.1.3.2.6 Submission DB

The Submission Database will be implemented in Azure Cosmos DB. It will utilise a standardised document-oriented JSON schema to ensure consistency in the management and presentation of submission data. Care will be taken to ensure reference data schemas can be presented (via the Submission API) to workflows and the rules engine(s) in a format to allow rapid processing of rules – please see the discussion in section 10.4 'Rules Engine'.

10.5.1.3.2.7 Submission Event DB

All service events relating to the lifecycle and management of submission data will be recorded, in the STARS Event schema, in Cosmos DB.

10.5.1.3.3 Service Interfaces

HTTP/REST/JSON: See Submission Data API, above. Submission data will be queryable through the API, using REST semantics. Refer to the STARS API documentation for more detail.

SQL: See Submission Database, above. The Submission DB will be queryable via Cosmos DB SQL DDL statements, depending on the calling user/service's access.

10.5.1.3.4 Service Security

The service will conform to the Security Architecture. Access to the Submission Service will be controlled by Azure RBAC, with all calling services identified and granted access through Managed Identities.

10.5.1.4Provider Services

Refer to section 11.1.3 'Provider Service' for a technical description.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 128 of 1295

10.5.1.4.1 Service Description

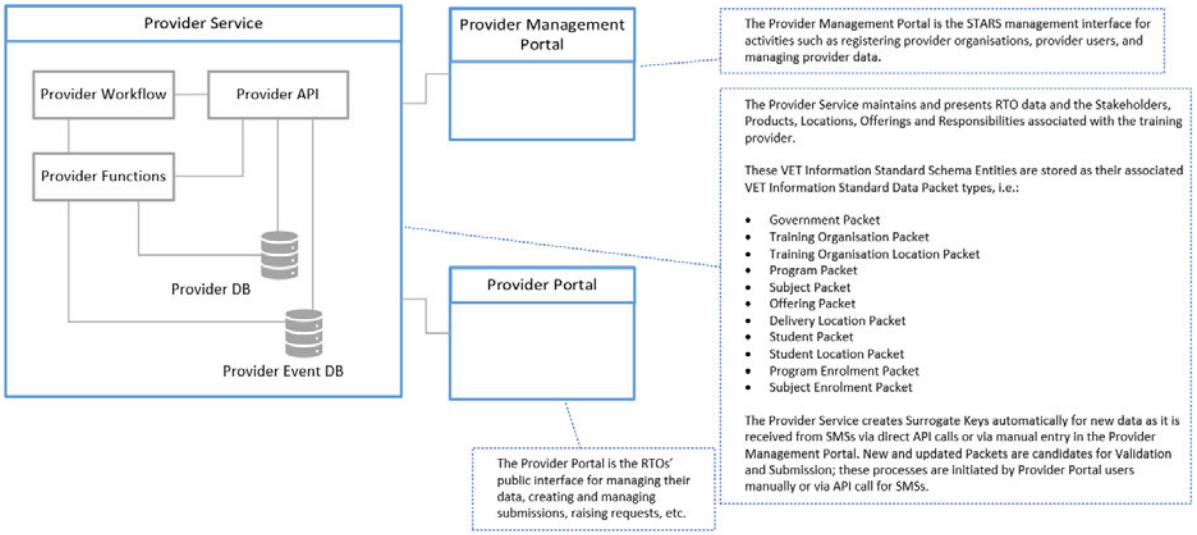


Figure 38: Provider Service (B:12)

The Provider Service acts as the gateway between Training Providers and STARS. Amongst other responsibilities, it:

- Provides the entry point for submissions from SMSs using the STARS public APIs
- Stores and manages VET data for micro-RTOs using the Provider Portal.

When it receives a request to submit activity via the Portal, it extracts relevant activity data from the Provider DB and formats it into a submission request in VET-IS JSON schema format for transmission to the Submission Queue.

The Provider Service maintains and presents RTO data and the Stakeholders, Products, Locations, Offerings and Responsibilities associated with the training provider.

These VET Information Standard Schema Entities are stored as their associated VET Information Standard Data Packet types, i.e.:

- Government Packet
- Training Organisation Packet
- Training Organisation Location Packet
- Program Packet
- Subject Packet
- Offering Packet
- Delivery Location Packet
- Student Packet
- Student Location Packet
- Program Enrolment Packet
- Subject Enrolment Packet

The Provider Service creates Surrogate Keys automatically for new data as it is received from SMSs via direct API calls or via manual entry in the Provider Management Portal. New and updated Packets are candidates for Validation and Submission; these processes are initiated by Provider Portal users manually or via API call for SMSs.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 129 of 1295

The Provider Portal provides the user interface for entry of both slowly changing date (Provider details, locations, products, offerings, students, etc.) and activity data (enrolments, subjects, programs). It communicates with the Provider Service.

The Provider Management Portal is the STARS management interface for activities such as registering provider organisations, provider users, and managing provider data.

10.5.1.4.2 Service Components

10.5.1.4.2.1 Provider Workflow

The Provider Service internal processes will be controlled by one or more Azure Logic App workflows.

10.5.1.4.2.2 Submission Functions

The Provider Service code-based functionality will be implemented as Azure Function Apps. These functions will support the operation of the service's workflows as well as functionality provided through the Provider Portal and the Provider Management Portal. For example, the portal will have functions to manage provider student information, the actual code to implement this at the database level will be within the Function Apps.

10.5.1.4.2.3 Provider API

The Provider API will be implemented as an Azure Function App, utilising HTTP endpoints.

The API will allow other services to flexibly query and extract provider data in JSON format, including information on the submissions themselves and all the submission lifecycle data, including submission status.

The API will also be used by the Portal user interfaces to service UI requests for data management.

The API will be defined during Work Package 2, pursuant to the design and implementation of the Provider DB and associated VET-IS SQL and JSON schemas.

The API will conform to the STARS Design Principles and the STARS API Design Document.

10.5.1.4.2.4 Portal DB

The Portal Database will be implemented in Azure Cosmos DB. It will utilise a standardised document-oriented JSON schema to ensure consistency in the management and presentation of provider data. Care will be taken to ensure provider data schemas can be presented (via the Provider API) to UIs to allow operators to manage and query provider information.

The API will conform to the STARS Design Principles and the STARS API Design Document.

10.5.1.4.2.5 Provider Event DB

All service events relating to the lifecycle and management of provider data will be recorded, in the STARS Event schema, in Cosmos DB.

10.5.1.4.3 Service Interfaces

HTTP/REST/JSON: See Provider Data API, above. Provider data will be queryable through the API, using REST semantics. Refer to the STARS API documentation for more detail.

SQL: See Provider Database, above. The Provider DB will be queryable via Cosmos DB SQL DDL statements, depending on the calling user/service's access.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 130 of 1295

10.5.1.4.3.1 Provider Portal

The Provider Portal is a Single Page Application (SPA) developed in Angular utilising Progressive Web Application (PWA) technologies. PWAs provide native app like experience on mobile devices without requiring distribution and installation via App Stores, which will result in a significant saving in administration and ongoing support costs.

The Provider Portal will allow micro-RTOs to manage their data, create and manage submissions, raise exemption, or support requests and receive notifications.

It will also allow all RTOs to register with STARS, view and accept Terms and Conditions, view the history of submissions including student searches, and it facilitates a defined set of bulk update capabilities.

10.5.1.4.3.2 Provider Management Portal

The Provider Management Portal is a Single Page Application (SPA) developed in Angular utilising Progressive Web Application (PWA) technologies.

The Provider Management Portal is the STARS management interface for activities such as registering provider organisations, provider users, and managing provider data.

10.5.1.4.4 Service Security

The service will conform to the Security Architecture. Access to the Provider Service will be controlled by Azure RBAC, with all calling services identified and granted access through Managed Identities.

10.5.2 Submission Pipeline Services

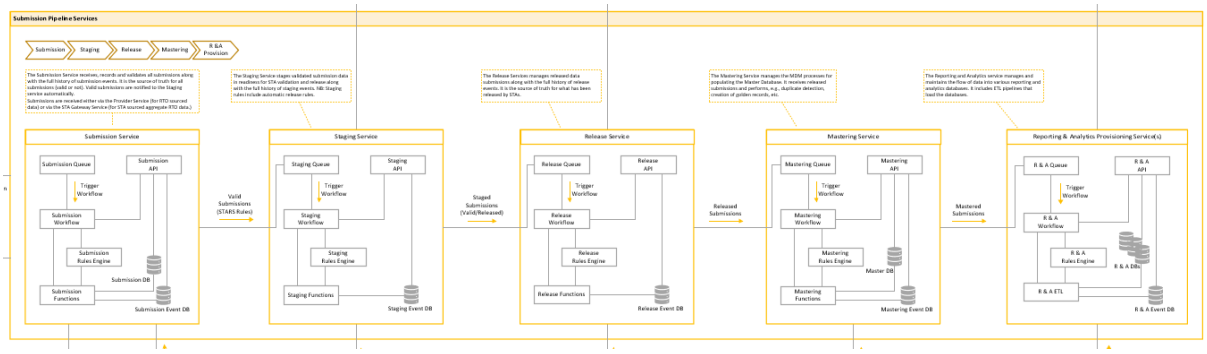


Figure 39: Submission Pipeline Services (B:12)

The Submission Pipeline Services are at the heart of STARS. They are responsible for receiving, validating, staging, releasing and mastering submissions received both from RTOs and STAs.

It underpins the main STARS business process pipelines:

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 131 of 1295

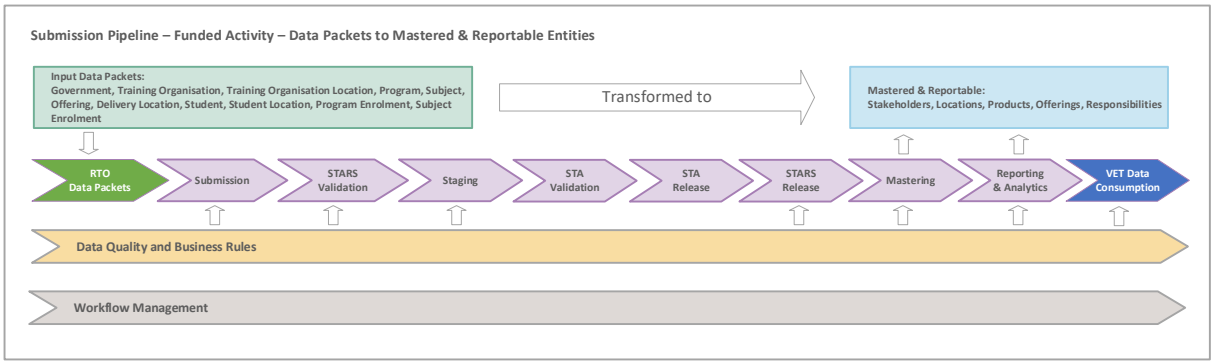


Figure 40: Funded Pipeline (B:3)

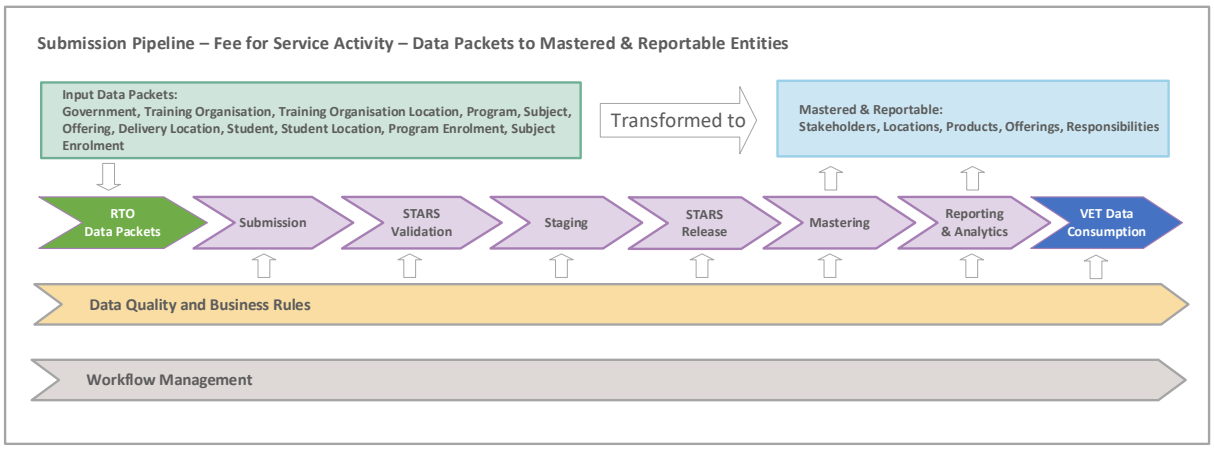


Figure 41: FFS Pipeline (B:3)

(Please refer to section 9.4 'Submission Pipelines' for further details)

The Submission Pipeline Services, along with the Validation Services support the main STARS business processes, described in detail in section 9.5 'Business Processes – To Be'. The BPMN diagrams that illustrate the processes include swim lanes to represent the various submission pipeline services along with their interactions:

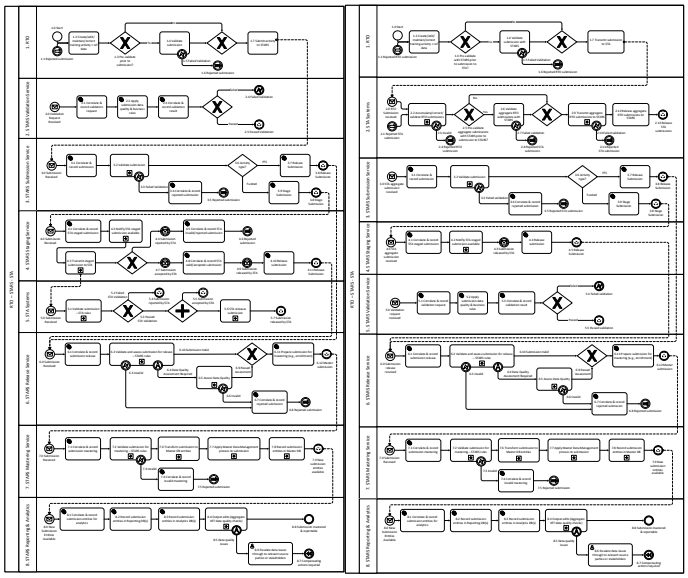


Figure 42: BPMN Diagrams

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 132 of 1295

10.5.2.1Submission Pipeline Service Interactions

All the submission pipeline services are loosely coupled – that is, they do not directly interact with each other when processing submission. Instead, each service contains an input message queue that the service constantly monitors. When an incoming message is received, the service acts on the message, performing its relevant function. When the service is finished it places a message in a queue reflecting the processing outcome. Note that the service is 'unaware' of what service sent the incoming message, nor what service will receive the outgoing message.

For example, as described further below, the Staging Service monitors a queue for notifications that a submission has been received and passed validation. The Staging Service extracts the URL of the submission from the message and uses the URL to retrieve the full submission information from the Submission Service. It then hands-off the submission to the relevant STA for further validation, processing, and release. When the Staging Service receives a release notification from the STA, it then creates a 'submission released' message (including the URL of the submission) and places it on an outbound queue. In this instance the outbound queue is configured as the inbound queue for the Release Service – the message sent by the Staging Service then triggers the Release workflows.

It should be noted that individual packets can be proceed through the submission pipeline independent of other packets. For example, if many SubjectEnrolment packets are received in a single Submission, those that pass validation can proceed further along the pipeline, while the ones that failed validation will not proceed. There are some cross-packet validation rules, which may affect progress of packets related to the same activity, though.

10.5.2.2Submission Service

Please refer to section 10.5.1.3 'Submission Service' for details.

The Submission Service plays a core role within the pipeline where it receives and processes raw information. It also provides submission information to other services both within the pipeline and elsewhere.

The submission receives submissions via the message bus from both the Provider Service (e.g., from RTOs) and from the STA Batch Submission service.

10.5.2.3Staging Service

Refer to section 11.1.6 'Staging Service' for a technical description.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 133 of 1295

10.5.2.3.1 Service Description

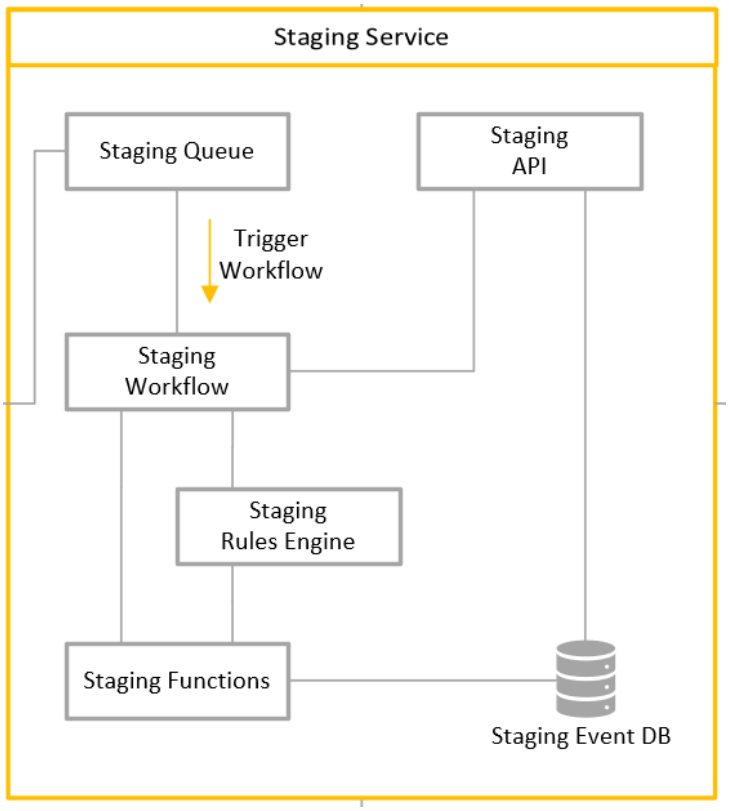


Figure 43: Staging Service (B:12)

The Staging Service manages the staging validated submission data in readiness for STA validation, processing and release. It's primary role is to orchestrate the exchange of submission data between the submission pipeline and STA systems, utilising STA-specific workflows and associated workflow business rules. This design refers to STAs performing these activities, but this role may be fulfilled by NCVER for Fee For Service data.

10.5.2.3.2 Service Components

10.5.2.3.2.1 Staging Queue

The Staging Service does not receive submissions directly; a message containing submission information is placed into the Staging Queue by the Submission Service once a submission is validated. This information will include the URL for the submission, allowing the submission data to be retrieved from the Submission Service.

The Staging Queue will be implemented within Azure Service Bus. It is likely that multiple queues will be utilised to support various STARS submission operational purposes (e.g., to support different VET-IS versions, or different STA business processes.) In this regard, the Staging Queue is a 'logical' construct that supports loose coupling between the Staging Service and any upstream services that are initiating submissions into the pipeline.

10.5.2.3.2.2 Staging Workflow

The Staging Service internal processes will be controlled by one or more Azure Logic App workflows.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 134 of 1295

The Staging Queue is monitored by the Submission Workflow Logic App(s). As submission information arrives in the queue, the workflow will record the submission information in the Staging Event DB, prior to initiating the relevant STA staging workflows.

The Staging Service will extract relevant submission data from the Submission Service, utilising the submission URL, in VET-IS JSON format. This data will be then sent to the relevant STA, via the STA Gateway Service. Depending on circumstances, the submissions may be transformed to AVETMISS 8 format. STAs may also require that submissions are batched along with other submissions relating to the current reporting period – in this instance the Staging Service will query the Submission Service for all related cumulative STA submissions for the period and send them as a batch to the STA via the STA Gateway Service.

The various scenarios for exchanging VET data with STAs during staging will be controlled by multiple Logic App workflows – these will be developed, updated and deployed throughout the life of the project and when STARS is in production.

When relevant, STAs will validate submissions using their own internal systems. Validation results will be transmitted back to the staging service, utilising the STA Gateway Service.

STAs will then release staged submissions, again through the STA Gateway Service – please note that release messages are handled by the Release Service (below).

Mechanisms will vary, but the current intent is that the STA Gateway Service will call the Staging API (below) to inform the Staging Service of STA validation results. The gateway will have a consistent interface on the STARS side (i.e., it sends and receives messages using STARS schemas, transports and protocols), but it has various adaptors on the STA side to facilitate data and message exchange with STA internal systems.

The details are scenario specific, but generally STA validation results are returned to the upstream service via ‘call-back’ URLs hosted within the upstream service, with the call-backs uniquely associated with each input submission.

When the Staging Service receives a successful validation notification from the STA, it then creates a ‘submission validated message (including the URL of the submission) and places it on an outbound queue. In this instance the outbound queue is configured as the inbound queue for the Release Service – the message sent by the Staging Service then triggers the Release workflows (below).

10.5.2.3.2.3 Staging Functions

The Staging Service code-based functionality will be implemented as Azure Function Apps. These functions will support the operation of the service’s workflows as well as functionality provided through the Pipeline Management Portal.

10.5.2.3.2.4 Staging Rules Engine

The Staging Rules Engine, and any associated rules, is intended to assist with the operation of the Staging Workflows. Please note that this does not include submission validation rules – these are handled by the Validation Service. The staging rules engine role is to apply any business rules associated with controlling the flow of VET data between STARS and STA.

10.5.2.3.2.5 Staging API

The Staging API will be implemented as an Azure Function App, utilising HTTP endpoints.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 135 of 1295

The API will allow other services to flexibly query and extract staging data in JSON format, including information on the staging lifecycle data, including STA submission validation and release status.

The API will be defined during Work Package 2, pursuant to the design and implementation of the Staging Event DB and associated VET-IS SQL and JSON schemas.

Care will be taken to ensure staging data can be presented via the API to workflows and the rules engine(s) in a format to allow rapid processing of rules – please see the discussion in section 10.4 'Rules Engine'.

The API will conform to the STARS Design Principles and the STARS API Design Document.

10.5.2.3.2.6 Staging Event DB

All service events relating to the lifecycle and management of staging and staged data will be recorded, in the STARS Event schema, in Cosmos DB.

10.5.2.3.3 Service Interfaces

HTTP/REST/JSON: See Staging API, above. Staging data will be queryable through the API, using REST semantics. Refer to the STARS API documentation for more detail.

SQL: See Staging Database, above. The Staging DB will be queryable via Cosmos DB SQL DDL statements, depending on the calling user/service's access.

10.5.2.3.4 Service Security

The service will conform to the Security Architecture. Access to the Staging Service will be controlled by Azure RBAC, with all calling services identified and granted access through Managed Identities.

10.5.2.4 Release Service

Refer to section 11.1.7 'Release Service' for a technical description.

DXC TECHNOLOGY
Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 136 of 1295

10.5.2.4.1 Service Description

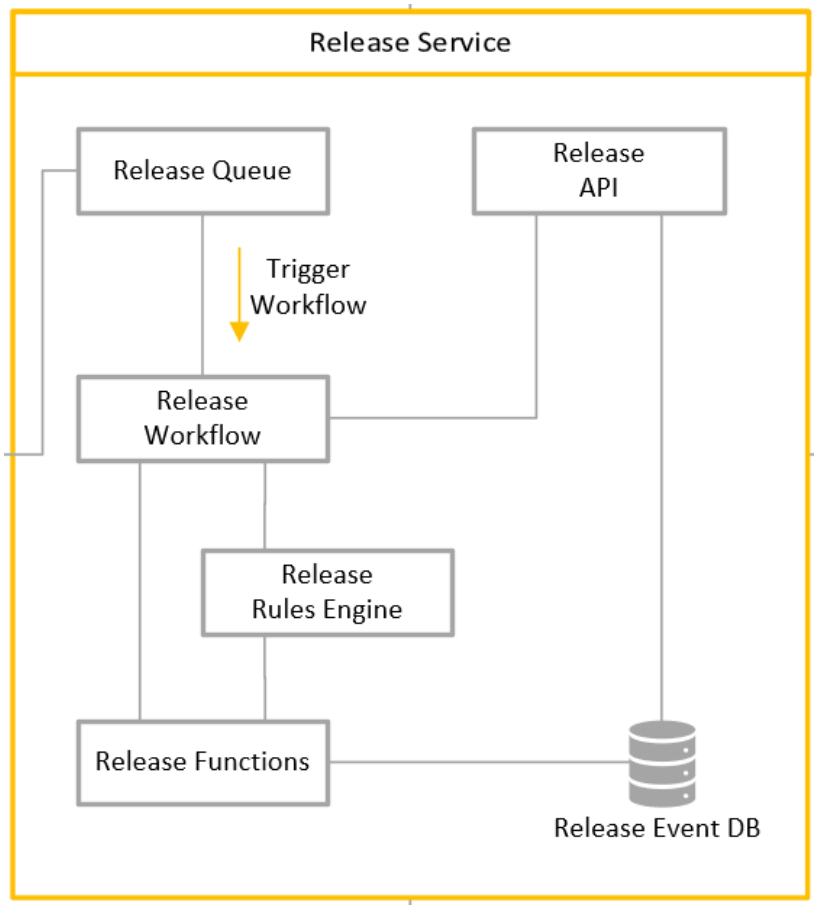


Figure 44: Release Service (B:12)

The Release Services manages released data submissions along with the full history of release events. It is the source of truth for what has been released by STAs.

The Release Service receives messages from the Staging Service about submissions that have been successfully validated by STAs. However, the Release Service will not process the submission until it has received a release message from the STA, noting that some STAs will wish to have submissions automatically released once they are validated.

Where an explicit release is required, the Release Workflows will raise automatic alerts to STARS System Operators and STA System Operators when a release is overdue.

10.5.2.4.2 Service Components

10.5.2.4.2.1 Release Queue

The Release Service does not receive submissions directly; a message containing submission information is placed into the Release Queue by the Staging Service once a submission is validated by an STA. This information will include the URL for the submission, allowing the submission data to be retrieved from the Submission Service.

The Release Queue will be implemented within Azure Service Bus. It is likely that multiple queues will be utilised to support various STARS submission operational purposes (e.g., to support different VET-IS versions, or different STA business processes.) In this regard, the Release Queue is a 'logical'

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 137 of 1295

construct that supports loose coupling between the Release Service and any upstream services that are initiating submissions into the pipeline.

10.5.2.4.2.2 Release Workflow

The Release Service internal processes will be controlled by one or more Azure Logic App workflows.

The Release Queue is monitored by the Submission Workflow Logic App(s). As submission information arrives in the queue, the workflow will record the submission information in the Release Event DB, prior to initiating the relevant release workflows.

The various scenarios for releasing VET data by STAs will be controlled by multiple Logic App workflows – these will be developed, updated and deployed throughout the life of the project and when STARS is in production.

When desired, submissions will be automatically released, depending on STA requirements.

Otherwise, the receipt of a submission message from the Staging Service, the Release Workflow will initiate configurable STA release deadline timers. When the timers have expired and a release message has not been received from an STA, the Release Workflow will automatically raise an alert to the relevant STARS system operators and STA system operators (via the Notification Service.)

STAs release staged submissions through the STA Gateway Service.

Mechanisms will vary, but the current intent is that the STA Gateway Service will call the Release API (below) to inform the Release Service of STA releases. The gateway will have a consistent interface on the STARS side (i.e., it sends and receives messages using STARS schemas, transports and protocols), but it has various adaptors on the STA side to facilitate data and message exchange with STA internal systems.

When the Staging Service receives a release notification from the STA, it then creates a ‘submission released’ message (including the URL of the submission) and places it on an outbound queue. In this instance the outbound queue is configured as the inbound queue for the Mastering Service – the message sent by the Release Service then triggers the Mastering workflows (below).

10.5.2.4.2.3 Release Functions

The Release Service code-based functionality will be implemented as Azure Function Apps. These functions will support the operation of the service’s workflows as well as functionality provided through the Pipeline Management Portal.

10.5.2.4.2.4 Release Rules Engine

The Release Rules Engine, and any associated rules, is intended to assist with the operation of the Release Workflows. Please note that this does not include submission validation rules – these are handled by the Validation Service. The release rules engine role is to apply any business rules associated with controlling the flow of VET data between STARS and STA, and to identify any data quality issues with released data prior to mastering.

10.5.2.4.2.5 Release API

The Release API will be implemented as an Azure Function App, utilising HTTP endpoints.

The API will allow other services to flexibly query and extract release data in JSON format, including information on the release lifecycle data, including STA release status.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 138 of 1295

The API will be defined during Work Package 2, pursuant to the design and implementation of the Release Event DB and associated VET-IS SQL and JSON schemas.

Care will be taken to ensure release data can be presented via the API to workflows and the rules engine(s) in a format to allow rapid processing of rules – please see the discussion in section 10.4 'Rules Engine'.

The API will conform to the STARS Design Principles and the STARS API Design Document.

10.5.2.4.2.6 Release Event DB

All service events relating to the lifecycle and management of release and release meta data will be recorded, in the STARS Event schema, in Cosmos DB.

10.5.2.4.3 Service Interfaces

HTTP/REST/JSON: See Release API, above. Release lifecycle data will be queryable through the API, using REST semantics. Refer to the STARS API documentation for more detail.

SQL: See Release Database, above. The Release DB will be queryable via Cosmos DB SQL DDL statements, depending on the calling user/service's access.

10.5.2.4.4 Service Security

The service will conform to the Security Architecture. Access to the Release Service will be controlled by Azure RBAC, with all calling services identified and granted access through Managed Identities.

10.5.2.5 Mastering Service

Refer to section 11.1.8 'Mastering Service' for a technical description.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 139 of 1295

10.5.2.5.1 Service Description

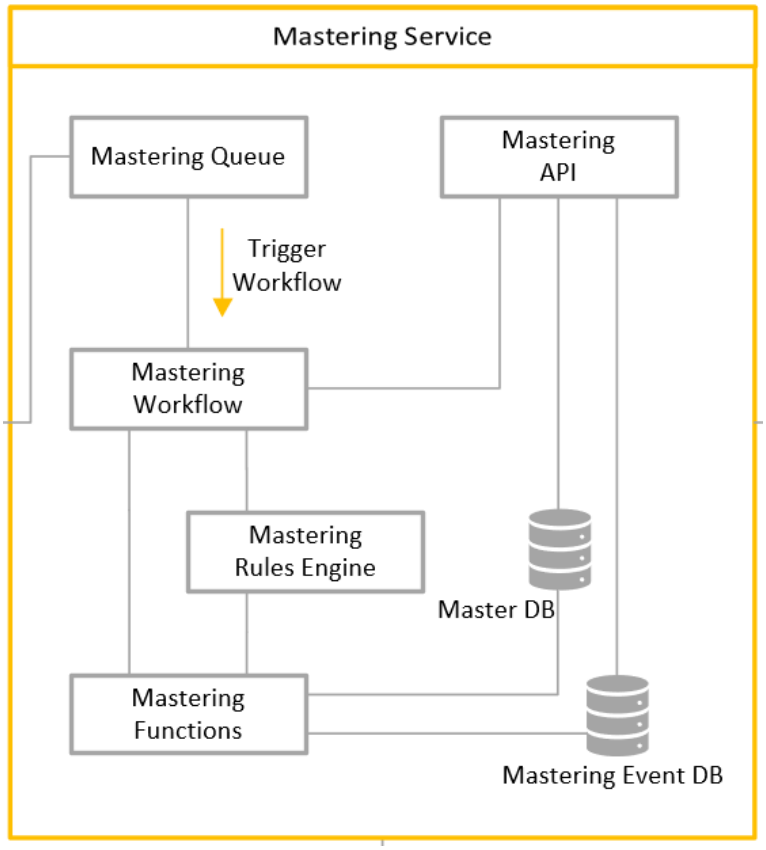


Figure 45: Mastering Service (B:12)

Note that the Mastering Service is distinct from the Master Data Service.

The Mastering Service's primary role is to manage the MDM processes for populating the Master Database. It includes workflows that manage and process silver-level data into the Master STA partitions and manage and create gold-level data into the Master National partition.

The gold-level oriented workflows involve activities such duplicate detection, data merging, creation of golden records, etc. These workflows are likely to involve some manual oversight and will take time to complete.

The silver-level oriented workflows will be designed to operate quickly. This is so as to allow STAs faster access to their mastered data to facilitate their internal operational requirements.

Please see sections 8.5 'Master Data Management Approach' and 8.14.6 'Master Database' for more details of the processes and data managed by the Mastering Service.

10.5.2.5.2 Service Components

10.5.2.5.2.1 Mastering Queue

The Mastering Service does not receive submissions directly upstream services; information about released submissions are placed into the Mastering Queue by the Release Service, including the URLs in the Submission Service for the released submissions. The URLs are used by the Mastering Service to extract submission data from the Submission Service, to be mastered.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 140 of 1295

The Mastering Queue will be implemented within Azure Service Bus. It is likely that multiple queues will be utilised to support various STARS submission operational purposes (e.g., to support different VET-IS versions, or different STA or NCVER business processes.) In this regard, the Mastering Queue is a 'logical' construct that supports loose coupling between the Mastering Service and any upstream services that are processing submissions in the pipeline.

10.5.2.5.2.2 Mastering Workflow

The Mastering Service internal processes will be controlled by one or more Azure Logic App workflows.

The Mastering Queue is monitored by the Mastering Workflow Logic App(s). As submission information arrive in the queue, the workflow will record the submission in the Mastering Event DB, prior to initiating the relevant mastering workflows.

Broadly, the mastering workflows will be in two halves; one oriented to production of gold-level data and one oriented to production of silver-level data.

The gold-level workflows will be oriented around managing the flow of submission data into the various MDM pipelines and processes, described in section 8.5 'Master Data Management Approach', with gold-level national data appearing in the Master – National database partition.

The silver-level workflows will be oriented around managing the flow of submission data directly into the relevant Master – STA database partitions. The emphasis is on speed of processing, with submission data only being transformed to the extent required for consistency with STA business requirements and data presentation.

When submissions have been successfully mastered, the Mastering Service raises a message to the R & A Provisioning Service.

Mastering failures or exceptions will be result in automatic alerts sent to STARS system operators (via the Notifications Service) for manual intervention.

10.5.2.5.2.3 Mastering Functions

The Mastering Service code-based functionality will be implemented as Azure Function Apps. These functions will support the operation of the service's workflows as well as functionality provided through the Pipeline Management Portal.

10.5.2.5.2.4 Mastering Rules Engine

The Mastering Rules Engine, and any associated rules, is intended to assist with the operation of the Mastering Workflows. The rules engine will assist with both applying any business rules associated with processing of submissions and any bulk MDM data quality rules processing.

10.5.2.5.2.5 Mastering API

The Mastering API will be implemented as an Azure Function App, utilising HTTP endpoints.

The API will allow other services to flexibly query and extract mastering data in JSON format, including information on mastering lifecycle data, including the status of submissions through the MDM processes.

The API will be defined during Work Package 2, pursuant to the design and implementation of the Mastering Event DB and associated VET-IS SQL and JSON schemas.

DXC TECHNOLOGY
Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 141 of 1295

Care will be taken to ensure mastering data can be presented via the API to workflows and the rules engine(s) in a format to allow rapid processing of rules – please see the discussion in section 10.4 'Rules Engine'.

The API will conform to the STARS Design Principles and the STARS API Design Document.

10.5.2.5.2.6 Master DB

Please see sections 8.14.6 'Master Database' and 10.5.1.1 'Master Data Service'.

10.5.2.5.2.7 Mastering Event DB

All service events relating to the lifecycle and management of mastering submissions will be recorded, in the STARS Event schema, in Cosmos DB.

10.5.2.5.3 Service Interfaces

HTTP/REST/JSON: See Mastering API, above. Mastering lifecycle data will be queryable through the API, using REST semantics. Refer to the STARS API documentation for more detail.

SQL: See Submission Database, above. The Submission DB will be queryable via Cosmos DB SQL DDL statements, depending on the calling user/service's access.

10.5.2.5.4 Service Security

The service will conform to the Security Architecture. Access to the Mastering Service will be controlled by Azure RBAC, with all calling services identified and granted access through Managed Identities.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 142 of 1295

10.5.2.6Reporting and Analytics Provisioning Service

See section 11.1.9 ‘Reporting and Analytics Provisioning Service’ for a technical description.

10.5.2.6.1 Service Description

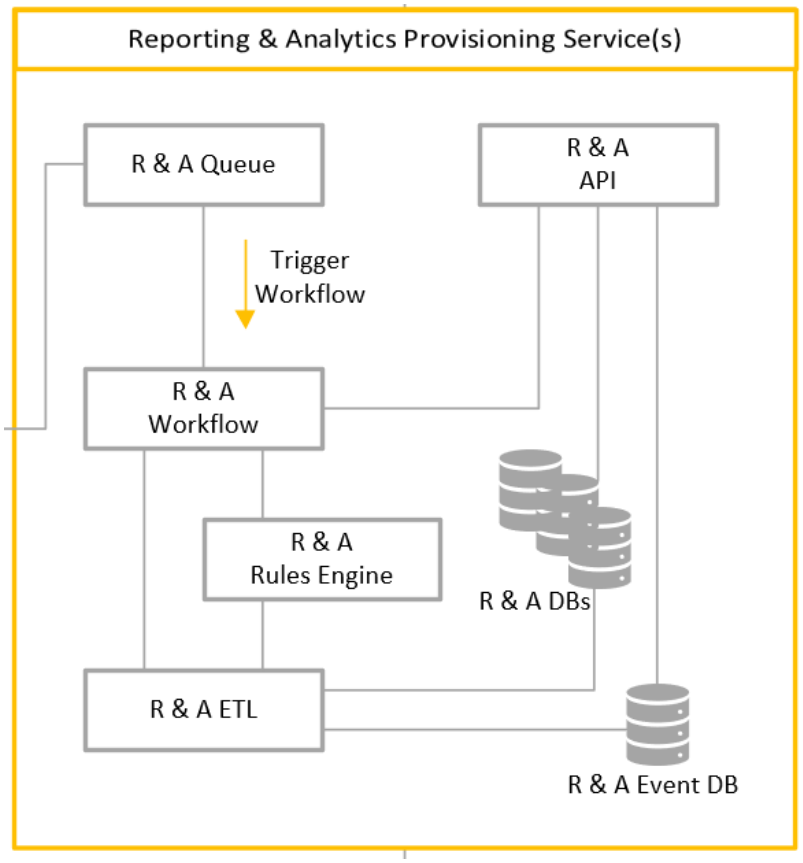


Figure 46: Reporting and Analytics Provisioning Service(s) (B:12)

Note that the Reporting and Analytics Provisioning Service is distinct from the Reporting and Analytics Services.

The R & A Provisioning Service’s primary role is to manage the ETL processes for populating the various reporting and analytics databases. The service receives notification messages that data has been mastered, which may result in one or more ETL pipelines being initiated to extract, transform and load mastered data into downstream data consumer reporting and analytics databases.

Please see section 8.9 ‘Reporting and Analytics Approach’ for more details of the processes and data managed by the R & A Provisioning Service.

10.5.2.6.2 Service Components

10.5.2.6.2.1 R & A Queue

The R & A Provisioning Service does not receive mastered information directly from upstream services; information about mastered data is placed into the R & A Provisioning Queue by the Mastering Service, including the URLs in the Mastering Service about the mastering metadata. The URLs are used by the R & A Provisioning Service to extract mastering metadata from the Mastering Service, to determine if or when a provisioning workflow is to be initiated.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 143 of 1295

The R & A Provisioning Queue will be implemented within Azure Service Bus. It is likely that multiple queues will be utilised to support various STARS submission operational purposes (e.g., to support different VET-IS versions, or different STA or NCVER business processes.) In this regard, the R & A Provisioning Queue is a 'logical' construct that supports loose coupling between the R & A Provisioning Service and any upstream services that are mastering data in the pipeline.

10.5.2.6.2.2 R & A Workflow

The R & A Provisioning Service internal processes will be controlled by one or more Azure Logic App workflows.

The R & A Queue is monitored by the R & A Provisioning Workflow Logic App(s). As mastering metadata information arrive in the queue, the workflow will record the metadata in the R & A Event DB, prior to initiating the relevant provisioning workflows.

The workflows will initiate and monitor Data Factory ETL pipelines that will extract, transform and load relevant mastered data into downstream data consumer reporting and analytics databases.

Provisioning failures or exceptions will be result in automatic alerts sent to STARS system operators (via the Notifications Service) for manual intervention.

10.5.2.6.2.3 R & A Functions

The R & A Provisioning Service code-based functionality will be implemented as Azure Function Apps. These functions will support the operation of the service's workflows as well as functionality provided through the Pipeline Management Portal.

10.5.2.6.2.4 R & A Provisioning Rules Engine

The R & A Provisioning Rules Engine, and any associated rules, is intended to assist with the operation of the R & A Provisioning Workflows. The rules engine will assist with both applying any business rules associated with provisioning R & A data and any bulk data quality rules processing.

10.5.2.6.2.5 R & A Provisioning API

The R & A Provisioning API will be implemented as an Azure Function App, utilising HTTP endpoints.

The API will allow other services to flexibly query and extract provisioning data in JSON format, including information on provisioning lifecycle data, including the status of data through the provisioning processes.

The API will be defined during Work Package 2, pursuant to the design and implementation of the R & A Event DB, associated VET-IS SQL and JSON schemas, and when R & A requirements are determined.

The API will conform to the STARS Design Principles and the STARS API Design Document.

10.5.2.6.2.6 R & A DBs

Please see section 8.9 'Reporting and Analytics Approach'.

10.5.2.6.2.7 R & A Event DB

All service events relating to the lifecycle and management of provisioning R & A data will be recorded, in the STARS Event schema, in Cosmos DB.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 144 of 1295

10.5.2.6.3 Service Interfaces

HTTP/REST/JSON: See Mastering API, above. Mastering lifecycle data will be queryable through the API, using REST semantics. Refer to the STARS API documentation for more detail.

SQL: See Submission Database, above. The Submission DB will be queryable via Cosmos DB SQL DDL statements, depending on the calling user/service's access.

10.5.2.6.4 Service Security

The service will conform to the Security Architecture. Access to the Mastering Service will be controlled by Azure RBAC, with all calling services identified and granted access through Managed Identities.

10.5.3 Decision Services

Refer to section 11.1.10 'Decision Services' for a technical description.

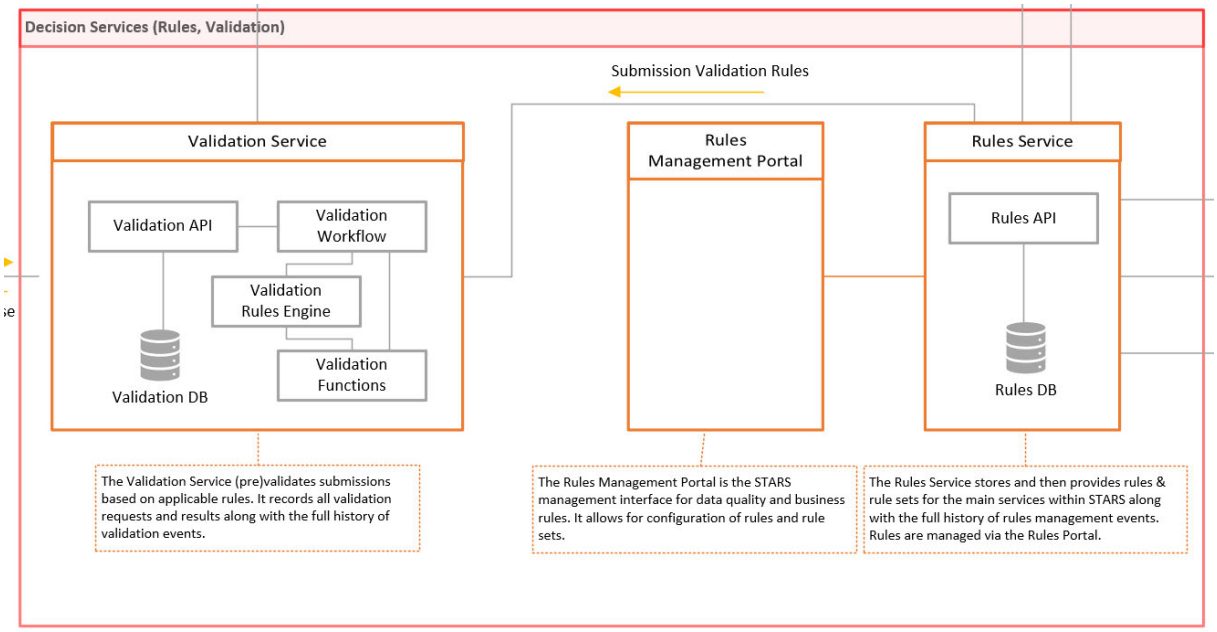


Figure 47: Decision Services (B:12)

The Decision Services provide the means for Submission Pipeline Services, and their embedded workflows, to control the flow of information within the system. These include both submission validation services and rules engine services.

DXC TECHNOLOGY
Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 145 of 1295

10.5.3.1 Rules Services

10.5.3.1.1 Service Description

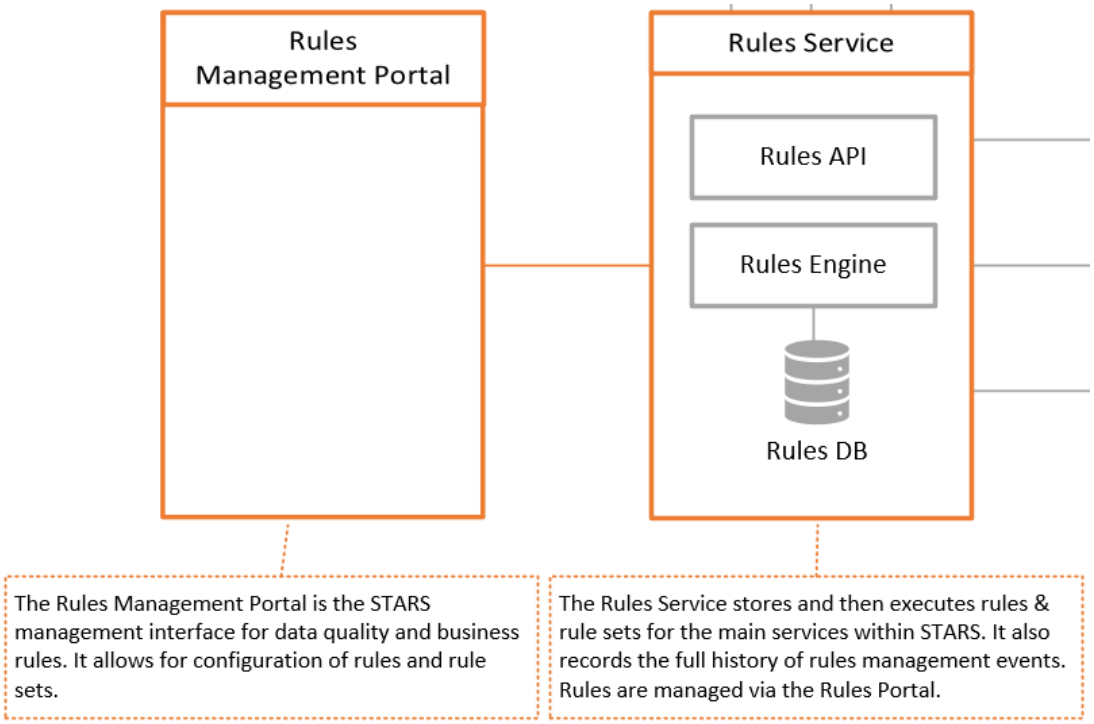


Figure 48: Rules Service (B:12)

The Rules Service manages and provides data quality and business rules and rule sets for the main services within STARS along with the full history of rules management events. Rules are managed via the Rules Management Portal. The Rules Service encapsulates the STARS Rules Engine, which executes rules on behalf of client solution services.

The Rules Management Portal is the STARS management for data quality and business rules. It allows for configuration of rules and rule sets.

Please see section 9.6.2 ‘Rules Engine’ for a discussion on how the STARS Rules Engine is configured and used.

The Rules Service functionality will encapsulate that of the STARS Rules Engine (*NRule/Drools*).

10.5.3.1.2 Service Components

10.5.3.1.2.1 Rules API

The Rules API will be implemented as an Azure Function App, utilising HTTP endpoints. The API will be defined during Work Package 2, pursuant to the choice, design and implementation of the STARS Rules Engine.

The API will present endpoints for submitting data for execution during rules sessions and for querying the results of rules sessions.

The API will conform to the STARS Design Principles and the STARS API Design Document.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 146 of 1295

10.5.3.1.2.2 Rules Engine

The Rules Engine executes rule sets against VET data provided through the Rules API.

Please see section 9.6.2 'Rules Engine' for a discussion on how the STARS Rules Engine is configured and used.

10.5.3.1.2.3 Rules DB

The Rules DB represents the rules repository for the STARS Rules Engine. It will contain rule and rule set configuration information as well as rule DSL scripts.

10.5.3.1.3 Service Interfaces

HTTP/REST/JSON: See Rules API, above. Rules configuration data and data to be evaluated will be managed through the API, using REST semantics. Refer to the STARS API documentation for more detail.

10.5.3.1.4 Service Security

The service will conform to the Security Architecture. Access to the Rules Service will be controlled by Azure RBAC, with all calling services identified and granted access through Managed Identities.

10.5.3.2Validation Service

10.5.3.2.1 Service Description

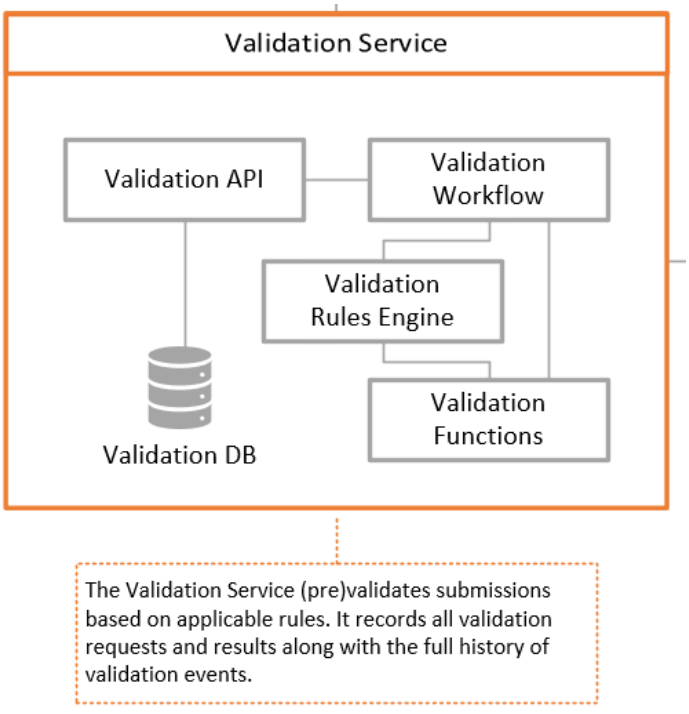


Figure 49: Validation Service (B:12)

The Validation Service (pre)validates submissions based on applicable rules (sourced from and run by the Rules Service). It records all validation requests and results along with the full history of validation events.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 147 of 1295

The Validation Service validates submissions using the STARS Rules Engine, but also call out to external STA-based validation services. External validation is controlled by the Validation Workflows, with specific workflows configured to cater for different STA external validation requirements.

Note that Validation Service managed external validation is separate and distinct from that managed by the Staging Service. The Validation Service external STA validation scenario applies to single submissions and are made in real-time through suitable STA webservices. The Staging Service STA validations are based around batched submissions that may take an extended period to process and then return results.

10.5.3.2.1.1 Validation API

The Validation API will be implemented as an Azure Function App, utilising HTTP endpoints.

The API will allow other services to present submission information in JSON format for submission validation. The information will include the URLs of the relevant submission data held within the Submission Service, to allow retrieval of the submission data during validation processing.

The API will be defined during Work Package 2, pursuant to the design and implementation of the Validation DB and associated VET-IS SQL and JSON schemas.

Care will be taken to ensure validation requests and responses can be presented via the API to workflows to allow rapid processing of submission validation requests.

The API will conform to the STARS Design Principles and the STARS API Design Document.

10.5.3.2.1.2 Validation Workflow

The Validation Service internal processes will be controlled by one or more Azure Logic App workflows.

Upon receipt of a validation request by the Validation API, a workflow will be triggered. The workflow will examine the metadata within the request to determine the nature of the request and decide which national and/or STA validation workflows apply.

The validation workflows will route the request through the appropriate data quality and business rule sets, managed and executed by the Rules Service. During this process, the workflows will retrieve the relevant submission data from the Submission Service, passing the data into the Rules Service via the Rules API.

When the Rules Service has finished and returned the rules execution results, the workflows collate and format the results in a form suitable for reporting back to the calling service.

When required, the validation workflows can also route submission data to STA validation web services, via the STA Service Gateway. Returned STA validation results will also be collated with the results of any STARS-hosted rules.

When relevant, STAs will validate submissions using their own internal systems. Validation results will be transmitted back to the staging service, utilising the STA Gateway Service.

Mechanisms will vary, but the current intent is that the STA Gateway Service will call the Validation API (below) to inform the Validation Service of STA validation results. The gateway will have a consistent interface on the STARS side (i.e., it sends and receives messages using STARS schemas,

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 148 of 1295

transports and protocols), but it has various adaptors on the STA side to facilitate data and message exchange with external STA validation systems.

The details are scenario specific, but generally validation results are returned to the upstream service via 'call-back' URLs hosted within the upstream service, with the call-backs uniquely associated with each input submission request.

10.5.3.2.1.3 Validation Functions

The Validation Service code-based functionality will be implemented as Azure Function Apps. These functions will support the operation of the service's workflows.

10.5.3.2.1.4 Validation Rules Engine

The Validation Rules Engine, and any associated rules, is intended to assist with the operation of the Validation Workflows. Please note that this does not include submission validation rules – these are handled by the Rules Service directly. The validation rules engine role is to apply any business rules associated with controlling the flow of VET data between STARS and STA.

10.5.3.2.1.5 Validation DB

All service events relating to the lifecycle and management of validation requests and validation results will be recorded, in the STARS Event schema, in Cosmos DB.

10.5.3.2.2 Service Interfaces

HTTP/REST/JSON: See Validation API, above. Validation requests and results will be made through the API, using REST semantics. Refer to the STARS API documentation for more detail.

SQL: See Validation Database, above. The Validation DB will be queryable via Cosmos DB SQL DDL statements, depending on the calling user/service's access.

10.5.3.2.3 Service Security

The service will conform to the Security Architecture. Access to the Validation Service will be controlled by Azure RBAC, with all calling services identified and granted access through Managed Identities.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 149 of 1295

10.5.4 STA Services

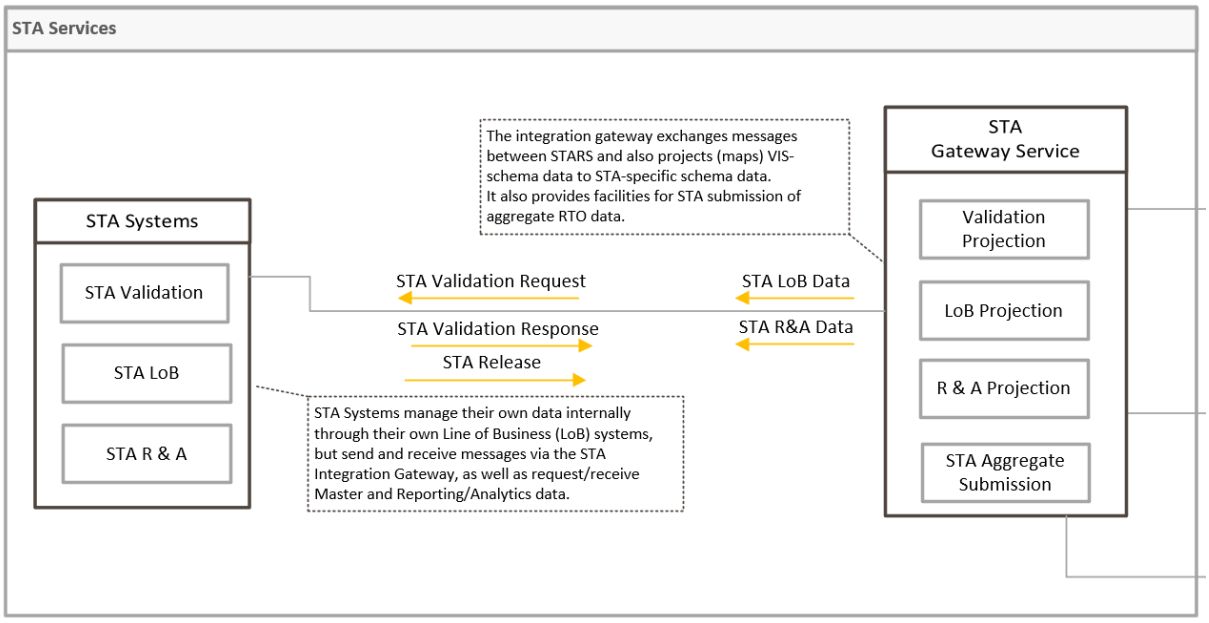


Figure 50: STA Service (B:12)

STA Services are provided to facilitate the exchange of VET data between STARS and STA systems. STA integration and data exchange detailed requirements are not yet provided, accordingly the discussions and descriptions in this section are general in nature.

10.5.4.1STA Gateway Service

10.5.4.1.1 Service Description

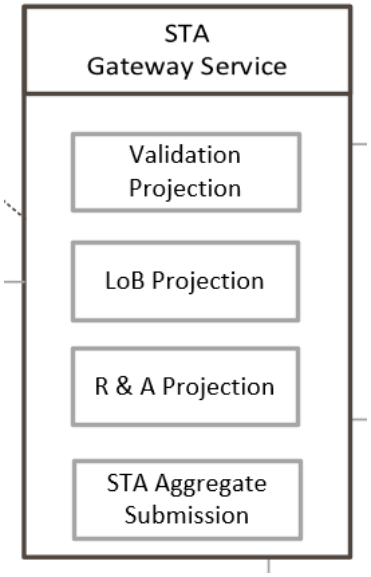


Figure 51: STA Gateway Service (B:12)

The STARS solution design currently proposes that, in the absence of requirements, that a 'logical' service model be identified to address some of the 'known-unknowns' surrounding integration and data exchange between STARS and STA systems.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 150 of 1295

This service model is based on implementing one or more 'gateway' services that can translate and transform STARS messages and data into formats that can be understood by STA systems, and vice-versa.

There are two broad areas to consider:

Message Interchange

It is likely that STA systems will be heterogenous across jurisdictions regarding technology stacks and capabilities. In this case, it is proposed that the Gateway Service conforms to STARS messaging and webservice transports and protocols on the STARS side and converts information/messages to and from STARS format into the relevant STA formats, delivered through the STA transports.

Data Interchange

STAs have indicated that it will take time to transform their systems to utilise VET-IS format data, and to be able to receive and process data in real time. For the short-to-medium term, it is likely that STA systems will require batched data in AVETMISS format. To this end, STARS will provide ETL-style data delivery pipelines that convert STARS internal VET-IS format data to batched AVETMISS data, for transport to STAs.

The STA transports and protocols are yet to be specified; however, it is most likely that STARS will deliver data to STAs using Azure Data Factory ETL pipelines.

10.5.5 Data Consumption Services

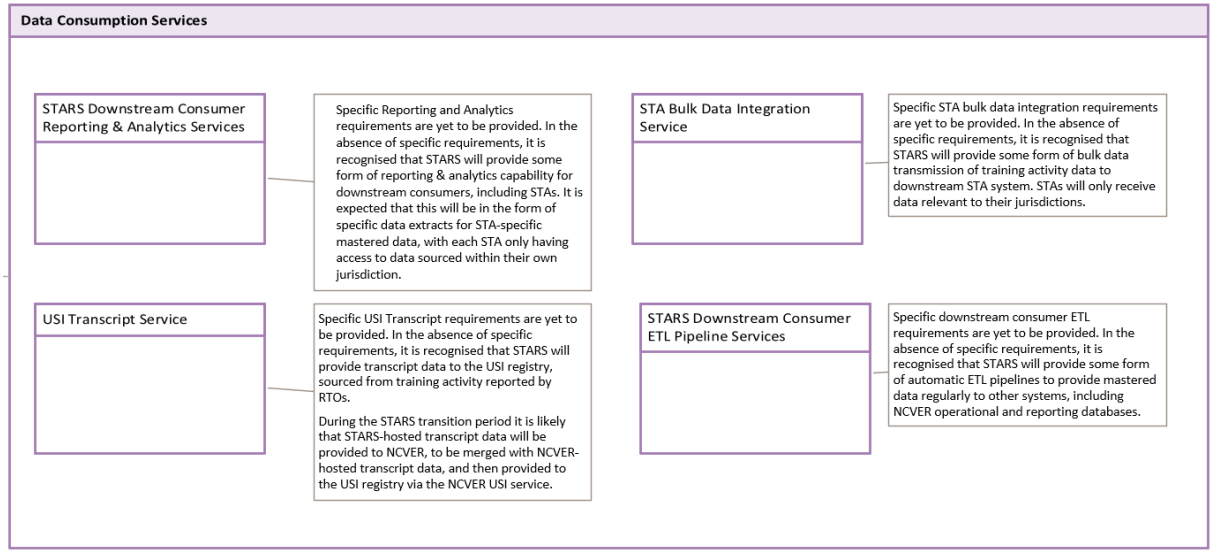


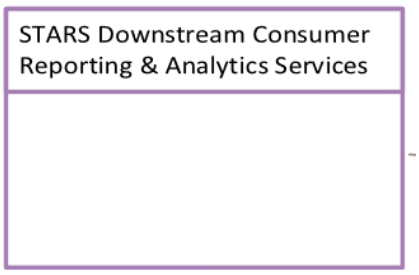
Figure 52: Data Consumption Services (B:12)

Data Consumption Services are provided to facilitate the exchange of VET data between STARS and downstream VET data consumers and systems. Data consumption detailed requirements are not yet provided, accordingly the discussions and descriptions in this section are general in nature.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 151 of 1295

10.5.5.1 Downstream Data Consumer Reporting and Analytics Services

10.5.5.1.1 Service Description

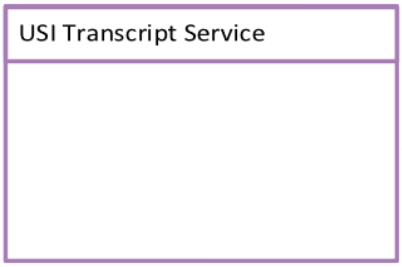


Specific Reporting and Analytics requirements are yet to be provided. In the absence of specific requirements, it is recognised that STARS will provide some form of reporting & analytics capability for downstream consumers, including STAs. It is expected that this will be in the form of specific data extracts for STA-specific mastered or staged data, with each STA only having access to data sourced within their own jurisdiction.

Figure 53: Downstream Consumer Reporting & Analytics Services (B:12)

10.5.5.2 USI Transcript Service

10.5.5.2.1 Service Description



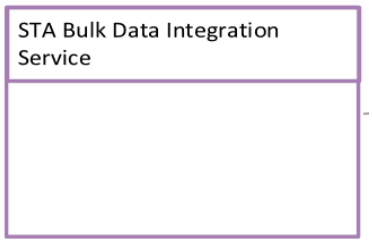
Specific USI Transcript requirements are yet to be provided. In the absence of specific requirements, it is recognised that STARS will provide transcript data to the USI registry, sourced from training activity reported by RTOs.

During the STARS transition period it is likely that STARS-hosted transcript data will be provided to NCVER, to be merged with NCVER-hosted transcript data, and then provided to the USI registry via the NCVER USI service.

Figure 54: USI Transcript Service

10.5.5.3 STA Bulk Data Integration Service

10.5.5.3.1 Service Description

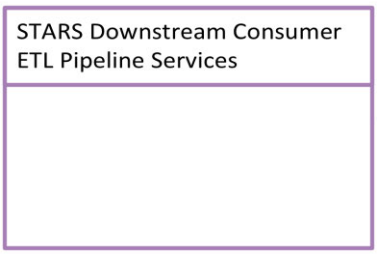


Specific STA bulk data integration requirements are yet to be provided. In the absence of specific requirements, it is recognised that STARS will provide some form of bulk data transmission of training activity data to downstream STA system. STAs will only receive data relevant to their jurisdictions.

Figure 55: STA Bulk Data Integration Service

10.5.5.4 Data Consumer ETL Pipeline Services

10.5.5.4.1 Service Description



Specific downstream consumer ETL requirements are yet to be provided. In the absence of specific requirements, it is recognised that STARS will provide some form of automatic ETL pipelines to provide mastered or staged data regularly to other systems, including NCVER operational and reporting databases.

Figure 56: Downstream Consumer ETL Pipeline Services (B:12)

DXC TECHNOLOGY
Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 152 of 1295

10.5.6 Support Services

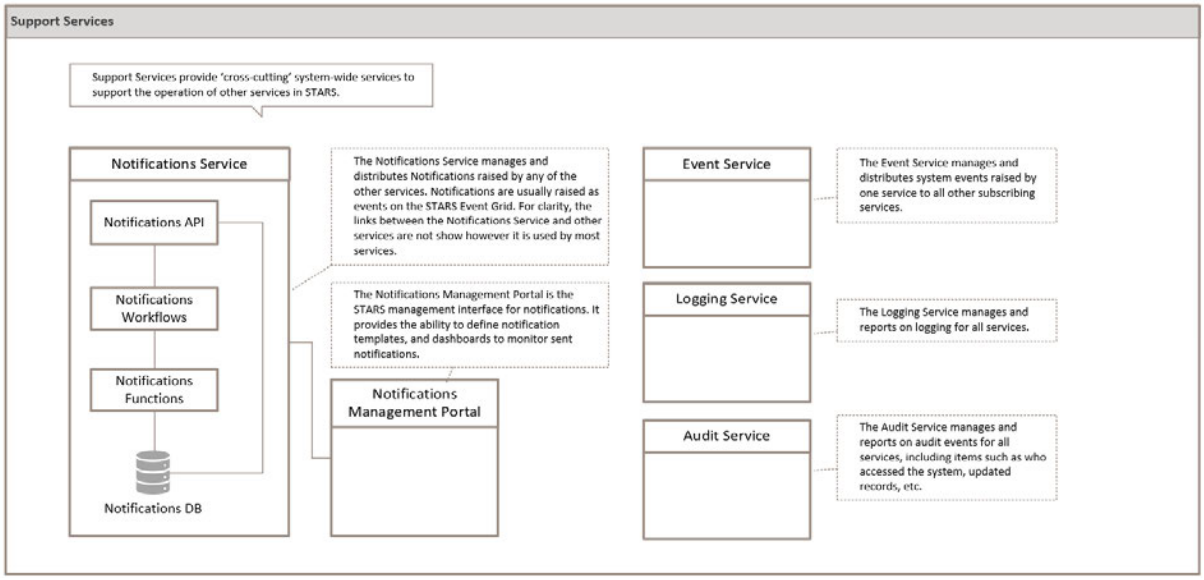


Figure 57: Support Services (B:12)

Support Services provide 'cross-cutting' system-wide services to support the operation of other services in STARS.

10.5.6.1 Notifications Service

Refer to section 11.1.4 'Notification Service' for a technical description.

10.5.6.1.1 Service Description

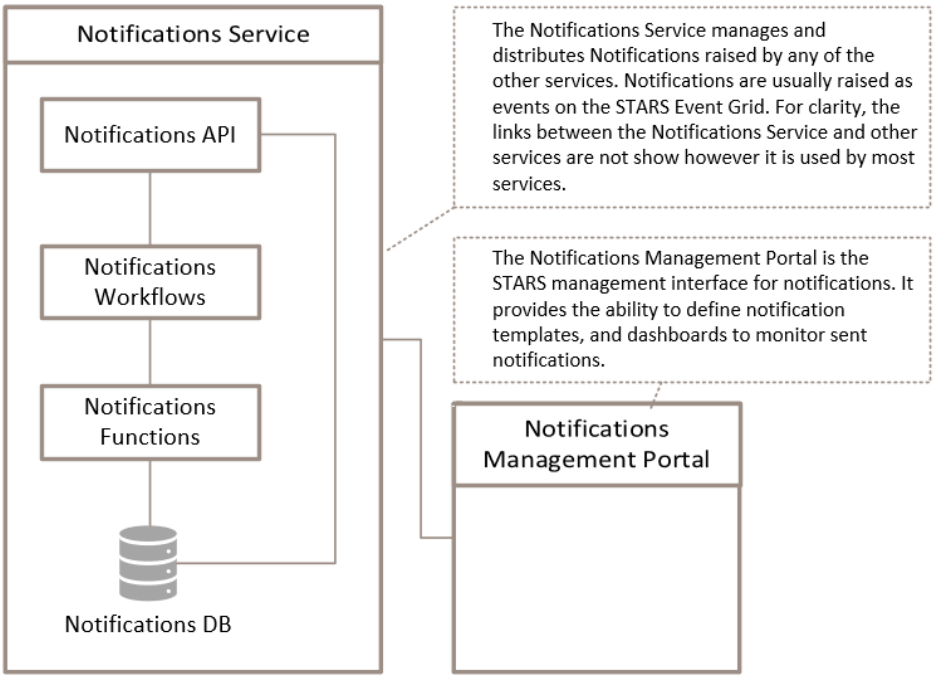


Figure 58: Notifications Service (B:12)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 153 of 1295

The Notifications Service manages and distributes Notifications raised by any of the other services. Notifications are usually raised as events on the STARS Event Grid. For clarity, the links between the Notifications Service and other services are not show however it is used by most services.

The Notifications Management Portal is the STARS management interface for notifications. It provides the ability to define notification templates, and dashboards to monitor sent and received notifications.

10.5.6.1.2 Service Components

10.5.6.1.2.1 Notifications Workflow

The Notification Service internal processes will be controlled by one or more Azure Logic App workflows.

Notification workflows will be triggered either by direct calls to the Notifications API, through notification events raised in the Event Service (Azure Event Grid), or on a schedule. Scheduled notifications will work with the Rules Service to provide a way for time-based notifications to be identified and actioned.

The workflows will manage the creation and transmission of notifications to the receiving parties, through the various transports and protocols (e.g., via email, SMS, or to the Provider Portal).

10.5.6.1.2.2 Notification Functions

The Notification Service code-based functionality will be implemented as Azure Function Apps. These functions will support the operation of the service's workflows as well as functionality provided through the Notifications Management Portal. For example, the portal will have functions to manage notification templates, the actual code to implement this at the database level will be within the Function Apps.

10.5.6.1.2.3 Notifications API

The Notifications API will be implemented as an Azure Function App, utilising HTTP endpoints.

The API will allow other services to flexibly query and provide notification data in JSON format, including information on the notifications themselves and all the notifications lifecycle data, including transmission and receipt status.

The API will also be used by the Management Portal user interfaces to service UI requests for notifications management.

The API will be defined during Work Package 2, pursuant to the design and implementation of the Notifications DB and associated JSON schemas.

The API will conform to the STARS Design Principles and the STARS API Design Document.

10.5.6.1.2.4 Notifications DB

The Portal Database will be implemented in Azure Cosmos DB. It will utilise a standardised document-oriented JSON schema to ensure consistency in the management and presentation of notifications data.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 154 of 1295

10.5.6.1.3 Service Interfaces

HTTP/REST/JSON: See Notifications API, above. Notifications data will be queryable and sent through the API, using REST semantics. Refer to the STARS API documentation for more detail.

SQL: See Notifications Database, above. The Notifications DB will be queryable via Cosmos DB SQL DDL statements, depending on the calling user/service's access.

10.5.6.1.3.1 Notifications Management Portal

The Notifications Portal is the management interface for the Notifications Service.

The Notifications Management Portal will allow STARS system operators to define notifications and notification templates. It will also allow operators to view the history and status of notifications.

10.5.6.1.4 Service Security

The service will conform to the Security Architecture. Access to the Notifications Service will be controlled by Azure RBAC, with all calling services identified and granted access through Managed Identities.

10.5.6.2 Event Service

10.5.6.2.1 Service Description

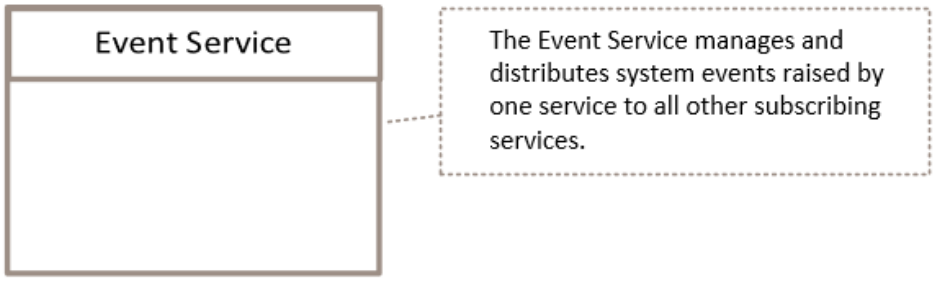


Figure 59: Event Service (B:12)

The Event Service manages and distributes system events raised by one service to all other subscribing services. Multiple services across STARS that will raise events to inform the wider system of process outcomes. The two primary event sinks will be the Notification Service and the Audit Service.

The Event Service is not discussed in detail in this section – please refer to sections 7 'Solution Architecture & Design Principles' and 11 ''Technical Architecture and Design' for further information.

The Event Service will be implemented using Azure Event Grid.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 155 of 1295

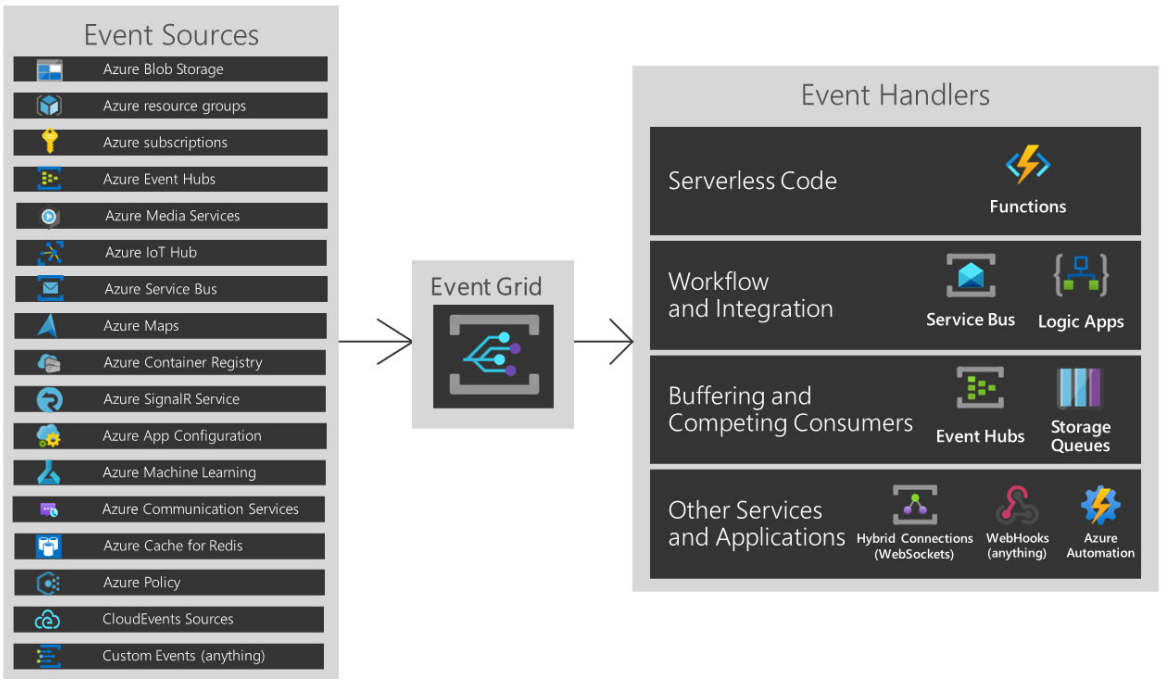


Figure 60: Azure Event Grid

10.5.6.3 Logging Service

10.5.6.3.1 Service Description

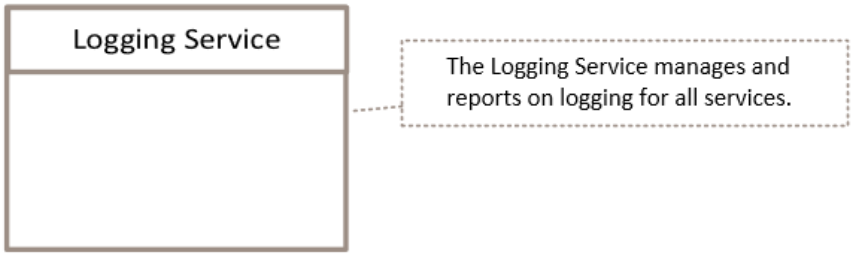


Figure 61: Logging Service (B:12)

The Logging Service manages and reports on logging for all services.

Multiple services across STARS that will perform detailed and structured logging to inform systems monitoring and fault diagnosis.

The Logging Service is not discussed in detail in this section – please refer to sections 7 'Solution Architecture & Design Principles' and 11 ''Technical Architecture and Design' for further information.

The Logging Service will be implemented using Azure Monitor and Azure Log Analytics. STARS will utilise a central, shared Log Analytics workspace outside of the Landing Zone Subscriptions, managed by OOBE.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 156 of 1295

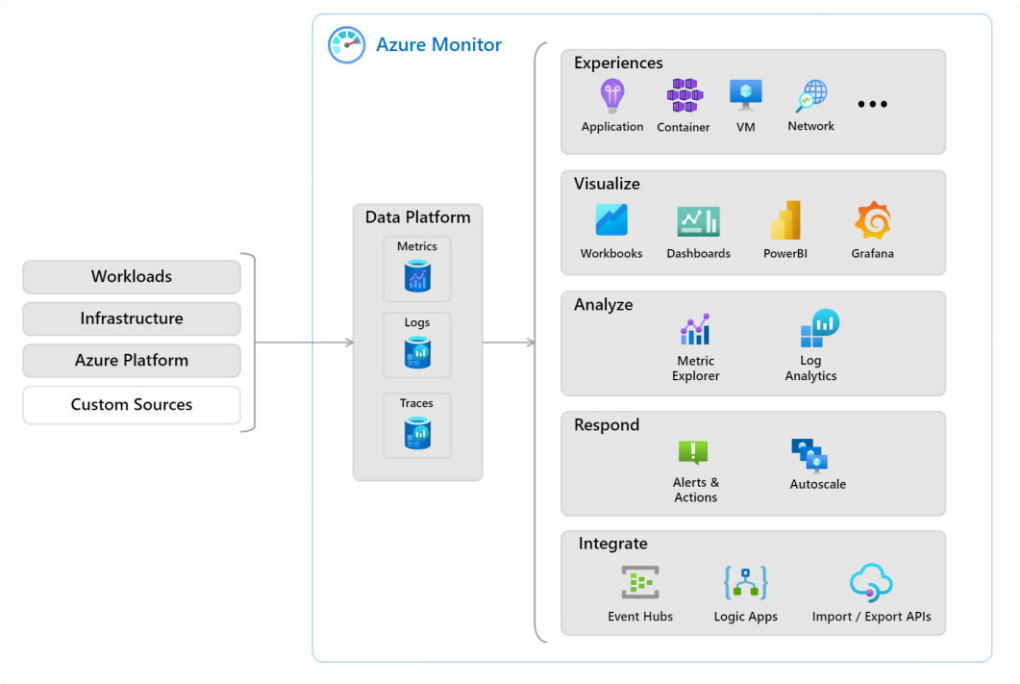


Figure 62: Azure Monitor

10.5.6.4Audit Service

10.5.6.4.1 Service Description

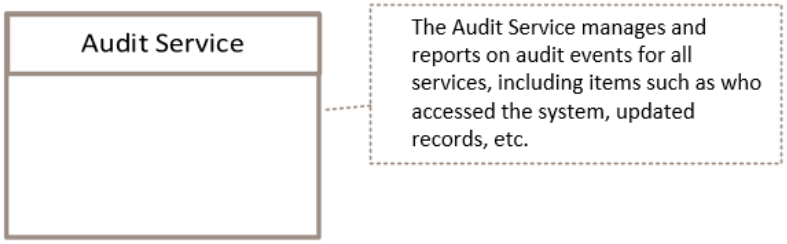


Figure 63: Audit Service (B:12)

The Audit Service manages and reports on audit events for all services, including items such as who accessed the system, updated records, etc.

The Audit Service is based on Azure of security logging and audit and will be implemented using several Azure Services – please refer to sections 7 ‘Solution Architecture & Design Principles’ and 11 ‘‘Technical Architecture and Design’ for further information.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 157 of 1295

10.5.7 Identity and Access Management Services

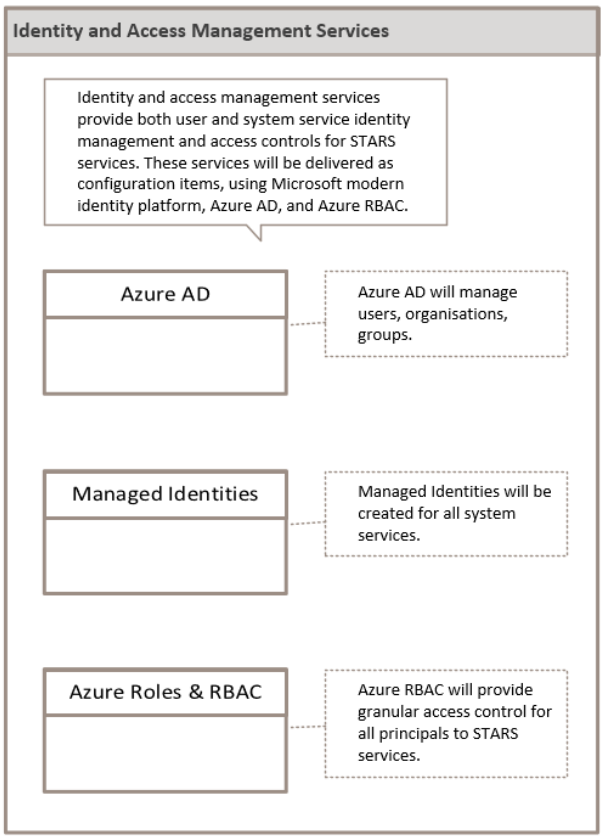


Figure 64: Identity and Access Management Services (B:12)

Identity and access management services provide both user and system service identity management and access controls for STARS services. These services will be delivered as configuration items, using Microsoft modern identity platform, Azure AD, and Azure RBAC.

MyGovID will be used as an identity provider for human and system users that will be consuming STARS services. The identity management capability also covers internal identities of components within the STARS system.

Please refer to sections 7 ‘Solution Architecture & Design Principles’ and 12 ‘Security Architecture and Design’ for further information.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 158 of 1295

11 Technical Architecture and Design

Please note that many detailed requirements are not yet provided surrounding lower-level STARS capabilities, processes, workflows and rules. Accordingly, the solution design proposes a set of flexible approaches that are most likely able to support detailed requirements as they are created or evolved during the life of the project (subject to the project scope change management process), and beyond.

Please also note that analysis and design work for workflows, rules, message queues, event topics, data containers, etc., will commence during Work Package 2 and continue during the project.

The STARS solution will be implemented within an Azure infrastructure environment provided by oobe. Their Perimeta for Azure solution defines the broad security and networking landscape in which the STARS solution will operate.

At a high level, the Perimeta for Azure solution requires compliance in areas of Identity, Network, Availability, Encryption, Monitoring & Compliance and other general areas such as Azure Resource Management, Management Groups, Subscriptions and Resource Groups.

Details of the Perimeta for Azure solution, in the STARS context can be found in oobe's VDS – Perimeta for Azure – Solution Design document and other associated documents.

The Landing Zone architecture defined by the Perimeta for Azure solution is as follows:

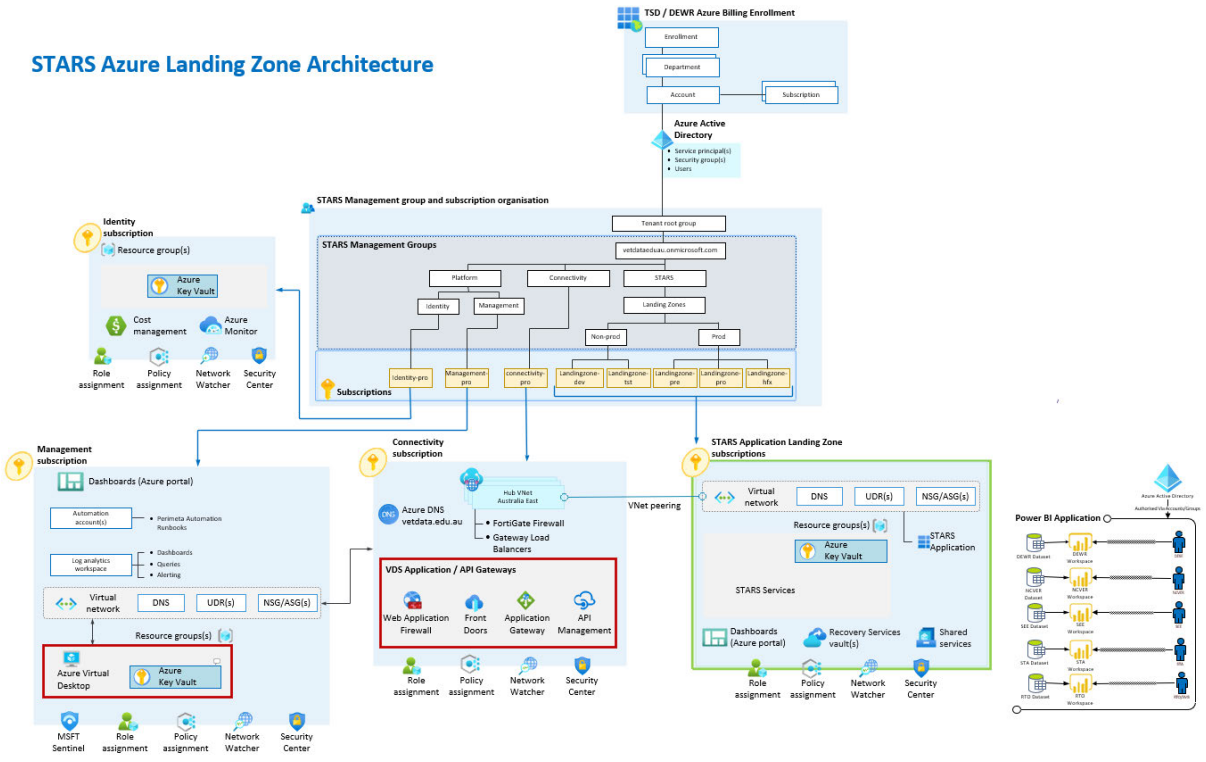


Figure 65: STARS Azure Landing Zone Architecture (I:14)

Some key things to note from this architecture that relate to the STARS solution design:

- The STARS services will be deployed inside the Landing Zone subscriptions provisioned for each environment.
- All network traffic is expected to pass through the FortiGate Firewall

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 159 of 1295

- API Management is outside of the Landing Zone subscriptions, but it is expected that a separate APIM instance will be available for each environment
- Web Application Firewall, Front Doors and Application Gateways are also outside of the Landing Zone Subscriptions and will be key components of the High Availability strategy.
- A central Log Analytics Workspace is provided in the Management Subscription and is the expected destination for all system logs.

11.1 Solution Services

This section lays out the various Azure-based PaaS services that will be used by each of the Solution Services described in section 10 'Software Architecture and Design'. In general, each service will be implemented in its own Azure Resource Group (RG).

Resource Groups will be deployed via their own Azure DevOps deployment pipelines, using ARM Bicep templates.

The entire STARS solution will be multi-region deployable and will use standard Azure High-Availability features. This does have ramifications on the configuration of the services – multi-region configuration and HA considerations are described in section 7 'Solution Architecture & Design Principles'. This section covers off both how individual supporting services and the STARS application code are configured for HA (e.g., Azure Gateway, Azure APIM, Azure SQL, Cosmos DB, Azure Functions).

Each of these services will use a consistent approach for similar patterns of communication and execution. The patterns used will be documented in detail in future Detailed Analysis and Design activities, but at a high level, the following descriptions represent the architectural decisions made based the current understanding of the solution requirements, as they have been assessed to be the best tools for the job at hand.

- APIs will be implemented as Open API specifications through APIM, with http-triggered Function Apps implemented the backend logic.
- Workflow orchestration will be implemented in Logic Apps
- Document data will be stored in Azure CosmosDB
- Relational data will be stored in Azure SQL Database
- Function Apps will be used to provide additional computational logic required within the service workflows.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 160 of 1295

11.1.1 Master Data Service

Refer to section 10.5.1.1 ‘Master Data Service’ for a functional description.

11.1.1.1 Diagram

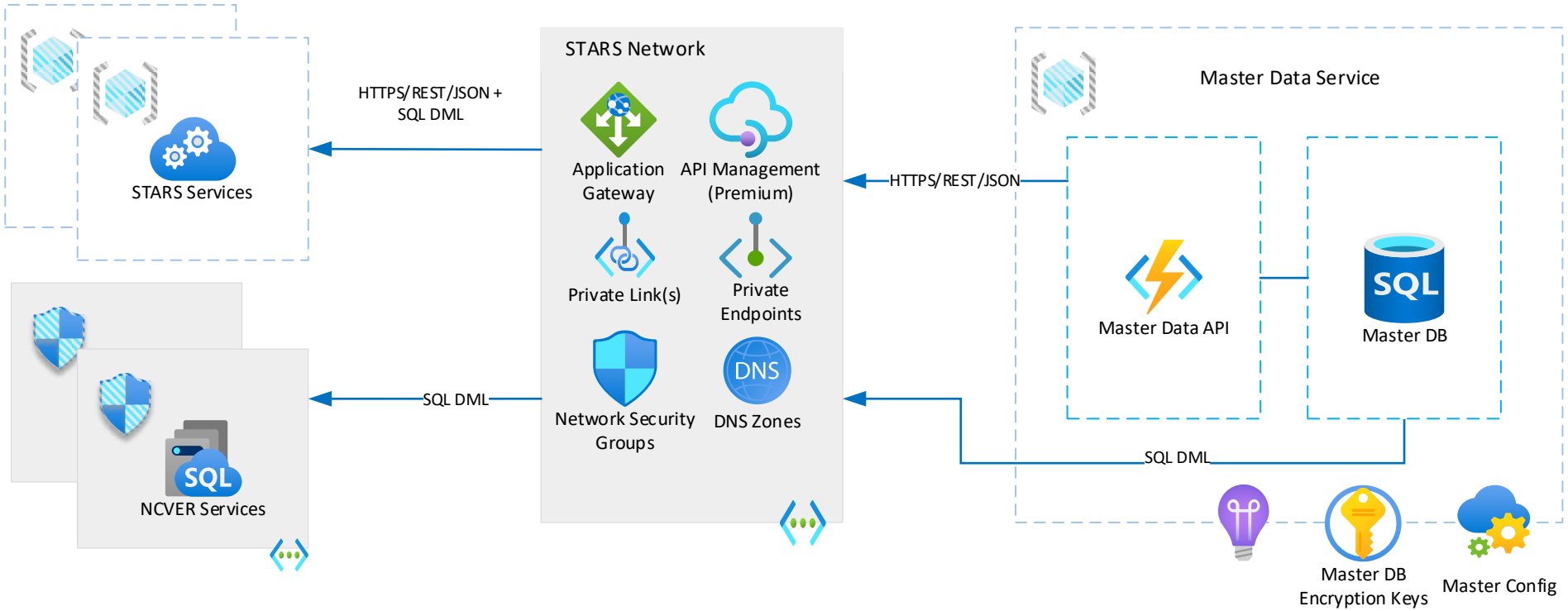


Figure 66: Master Data Service (I:2)

11.1.1.2 Description

Component	Azure Service	Description
Master Data API	Azure Functions	The Master Data API is an HTTP-triggered function app. Master data operations will be mapped to relevant function app endpoints. Please see STARS API Design document (in development) for details of API endpoints. The API will be factored around VET-IS Data Packet entities (e.g., Student, Enrolment).
Master DB	Azure SQL	The Master Database is one (or more) Azure SQL database instance that stores data in VET-IS Physical Schema format (e.g., Stakeholder/Responsibility).
Master DB Encryption Keys	Azure Key Vault	Master DB will encrypt data at rest; encryption keys to be held in Key Vault.
Master Config	Azure App Config	Any RG-specific config, including for function apps, to be held in a common Azure App Config.
STARS Network	API Management	Primary access to Master Data API is via HTTPS/REST/JSON, mediated through APIM.

11.1.2 Reference Data Service

Refer to section 10.5.1.2 ‘Reference Data Services’ for a functional description.

11.1.2.1 Diagram

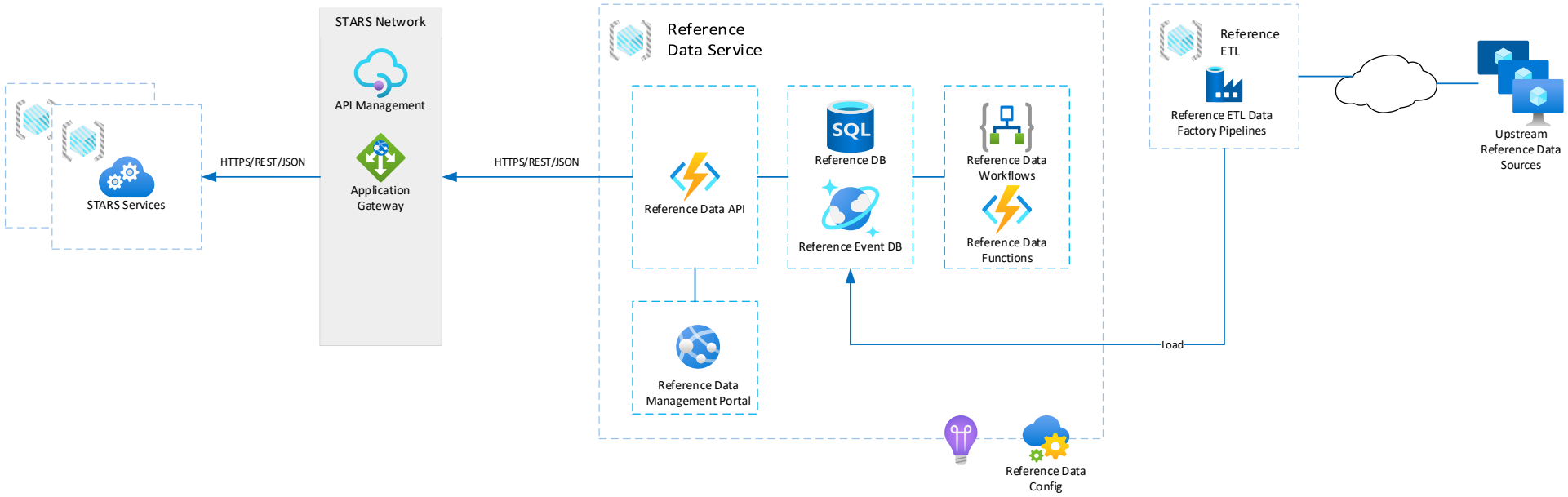


Figure 67: Reference Data Service (I:3)

11.1.2.2 Description

Component	Azure Service	Description
Reference Data API	Azure Functions	The Reference Data API is an HTTP-triggered function app. Reference data operations will be mapped to relevant function app endpoints. Please see STARS API Design document (in development) for details of API endpoints. The API will be factored around VET reference entities (e.g., course & subject codes, RTO delivery locations).

Component	Azure Service	Description
Reference Data Management Portal	App Services	The portal is an SPA that manages reference data, allowing for invalid or missing data to be edited.
Reference DB	Azure SQL	The Reference Database is an Azure SQL database instance that stores reference data in STARS Reference Physical Schema (detailed requirements yet to be provided.)
Reference Event DB	Azure Cosmos DB	The Reference Event Database is an Azure Cosmos database instance that stores all reference data management and lifecycle events.
Reference Data Workflows	Azure Logic Apps	Any business processes or workflows for reference data will be implemented in Azure Logic Apps (detailed requirements yet to be provided.)
Reference Data Functions	Azure Functions	The Reference Data Functions provide the coded logic required for any reference data workflows. Please see STARS API Design document (in development) for details of API endpoints.
Reference Data Config	Azure App Config	Any RG-specific config, including for function apps, to be held in a common Azure App Config.
Reference Data ETL Pipelines	Azure Data Factory	Reference data will be sourced at regular intervals from several upstream systems, e.g., TGA, Australia Post.
STARS Network	API Management	Primary access to Reference Data API is via HTTPS/REST/JSON, mediated through APIM.

11.1.3 Provider Service

Refer to section 10.5.1.4 ‘Provider Services’ for a functional description.

11.1.3.1 Diagram

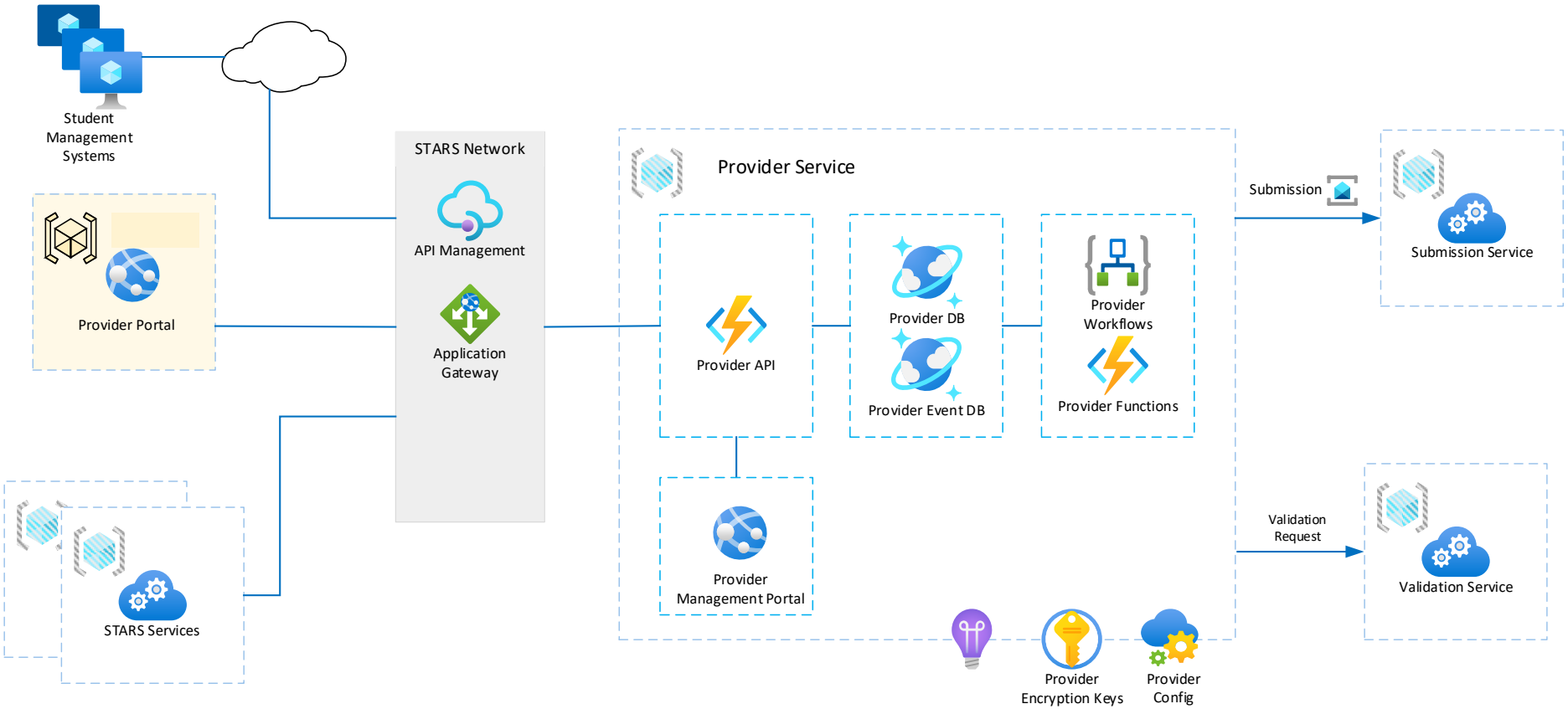


Figure 68: Provider Service (I:4)

11.1.3.2 Description

Component	Azure Service	Description
Provider API	Azure Functions	The Provider API is an HTTP-triggered function app. Provider data operations will be mapped to relevant function app endpoints. Please see STARS API Design document (in development) for details of API endpoints. The API will be factored around VET-IS data packet entities (e.g., students, enrolments, delivery locations). Of note is that VET data is submitted via the Provider API, as HTTP/REST/JSON (or files – TBC) and passed on to the Submission Service for processing.
Provider Management Portal	App Services	The management portal is an SPA that manages provider data, used by STARS system operators.
Provider Portal	App Services	The provider portal is an SPA that allows RTOs to manage their VET-IS training activity data. The Provider Portal is public-facing and will be hosted in the STARS DMZ.
Provider DB	Azure Cosmos DB	The Provider Database is an Azure Cosmos DB database that stores provider data in STARS Provider Physical Schema (detailed requirements yet to be provided.)
Provider Event DB	Azure Cosmos DB	The Provider Event Database is an Azure Cosmos database instance that stores all provider data management and lifecycle events.
Provider Workflows	Azure Logic Apps	Any business processes or workflows for provider data will be implemented in Azure Logic Apps (detailed requirements yet to be provided.) Of note is that VET data is submitted via the Provider API, as HTTP/REST/JSON (or files – TBC) and passed on to the Submission Service for processing.

Component	Azure Service	Description
		Submission data is sent as Service Bus messages (Queue is within the Submission RG.)
Provider Functions	Azure Functions	The Provider Functions provide the coded logic required for any provider data workflows. Please see STARS API Design document (in development) for details of API endpoints.
Provider Encryption Keys	Azure Key Vault	Provider data will be encrypted at rest, encryption keys to be held in Key Vault.
Provider Config	Azure App Config	Any RG-specific config, including for function apps, to be held in a common Azure App Config.
Provider Messaging	Service Bus	Provider sourced VET data is handed off to the Submission Service via Service Bus messages.
STARS Network	API Management	Primary access to Reference Data API is via HTTPS/REST/JSON, mediated through APIM.

11.1.4 Notification Service

Refer to section 10.5.6.1 ‘Notifications Service’ for a functional description.

11.1.4.1Diagram

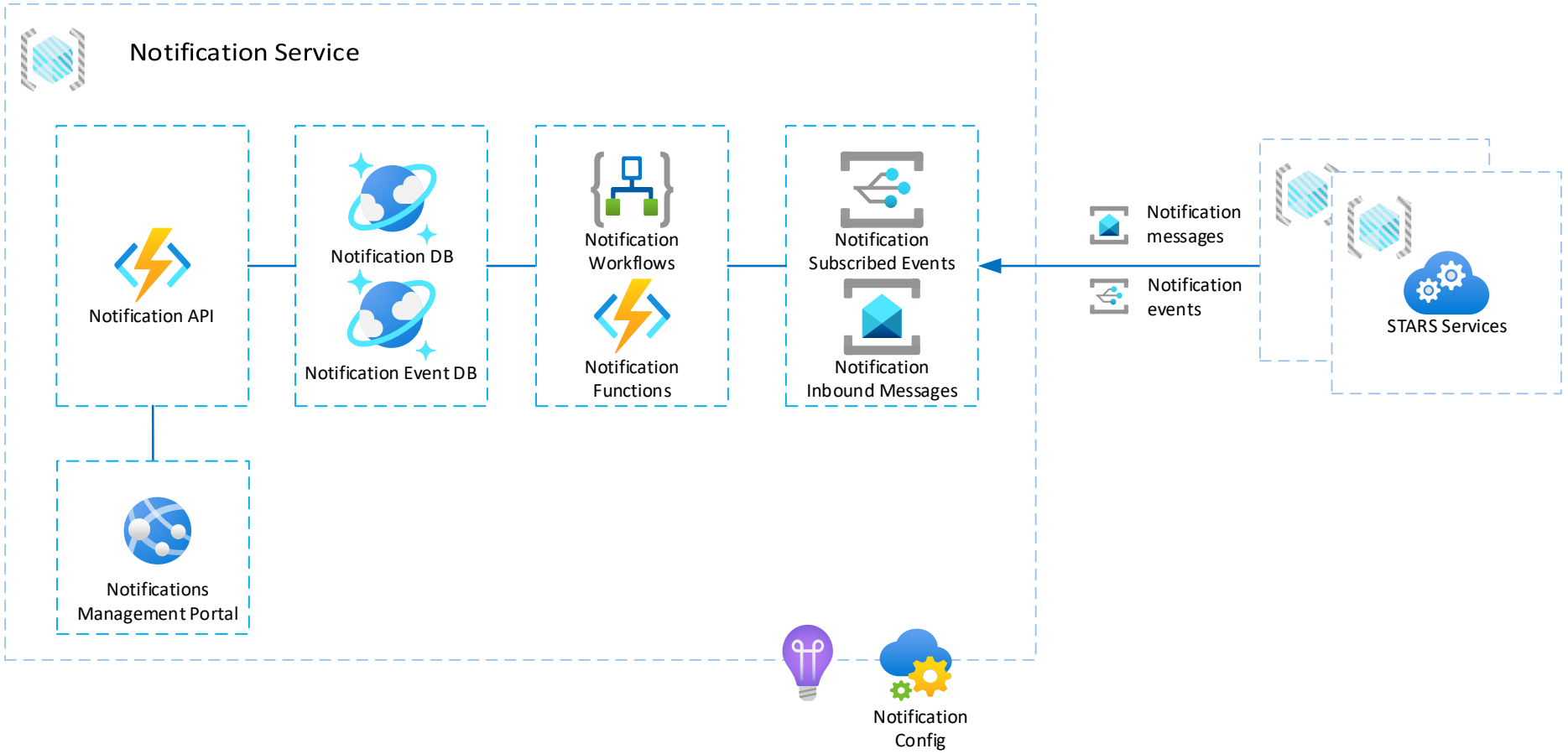


Figure 69: Notification Service (I:5)

11.1.4.2 Description

Component	Azure Service	Description
Notifications API	Azure Functions	The Notifications API is an HTTP-triggered function app. Notification data operations will be mapped to relevant function app endpoints. Please see STARS API Design document (in development) for details of API endpoints.
Notifications Management Portal	App Services	The management portal is an SPA that manages notifications data, used by STARS system operators.
Notification DB	Azure Cosmos DB	The Notifications Database is an Azure Cosmos DB database that stores notifications data in STARS Notification Physical Schema (detailed requirements yet to be provided.) This includes items such as Notification templates.
Notification Event DB	Azure Cosmos DB	The Notification Event Database is an Azure Cosmos database instance that stores all notification data management and lifecycle events.
Notification Workflows	Azure Logic Apps	Any business processes or workflows for notification data will be implemented in Azure Logic Apps (detailed requirements yet to be provided.) Of note is that Notifications may be submitted either via the Notifications API or as messages via Service Bus. Notifications workflows may also subscribe to STARS system-wide Events via Event Grid.
Notification Functions	Azure Functions	The Notification Functions provide the coded logic required for any provider data workflows. Please see STARS API Design document (in development) for details of API endpoints.

Component	Azure Service	Description
Notification Config	Azure App Config	Any RG-specific config, including for function apps, to be held in a common Azure App Config.
Notification Messaging	Service Bus	Notifications may be raised as messages from other STARS Services on the Notification RG hosted Service Bus Queue(s).
Notification Events	Event Grid	Notifications may be created arising from events raised by other STARS Services through the STARS system-wide Event Grid.
STARS Network	API Management	Primary access to Notification API is via HTTPS/REST/JSON, mediated through APIM.

11.1.5 Submission Service

Refer to section 10.5.1.3 ‘Submission Service’ for a functional description.

11.1.5.1 Diagram

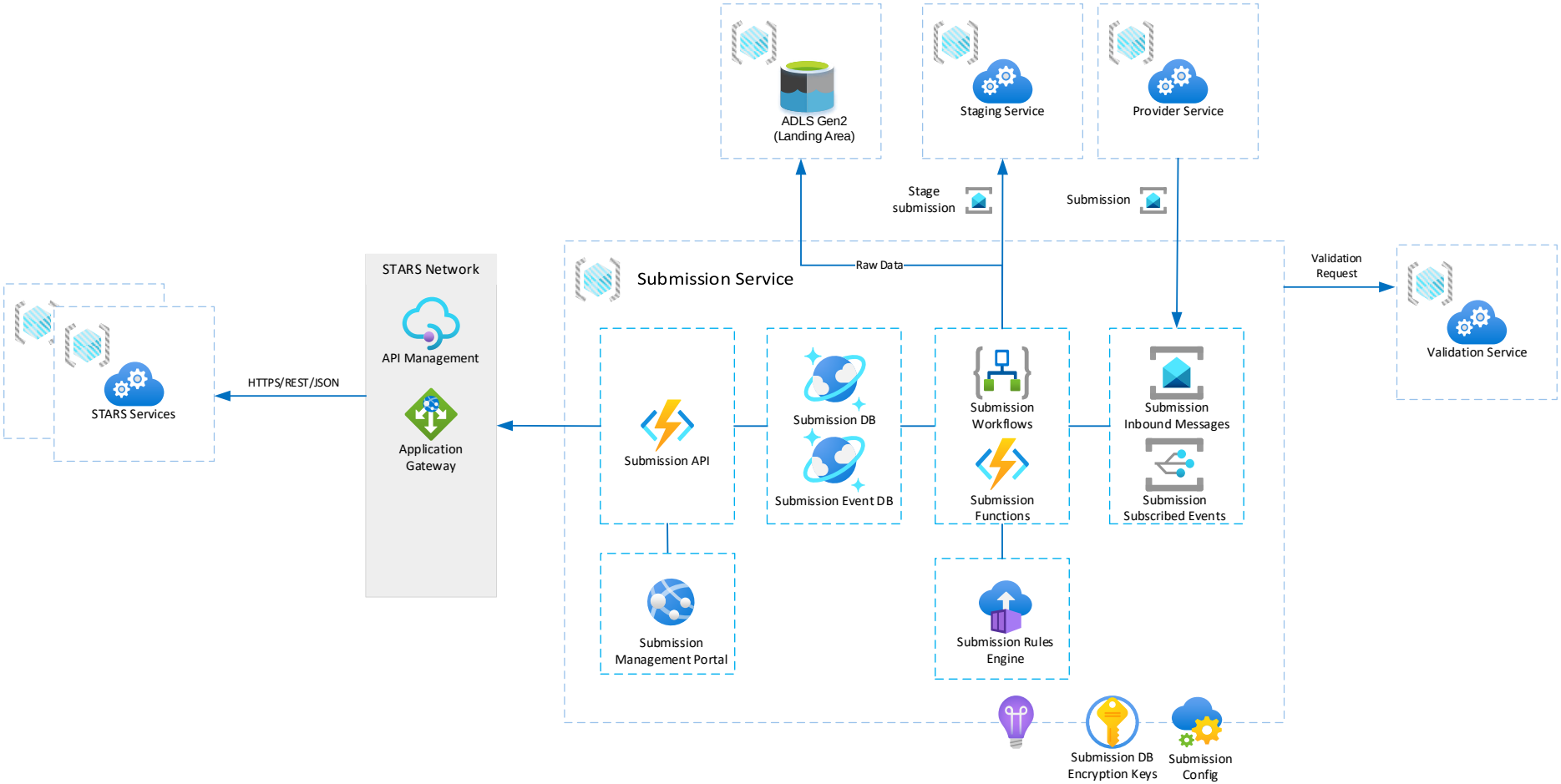


Figure 70: Submission Service (I:6)

11.1.5.2 Description

Component	Azure Service	Description
Submission API	Azure Functions	The Submission API is an HTTP-triggered function app. Submission data management and query operations will be mapped to relevant function app endpoints. Please see STARS API Design document (in development) for details of API endpoints. The API will be factored around VET-IS data packet entities (e.g., students, enrolments, delivery locations).
Submission Management Portal	App Services	The management portal is an SPA that manages submission pipeline operations, used by STARS system operators.
Submission DB	Azure Cosmos DB	The Submission Database is an Azure Cosmos DB database that stores submission data in VET-IS Data Packet Schema (detailed requirements yet to be provided.)
Submission Event DB	Azure Cosmos DB	The Submission Event Database is an Azure Cosmos database instance that stores all submission pipeline, data management and lifecycle events.
Submission Workflows	Azure Logic Apps	Any business processes or workflows for submission data will be implemented in Azure Logic Apps (detailed requirements yet to be provided.) Of note is that VET data is submitted by Provider API, as JSON and passed on to the Submission Service for processing. Submission data is sent as Service Bus messages (Queue is within the Submission RG.)
Submission Functions	Azure Functions	The Submission Functions provide the coded logic required for any provider data workflows.

Component	Azure Service	Description
		Please see STARS API Design document (in development) for details of API endpoints.
Submission Rules Engine	Azure Container Instances	The submission rules engine supports submission workflow-level business rules management (i.e., not submission data-level rules validation). Container Instances will host and run the rules engine and associated internal infrastructure in a container.
Submission Encryption Keys	Azure Key Vault	Submission data will be encrypted at rest, encryption keys to be held in Key Vault.
Submission Config	Azure App Config	Any RG-specific config, including for function apps, to be held in a common Azure App Config.
Submission Messaging	Service Bus	Provider sourced VET data is handed off to the Submission Service via Service Bus messages.
Submission Events	Event Grid	The Submission service may respond to events raised by other STARS Services through the STARS system-wide Event Grid, e.g., to raise compensating transactions for failures due to downstream or supporting services.
STARS Network	API Management	Primary access to Submission API is via HTTPS/REST/JSON, mediated through APIM.
STARS Data Lake	Azure Data Lake	All 'raw' submission data will be stored in ADLS for future (TBD) data analysis.

11.1.6 Staging Service

Refer to section 10.5.2.3 'Staging Service' for a functional description.

11.1.6.1Diagram

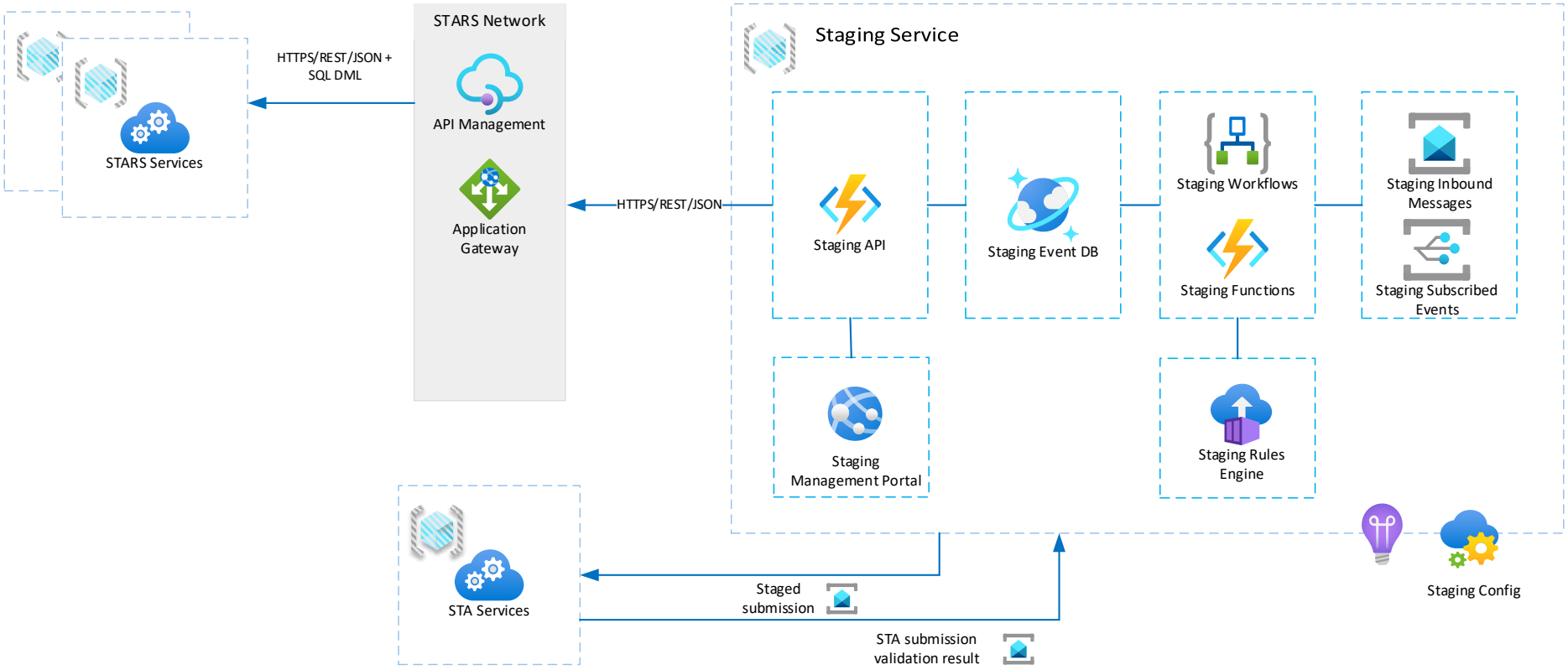


Figure 71: Staging Service (I:7)

11.1.6.2 Description

Component	Azure Service	Description
Staging API	Azure Functions	The Staging API is an HTTP-triggered function app. Staging data management and query operations will be mapped to relevant function app endpoints. Please see STARS API Design document (in development) for details of API endpoints.
Staging Management Portal	App Services	The management portal is an SPA that manages staging pipeline operations, used by STARS system operators.
Staging Event DB	Azure Cosmos DB	The Staging Event Database is an Azure Cosmos database instance that stores all staging pipeline, data management and lifecycle events.
Staging Workflows	Azure Logic Apps	Any business processes or workflows for staging will be implemented in Azure Logic Apps (detailed requirements yet to be provided.)
Staging Functions	Azure Functions	The Staging Functions provide the coded logic required for any submission workflows. Please see STARS API Design document (in development) for details of API endpoints.
Staging Rules Engine	Azure Container Instances	The Staging rules engine supports staging workflow-level business rules management (i.e., not data-level rules validation). Container Instances will host and run the rules engine and associated internal infrastructure in a container.
Staging Config	Azure App Config	Any RG-specific config, including for function apps, to be held in a common Azure App Config.
Staging Messaging	Service Bus	Validated VET data is handed off to the Staging Service via Service Bus messages that include the URL(s) of the data within the Submission Service.

Component	Azure Service	Description
Staging Events	Event Grid	The Staging service may respond to events raised by other STARS Services through the STARS system-wide Event Grid, e.g., to raise compensating transactions for failures due to downstream or supporting services.
STARS Network	API Management	Primary access to Staging API is via HTTPS/REST/JSON, mediated through APIM.

11.1.7 Release Service

Refer to section 10.5.2.4 ‘Release Service’ for a functional description.

11.1.7.1 Diagram

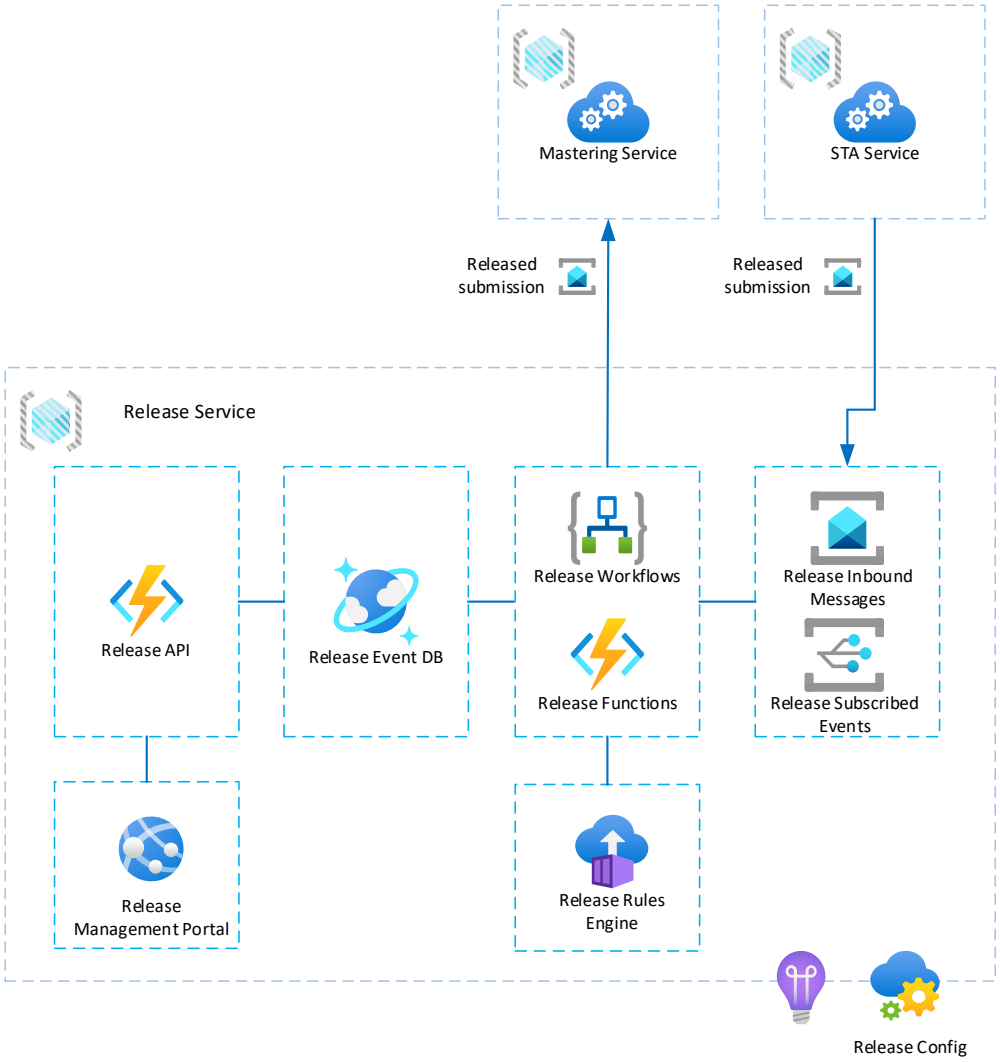


Figure 72: Release Service (I:8)

11.1.7.2 Description

Component	Azure Service	Description
Release API	Azure Functions	The Release API is an HTTP-triggered function app. Release data management and query operations will be mapped to relevant function app endpoints. Please see STARS API Design document (in development) for details of API endpoints.
Release Management Portal	App Services	The management portal is an SPA that manages release pipeline operations, used by STARS system operators.
Release Event DB	Azure Cosmos DB	The Release Event DB is an Azure Cosmos database instance that stores all Release pipeline, data management and lifecycle events.
Release Workflows	Azure Logic Apps	Any business processes or workflows for Release will be implemented in Azure Logic Apps (detailed requirements yet to be provided.)
Release Functions	Azure Functions	The Release Functions provide the coded logic required for any provider data workflows. Please see STARS API Design document (in development) for details of API endpoints.
Release Rules Engine	Azure Container Instances	The Release rules engine supports release workflow-level business rules management (i.e., not data-level rules validation). Container Instances will host and run the rules engine and associated internal infrastructure in a container.
Release Config	Azure App Config	Any RG-specific config, including for function apps, to be held in a common Azure App Config.
Release Messaging	Service Bus	Released VET data is notified to the Release Service via Service Bus messages from the STA Service that include the URL(s) of the data within the Submission Service.

Component	Azure Service	Description
		The Release Service in turn hands off control to the Mastering Service, during it's workflows, via messages to the Mastering Service inbound Service Bus Queue(s).
Release Events	Event Grid	The Release service may respond to events raised by other STARS Services through the STARS system-wide Event Grid, e.g., to raise compensating transactions for failures due to downstream or supporting services.
STARS Network	API Management	Primary access to Release API is via HTTPS/REST/JSON, mediated through APIM.

11.1.8 Mastering Service

Refer to sections 10.5.2.5 'Mastering Service' and 8.5 'Master Data Management Approach' for functional descriptions.

11.1.8.1Diagram

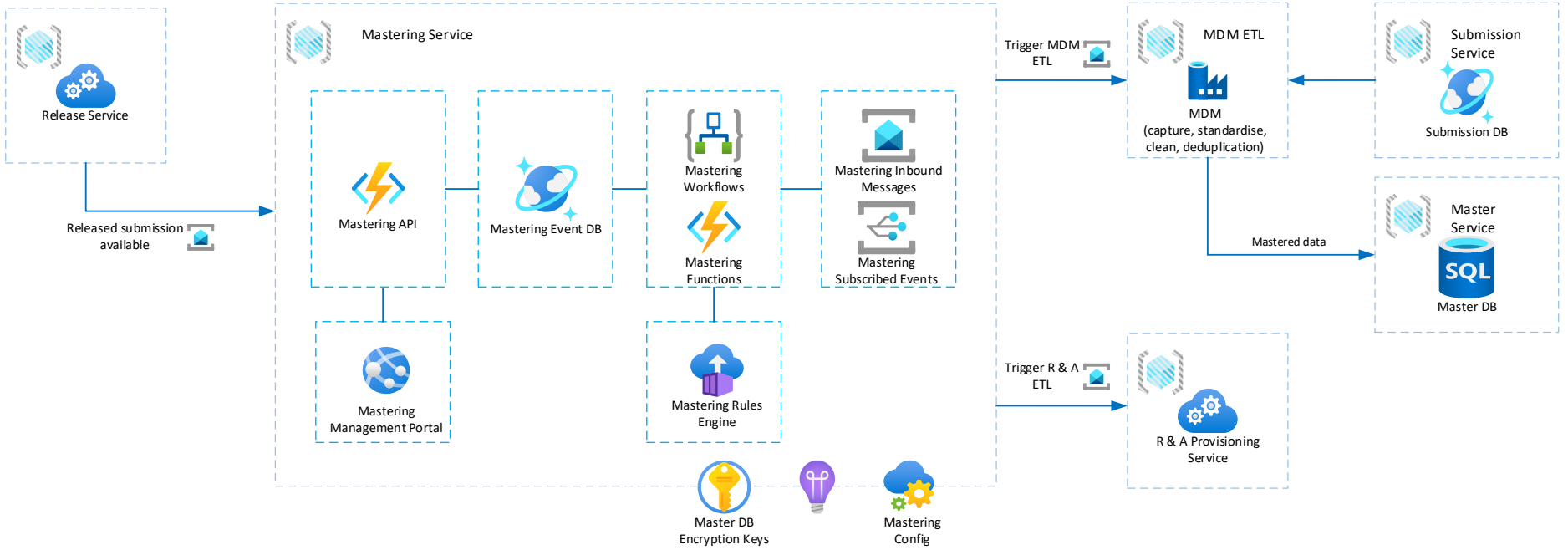


Figure 73: Mastering Service (I:9)

11.1.8.2Description

Component	Azure Service	Description
Mastering API	Azure Functions	The Mastering API is an HTTP-triggered function app. Mastering management and query operations will be mapped to relevant function app endpoints. Please see STARS API Design document (in development) for details of API endpoints.

Component	Azure Service	Description
Mastering Management Portal	App Services	The management portal is an SPA that manages mastering pipeline operations, used by STARS system operators.
Mastering Event DB	Azure Cosmos DB	The Mastering Event Database is an Azure Cosmos database instance that stores all mastering pipeline management and lifecycle events.
Mastering Workflows	Azure Logic Apps	Any business processes or workflows for mastering data will be implemented in Azure Logic Apps (detailed requirements yet to be provided.)
Mastering Functions	Azure Functions	The Mastering Functions provide the coded logic required for any mastering workflows. Please see STARS API Design document (in development) for details of API endpoints.
Mastering Rules Engine	Azure Container Instances	The Mastering rules engine supports mastering workflow-level business rules management (i.e., not Mastering data-level rules validation). Container Instances will host and run the rules engine and associated internal infrastructure in a container.
Master DB Encryption Keys	Azure Key Vault	Master data will be encrypted at rest, encryption keys to be held in Key Vault.
Mastering Config	Azure App Config	Any RG-specific config, including for function apps, to be held in a common Azure App Config.
Mastering Messaging	Service Bus	Messages informing the Mastering Service that data has been released will be received by the service as Service Bus messages, in the Mastering Queue(s). The mastering workflows will inform the R & A Provisioning Services when data is mastered via outbound Service Bus messages.

Component	Azure Service	Description
Mastering Events	Event Grid	The Mastering service may respond to events raised by other STARS Services through the STARS system-wide Event Grid, e.g., to raise compensating transactions for failures due to downstream or supporting services.
MDM ETL	Azure Data Factory	Most MDM processing will occur in ADF pipelines. The pipelines will be triggered and overseen by the mastering Logic Apps workflows. See section 8.5 ‘Master Data Management Approach’ for more information.
STARS Network	API Management	Primary access to Mastering API is via HTTPS/REST/JSON, mediated through APIM.

11.1.9 Reporting and Analytics Provisioning Service

See section 10.5.2.6 ‘Reporting and Analytics Provisioning Service’ for a functional description.

11.1.9.1Diagram

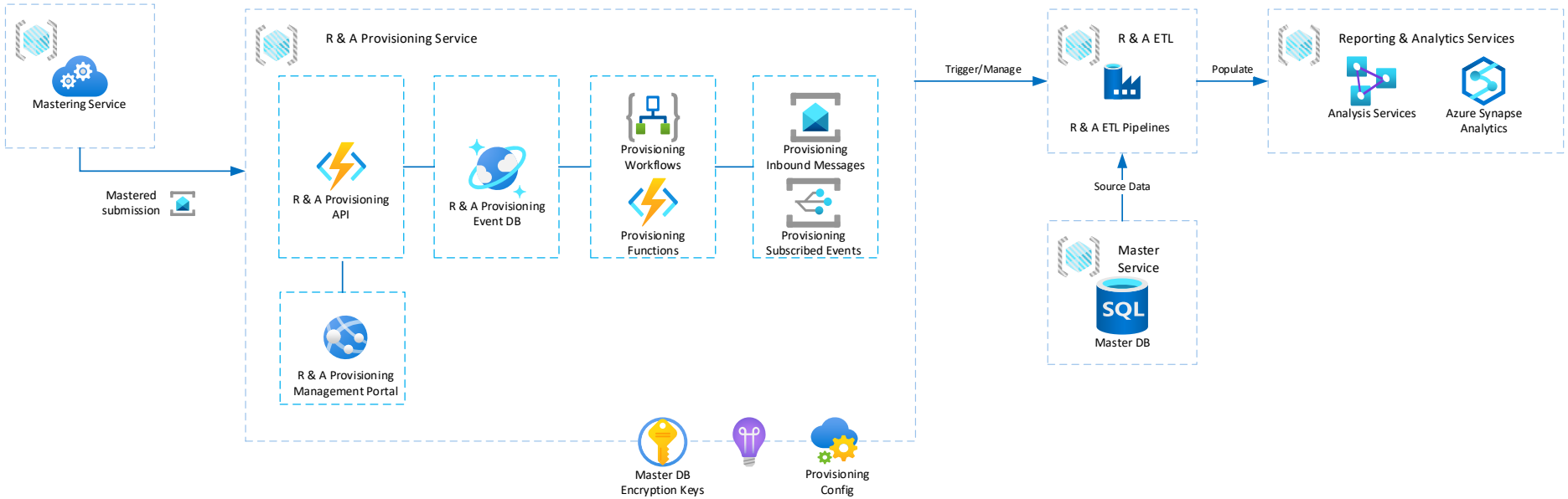


Figure 74: Reporting and Analytics Provisioning Service (I:10)

11.1.9.2Description

Abbreviation: Reporting & Analysis Provisioning Service = RA-P

Component	Azure Service	Description
RA-P API	Azure Functions	The RA-P API is an HTTP-triggered function app. RA-P management and query operations will be mapped to relevant function app endpoints. Please see STARS API Design document (in development) for details of API endpoints.

Component	Azure Service	Description
RA-P Management Portal	App Services	The management portal is an SPA that manages RA-P pipeline operations, used by STARS system operators.
RA-P Event DB	Azure Cosmos DB	The RA-P Event Database is an Azure Cosmos database instance that stores all RA-P pipeline management and lifecycle events.
RA-P Workflows	Azure Logic Apps	Any business processes or workflows for RA-P will be implemented in Azure Logic Apps (detailed requirements yet to be provided.)
RA-P Functions	Azure Functions	The RA-P Functions provide the coded logic required for any RA-P workflows. Please see STARS API Design document (in development) for details of API endpoints.
RA-P Rules Engine	Azure Container Instances	The RA-P rules engine supports RA-P workflow-level business rules management (i.e., not RA-P data-level rules validation). Container Instances will host and run the rules engine and associated internal infrastructure in a container.
Master DB Encryption Keys	Azure Key Vault	Master data will be encrypted at rest, encryption keys to be held in Key Vault.
RA-P Config	Azure App Config	Any RG-specific config, including for function apps, to be held in a common Azure App Config.
RA-P Messaging	Service Bus	Messages informing the RA-P Service that data has been mastered will be received by the service as Service Bus messages, in the RA-P Queue(s). The RA-P workflows will inform the R & A Provisioning Services when data is mastered via outbound Service Bus messages.
RA-P Events	Event Grid	The RA-P service may respond to events raised by other STARS Services through the STARS system-wide Event Grid, e.g., to raise compensating transactions for failures due to downstream or supporting services.

Component	Azure Service	Description
RA-P ETL	Azure Data Factory	Most provisioning of R & A databases will occur in ADF pipelines. The pipelines will be triggered and overseen by the RA-P Logic Apps workflows. See section 8.9 ‘Reporting and Analytics Approach’ for more information.
STARS Network	API Management	Primary access to RA-P API is via HTTPS/REST/JSON, mediated through APIM.

11.1.10 Decision Services

Refer to sections 10.4 ‘Rules Engine’ and 10.5.3 ‘Decision Services’ for a functional description.

11.1.10.1 Diagram

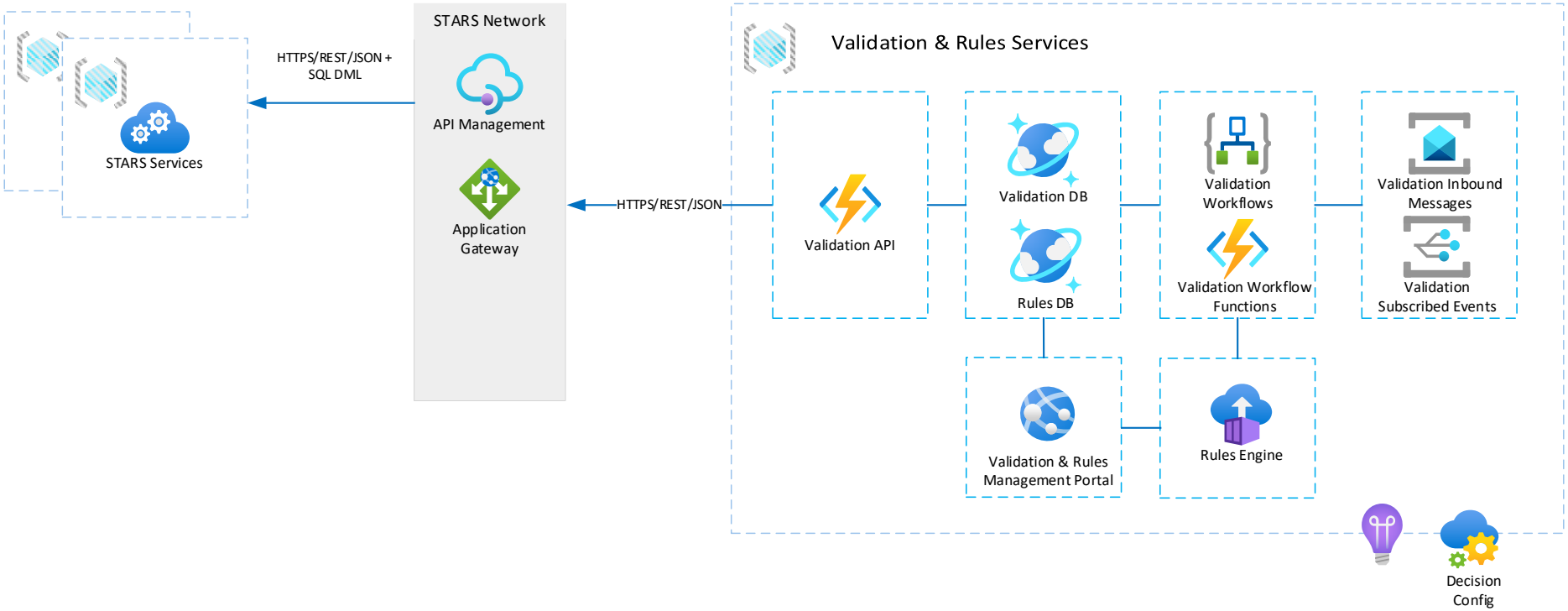


Figure 75: Decision Services (I:11)

11.1.10.2 Description

Component	Azure Service	Description
Validation API	Azure Functions	The Validation API is an HTTP-triggered function app. Validation management and query operations will be mapped to relevant function app endpoints. Please see STARS API Design document (in development) for details of API endpoints.
Validation & Rules Management Portal	App Services	The management portal is an SPA that manages validation operations and rule configuration, used by STARS system operators.
Validation DB	Azure Cosmos DB	The Validation Database is an Azure Cosmos database instance that stores all validation outcomes.
Rules DB	Azure Cosmos DB	The Rules Database is an Azure Cosmos database instance that stores all rule configuration.
Validation Workflows	Azure Logic Apps	Any business processes or workflows for validating data will be implemented in Azure Logic Apps (detailed requirements yet to be provided.) It is likely that many workflows will be developed to suit STA requirements.
Validation Workflow Functions	Azure Functions	The Validation Functions provide the coded logic required for any validation workflows. Please see STARS API Design document (in development) for details of API endpoints.
Rules Engine	Azure Container Instances	The Rules Engine supports validation workflow-level business rules management and data quality rules. Container Instances will host and run the rules engine and associated internal infrastructure in a container.
Decision Config	Azure App Config	Any RG-specific config, including for function apps, to be held in a common Azure App Config.

Component	Azure Service	Description
Validation Messaging	Service Bus	Messages requesting the Validation Service to validate will be received by the service as Service Bus messages, in the Validation Queue(s).
Validation Events	Event Grid	The Validation service may respond to events raised by other STARS Services through the STARS system-wide Event Grid, e.g., to raise compensating transactions for failures due to downstream or supporting services.
STARS Network	API Management	Primary access to Validation API is via HTTPS/REST/JSON, mediated through APIM.

11.2 Azure Data Estate

11.2.1 Azure Data Estate Overview

The diagram below shows high-level overview of the Azure data services used with STARS and their broad roles and relationships:

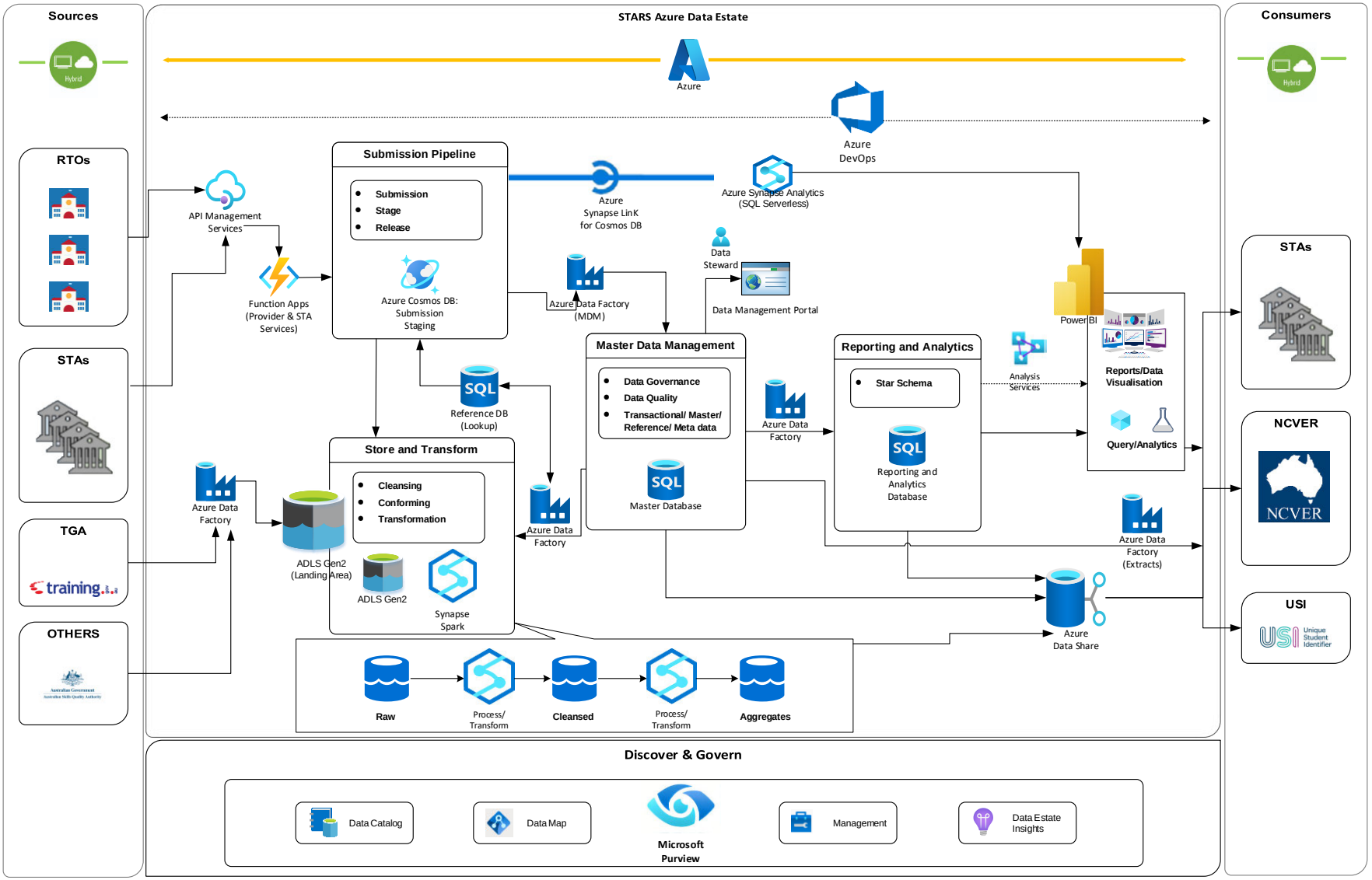


Figure 76: STARS Azure Data Estate (I:12)

11.2.2 Components Overview

This section describes the Azure data-oriented components that will be utilised in STARS.

11.2.2.1 Data Sources

These are the external business systems that generate transactional and reference data. Data is loaded from these systems into the STARS Azure platform. These systems may be a combination of JSON documents, relational, flat-file, or external datasets. These source systems include:

- Registered Training Organisations (RTO)
- State Training Authority (STA)
- Training.gov.au (TGA)
- Australian Skills Quality Authority (ASQA)

11.2.2.2 Azure Cosmos Database

Azure Cosmos DB is a fully managed NoSQL database for modern app development. In STARS solution, Azure Cosmos DB will be used for storing submissions and results within a document-oriented database. This provides a cost effective and highly performant datastore to be used to store the API responses in JSON format.

This NoSQL database also allows a lot of flexibility both with triggering further processing and as a source of data for later analytics processing.

11.2.2.3 Azure Data Factory

Azure Data factory is a hosted service facilitating data movement and transformation. This service should be used in the STARS for moving data between storage types and locations, integrating Databricks notebooks into data engineering processes. It will also be used for scheduling data loads and data engineering processes.

11.2.2.4 Azure Data Lake Gen2 Storage

ADLS is an Azure hosted mass storage service and is the preferred storage engine for Databricks due to its support for hierarchical namespace, a data storage technique intended to support parallel processing. When paired with Databricks and distributed compute processes, it is capable of significant performance when executing large datasets.

Storage accounts will be provisioned within Azure in the same resource group as any corresponding Databricks workspace and Key vault services.

In STARS solution, the Azure Data Lake will be used primarily as a landing area for receiving documents from external sources like ASQA, TGA as well storing raw JSON files submitted by RTOs.

11.2.2.5 Synapse Spark

Synapse Spark is the cloud-based data engineering tool used for the management and analysis of data; this tool is based on Apache Spark on Azure. This will be used for processing and transforming massive quantities of data, it allows commonly used programming languages like Python, Scala, .Net and SQL to be used.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 190 of 1295

In STARS, this is an optional service and can be used for processing massive data files if required. Also, it can serve as a future platform for building, training, deployment, and management of AI/ML models in case requirement arises.

11.2.2.6 Azure SQL Database

Azure SQL Database is a fully managed platform as a service (PaaS) database engine. This scalable cloud database engine used for all data stores within the STARS system i.e., implementing Master Data Management, Reference, and Reporting & Analytics databases.

11.2.2.7 Store and Transform

Once data has landed in Azure Data Lake, data is cleansed and transform (clean, sort, merge, join, etc.). The ingested data in Azure Databricks as Notebook activity steps in data factory pipelines will enable analytical reporting at different stages.

- Raw - Initial data
- Filtered, Cleansed & Augmented
- Business-level Aggregates

11.2.2.8 Master Data Management

Master Data Management is the technology, tools and processes that will ensure master data is coordinated across the department. MDM will provide a unified master data service that provides accurate, consistent, and complete master data across the department and to partners. In the STARS system master data management (MDM) will be implemented using native Azure solutions. STARS will use Azure Data Factory, Microsoft Purview, Azure Logic App, and Azure SQL Database to implement MDM practices.

11.2.2.9 Azure Synapse Link

Azure Synapse Link for Azure Cosmos DB will enable near real time analytics over operational data in Azure Cosmos DB submission staging database for the consumers such as STAs and NCVER. Azure Synapse Link creates a tight seamless integration between Azure Cosmos DB and Azure Synapse Analytics.

11.2.2.10 Azure Synapse SQL Serverless

Serverless SQL pool in Azure Synapse Analytics is a distributed data processing system, built for large workloads. Azure Synapse serverless SQL pools are used to query data in a data lake, Cosmos DB via T-SQL and SQL Server endpoint. This will enable consumers of STARS to get insights, explore and analyse data in near real time from Azure Cosmos submission staging database.

11.2.2.11 Azure Analysis Services (optional)

Azure Analysis services is a fully managed service that provides data modelling capabilities. The semantic model is loaded into Analysis Services. This is an optional service for STARS solution. Power BI Premium is recommended as Microsoft goal is for Power BI Premium to be a superset of Azure Analysis Services, in which case we will not need Azure Analysis Services.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 191 of 1295

11.2.2.12 Azure Data Share

This feature provides secure, automated data replication to authorised consumers like NCVER and STAs via the Azure Data share capability. This will be used to replicate authorised data to other departments who also use Azure Data Share without the need to manually build data transfer pipelines.

Azure Data Share currently offers snapshot-based sharing and in-place sharing, depending upon the requirements one or both features could be used.

11.2.2.13 Power BI

This is a suite of business analytics tools that will deliver insights. It will connect multiple data sources, simplify data preparation, and drive ad-hoc analysis. In the STARS solution, Power BI tool will provide Data visualisation capabilities across all the staging/datastore layers and a self-service model for all the data visualisation reporting. It will also produce reports and publish then for consumption on the web and across mobile devices.

11.2.2.14 Azure DevOps

Azure DevOps in STARS will provide Continuous Integration and Continuous Delivery (CI/CD) pipelines to test and build code – constantly and consistently – and ship to any target environment.

11.2.2.15 Microsoft Purview

The STARS solution improves data governance and empowers authorised data consumption via integration with Microsoft Purview. Microsoft Purview is an Azure service that provides core data governance capabilities including data catalogue, business glossary, data lineage and data classification features.

Microsoft Purview is a key component in STARS Data Governance solution and will be used to ensure that the STAR Solution aligns with the DEWR Data Management Framework.

11.3 CI/CD Pipeline Design

The approach the CI/CD pipeline is documented in the Delivery and Support Plan (k), but at a high level the process is as follows:

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 192 of 1295

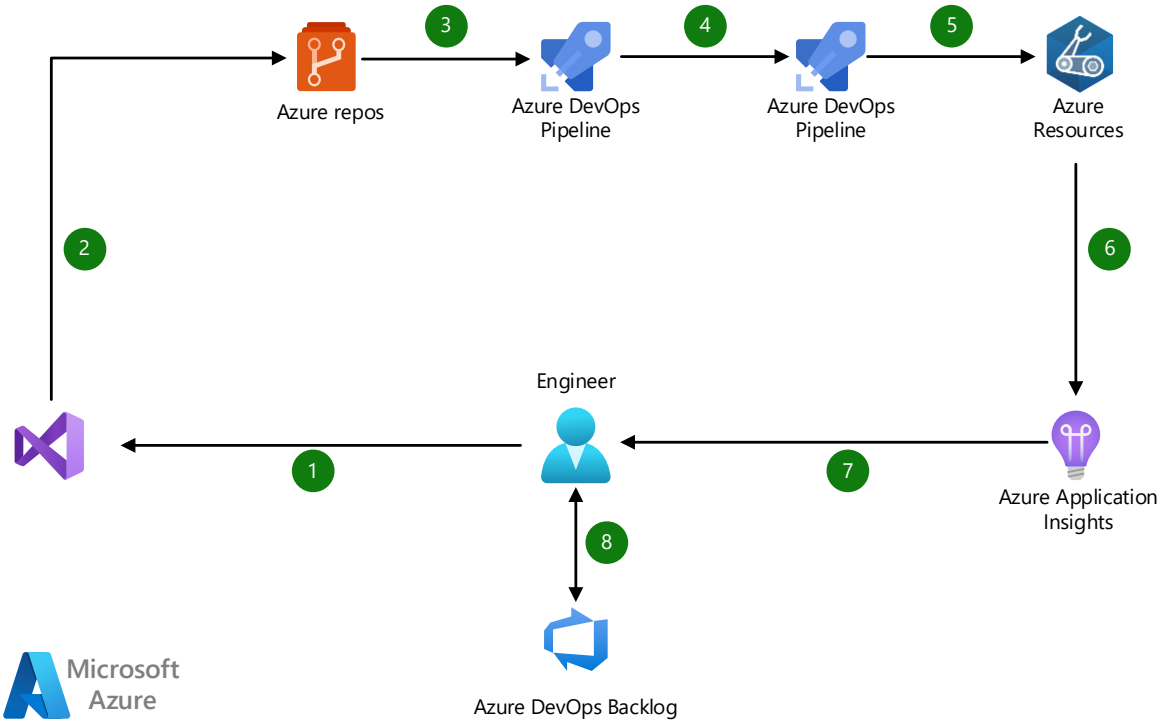


Figure 77: CI/CD Process (I:16)

1. Change application source code.
2. Commit application code and related files.
3. Continuous integration triggers application build and unit tests.
4. Continuous deployment trigger orchestrates deployment of application artifacts with environment-specific parameters.
5. Deployment to the relevant Landing Zone Subscription(s) in Azure.
6. Azure Application Insights collects and analyses health, performance, and usage data.
7. Review health, performance, and usage information.
8. Update backlog item.

To deploy to the Landing Zone Subscriptions in Azure, the Build Agents used to perform the deployment tasks need to have access to the VNet for the deployment target. This requires a build agent to be deployed within the private network and configured within an Agent Pool in DevOps.

The details of the Build Agent deployment will be confirmed during Infrastructure Design Activities outlined in the Contract Master Schedule, in collaboration with oobe, but is likely to look like the following:

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 193 of 1295

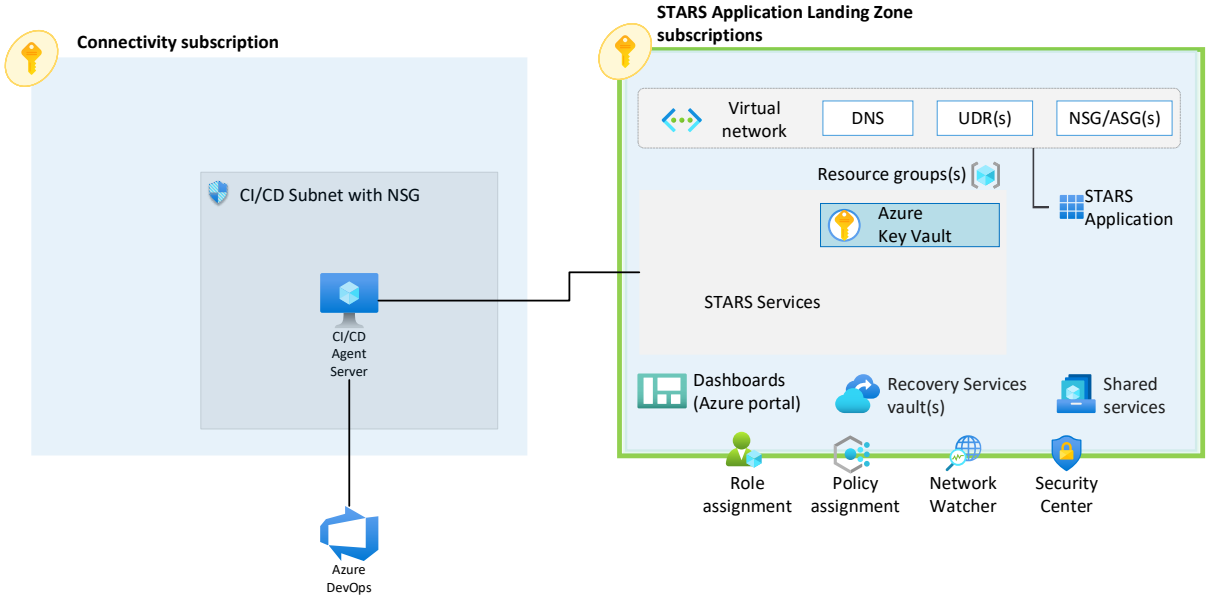


Figure 78: Build Agent Architecture (I:15)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 194 of 1295

12 Security Architecture and Design

12.1 Overview and Purpose

The solution design proposes a set of flexible approaches that are most likely able to support detailed requirements as they are created or evolved during the life of the project (subject to the project scope change management process), and beyond.

This Security Architecture and Design section addresses the security requirements for the STARS solution.

The Security Architecture is described by:

- Outlining the underlying security design principles being used;
- Performing a Security Boundary analysis to examine the security of data travelling across the boundary; and
- Outlining the Security Functions of the STARS System which are guided by the ACSC's cyber security principles: Govern, Protect, Detect and Respond[24].

12.2 Security Design Principles

The security design principles used for the STARS Security Architecture and Design are to ensure that the Confidentiality, Integrity and Availability of the system are maintained over its lifetime. Additionally, to achieve this, the following principals will be applied:

- Minimise attack surface area;
- Establish secure defaults;
- Least privilege;
- Defence in depth;
- Fail securely;
- Separation of duties; and
- Keep it simple.

Additionally, during the development of STARS, applicable ISM controls will be selected, implemented, and assessed. This will also consider Essential Eight Mitigation Strategies[25] which were primarily designed to protect Microsoft Windows-based internet-connected networks.

ISM Security Controls published on 16th June 2022 will be used in Azure DevOps and be linked to STARS Functional Requirements. This will help to identify any security functionality that is required to be developed and not already provided for.

During the assessment phase of STARS Azure DevOps will be able to show a real-time dashboard of the system's security control statuses as they are assessed.

ISM Security Controls in DevOps are tagged with the following assessment scope values which are either the system itself or supporting organisations:

- *STARS* to denote the control is covered under the development and operation of STARS;
- *Microsoft* to denote control coverage is under the IRAP of Microsoft Azure services;
- *oobe* covers the setup and operation of the Azure subscriptions as well as the virtualised development environment;

[24] https://www.cyber.gov.au/acsc/view-all-content/advice/cyber-security-principles
[25] https://www.cyber.gov.au/acsc/view-all-content/publications/essential-eight-maturity-model

DXC TECHNOLOGY
Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 195 of 1295

- *NCVER* covers the system operator organisation (largely a governance scope);
- *DEWR* covers the system owner as it is expected DEWR will be responsible for issuing the Approval to Operate for release 1 (note that the NCVER and the DEWR assessment scopes are the same and are tagged as NCVER only in Dev Ops and in the Security Traceability Matrix);
- *DXC* covers mainly software development controls; and
- *Secops* covers ongoing cyber security monitoring and response.

Most ISM Security Controls tagged with STARS will have a relationship with one or more STARS Functional Requirements.

12.3 Security Boundary

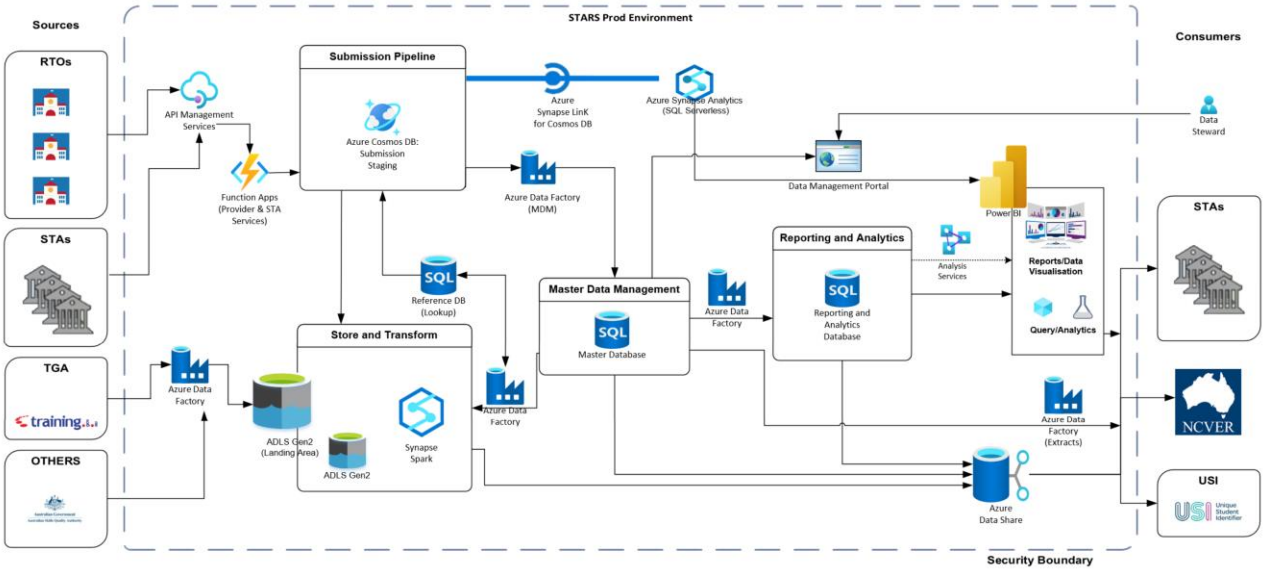


Figure 79: STARS Workload Production Security Boundary Architecture

This is the security boundary of the high-level design architecture of the STARS production environment or core system. The purpose of this diagram is to outline the dataflows and user interactions with this system.

Figure 79 is to be used in conjunction with Figure 83 which outlines the entire STARS Solution including various landing zones and security functions to support the core system. The Detailed Design will outline the permitted network paths and users in more detail.

All Azure services used to build the core system have been assessed in the 2021 Microsoft Azure IRAP report. No issues with the services outlined above have been identified so far. This, however, is subject to detailed analysis that will form the STARS System Security Plan. More information about the services in use are outlines in section 12.3.6 below.

A core function of securing system boundary is the use of oobe Perimeta and the Microsoft subscriptions managed by oobe in commissioning the Azure tenancy. Instances of the core system is deployed into the subscription landing zones noted by the green boxes in Figure 80 below.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 196 of 1295

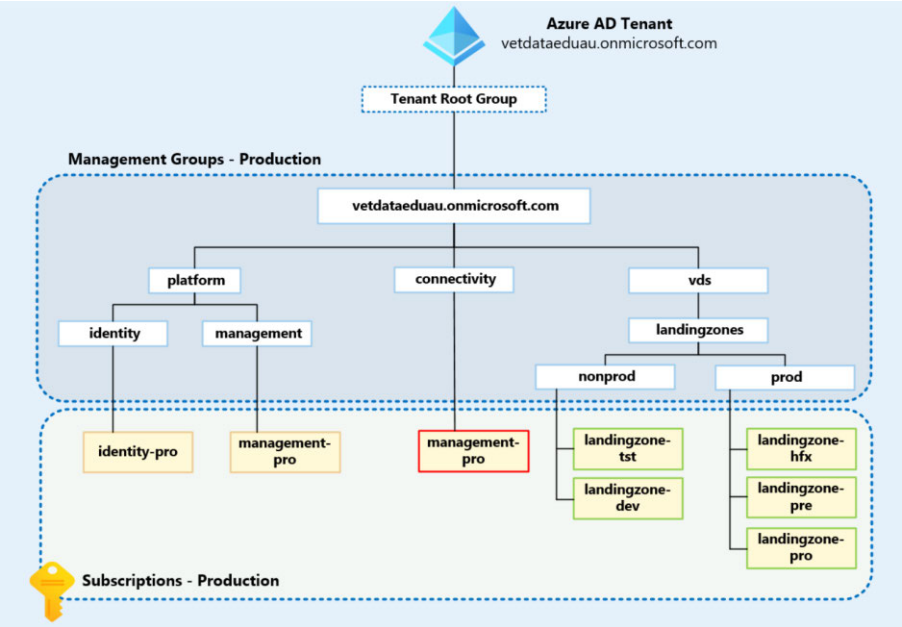


Figure 80: Overview of STARS tenancy and subscriptions

Figure 80 describes the Management Groups provisioned for STARS as part of oobe's Perimeta for Azure solution. The Management Groups provide flexibility for organising policy, access control and compliance, across multiple subscriptions as well as logical boundaries between different environments and resources.

The Microsoft subscriptions in use are:

Type	Purpose	Name
Platform	Connectivity	connectivity-pro
Platform	Identity	identity-pro
Platform	Management	management-pro
Landing Zone	Development	landingzone-dev
Landing Zone	Test	landingzone-tst
Landing Zone	Hotfix	landingzone-hfx
Landing Zone	Pre-Production	landingzone-pre
Landing Zone	Production	landingzone-pro

The Connectivity subscription provides a centralised networking and enforcement capability and does not run any of the client (STARS) workloads.

The Identity subscription provides key storage, cost management and AAD services.

The Management subscription provides administrative tools and supporting infrastructure such as management services, backup, monitoring, patch automation and compliance enforcement.

The Landing Zone subscriptions are the zones for STARS workloads.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 197 of 1295

12.3.1 System Scope

Security control assessment will be inherited from oobe Perimeta for Azure IRAP assessment which will inherit security control assessment from the 2021 Microsoft Azure IRAP assessment. There will be additional security controls that will be required to be assessed for STARS.

12.3.2 Physical Security

STARS will be deployed in Australian hosted Azure tenancies. STARS is currently planned to be deployed into Australia East (Primary) and Australia Southeast (Secondary). The 2021 Azure IRAP report has not highlighted any issues. During the system security assessment, it is expected that SCEC reports exist for the data centres used and will be used to accredit the zone for use by STARS.

From the IRAP report: Azure Datacentre Engineers manage the physical security of the DC data hall premises. DC protections have multiple layers to provide an extensive physical protection presence, this includes:

- Access Approval, involving personnel background checks and systems checks;
- Perimeter requirements, specifically Perimeter fencing, front entrance gate, one defined access point, video coverage and ongoing roaming patrols;
- Building controls, such as two-factor authentication with biometrics, no building signage, and 24x7x365 security operations; and
- Server environment management, video coverage of racks (front and back) employee and contractor vetting, metal detectors, an Inability to identify location of specific customer data; and secure destruction devices.

12.3.3 Identity Management

Two to three systems will be used for user's authentication for identity management:

1. STARS Webapp Authentication – Should be myGovID for RTOs, via the Services Australia Digital Identity Exchange service. Relationship attributes for individuals logging into the STARS web app can be handled via organisational relationships in Relationship Authorisation Manager (RAM)[26];
2. SMS authentication – RAM M2M authentication can be leveraged here to ensure only authenticated SMS providers can send communications via our API gateway; and
3. Azure Active Directory (AAD) – For Azure System Administrators to manage the subscription environments.

Multifactor Authentication (MFA) can be handled by either myGovID smartphone app PIN (or biometric authentication) for normal users or by Azure tenancy conditional access configuration with Microsoft Authenticator for Azure privileged users.

The user authentication flow will be initiated by STARS, link to myGovID using Azure Active Directory B2C[27] providing user email address and the business identifier (already configured in myGovID) and utilise the Australian Government Trusted Digital Identity Framework[28].

[26] https://info.authorisationmanager.gov.au
[27] https://www.digitalidentity.gov.au/sites/default/files/2021-06/tdif-06b-openid-connect-1.0-profile-release-4-v1.6.pdf
[28] https://docs.microsoft.com/en-us/azure/active-directory-b2c/authorization-code-flow

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 198 of 1295

Authorisation will be a distributed RBAC model managed inside AAD by the various stakeholder organisations and overseen by NCVER. See section 12.3.9 for further information about the STARS roles. In addition, the oobe has standard roles deployed in the Azure tenancy.

The roles create by oobe are outlined in the *VDS – Perimeta for Azure – ABAC v1.0* document in table 5 in the *Role-based Access Control (RBAC)* section. There are 233 roles outlined in this table for use across the various landing zones and Azure services in the deployment. These roles include various database, owner, contributor, developer, security, admin (global and service specific), and backup roles.

12.3.4 Network Perimeter

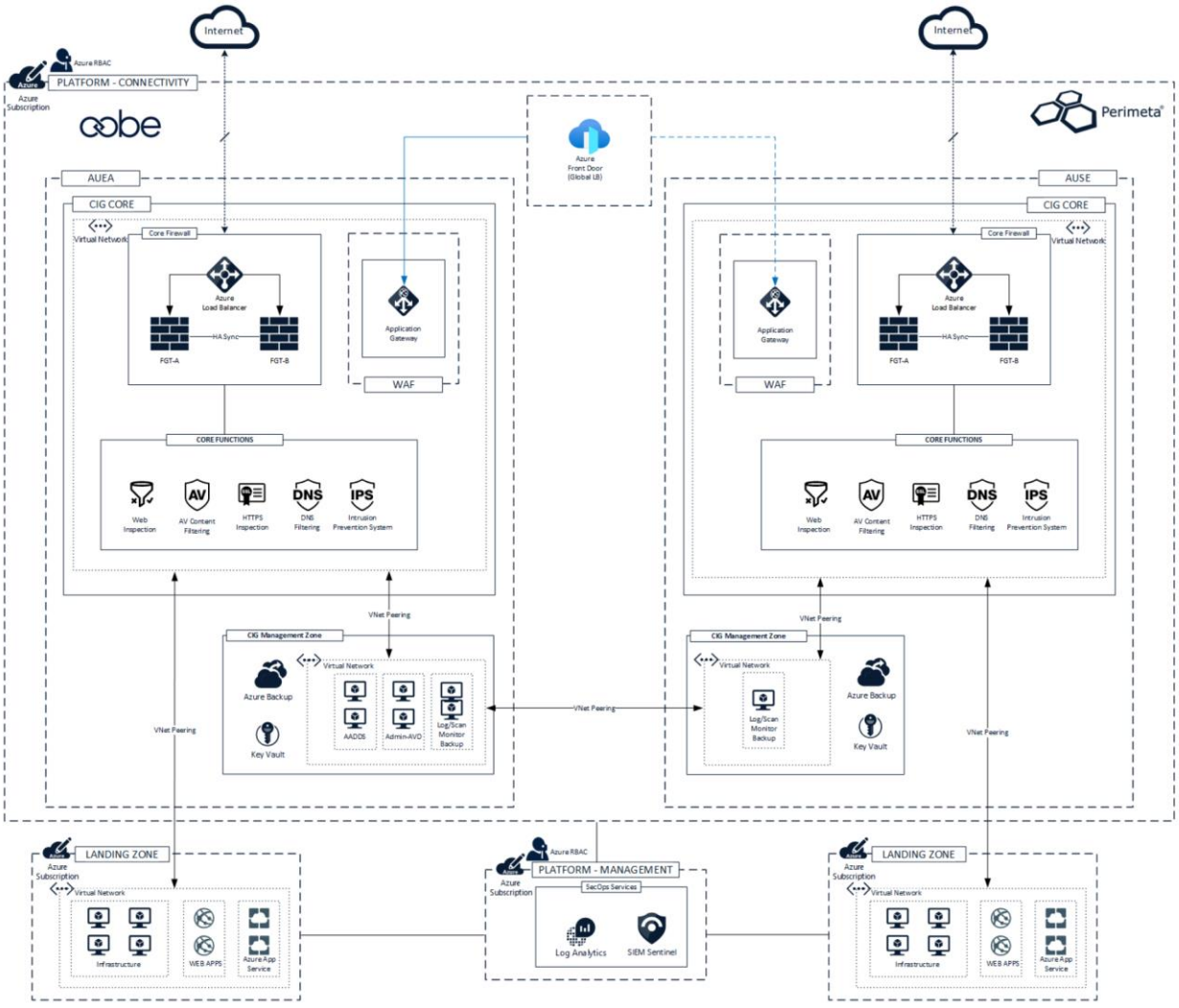


Figure 81: VDS Cloud Internet Gateway High Level Architecture

The network perimeter or connectivity zone is being setup using oobe - Perimeta for Azure and utilises:

- Azure Web Application Firewalls (WAF);
- Azure Front Doors (load balancing – non regional for content distribution);
- Azure Application Gateway (which performs load balancing); and
- Azure API Management.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 199 of 1295

The internet gateway security will be assessed as part of the STARS security documentation set and the above diagram will be updated to reflect the resulting architecture.

12.3.5 Networking

The STARS subscriptions rely on Azure Virtual Network (VNet) to implement networking. The Connectivity subscription is setup by oobe using Perimeta for Azure. Connectivity between the different landing zones will be strictly governed by user-defined routes and network security groups. This will be documented further during configuration and setup of the Azure infrastructure. The following diagram shows the high-level network overview provided by the Perimeta for Azure solution:

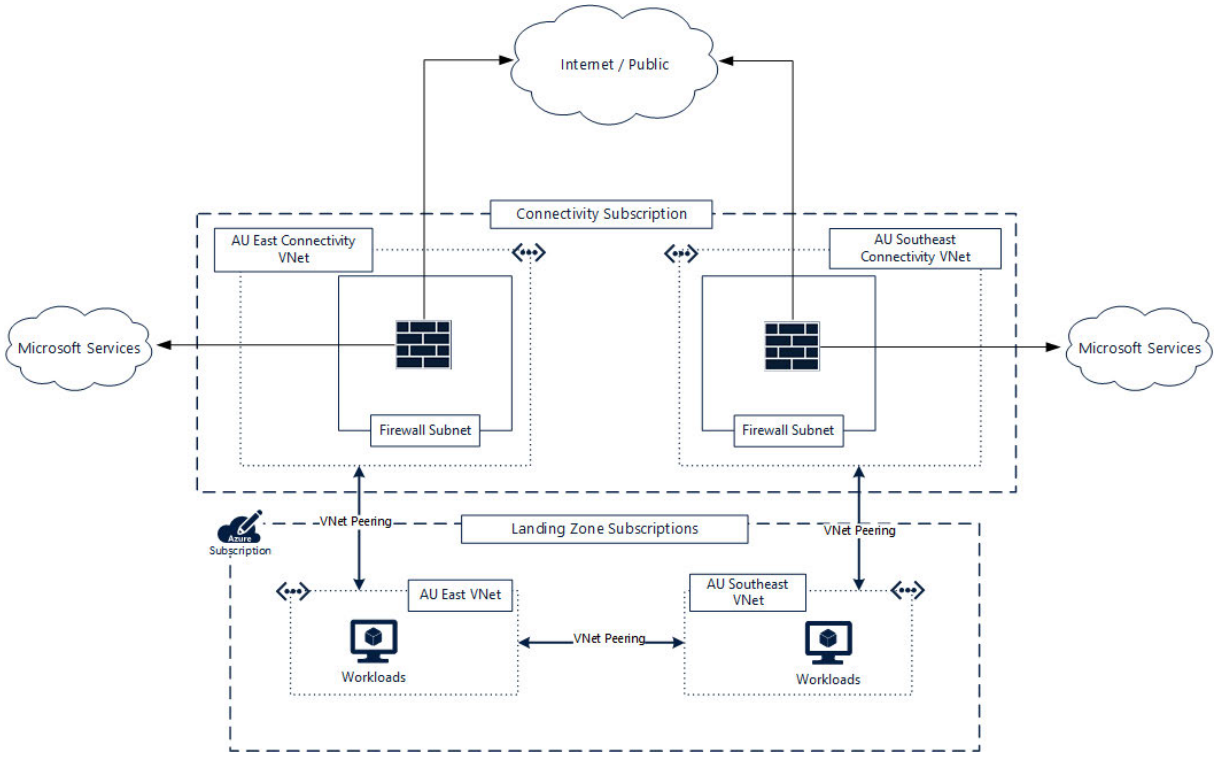


Figure 82 Perimeta for Azure Network Overview

12.3.6 PaaS Services and Applications

The list of services used by STARS include (for a full list see section 11.1 above):

- Azure Functions
- Azure SQL (Stored Procedures)
- Azure Logic Apps
- Power BI;
- Azure SQL Database;
- Azure Data Factory;
- Azure Cosmos DB; and
- Azure API Management.

Security for the API implementation will be guided by https://api.gov.au/sections/api-security.html.

Security assurance within the application layer will be managed through:

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 200 of 1295

- Test security use cases which will be based on the inverse of the Solution Security Matrix – to ensure users do not have unauthorised access to functionality/data;
- Assigning applicable ISM Security Controls to Functional Requirements; and
- Additional security penetration test and/or configuration review (if required).

12.3.7 Encryption Key Management

Any encryption keys in use for the Azure Subscriptions will be managed in Azure Key Vault and restricted users who are assigned the role to access each vault. Each Landing Zone subscription (each environment) will have its own Key Vault instance.

12.3.8 Data

Sovereignty of data is ensured through use of Australian based data centres (Microsoft Australia) and all users/administrators that are based in Australia.

For data consumption, some use cases will require data to be de-identified, or confidentialised. In such cases, STARS will prepare the confidential/de-identifiable unit record file/s for exporting of data. In this instance, the Data Factory workflows/pipelines will include stages that deidentify data (e.g., removing names).

Note that the solution includes creation of Power BI restricted workspaces and will include Power BI Row-level security, roles, etc.

12.3.9 User Roles and Access to Data

The types of application users interacting with STARS (from the security boundary architecture diagram) include:

- RTO/SMS (Registered Training Organisation/Student Management System)
- TGA (training.gov.au)
- USI (Universal Student Identifier)
- DEWR (Department of Employment and Workplace Relations)
- NCVER (National Centre for Vocational Education Research)
- SEE (Skills for Education and Employment)
- STA (State Training Authority)

Different application user types referenced in the requirements include:

- ASQA (Australian Skills Quality Authority)
- Authorised Users
- Consumer
- Data Custodian
- Funder
- Jurisdiction Data Custodian
- National Data Custodian
- Other Collection Agency
- Regulator
- SMS Organisation
- STA (State Training Authority)
- System Operator
- Training Organisation
- VDS Users
- VET Provider

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 201 of 1295

Other types of users include at the tenancy levels:

- Azure administrators (oobe);
- Azure administrators (System Operator);
- Azure developers (DXC, DEWR);

A more detailed overview of users' access is covered under the Security Solution Matrix.

A summary of the Security Solution Matrix is that it shows which users will be allowed access to the STARS functionalities and VET data based on their roles. High level web user interface functions include:

1. Register
2. Login
3. Logout
4. Manage Users
5. Manage Users Access
6. Organisation Management
7. Search
8. Search Report
9. Search Export
10. Tools and Resources
11. Display Notification
12. View (Notification) Reports
13. Display Admin Menu
14. Display Main Menu
15. Display Tools/Resources
16. Client Support Function
17. Display Reporting Menu
18. VET Regulator functions
19. Enter/upload data
20. View Reports
21. Release Data
22. Manage Reference Data
23. Manage Business Rules
24. Data Quality functions
25. STA Data Record Append

Some functionalities will be accessible to all end users such as viewing reports on public datasets. Relevant STAs will have access to state-specific functionalities and VET providers (will have access to their vocational education training data (whether staged or mastered). NCVER and the Department of Employment and Workplace Relation (DEWR) will have relevant oversight of national data for data quality and analytic reporting purposes. VET Regulator will have access to VET regulator specific functions. At a high level, each organisation type will have data users who submit/resubmit data, view validation outcomes, receive notifications/reminders. Organisational Administrators will have privilege to manage their organisational details and their organisational users and user access.

APIs interactions with internal and external components will be managed through the API Management. APIs will have different API IDs allow API authentication and adopt claims-based authentication allowing APIs to access appropriate data for retrieval and update purposes.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 202 of 1295

12.3.10Secured Administration and Development

Security of the administration environment and development environments will be managed through security approved Azure Virtual Desktop environments within the Azure subscription for STARS.

The secured jump hosts for access to the AVD environment will be DEWR managed virtual machines.

12.4 Security Functions

The security functions are outline in Figure 83 below. In this Security Functions section items in the architecture diagrams are described in detail and are noted by bold text.

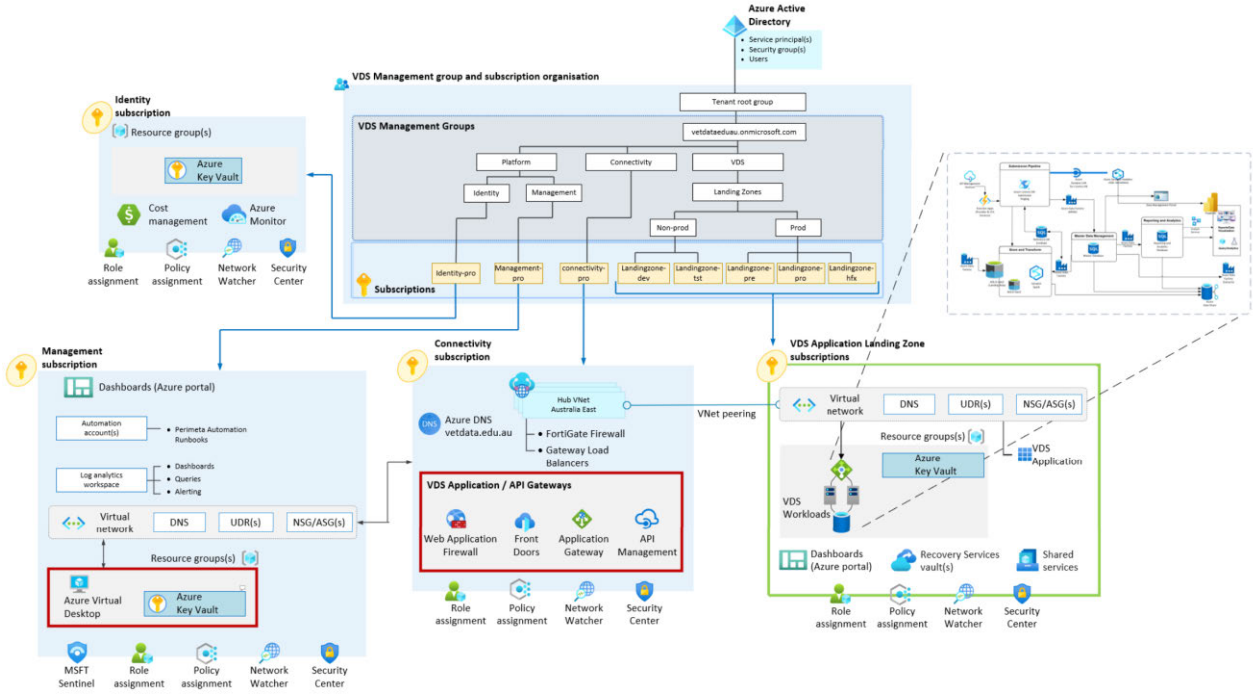


Figure 83: STARS Security Function Architecture

12.4.1 Govern

The System Operator should manage STARS security risks through understanding and accepting the documentation authorisation brief which outlines any risks that have been identified and documented in the STARS Security Risk Management Plan, which will be developed during the STARS implementation.

To guide the Security Architecture phase, a Threat Risk Assessment has been completed using the STARS System background documentation and Functional Requirements listed in the Master Requirements document.

High level overview of the security operation of STARS is through incident monitoring through the utilisation of **Microsoft Sentinel** by the System Operator's Security Operations Centre staff and its oversight by delegated management.

The attack surface area for external malicious actors includes:

- a publicly accessible web and API interface;

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 203 of 1295

- Azure Data Share;
- Azure Power BI interface;
- compromised 3rd party Student Management systems (indirect); and
- compromised 3rd party workstations and mobile devices which embed malware into data import extracts such as excel documents (indirect).

The attack surface area for trusted insiders includes:

- tampering or configuration errors with the Azure tenancies and STARS application;
- tampering with the STARS data;
- inappropriate access to and leaking of student's personal identity and other sensitive information; and
- IaaS vulnerabilities.

Cyber security standards being applied during the development and operation of STARS include:

- Australian Government Protective Security Policy Framework (PSPF);
- Australian Government Information Security Manual (ISM);
- Australian Privacy Principles (APP);
- Open Web Application Security Project (OWASP) guides; and
- Australian Government Open API Standards.

ISO27001 is not a standard being applied to STARS however is a standard that the Department uses for accrediting external IT systems and providers in the Department's "Right Fit For Risk Cyber Security Accreditation" approach[29]. ISO27001 is a very useful standard which is generally applied to at an organisational to uplift the cyber security capability and to manage the associated risks.

Note that the PSPF is design for Australian Government entities to protect their people, information and assets. NCVER, as a government owned not-for-profit Australian company, does not have the same legal obligations to comply with the PSPF. However, STARS will be designed to comply with the PSPF, especially with respect to, information security:

- PSPF Policy 8: Sensitive and classified information;
- PSPF Policy 9: Access to information;
- PSPF Policy 10: Safeguarding data from cyber threats; and
- PSPF Policy 11: Robust ICT Systems.

Governance of STARS culminates with the Approval To Operate (ATO). There are several security artefacts for the VDS program to achieve this approval for STARS:

- STARS Solution Security Matrix for system users;
- STARS System Security Plan Annex (also known as ISM Statement of Applicability or SoA);
- STARS System Security Plan;
- STARS Incident Response Plan;
- STARS Continuous Monitoring Plan;
- STARS System Risk Management Plan;
- STARS Access Policy and Procedures;
- STARS User Access Register; and
- STARS Event Logging Policy and Procedures.

Prior to ATO, it may be decided for STARS to undergo the following additional assessment for assurance:

[29] https://www.dewr.gov.au/right-fit-risk-cyber-security-accreditation

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 204 of 1295

- Penetration Test;
- System Configuration Review;
- IRAP Stage 1 and Stage 2 audit; and/or
- SOC2 Type 2 audit.

STARS will be consistent with NCVER's Privacy Policy[30] (which is aligned with the Australian Privacy Principles and the Privacy ACT) and implement appropriate incident response procedures in event of a data privacy breach.

12.4.2 Protect

Controls to reduce risks and prevent incidents from occurring. These can be in the form of technical, physical and administrative controls.

Security Protection Functions within the STARS architecture include:

- *Microsoft Antimalware for Azure* or *Microsoft Defender for Cloud* which blocks malware from execution for both exporting and importing of data;
- Boundary **FortiGate Firewalls** with tight Firewall ACLs for both ingress and egress traffic;
- **Web Application Firewalls**;
- Azure user roles restriction utilising **Azure Active Directory**;
- Key management with **Azure Key Vault**;
- Tenancy configuration compliance with **Azure Policy Cloud and Compliance Management** which covers prevention of public access by default, enforcing use of private subnets, use of HTTPS, permitted services and regions*;*
- Data segregation (including prevention of traffic between various landing zone subscriptions);
- Data version control;
- Data backup with *Azure Backup*;
- MFA using Microsoft Authenticator for Azure tenancy users and mobile SMS or other authenticator for application users using MyGov ID.
- Tenancy configuration compliance with **Azure Policy Cloud and Compliance Management** which covers prevention of public access by default, enforcing use of private subnets, use of HTTPS, permitted services and regions*;*

Perimeta backs up Microsoft Sentinel logs into a backup server.

Once the Solution Design is finalised there will be a need to hold a workshop to identify the data backup requirements for STARS data.

12.4.3 Detect

Detecting and understanding cyber security events to identify cyber security incidents.

Security Detection Functions include:

- Logging in **Azure Monitor** which covers all state changes of the data and system;
- Networking availability issues with **Azure Network Watcher**;
- Security monitoring with **Microsoft Defender for Cloud** (formerly *Azure Security Center*);
- Alerts in **Microsoft Sentinel**.

[30] https://www.ncver.edu.au/privacy

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 205 of 1295

Note that there is a Log Analytics Workspace in AU EAST. All logs (including security related) are forwarded to this workspace. Sentinel ingests these logs from the log workspace for further analysis.

Note there will be a requirement to link application-level events and logs to the tenancy level tools (Azure Monitor and Microsoft Sentinel) that will need to be scoped out further between oobe and DXC.

12.4.4 Respond

Detecting and understanding cyber security events to identify cyber security incidents requires someone authorised by the System Operator to respond to alerts.

Security Response Functions:

- Identification and assigning of roles;
- Incident Response Plan;
- Incident Response Tracking Register built into solution with System Operator oversight.

Security incident responses can be handled via configurable actions within Sentinel.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 206 of 1295

13 Software Development Kit (SDK)

An SDK will be developed for both the .NET and Java technology stacks to facilitate the submission of VET-IS Data Packets from Student Management Systems (SMS) into the STARS Provider Service.

The SDK will be similar across both stacks, that is, DXC will endeavour to create class libraries that are as identical as possible in form and function, subject to differences in programming languages.

The SDK will be refactored and versioned as necessary as the STARS API changes (see the paragraph below.)

The SDK will include libraries (i.e., DLLs or JARs) that:

- model the STARS domain (e.g., VET information standard entities);
- provide client-side validation of STARS domain classes (e.g., mandatory fields, valid combinations of field values);
- provide easy-to-use façade service classes to transmit information to and from the STARS HTTP/REST/JSON API webservices (i.e., the Provider Service);
- model and handle responses (e.g., report success/fail, detailed error/success classes);
- wrap web service calls to the STARS webservices via the stacks' native HTTP client libraries. They differ between .NET and Java – the SDK will present a uniform approach;
- wrap and manage credentials and authentication. The requirements call for MyGovID machine-machine authentication; the seller will implement libraries and configuration management (below) to enable this; and
- provide classes that allow for platform-specific configuration, e.g., environment configurable settings for webservice URLs, secure locations for MyGovID certificates, etc.

13.1 Distribution of versioned SDK artefacts

SDK libraries will be made available to SMS developers via the Azure DevOps Artifacts feature. Measures will be taken to must ensure that access to the DevOps Artifacts repositories is controlled, most likely through an OpenID-Connect based authentication mechanism.

As the SDK is updated to reflect changes to the STARS API, new versions of the SDK libraries will be released into the artefact repositories. This will be performed as the last step in an automatic build pipeline for the SDK code.

Releases into 'production-ready' repositories will be gated.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 207 of 1295

14 Software Development Tools and Practices

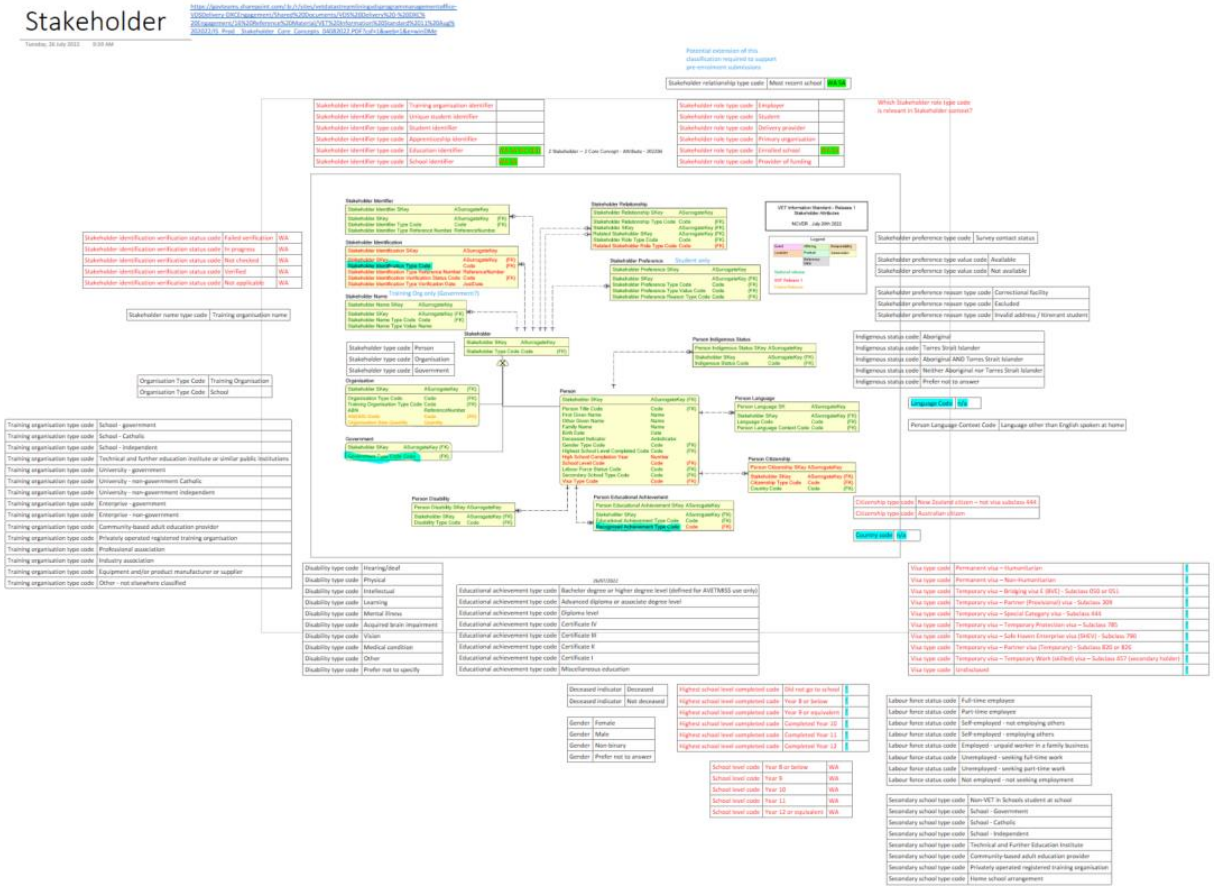
The development tools and practices are detailed in the Delivery and Support Plan.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 208 of 1295

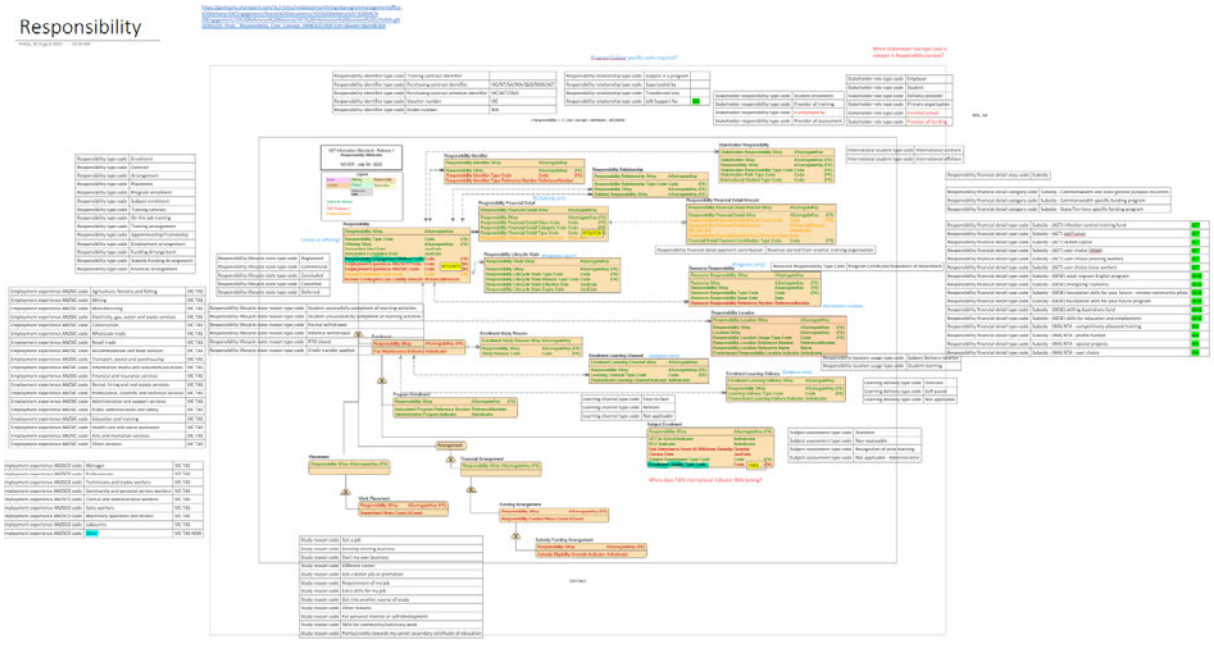
Appendix A. VET-IS Logical Model Entity Relationship Diagrams

Please refer to the Visio file Attachment 6 – VET-IS ERD – D.0.2.zip for a full-size representation.

Stakeholder

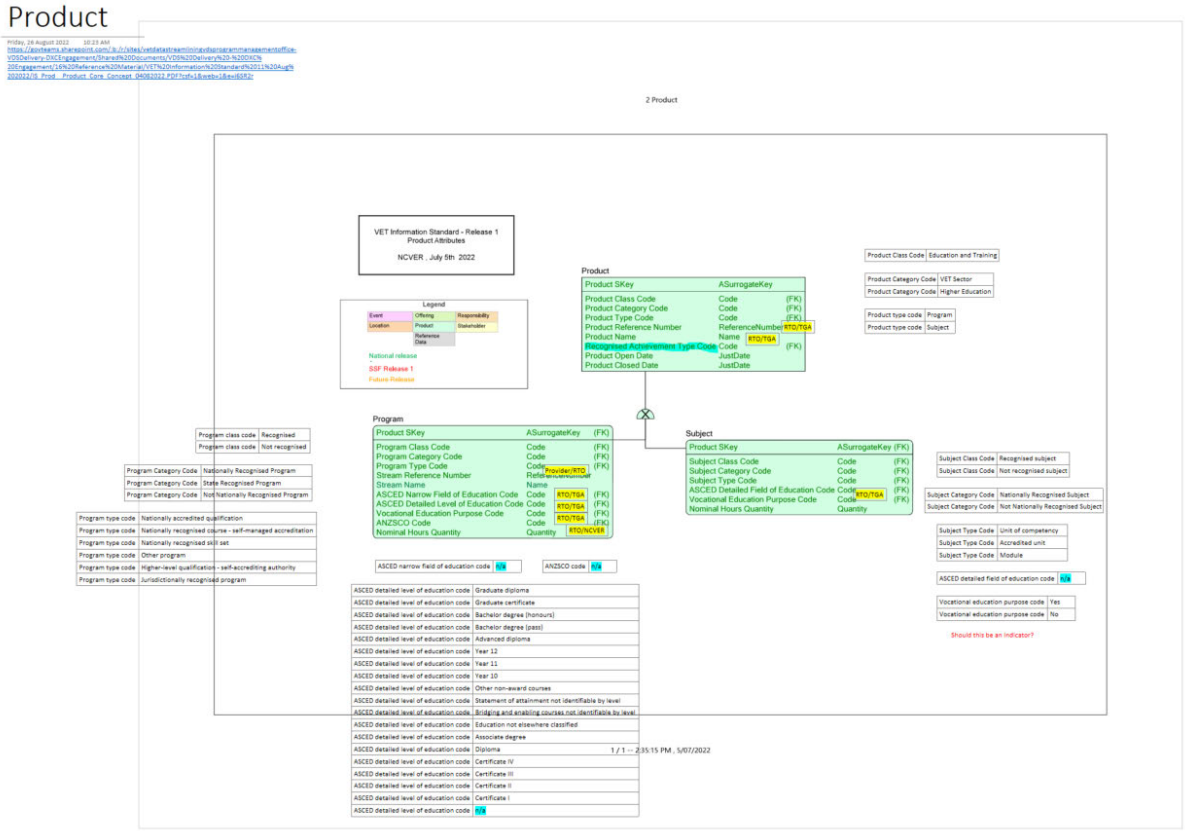


Responsibility

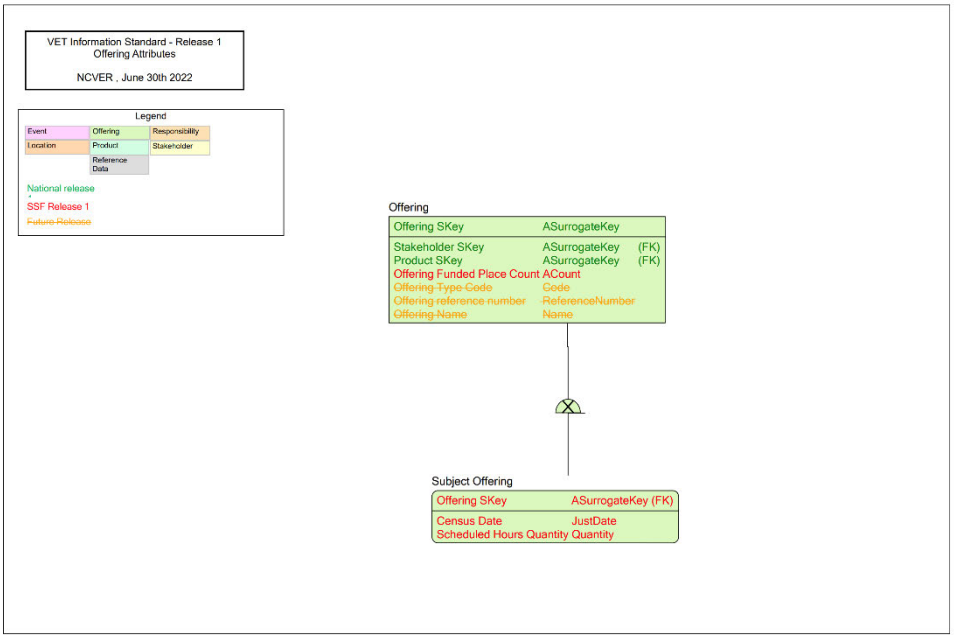


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 209 of 1295

Product

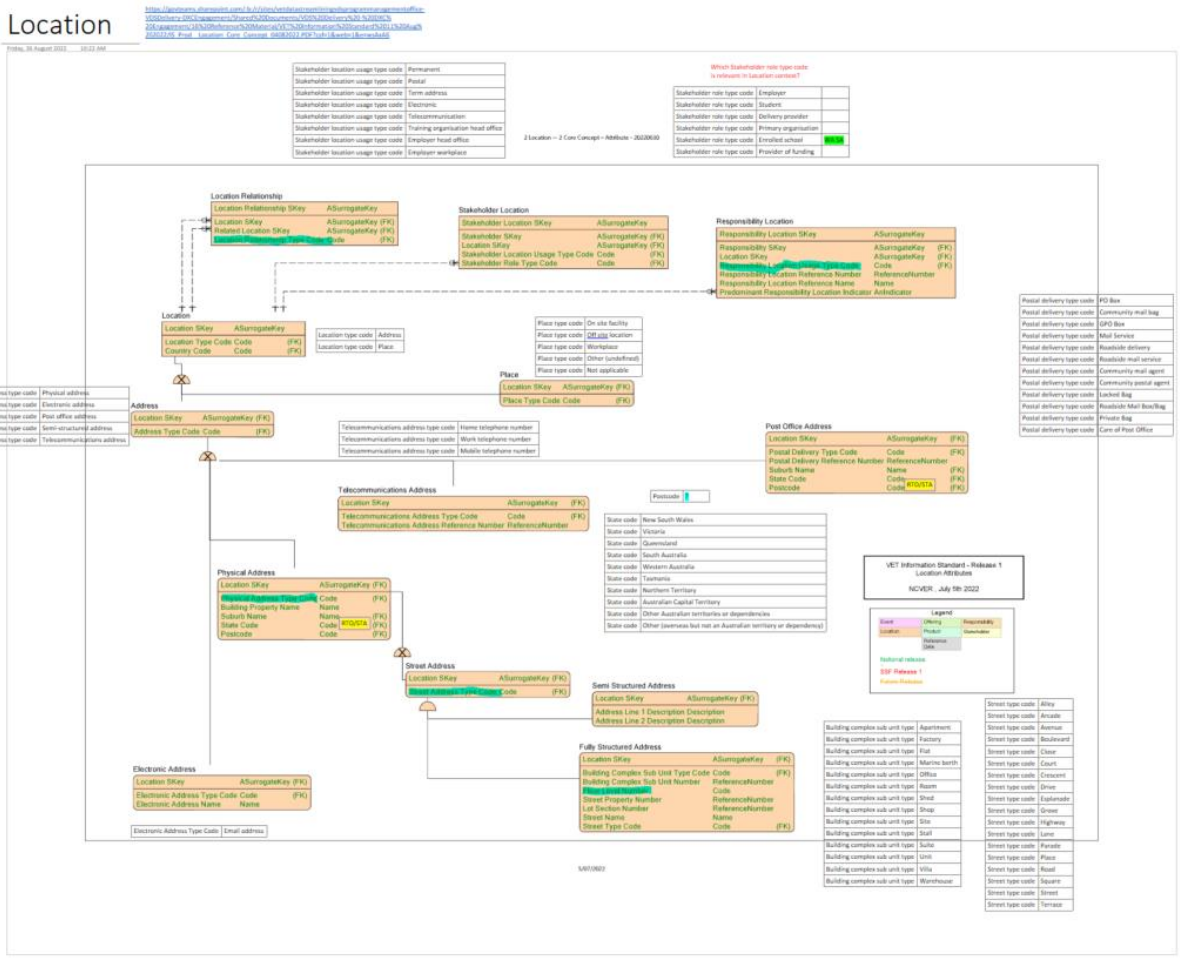


Offering



Location

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 210 of 1295



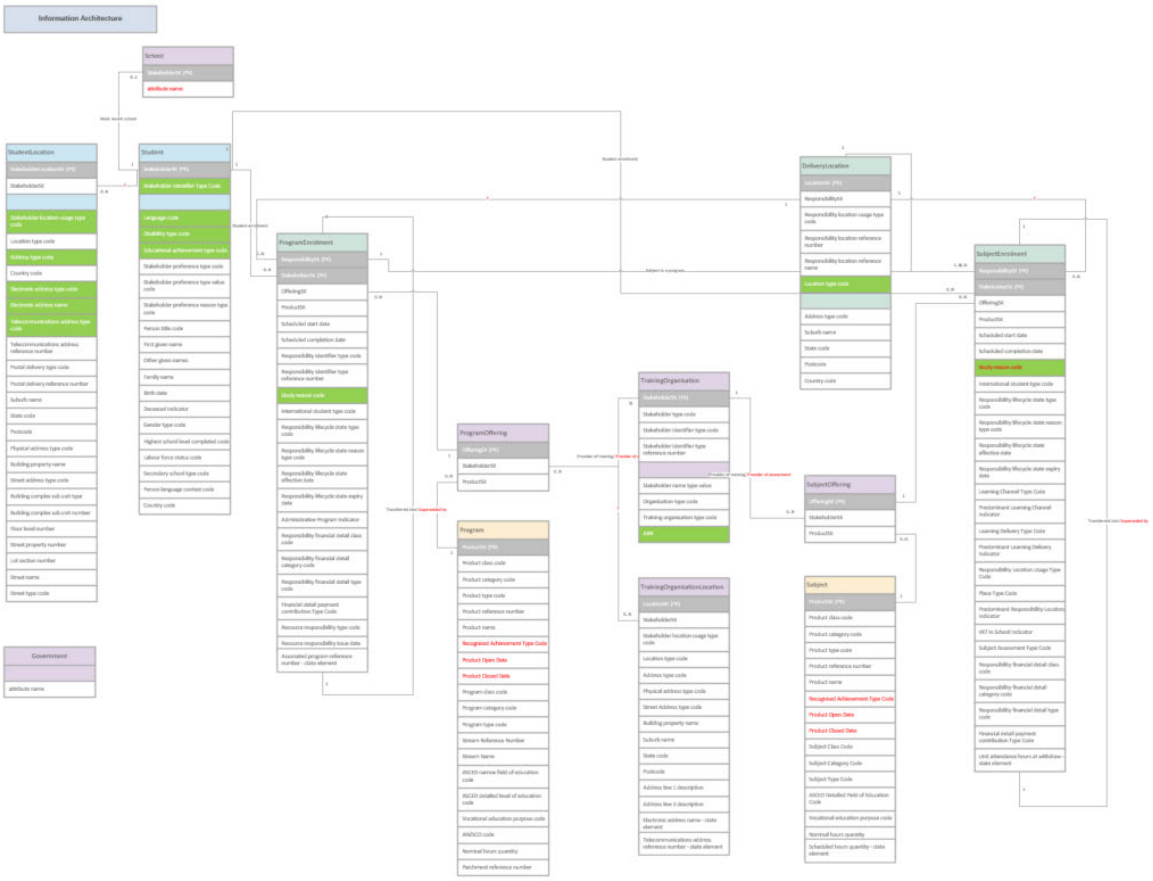
Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 211 of 1295

Appendix B. VET-IS Data Packet Logical Entity Relationship Diagram

The following high level logical ERD diagram show the structure and relationship between the VET-IS data packets.

Note that green fields contain repeating elements. These will be represented in the associated JSON schemas as JSON lists.

Please refer to the Visio file Attachment 4 – Data Packet ERD – D.0.2.vsdx for a full-size representation.



Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 212 of 1295

Appendix C. Stakeholder Overview

For the project, stakeholders are defined as individuals and groups that are impacted by, or can influence delivery of the program's outputs, outcomes and benefits. Stakeholders will be identified as the project progresses and grouped according to their relevant workstream.

As the project progresses, the project team will identify more stakeholders and stakeholders may be recommended to engage by other stakeholder groups. In these instances, the project team will assess the stakeholders input and value to the project.

A stakeholder engagement log will be developed to capture stakeholder roles, engagement objectives, as well as actions and key decisions.

The figure below illustrates the complexity of the VET stakeholder ecosystem. This will be enhanced and refined as further stakeholder identification and analysis continues.

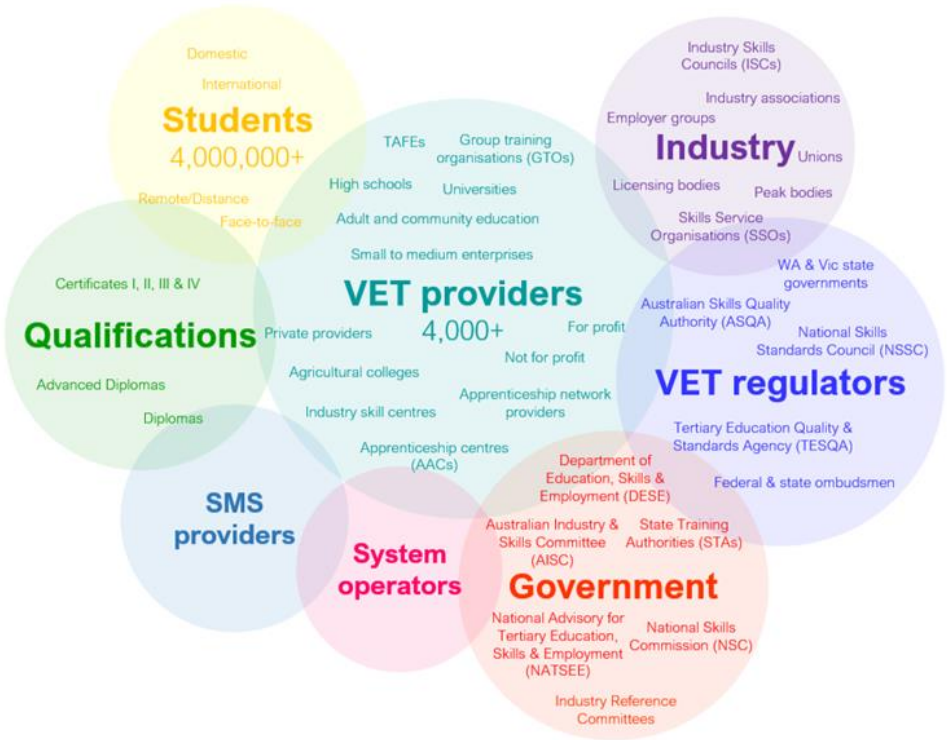


Figure 84: VET Stakeholder Map

There are several types of stakeholders that are impacted by the project. Some of these are users of the solution and understanding how they interact with the current process is critical to user-centred design. As part of the Stakeholder Identification and Assessment, the different stakeholder types (including users) will be identified and assessed.

The table below lists all the identified user types that are expected for the project. The range and types of stakeholders may be expanded upon as the project matures.

Table 14: Stakeholder Users of the STARS System

User	Description
VET Provider	A registered or a non-registered training organisation that submits VET data for compliance purposes. This user would store and consume data from the System

DXC TECHNOLOGY
Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 213 of 1295

Student Management System (SMS)	A provider of a Student Management System that enables one or more VET Providers to submit and validate their data. This user would integrate with the system for consuming and storing data.
Systems Operator	The Systems Operator is responsible for the overall development, integration, modification, operation, maintenance and retirement of the STARS System. This user would not user the system like most, but would need to maintain and enhance the system therefore still requiring consideration (training, knowledge transfer, etc)
State Training Authority (STA)	A state and territory government that has a training authority to administer the vocational education and training (VET). This user would use the system for data consumption to make decisions such as allocating funds, registering training organisations and accrediting courses.
VET Regulator	The VET regulator (registering authority or registering body) that supervises the registered training providers under a legislated jurisdiction. This user may use the system for data consumption services to identify further information regarding a VET Provider.
Commonwealth	An Australian government department or agency that is the national policy leader for VET sector policy and programs. This user would use the system for data consumption services for making decisions about VET within Australia.
Commonwealth Entities	Commonwealth Entities that support the VET Sector stakeholders and provide guidance, policy, governance, reporting and research eg. Training.gov.au, NCVER, ASQA.gov.au

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 214 of 1295

Appendix D. Glossary

Acronym/ Abbreviation	Definition
API	Application Programming Interfaces
AVETMISS	Australian Vocational Education and Training Management Information Statistical Standard
BABOK	Business Analysis Body of Knowledge V3.0
BPMN	Business Process Modelling Notation 2.0
BR	Business Requirements
DEWR	Department of Employment and Workplace Relations
DPT	DEWR Program Team
DTO	Data Transfer Objects
DXC	STARS Solution Provider
FFS	Fee For Service
FR	Functional Requirement
IAM	Identity and Access Management
IIBA	International Institute of Business Analysis
Error! Reference source not found.	Information Technology
Error! Reference source not found.	Information Technology Infrastructure Library
LoB	Line-of-Business
MDM	Master Data Management
NCVER	National Centre for Vocational and Education Research
NFR	Non-Functional Requirement
Error! Reference source not found.	Organisational Change Management
OCM	Organisational Change Management
ODIA	Offer Definition Improvement Activity
RTO	Registered Training Organisation
SMS	Student Management System
SMS	Student Management System
SOA	Service Oriented Architecture
STA	State Training Authority
STARS	Student Training & Activity Reporting System
TGA	Training.gov.au
TO	Training Organisation
TVA	Total VET Activity
USI	Unique Student Identifier
UX	User Experience
VDS	VET Data Streamlining

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 215 of 1295

Acronym/ Abbreviation	Definition
VET	Vocational Education & Training
VET-IS	VET Information Standard
WoG	Whole of Government

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 216 of 1295

Appendix E. Enterprise Data Model (An Example of Data Governance)

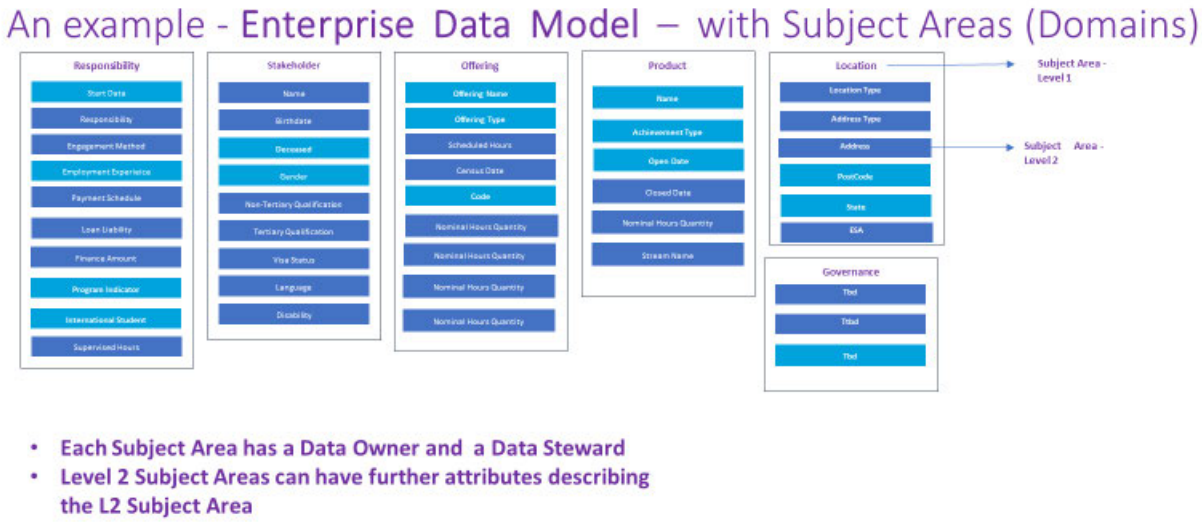


Figure 85: Data governance example

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 217 of 1295

OFFICIAL

Appendix F. BPMN Notation

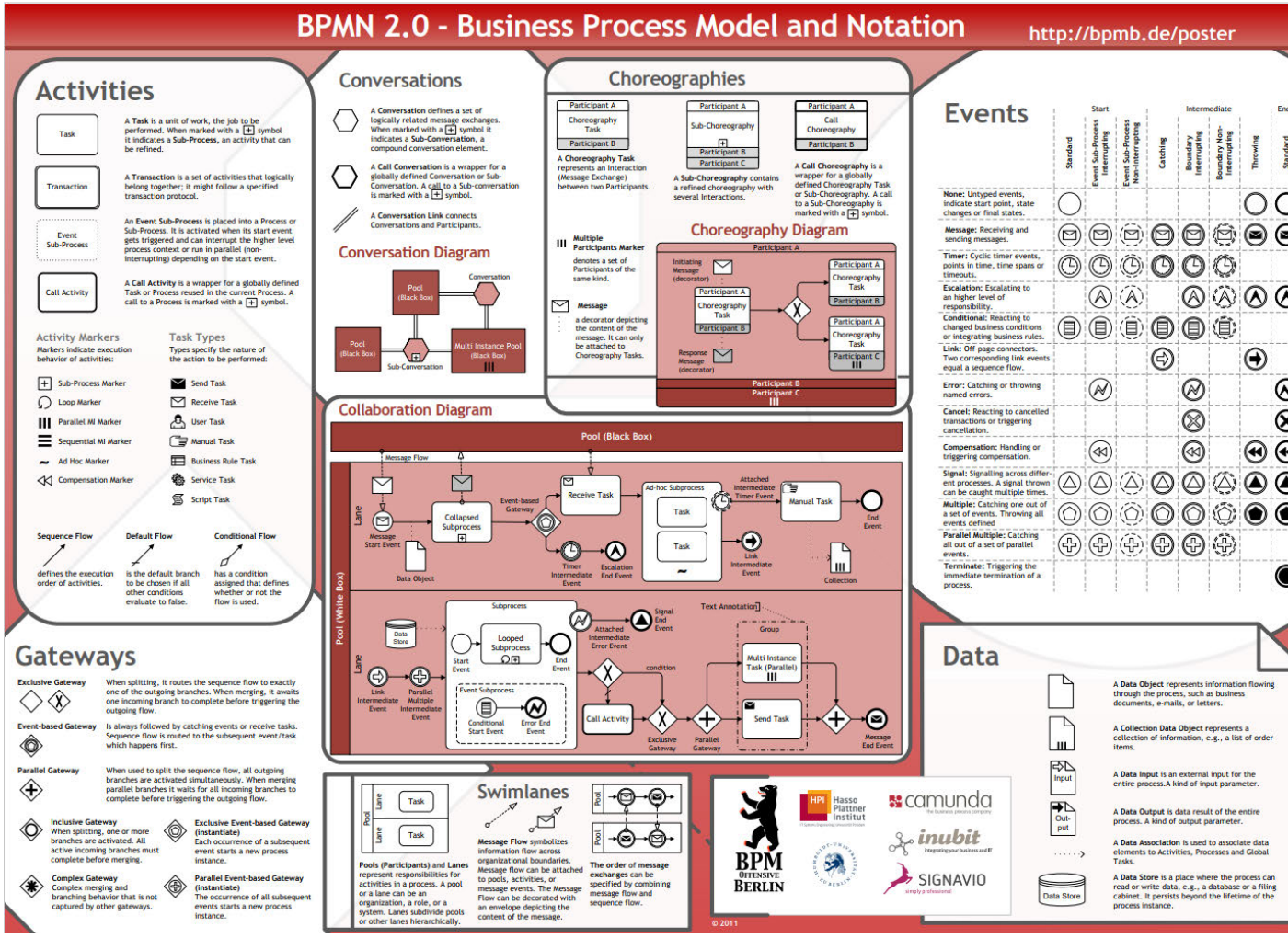


Figure 86: BPMN Notation

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 218 of 1295

Appendix G. Requirements Traceability Matrix

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR001a	Data Submission	Submit or update VIS data	Data submission via API	A training organisation must be able to submit/resubmit a dataset in real-time to the VDS system via a Student Management System (SMS). SMS and training organisations should be able to setup API unaided based on the API interface design (set out in Annexure B) documentation and the training provided by the seller.	9.5.2.2.3 10.2.1 10.5.1.4.2.3 11.1.3
FR001b	Data Submission	Submit or update VIS data	Data entry via user interface	A training organisation must be able to submit/resubmit a dataset in real-time to the VDS system via an adaptive Web User Interface, including options to upload (per FR2) or to directly enter data.	9.5.2.2.1 10.2.1 10.2.2 10.5.1.4.3.1 11.1.3
FR001c	Data Submission	Submit or update VIS data	Data entry via user interface	A training organisation must be able to submit/resubmit a dataset in real-time to the VDS system via a Mobile Application	
FR002	Data Submission	Submit or update VIS data	Data submission formats through user interface	The VDS system must allow a data to be imported in the following formats: • CSV • Excel • Text • JSON • Other dataset formats as determined such as XML	9.6.1 10.2.1
FR003	Data Submission	Submit or update VIS data	Re-submitting data	A VET Provider must be able to submit/re-submit (including correcting historical data) at any point in time all data specified in the VET Information Standard. A VET Provider must be able to re-submit data that has previously been validated at anytime. The system must be able to	9.5.2 10.5.2 10.5.1.3 10.5.1.4 11.1.3

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
				distinguish which record is latest.	
FR004	Validation and Data Quality	Store data in Staging DB	Store data for Validation	The VDS system must store data submitted by a training organisation in a staging database for data validation.	9.5.2.2.4 10.5.2.3 11.1.5 11.1.6
FR005a	Validation and Data Quality	Validate Data submitted by a training organisation against National and Reference data validation rules	Application of validation rules based on dataset	The VDS system must validate data submitted by a training organisation against the VDS National/Reference Data rules. This includes: • Validation from a consistent rule set, regardless of which method was used to enter/upload the data	9.5.2.2.2 10.4 10.5.3.2
FR005b	Validation and Data Quality	Validate Data submitted by a training organisation against National and Reference data validation rules	Application of validation rules to prevent duplicate records	The VDS system must validate the data submitted by a training organisation against the VDS National/Reference Data rules. This includes: • The VDS system will be able to distinguish whether data submitted reflects a new or existing record in order to prevent duplicate records.	8.14.6 8.3 8.5 9.5.2.2.6 9.5.3.2.7 10.5.1.1.2.1

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR006	Notifications	Send error notification to Provider and prompt to resubmit	National rules error notification	The VDS system must notify a training organisation of the VDS National/Reference Data validation result (errors/warnings) in near real-time via: • Student Management System • Adaptive Web User Interface • Mobile App [Mobile App out of scope]	9.3.2.2 9.5.2.2.2 9.5.3.2.3 9.6.1 10.5.3.2.1.2
FR008a	Validation and Data Quality	Validate the Data submitted by a training organisation against the STA/Funder validations	STA/Funder Validation	The VDS system must validate data submitted by a training organisation against the STA/Funder validation rules. This includes: • Validation from a consistent rule set, regardless of which method was used to enter/upload the data	9.3.2.2 9.5.2.2.2 9.5.3.2.3 9.6.1 10.5.3.2.1.2
FR008b	Validation and Data Quality	Validate the training organisation Dataset against the STA/Funder validations	STA/Funder Validation	The VDS system must validate the data submitted by a training organisation against the STA/Funder validation rules. This includes: • The VDS system will be able to distinguish whether data submitted reflects a new or existing record in order to prevent duplicate records.	8.14.6 8.3 8.5 9.5.2.2.6 9.5.3.2.7 10.5.1.1.2.1
FR008c	Validation and Data Quality	Validate the training organisation Dataset against the STA/Funder	STA/Funder Validation	The VDS system must validate the data submitted by the training organisation against the STA/Funder validation rules. • training organisations can see which STA(s) is validating their data. The reason for this is transparency.	10.2.2 10.5.1.4.1 10.5.1.4.3.1

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 221 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
		validations			
FR009a	Validation and Data Quality	Validation response	Real - Time validation results	The VDS system must validate data submitted by training organisation against the STA/Funder Data validation and provide a result (warnings/errors) in near real-time via the interface the submitter used e.g.: • SMS • Adaptive Web User Interface • Mobile App (out of scope)	9.3.2.2 9.5.2.2.2 9.5.3.2.3 9.6.1 10.5.2.3 10.5.3.2.1
FR009b	Validation and Data Quality	Validation response	Real - Time validation results	The VDS system provides a messaging/brokering service which interfaces (via API) with state-managed systems, which run additional state-specific validation rules (including demographics, progress, contracts and eligibility checks against state-managed systems) and provide a response from the STA through the VDS system and to the training organisation. • If STA/funder checks are outside VDS system and not completed within an agreed timeframe, the STA/funder validation response may be separate to national validation response.	9.3.2.2 9.5.2.2.2 9.5.3.2.3 9.6.1 10.5.2.3 10.5.3.2.1 10.5.4.1

OFFICIAL

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR010	Validation and Data Quality	Validation error	Reject data	The VDS system must accept records that have passed validation checks, and reject only records that have failed validation checks, during submission. • The VDS system will notify validation errors/warnings/messages in near real time at the time of submission, where possible	10.5.2 10.5.3.2
FR011	Validation and Data Quality	Submit Data to VDS	Submit data	A training organisation must authorise the submission of data into the VDS system. • If the time limit for submission has been reached, the VDS system will notify the Provider if data has completed a validation check, but has not been authorised for submission.	10.5.1.4 10.5.3.1 10.5.6.1
FR012	Data Storage	Store Data in Staging DB	Store validated data	The VDS system must store (time period tbd) successfully validated Data submitted by a training organisation containing Government Funded activity in the Staging Database.	10.5.6.1
FR013a	Data Storage	Access and use of data in Staging DB	Access and use of data in Staging DB	The VDS system restricts access to data in the Staging Database based on access profiles. • The VDS system restricts access to data in the Staging Database to authorised users such as ensuring only the relevant STA/Funder, or ASQA has access	10.5.5 12.3.10
FR013b	Data Storage	Access and use of data in Staging DB	Access and use of data in Staging DB	A STA/Funder will be able to access, extract and report on data, including related validation outcomes, from the Staging Database through APIs, e.g. REST or Open API, ETL, and Database replication. This includes the ability to extract	8.14.5 8.14.6 10.5.2.3 10.5.4 10.5.5

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
				only the data that has changed since the previous validation.	
FR013c	Data Storage	Access and use of data in Staging DB	Access and use of data in Staging DB	The VDS system will support data to be exported from the Staging Database	10.5.4 10.5.5
FR014a	Data Storage	Conduct Funder QA check	STA/Funder QA	The VDS system must support an STA/Funder to quality assure data on activity they fund.	9.5.2.2.4 10.5.2.3
FR014b	Data Storage	Conduct Funder QA check	National Data QA	The VDS system must support the National Data Custodian to quality assuring data on fee for service activity.	9.5.2.2.7 9.5.3.2.8
FR014c	Data Storage	Conduct Funder QA check	Release data to Master DB	A STA/Funder must be able to authorise the release of a successfully validated and quality assured dataset into the Master Database. • If the time limit for release has been reached, the VDS system will notify the STA/Funder.	9.5.2.2.5 9.5.3.2.6 10.5.2.3 10.5.2.4
FR015	Data Storage	Store in Master DB	Store data in Master DB	The VDS system must store successfully validated and quality assured Data submitted by a training organisation in the Master Database.	9.5.2.2.6 9.5.3.2.7 10.5.1.1 10.5.2.5 11.1.1
FR016a	Data Storage	Perform aggregated data checks	Output Edits	The VDS System must perform output edits on the aggregated data when data is updated/refreshed in the Master Database which includes checking data correctness against aggregated	8.6 9.5.2.2.7 9.5.3.2.8 10.5.2.6

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
				historical VET data and/or usage of statistical analysis to detect data inconsistencies.	
FR016b	Data Storage	Perform Output Edits	Output Edits Action	The VDS system must perform output edits on the aggregated data in the Master Database when the system detects data inconsistency a) Identify the inconsistency b) Notify the Provider (fee for service) and the STA/Funder (government subsidised) of the inconsistency for resolution through an agreed process	8.6 9.5.2.2.7 9.5.3.2.8 10.5.2.6 10.5.6.1
FR017	Data Storage	Store in Master DB	Store output edited data	The VDS System must store the output edited data in the data warehouse (analytics databases).	9.5.2.2.7 9.5.3.2.8 10.5.2.6 10.5.6.1 11.1.9
FR018	Data Consumption and Reporting	Prepare unit record file/s	Unit Records	The VDS system must prepare the unit record file/s for exporting of data.	10.5.5 12.3.9
FR019	Data Consumption and Reporting	Prepare Confidentialised Unit Record File/s	Privacy	The VDS system must prepare the confidential/ de-identifiable unit record file/s for exporting of data. • The VDS system must be able to uniquely identify student level information without identifying the student themselves. • The STA/Funder must be able to create de-identified data. The original identifying data will remain accessible by the relevant STA/Funder.	10.5.5 12.3.9

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR020	Data Consumption and Reporting	Data Consumption	Export formats	The VDS system must allow a dataset to be exported from staging and master databases in the following formats: • CSV • Excel • Text • JSON • Other dataset formats as determined (potentially PDF)	10.5.5
FR021a	Data Consumption and Reporting	Data Consumption	Data Outputs	The VDS system must facilitate data consumption by authorised users via: • BI tools • APIs • ETL	10.5.5
FR021c	Data Consumption and Reporting	Data Consumption	Data Outputs	The VDS system must provide capability to output data from the Master Database at the aggregated level and unit record level, subject to privacy and confidentiality requirements.	10.5.5 12.3.9
FR022	Data Consumption and Reporting	Report and data cube catalogue	Report Types	The VDS system must provide pre-defined reports/data cube that are easily customisable and allow for filtering of data used in reports. This includes: filtering based on field values (e.g. outcome types), customising time periods (e.g. fiscal year, fiscal period, or custom date ranges), and the ability to report on longitudinal data that can be utilised for reporting on the Master Database.	10.5.5.1 11.1.9
FR023	Data Consumption and Reporting	Report BI & Analytics	BI and Analytics	The VDS system must provide a BI and Analytics capability that can be utilised for reporting on data from the Master Database. • This includes the ability to disaggregate the data. This is	10.5.5.1 11.1.9

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 226 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
				applicable for consumers who want to drill down to unit level records when they get aggregated data. Access to unit record data must be in accordance with privacy and confidentiality requirements.	
FR025	Data Consumption and Reporting	Data Consumption	Reporting based on time	The VDS system must allow reporting on point in time data stored in the VDS system Master Database.	10.5.5.1 11.1.9
FR026	VDS System and Data Governance	VDS system and Data Governance	Test environment	The System Operator must be able to test a data classification, business or validation rule in a test environment.	10.4.3

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR027	Validation and Data Quality	Validate Data submitted by a training organisation against National/Funder Reference data validation rules	Managed validation rules	A Systems Operator must be able to manage the National/Reference/Funder Data Validation Rules on behalf of rule owner (STA, NCVER etc): • Create • Read • Update • Delete • Copy • Activate • Deactivate	10.4.3 10.5.3.1
FR028	VDS System and Data Governance	VDS system and Data Governance	Test environment	A Custodian/STA must be able to test a business/validation rule in a testing environment.	See Delivery and Support Plan
FR029	VDS System and Data Governance	VDS system and Data Governance	Compliance Metrics	The System Operator must be able to configure compliance metrics in the VDS system on the behalf of the VET Regulator. E.g thresholds for data integrity metrics to be defined in business rules	10.4.6

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 228 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR030	Validation and Data Quality	Validations and checks	Validate Student enrolment	A SMS must be able to (but not required to) validate the student enrolment information on behalf of a training organisation in real-time.	10.5.3.2
FR031	Validation and Data Quality	Validations and checks	View and extract validation rules	The VDS system must allow the extraction of the National, Reference or STA/Funder Data Validation Rules. A STA/Funder will be able to view, distinguish, and extract the configured national and state-specific validation rules.	10.4.6
FR032a	VDS System and Data Governance	Access Control	Access Control	The VDS system must ensure that the appropriate data access controls (Role and Attribute Based Access Control) are implemented: • Access to Jurisdiction/data custodian specific data items will be controlled by that jurisdiction(s) • Access to National data will be controlled by the National Data Custodian	12.3.9
FR032b	Data Submission	Store in Master DB	Managed Data Access	The VDS system must ensure that the appropriate data access controls (Role and Attribute Based Access Control) are implemented in the Master Database. This includes the ability for a STA to manage access via the Systems Operator: • Based on data set type (i.e. aggregated and de-identified data sets, financial data, national core data, state based program data, and student and individual enrolment data) • Based on assigned user groups	12.3.9 8.13 8.9.2

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 229 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR032c	VDS System and Data Governance	Access Control	Method of access	Users can access data they are authorised to access database reporting capability for staging and mastered data via: • BI tools • standard reports or extracts as per FR020	8.9
FR033	VDS System and Data Governance	Terms and Conditions	T&C	The VDS system must send the current Terms and Conditions of using the VDS system to a user.	10.5.1.4.3.1
FR034	VDS System and Data Governance	Terms and Conditions	Accept T&C	The VDS system must require a user to accept the VDS system's Terms and Conditions prior to submitting data to the VDS system.	10.5.1.4.3.1
FR035	VDS System and Data Governance	Terms and Conditions	Acknowledgement of T&C acceptance	The VDS system must require a user has accepted the VDS system's Terms and Conditions prior to accepting data to the VDS system.	10.5.1.4.3.1
FR036	VDS System and Data Governance	Store in Master DB	Manage data in Master DB	The VDS System must provide the ability to view and trigger a request for update to the authorised source for data committed to the Master Database, in accordance with FR032a and FR032b.	9.5.2.2.7 9.5.3.2.8
FR037	Data Consumption and Reporting	Data Consumption	Reporting between timeframes	The VDS system must allow comparison of data between different time periods.	8.9.2
FR038	VDS System and	VDS system and Data	Data lineage	The VDS system must provide capability for data lineage so that consumers and System	8.5.2.2

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 230 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
	Data Governance	Governance		Operator know how data has changed in its lifecycle from source to consumption.	
FR039	VDS System and Data Governance	VDS system and Data Governance	Metadata	The VDS system must enable metadata management. DEWR is currently using Aristotle and Azure Purview for metadata management. The compatibility of these systems needs to be ensured.	8.5.2.2
FR040	Notifications	VDS system and Data Governance	Notifications	The VDS system must have the ability to add, update and change the notifications to users (error, validation, output edits, reminders): • Content of the notification • Delivery type of the notification • Frequency of the notification	10.5.6.1.3.1
FR041	Data Submission	Update student or enrolment information	Pre-population of data	The VDS system must provide a mechanism to enable the pre-population of student information and training activity data from data already held in the master database. • This will require the VDS system to seek and record permission/consent from the student authorising pre-population	10.5.1.4
FR042a	Data Submission	Submit Data to VDS	Activity data	The VDS system must be able to collect training activity data. For example, this includes: funded and unfunded training activity data, subject-only and program-only activity, estimated/forecast and actual training outcome data, and related fields e.g. unique ID which identifies the training contract in STA systems.	10.5.1.4

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 231 of 1295

OFFICIAL

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR042b	Data Submission	Submit Data to VDS	Non-accredited courses and training organisations	The VDS system must be able to collect non-accredited training activity data. • This will require additional reference data from STAs to support validations. For example, courses not recognised by TGA, and data from non-registered training organisations.	10.5.1.4 10.5.3.2
FR044	Validation and Data Quality	VDS system and Data Governance	Duplication removal	The VDS system must enable the identification of similar records (potential duplicates) for review and remove duplication, e.g. for students who do not have a USI or have more than one USI	8.3 8.5.1 8.5.2.1 8.14.6 10.5.2.5
FR045	Validation and Data Quality	Validation and check	Validation result history	The VDS System must store all validation results, including validation results for each previously submitted entry. This includes metadata describing the method and date/time of the submission. • Results to be kept in accordance with a retention policy	10.5.3.2.1.5
FR046	Data Consumption and Reporting	Data Consumption	Reporting on data submission and validation outcomes	The VDS system must have the ability to generate validation reporting that can be extracted. Standard reports on data validation will be readily available outlining the outcomes of attempted data submissions from training organisations, including success rates, reasons for validation failures, and related trends.	8.9.2
FR047	Validation and Data Quality	Submit Data to VDS	Usability help for Providers	The VDS system must provide supporting contextual information on the validation messages.	10.5.3

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 232 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR048	Data Submission	Submit Data to VDS	State and program data	The VDS system gathers data that distinguishes the relevant STA, program, funding type and funding source (e.g. Commonwealth, state-specific, Fee For Service local, Fee For Service international) that submitted data relates to.	8.14.3 10.5.1.4 10.5.1.3
FR049	Data Submission	Submit Data to VDS	Provisional data submission	A training organisation must be able to provisionally validate data, (i.e. view the outcomes of validation checks) without submitting the data for downstream processing.	10.5.1.4 10.5.3.2
FR050	Data Submission	Submit Data to VDS	Log of historical data and validation results	The VDS system must allow the relevant data custodians to: • View a log of validation results from previous submissions, including result for each entry • View data the training organisation has previously submitted to the VDS system	10.5.3.2 10.5.1.3
FR053	Validation and Data Quality	Validations and checks	Master tables	The VDS system must have the capability to exchange master tables / reference data from any other platform	10.5.5
FR054	Validation and Data Quality	Validations and checks	Reference data sets	The VDS system must validate the Data submitted by a training organisation against reference data sets, for example: • TGA • ANZSCO reference data • Program field of education • Subject field of education • Level of education • Language spoken at home • USI • Training Contract ID and Client ID	10.5.3.2

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 233 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
				• CARMS (NCVER) • address and postcode This includes state-based training programs that are not recognised by TGA.	
FR055a	VDS System and Data Governance	VDS system and Data Governance	Business rules	The VDS system must have capability to create, configure, edit, delete business rules and capturing justifications for these actions • The VDS system will have the ability to add or update business rules that trigger notifications. • The VDS system manages a central register of rules, including a unique ID for each discrete rule, and classification of the rule to help define the rule type, to enable the easy identification of validation outcomes and analysis when providing support to training organisations to submit valid data, and when troubleshooting rule issues with the System Operator. • The VDS system will advise users of all additions and updates to business rules. • The VDS sytem will support users to download business rules and updates.	10.5.3.1

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR055b	VDS System and Data Governance	VDS system and Data Governance	Updating business rules	The VDS system must allow business rules to be easily updated to reflect feedback from training organisations who identify rules that force submission of inaccurate data • The VDS system will have a process for data providers to report problems with the VIS that restrict high quality/accurate data.	10.4.3 10.5.3.1
FR056a	Notifications	VDS system and Data Governance	Notifications for due or overdue data	The VDS system must provide a comprehensive set of notifications: • The VDS system must identify records that are due or overdue to provide a notification to the training organisation and other users where such an entry is missing from the submitted data (e.g. Open Commitment data in IVETS)	10.5.6.1
FR056b	Notifications	VDS system and Data Governance	Notifications to user groups	The VDS system must provide a comprehensive set of notifications: • The VDS system will enable configuration of notification rules to notify specific user groups (e.g. a STA/Funder or other users) based on the validation outcome.	10.5.6.1
FR056c	Notifications	VDS system and Data Governance	Message exchange	The VDS system must provide a comprehensive set of notifications: • The VDS System must enable message exchange for errors, warnings, system notifications, etc.	10.5.6.1

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 235 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR056d	Notifications	VDS system and Data Governance	Notifications to training organisation when data is pending validation	The VDS system must provide a comprehensive set of notifications: • The VDS system must notify the user when data entered is pending validation and notify the user again when the data has been validated.	10.5.6.1
FR058	Data Storage	Conduct Funder QA check	Automate the release of data from Staging DB to Master DB	A STA/Funder must be able to perform manual QA checks for funded data in the Staging Database, after it has passed validation, and before it is released to the Master Database: • A STA/Funder must be able to automate the release of data from the Staging Database into the Master Database, after it has passed validation, i.e. bypassing the manual QA check step	10.5.2.3 10.5.2.4 9.5.2.2.4 9.5.3.2.2
FR059	VDS System and Data Governance	VDS system and Data Governance	Data labels and Identifiers	The VDS system must assign data labels and allow for data tagging. The VDS system must be able to export labelled data and provide identifiers to allow linkage with data from other systems like SEE, ADMS, NESM and "Skills, Education and Employment Data integration platform".	8.5.2.2
FR060	VDS System and Data Governance	VDS system and Data Governance	Record Connecting	The VDS system must provide capability to connect all records for a particular student	10.5.5 8.9 12.3.8

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 236 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR061	Validation and Data Quality	Send rejection notification to training organisation and prompt to resubmit	Cumulative validation	The VDS system must perform validation on the cumulative data held within the VDS system for identified data item before it hits the data warehouse (analytical database).	10.5.2.6 9.5.3.2.8 9.5.2.2.7 8.6
FR062	Data Consumption and Reporting	Data Consumption	Reporting on trends over time	The VDS system must allow reporting on trends in data over time stored in the VDS system Master Database for data integrity assessments/validations	8.9.2
FR064	Validation and Data Quality	VDS system and Data Governance	Reference data curation and enrichment	The VDS system must leverage best practice industry standards for reference data curation and management. If the reference data is imported into the VDS system, it must be updated whenever the source data is updated.	10.5.1.2
FR065	VDS System and Data Governance	VDS system and Data Governance	Reporting on system performance	The System Operator must be able to monitor and report on system performance, e.g. uptime, response time, etc.	10.5.6.3 11
FR068	Validation and Data Quality	Submit Data to VDS	Data Submission - USI dataset	The VDS system must provide regulators the ability to submit minimum datasets to support Student Transcripts for students/former students with a USI number	10.5.1.4
FR069	Validation and Data Quality	VDS system and Data Governance	Data standard version	The VDS system must associate every unit record with the version of the data standard it was validated with.	10.5.3.2

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 237 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR072	VDS System and Data Governance	VDS system and Data Governance	API authentication	The VDS system must allow API authentication by: • SMS and training organisations have different API IDs • SMS providers have an API token that allows them to submit data on behalf of their training organisations • training organisation can assign a token that to their SMS to validate and submit on their behalf	12.3.3
FR073	Notifications	VDS system and Data Governance	Reporting requirements where no activity has occurred	The VDS system must provide a mechanism to identify when data hasn't been submitted/updated within an expected timeframe.	10.5.6.1
FR074	Validation and Data Quality	Validations and checks	Data integrity adherence	The system must be flexible enough to identify incorrect data upfront and also have the functionaility to accept 'exceptional' data. For example: • Numerical value in a name • Name is NULL • Students are twins with same last name, DOB, address and other personal details	10.5.1.4 10.2.1
FR075	Validation and Data Quality	VDS system and Data Governance	Data Profiling	The VDS system must have the ablility to perform data profiling quickly, and to easily assess data quality trends on new datasets or new elements in existing datasets, for which there is no historical data to support trend analysis or the creation of data quality business rules. Data Profiles should be reportable and accessible to custodians of those datasets.	8.5

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 238 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR076	Data Submission	Submit Data to VDS	Other Collection agencies are able to provide bulk event data to VDS system using any version of the VET Information standard that is still accepted nationally.	The VDS system must allow another collection agency to upload data that does not exist in VDS system and the collection agency will receive validation responses on behalf of training organisation. This can be a partnering or non-partnering jurisdiction.	9.5.3
FR077	Data Submission	Submit Data to VDS	Other Collection agencies are able to send near real time event data to VDS system using any version of the VET Information standard that is still accepted nationally.	The VDS system will allow another collection agency to upload data that does not exist in VDS system and the collection agency will receive validation responses on behalf of training organisation. This can be a partnering or non-partnering jurisdiction.	9.5.3
FR078	Data Submission	Direct Entry	Search Student	As a Training Organisation User using data entry functionality, I want to be able to search for a student that is enrolled (or has previously enrolled) with my organisation so that I can view/maintain the student details	10.5.1.4
FR079	Data Submission	UI Direct Entry	View Student and Enrollment Info	As a Training Organisation User using data entry functionality, I want to be able to view the student information of a particular student associated with my organisation so that I can view/maintain the student details	10.5.1.4

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 239 of 1295

OFFICIAL

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR080	Data Submission	UI Direct Entry	Update of student info	As a Training Organisation User using data entry functionality, I want to be able to update the student information of a particular student associated with my organisation so that I can update the student details	10.5.1.4
FR081	Data Submission	UI Direct Entry	Update enrollment	As a Training Organisation User using data entry functionality, I want to be able to update the student enrollment information of a particular student associated with my organisation so that I can update the student details	10.5.1.4
FR082	Data Submission	UI Direct Entry	Bulk Enrollment	As a Training Organisation User using data entry functionality, I want to be able to batch enroll a list of students to selected program/subjects so that I do not need to enroll them one by one	10.5.1.4

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 240 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR083	Data Submission	UI Direct Entry	Bulk Update of Enrollment	As a Training Organisation User using data entry functionality, I want to be able to perform batch update for students involved in a particular training so that I do not need to update for each student one by one.	10.5.1.4
FR084	Data Submission	Data Resubmission	Handling of Data Resubmission	As a Training Organisation User (using data entry functionality or SMS system), I want the previous submission records of the same data to be treated/marked as historical so that those records do not need to be reprocessed again	10.5.1.4
FR085	Data Submission	Upfront Assessment of Need	Capturing whether UAN has been performed and date of assessment.	As a Training Organisation User providing community training (ACE Program), I want to be able to record whether I have completed the Upfront Assessment of Need (UAN) process and date of UAN so that I can submit this info to STA and get paid.	10.5.1.4
FR086	Data Submission	Upfront Assessment of Need	Capturing Pathway Outcome(whether the training has met the training objective (UAN))	As a Training Organisation User providing community training (ACE Program), I want to be able to record the pathway outcome in terms of how/whether the training objective has been met.	10.5.1.4
FR087	Data Consumption and Reporting	Training Organisation and SMS Provider Support	Training Organisation Support To move from Data Entry Functionality to SMS	As a Training Organisation user using data entry functionality, I want the system to be able to support the data migration for data related to my organisation from VDS system to a SMS system when I want to switch to use a Proper SMS system so that I do not need to manually input these data into SMS system.	10.5.1.4

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 241 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR088	Validation and Data Quality	Exception Management	Exceptional Data Handling (Auto reprocessing of latest records when any lookup data changes)	As a STA User, I want the system to auto reprocess the list of failed records (those without a newer record as a result of resubmission) where any lookup data changes could potentially resolve the errors (e.g reference data fixed or data package received out of order) so that those errors can now be auto resolved rather than relying on manual human intervention.	10.5.1.2 10.5.3.2
FR089	Validation and Data Quality	Exception Management	Exceptional Data Handling (Training Org request for review)	As a Training Organisation/STA user, I want to be able to request for review of the data submission violation so that my data can be accepted in exceptional scenario.	10.5.1.4 10.2.1 10.2.2

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 242 of 1295

OFFICIAL

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR090	Validation and Data Quality	Exception Management	Exceptional Data Handling (System Operator handling the request for review)	As a system operator, I want to be able to retrieve the list of records and errors/warnings that were requested for review by training organisation/STA so that I can identify the issue, resolve the problem and re-process the data	10.5.3.1
FR091	Validation and Data Quality	Exception Management	Exceptional Data Handling (STA User handling the request for review)	As a STA User, I want to be able to retrieve the list of records and errors/warnings that were requested for review by training organisation so that I can identify the issue, resolve the problem and re-process the data	10.5.3.1

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 243 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR092	Data Migration	Data Migration	National Reference Data	As a system operator, I want to migrate the current set of national reference data including the following: .Registered Training Organisations .Courses/qualifications/skills/modules/units and associated training organisations .Study Reason .Mode of Delivery .Outcome Identifer (National) .Funding Source (National) .Specific funding identifier .School type identifier??? .Commencing Program Identifier??	See Data Migration Plan
FR093	Data Migration	Data Migration	State Specific Reference Data	As a STA user, I want to migrate the current set of state specific reference including the following : .Non Registered Training Organisations .Training Organisation Delivery Locations (including Non-Registered and Registered organisations) as the location identifer is sourced from state specific system .Non-Accredited courses/qualifications/skills/modules/units and associated training organisations .Funding Source - STA (referenced in current NAT00120 file) .School Location ID	See Data Migration Plan
FR094	Data Migration	Data Migration	Data of Below the Line Fields	As a STA user, I need the data in current below the line fields to be migrated to VDS system (Currently some of the below the line fields do not appear in NCVER data though we report them) and we need to evaluate how those info would be	See Data Migration Plan

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 244 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
				supported during data migration	
FR095	Data Migration	Data Migration	Handling of overloaded fields	As a STA user, I want the data migration to map currently overloaded fields to the desired fields in new standard so that we do not overload the fields in new data	See Data Migration Plan
FR096	Validation and Data Quality	Training Org Reference Data	Management of Training Organisation Delivery Location via UI for non-Registered Training Org without SMS	As a non-Registered Training Organisation using data entry functionality, I want to be able to maintain the training delivery location of my organisation so that I can use them for data submission purpose.	10.5.1.4 10.5.1.2
FR097	Validation and Data Quality	Training Org Reference Data	Management of Training Organisation Delivery Location for Registered Training Org without SMS	As a Registered Training Organisation using data entry functionality, I want to be able to maintain the training delivery location of my organisation and keep source of truth in one system so that I do not need to manually maintain/update in multiple system/government agencies (e.g ASQA/TGA).	10.5.1.4 10.5.1.2
FR098	Validation and Data Quality	Training Org Reference Data	Management of Training Organisation Delivery Location for Training Org using SMS	As a Training Org using SMS, I want to be able to maintain my training delivery location in SMS and the information will be submitted/synced to VDS system so that I do not need to maintain the information manually in multiple systems.	10.5.1.4 10.5.1.2

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR099	Validation and Data Quality	National Reference Data	Management of Registered Training Organisation Data	As a system operator, I want the registered training organsations (and associated data) in TGA to be synced to VDS system so that no manual processing of the training organisation info is required in VDS system.	10.5.1.2
FR100	Validation and Data Quality	National Reference Data	Management of accredited Courses/Qualifications/Skills/Units and training organisation Scope	As a system operator, I want the accredited Courses/Qualifications/Skills/Units and training organisation Scope in TGA to be synced to VDS system so that no manual handling of the accredited course/qualification/skills/units is required in VDS system.	10.5.1.2
FR101	Validation and Data Quality	State Reference Data	Management of Non-Registered Training Organisation Data by STA	As a STA User, I want to be able to manage the list of Non-Registered Training Organisation data in the system and keep the source of truth in one system so that I do not need to maintain the information in multiple system manually (e.g, VDS system, State Specific System	10.5.1.4 10.5.1.2 10.5.4
FR102	Validation and Data Quality	State Reference Data	Management of Non-Registered Training Organisation Data by non-Registered Training Organisation Admin user	As a Non-Registered Training Organisation Admin user, I want to be able to manage my organisation info in the system (address, contact info etc) so that I do not need to raise requests to STA to maintain my info	10.5.1.4 10.5.4
FR103	Validation and Data Quality	State Reference Data	Management of Non-Accredited (State specific) Courses/Qualifications/Skill Sets/Modules/Units	As a STA user, I want to be able to manage (and keep the source of truth) of the non-Accredited Courses/Qualifications/Skills/Units for my state in only one system so that I do not need to maintain the info in multiple system manually (e.g, VDS system, State Specific Systems)	10.5.1.2 10.5.4

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 246 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR104	Validation and Data Quality	State Reference Data	Management of Funding Source - STA (Current NAT00120 Funding Source - State Training Authority Field)	As a STA user, I want to manage Funding Source - STA and keep the source of truth in one system only so that I do not need to manaully maintain the info in multiple systems (e.g VDS system, State Specific systems)	10.5.1.2 10.5.4
FR105	Validation and Data Quality	State Reference Data	Management of School Location ID (Used by SA when the training activity is for VET in School)	As a STA user, I want to manage the school locations and ID (for VET in school) and keep the source of truth in one system so that I do not need to maintain the info in multiple systems manually (e.g VDS system, State Specific Systems)	10.5.1.2 10.5.4
FR106	Data Consumption and Reporting	Reports	All Active Users of a training organisation for training organisation Admin	As a training organisation Admin, I want to be able to view the full list of active users related to my organisation so that I can know who are the active users and manage user access within my organisation. request of deactivation of any user when necessary	10.2.2 (last row)
FR107	Data Consumption and Reporting	Dashboards and reports	Contact method and person/people for a particular training organisation for STA	As a STA user, I want to be able to view the contact method and the person (people) for a particular training organisation that is delivering training in my state so that I can contact them when necessary.	10.2.2 (last row)
FR108	Data Consumption and Reporting	Dashboards and reports	Data Submitted for STA	As a STA user, I want to be able to view the data submitted by a particular training organisation that is delivering training in my state regardless of the method of data submission so that I am aware of what data has been submitted and the validation status.	Pipeline Management Uis 10.2.2 10.3.2 10.5.2.3 10.5.2.4 10.5.2.5 10.5.2.6

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 247 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR109	Data Consumption and Reporting	Dashboards and reports	Error and Warning Report for STA	As a data custodian, I want to be able to view the list of unresolved errors and warnings so that I am aware of the data quality and can follow up from there.	Pipeline Management Uis 10.2.2 10.3.2 10.5.2.3 10.5.2.4 10.5.2.5 10.5.2.6
FR110	Data Consumption and Reporting	Dashboards and reports	Data Submitted for Training Org	As a Training Organisation user, I want to be able to view the data submitted by my organisation so that I am aware of what data has been submitted and the validation status.	10.5.1.4
FR111	Data Consumption and Reporting	Dashboards and reports	Error and Warning Report for Training Org	As a Training Organisation user, I want to be able to view the list of unresolved errors and warnings related to my organisation so that I am aware of the data quality and can follow up from there.	10.5.1.4
FR112	Data Consumption and Reporting	Dashboards and reports	Training Organisation Delivery Location for training org	As a Training Organisation user, I want to be able to view the list of training delivery locations related to my organisation so that I can use them for data submission purpose.	10.5.1.4
FR113	VDS System and Data Governance	User Management	User Sign In	As a System User, I want to be able to sign in to the system so that I can perform the actions required.	10.5.1.4 12.3.3
FR114	VDS System and Data Governance	User Management	User Sign Out	As a System User, I want to be able to sign out from the system so that I do not remain logged on unnecessarily	10.5.1.4 12.3.3

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 248 of 1295

OFFICIAL

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR115	VDS System and Data Governance	User Management	Register New User	As a System Operator, I want to be able to register a new user and grant the necessary roles so that the user can access the system with relevant permissions	10.5.1.4 10.2.2 (last row) 12.3.9
FR116	VDS System and Data Governance	User Management	Maintaining User Roles	As a System Operator, I want to be able to maintain the roles related to the user so that the user can be granted/removed the necessary permissions.	12.3.9
FR117	VDS System and Data Governance	User Management	Deactivate User	As a System Operator, I want to be able to Deactivate an existing user so that the relevant user cannot access the system any more	10.2.2 (last row)
FR118	Notifications	VDS system and Data Governance	Notifications for Validations Completed during Transition Period	An email must be sent to the training organisation (person who initiated the upload) when validations are complete for a particular upload. The message must be sent to the email address recorded against the user's access profile.	10.5.6.1 10.5.3.2
FR119	Notifications	VDS system and Data Governance	Notifications for Changes in Fees during Transition Period	The training organisation must be informed of all changes in student fees. (All fee changes for students attending their training organisation under Smart and Skilled). Note: The training organisation must be alerted to the fee change at the time the fee is known to have changed. (Not at payment authorisation time. This is due to the fact that fees can change when no payment is due)."	Clarification required

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 249 of 1295

OFFICIAL

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR120	Validation and Data Quality	Validations and checks	Master tables	VDS system should perform state specific & state's program specific validations on the submitted data using up-to-date STAs reference data available from the system of record i.e. the frequency of reference data updates to VDS should be in sync with the freuqency of updates on reference data in the source system ex: Near real time update for commitment id status and associated reference data elements from TSNSW Skills Groups id's created and maintained for TSNSW (such as part Quals) should be made available to VDS system (likely extension of TGA data)	10.5.3.2 10.5.1.2 10.5.4
FR121	Validation and Data Quality	Validations and checks	Data integrity adherence	The data type definition for each data element that is submitted as part of VET training activity data should support data values that is accepted for each data elements as defined in the system of record For ex. SMS / USI registry is the master for student data ? considering this any data type , value definition for student information should be in sync wtih SMS and USI registry student details should accept: Numerical value in a name , students are twins etc.. Allow a third valid input for gender "@" or X and map to Not Specified for training activity :	10.5.3.2

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 250 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
				future dates to be accepted, referet to TSNSW FR96k	
FR122	Validation and Data Quality	Validations and checks	Validation Results	In continuation to TSNSW FR13 and FR045, as a training organisation, I would like to also have access to the errors & warnings generated from the most recent VET activity data submission stored and accessible when I relogin or if the session times out so that I can have access to the history of my submission status and be able to perform error corrections on the screen for the data and be able to resubmit the specific student activity record . Applicable only until the next subsequent submission Note: FR050 and FR45 looks similar and needs to be reviewed	10.5.1.4

OFFICIAL

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR123	Validation and Data Quality	Validations and checks	Validation Results	The error/warning/info messages must be granular and quote the data/reference item which is under error in reference to notification requirements in FR056b, FRo56c Note: For S&S program specfic VET student activity data the following data elements should not be exposed on the screen: • Payment Stage • Reject Id • Reject Reason • Addendum: o Reject Reason o Course of Action Reject ID will be concatenated at the end of the text message	10.5.3
FR124	Validation and Data Quality	Validate Transition Qualification	TSNSW - S&S Program Specific Validations	Transition Qualifications: When a student's qualification on the student input record does not match the qualification held against the Commitment ID. The system must ensure: • The new qualification is a valid transition qualification for the old qualification • The new qualification is on the training organisation approved scope (as per validations for the original qualification) • All validation checks including those for core + elective UoCs must be performed using the new qualification Note: VDS Solution must validate the units reported for the new qualification under the CID for the old qualification, if the provider transitions the student. Some providers may decide not to transition, but to teach out instead. The decision will be up to the training	10.5.3.2

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 252 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
				organisation. Note: This FR has reference data dependency TSNSW FR053a (in this case: commitment id + other reference data against the commitment id where TSNSWs iVETS is the system of record)	
FR125	Validation and Data Quality	Post Validation Flag latest actiity data	TSNSW - S&S Program Specific Validations	Once a student training activity data record is successfully validated, VDS system should be able to flag the latest valid activity information received for a student for use by TSNSWs milestone payment reporting.	10.5.2.3 10.5.4 9.5.2.2.4
FR126	Data Consumption and Reporting	Post validation data melding	Data melding post validation	VDS system should have the capability to complete the activity data records with relevant reference data required as soon as the validations are complete, so STA can review the full data set to release data for NCVER reporting For ACE: Append the following reference data to NAT0010 Training organisation type identifier Address first line Address second line Address – suburb, locality or town Postcode State identifier	9.5.2.2.5 9.5.2.2.6 9.5.3.2.6 9.5.3.2.7 10.5.1.2

OFFICIAL

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR127	Data Consumption and Reporting	Data Consumption	BI Reporting on Student Details	The SMS or UI must be able to search for students saved/submitted in activity file submissions by the training organisation that submitted the data only. The information displayed reflects current most recently updated/submitted student details.	10.5.1.4
FR128	Validation and Data Quality	Validations and checks	Near Real - Time error Notification	As a training organisation, I should have near realtime capabilitiy to view Errors and Warnings online especially when sanity checks stated in FR52 has failed; once the sanity checks are compelted and if no errors, system should notify the user with the message - Thank You for submitting your training activity data, it may taken between X to Y minutes for the files to be processed. with an acknolwedgement button.	10.5.1.4
FR129	Validation and Data Quality	Validations and checks	error Corrections & Resubmission	As a training organisation, I should have provision to be able to either resubmit the entire data set in which one of the activity data has got errors or be able to perform error corrections online and resubmit error corrections on a subset of activity records that has got errors or able to correct all errors against all activity records in error online in VDS for further validation.	10.5.1.4 10.5.1.3
FR130	Validation and Data Quality	Validations and checks	Validate via 2 Standards	System should be able to store all the information (Data elements) in the VET training activity data irrespective of the standard followed during data submissions so that the data has the same meaning and that the data consumer (for ex. STA	10.5.1.3 10.5.1.4

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 254 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
				/ NCVER) can consume data in the VET information standard they are still operating in. Note* system should be able to accomodate multiple sets of validations at any given point. The system will record the standard data was validated in, so future changes can be understood.	
FR131	Data Consumption and Reporting	Data Consumption	Activity Data Upload Capability during Transition Period	TSNSW must receive a full set of AVETMISS 8.0 compliant Training Activity Files including all the existing Data Elements. • System should provide File outputs containing Activity Data per program types Generic File Names required for ACE & S&S: • RTraining Organisation information (NAT00010) • Delivery venue location (NAT00020) • Program information (NAT00030) • Subject information (NAT00060) • Student demographics (NAT00080) • Student postal details (NAT00085) • Student disability (NAT00090) • Student prior education (NAT00100) • Subject enrolment (NAT00120) • Program completions (NAT00130) Note* These Files are specific to contracted RTraining Organisation's Funded Data. For FFS data, further details to be added.	8.11 10.5.5 10.5.4.1

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR132	Data Submission	Submit Data to VDS	Activity Data Upload Capability during Transition Period	System should be able to configure limits of Activity Data upload file sizes to support the peformance needs for real time processing	10.5.4.1 10.5.3.1
FR133	Data Submission	Submit Data to VDS	Activity Data Upload Capability during Transition Period	While Activity Data Files are being uploaded or processed via UI, system will not allow relevant training organisation's to re-upload files	10.5.4.1
FR134	Validation and Data Quality	Submit Data to VDS	Activity Data with future dates	System will allow training organisations to submit Activity Data with future start and end dates. • System allows training organisations to amend start and end dates	10.5.4.1 10.5.3.2
FR135	Notifications	VDS system and Data Governance	Notifications for pending actions	The VDS system must be able to send notifications to training organisations when there are pending actions required. For example, the training organisation must receive a reminder “x” days after valid data has been received for an training organisation, and no subsequent valid files have been uploaded. This email does not detail individual students. The training organisation must receive a reminder “x” "x" depends on the Program Type and relevant contracts with training organisations	10.5.6.1
10.2.2 (last row)	Notifications	VDS system and Data Governance	Notifications	Training organisation must have the capability (perhaps via GUI interface) to view up-to-date business glossary to consume easily and to understand the meaning of the data elements associated wtih data packets in the new VET information standard	8.4 8.9.2 11.2.2.15

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 256 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR137	Validation and Data Quality	Validations and checks	Validation	VDS system should continue to perform validations in the background even if the user who has submitted the records for validation has logged out of the system. If validation of a record is still being processed, the training organisation should not be able to re-submit the record, until processing is complete. Once the validations are complete, system should make provision for user should to view the validation results on next login and for all subsequent logins until another data submission.	10.5.3.2 11.1.10
FR138	Validation and Data Quality	Validate the Data submitted by a training organisation against the STA/Funder validations	Validation	If a training organisation is contracted with multiple STAs, VDS system should be able to isolate funded activity data that is specific to my STA and should apply my STAs specific state validation rules only on the activity data that is required and relevant for funding and NCVER reporting for my STA.	9.5.2.2.2 10.5.3.2
FR139	Validation and Data Quality	Validate the Data submitted by a training organisation against the STA/Funder's program specific validations	Validations	Each funded VET student activity record that is reported by a contracted training organisation after passing the sanity checks should pass the national validations confirming that the VET student activity records are free of any errors ahead of VDS solution performing the program and state specific validation checks.	10.5.1.3 9.5.2.2

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 257 of 1295

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
FR140	Validation and Data Quality	Valdiation Stages	Validations	The VET student activity data records once submitted by the contracted training organisations should be accessible by the STA and NCVER with appropriate validation check status set against each activity record. STAs will have access to data it funds across entire data lifecycle.	8.4 10.5.1.3.2.6
FR141	VDS System and Data Governance	Access Control	Access Control	The VDS Solution must allow for validations functionality to be disabled (put on hold) for individual training organisations at any point in time with specific criteria (eg specified validation rules, time turned off, specific training organisations/program). This is to support special exemption needs on contracted training organisations. Temporarily disabling validations must be approved and the system should capture the associated metadata (date & time; approver; requester; justification; associated records, threshold changes, related document links).	10.5.3.1
FR142	Validation and Data Quality	Validate the Data submitted by a training organisation against the STA/Funder's program specific	TSNSW - Program Specific Validations	Validation rules must include program specific validation rules that are only applied to contracted training organisations funded activity data specific to that program.	10.5.3.1 10.5.3.2

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 258 of 1295

OFFICIAL

FR ID	Category	Process	Requirement name	Requirement description	Design Reference
		validations			

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 259 of 1295

Appendix H. Solution Software Components – Changed Scope

This appendix lists the changed scope from the VDS Work Order relating to the STARS solution components, including assumptions.

Please refer to the VDS Work Order for details of the original deliverables and assumptions. These are reproduced here for convenience.

The STARS solution scope, including solution features, has been identified within the Business Requirements and Solution Requirements as submitted under Work Package 1, documented in the STARS Solution Requirements Document.

This Solution Design covers STARS Release 1 only. At various points in this document, it has been noted that no requirements have yet been provided for certain features. It is expected that in many cases, missing requirements will be developed during the life of the project, which will result in additional design and development work. This will be subject to the agreed project scope change management process.

Note that the features, complexities and counts listed below relate to only those items explicitly described within this solution design, and do not include items yet to have requirements provided.

Software Component Assumptions

Assumptions for system components that involve software and VDS Master Database development and configuration have been made using metrics, complexity rating, and quantities based on the requirements within the work order and the Data Schema Assumptions described in the Work Order.

In this section:

1. metrics are based on the component type, e.g., a code module, a configuration file, or a Master Data Management coding, configuration and integration item;

1. a ‘code module’ is context dependent but relates to code that operates on a single entity. For example, a microservice code module is a function that is responsible for handling a request to store a student entity, along with any supporting functions. An integration test code module is a suite of functions that test, for example, a set of read/write/delete functions for a data entity, or a set of validation functions for that entity. A web UI code module is a web page and its underlying event handling code;

2. a code file might contain several modules, and a code module might extend over several code files and may not have a 1:1 correspondence;

3. configuration files are infrastructure-as-code template files (e.g., Azure Resource Manager templates) or Azure DevOps CI/CD pipeline template files;

4. Master Data Management (**MDM**) coding, configuration and integration items relate to coding and configuring the VDS Master Database for the data entities and integrating it with appropriate ETL pipelines and solution business rule engine(s), utilising standard Microsoft Azure-based data management technologies and tools. Each item is based around the entity that is being coded, configured, and integrated with the VDS Master Database.

5. The lines of code (LOC) metric refers to the count of effective or executable code lines and is exclusive of comments and white space. For convenience, if measurements are required, a simple 10% factor will be included as being the comment or white space line count, e.g., a code

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 260 of 1295

module that is 900 lines long total will be taken as having 90 lines of comments or white space, with an effective LOC of 810 lines.

6. The schema metric refers the count of the number of data tables within the database.

Metrics and Complexity Ratings

The complexity metrics and ratings are as follows:

Metric	Complexity Rating	Assumption
Code	1	The code module involves <= 100 Lines of Code (LOC)
	2	The code module LOC is in the range: 100 < LOC <= 500
	3	The code module LOC is in the range: 500 < LOC <= 1000. Note that seller will refactor any code files so that code modules don't exceed 1000 LOC – this is best practice and ensures future maintainability.
Config	1	The configuration file involves <= 100 Lines of Code (LOC); A pipeline configuration files contain <= 10 tasks; An infrastructure configuration template file contain <= 10 resources;
	2	The configuration file LOC is in the range 100 <= LOC < 200; A pipeline configuration file task count lies in the range: 10 < task count <= 20 tasks; An infrastructure configuration template file resource count is in the range: 10 <= resource count < 20;
	3	The configuration file LOC is in the range 200 <= LOC < 500; A pipeline configuration files task count lies in the range: 10 < task count < 20; An infrastructure configuration template file resource count is in the range: 20 <= resource count < 30; Note: configuration files that exceed these counts will be refactored to be smaller files – this is best practice and ensures future maintainability.
MDM	1	The Master Data Management coding, configuration and ETL integration involves a simple one entity/one table configuration.
	2	The Master Data Management coding, configuration and ETL integration involves up to between 2 and 10 entities/10 tables configuration.
	3	The Master Data Management coding, configuration and ETL integration involves between 11 and 20 entities/tables configuration. Note: MDM configurations that involve more than 20 entities will be refactored to smaller configurations – this is best practice and ensures future maintainability.

OFFICIAL

Metric	Complexity Rating	Assumption
Report	1	The report is based on 1 data table and one presentation form.
	2	The report data tables and presentation forms count lies in the range 2 <= table/form count < 5 presentation forms.
	3	The report is based 10 or less or more data tables and 10 or less presentation forms.
Schema	1	The tables within the schema contain an average of less than 10 attributes with simple 1:N relationships, no check constraints nor triggers.
	2	The tables within the schema contain an average between 10 and 20 attributes with simple 1:N relationships, no check constraint nor triggers.
	3	The tables within the schema contain <= 50 attributes, or have N:M relationships, check constraints or triggers.
Workflow	1	Workflows contain only one decision branch and up to 3 input/output connectors.
	2	Workflows contain up to 5 decision branches and up to 5 input/output connectors.
	3	Workflows contain <= 10 decision branches and <= 5 input/output connectors, or contain coded decision logic.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 262 of 1295

Software Component Assumptions Tables

The following table details the assumptions regarding the work packages and associated work streams, components, quantities involved and the complexity rating. These assumptions have been made based on the requirements within this work order.

These assumptions are in addition to assumptions for Data & Reporting and Data Migration.

All the code modules include unit test code (that is, code that tests only that module.) This is separate to integration tests that test multiple modules in combination – these are listed as separate items below.

The Config items include setting up required Azure resources for the system and setting up CI/CD build and deployment pipelines for the component.

The quantities refer to the number of, for example, code modules for the deliverable. For example, the Data Submission Microservice assumes 87 code modules (representing each service endpoint, responsible for read/write/soft delete). Each of these code modules has a complexity rating of '2', meaning that each of the 87 code modules LOC range is assumed to lie between 100 and 500 lines of code, i.e., there are up to 87 * 500 = 43,500 lines of code assumed for the Data Submission Microservice.

The notes describe the basis for the assumption – refer to the Data Schema Assumption section above for a list of assumed data entities that have informed this set of assumptions.

Work Order Solution Feature Scope (Original Scope)

Table 15: Work Order Components and Associated Assumptions (Original Solution Scope)

Key: R – Read, W – Write, SD – Soft Delete, MDM – Master Data Management

Work Package	Solution Feature	Type	Component	Quantity	Metric	Complexity Rating	Notes
PM 3 **Work Package 3 - Data Submission Stage 1 (Submission)**							
	Data Submission Services	Microservice	Microservice Endpoint	87	Code	2	Based on 29 entities, 3 endpoints per entity (R/W/SD)
			Integration Tests	29	Code	1	Based on 29 entities, tests across each endpoint

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

Work Package	Solution Feature	Type	Component	Quantity	Metric	Complexity Rating	Notes
			CI/CD Pipelines	4	Config	1	Based on 4 services for each of the logical entities
	Data Submission UI	Web Application	Web Page	10	Code	2	Single Page App, based on current NCVER portal, calling Data Submission Services
			Integration Tests	1	Code	1	Based on one application
			CI/CD Pipelines	1	Config	1	Based on one application
	Data Submission Rules Engine	Microservice	Microservice Endpoint	29	Code	2	Based on 29 entities, 1 endpoint per entity (initial validation rules processing)
			Integration Tests	29	Code	1	Based on 29 entities, test all rules
			CI/CD Pipelines	1	Config	1	Based on one validation service
	Master Data Management (Data Submission)	MDM	MDM Entity Coding and Configuration	29	MDM	1	Based on 29 entities, Master Database coding, config & integration for submitted data
	Integration Support Services SDK (Data Submission)	Code Library	API Functions - .NET	58	Code	1	Based on 29 entities, 2 endpoints per entity (R/W)
			API Functions - Java	58	Code	1	Based on 29 entities, 2 endpoints per entity (R/W)
			Integration Tests	29	Code	2	Based on 29 entities, integration tests for Java & .NET
PM4 **Work Package 4 - Data Submission Stage 2 (Data Validation)**							
	Data Validation Services	Microservice	Microservice Endpoint	29	Code	2	Based on 29 entities, 1 endpoint per entity (final validation request)
			Integration Tests	29	Code	1	Based on 29 entities
			CI/CD Pipelines	1	Config	1	Based on one validation service
	Data Validation UI	Web Application	Web Page	5	Code	2	Single Page App, calling Data Validation Services

Work Package	Solution Feature	Type	Component	Quantity	Metric	Complexity Rating	Notes
			Integration Tests	1	Code	1	Based on one application
			CI/CD Pipelines	1	Config	1	Based on one application
	Data Validation Rules Engine	Microservice	Microservice Endpoint	29	Code	2	Based on 29 entities, 1 endpoint per entity (final validation rules processing)
			Integration Tests	29	Code	1	Based on 29 entities, test all rules
			CI/CD Pipelines	1	Config	1	Based on one service
	Master Data Management (Data Validation)	MDM	MDM Entity Coding and Configuration	29	MDM	1	Based on 29 entities, Master Database coding, config & integration for validated data
	Integration Support Services SDK (Data Validation)	Code Library	SDK API - .NET	29	Code	2	Based on 29 entities, 1 endpoint per entity (R)
			SDK API - Java	29	Code	2	Based on 29 entities, 1 endpoint per entity (R)
			Integration Tests	29	Code	2	Based on 29 entities, integration tests for Java & .NET
PM5 **Work Package 5 – Data Release, Data Delivery**	**Data Release**						
	Data Release Services	Microservice	Microservice Endpoint	58	Code	2	Based on 29 entities, 2 endpoint per entity (r/w)
			Integration Tests	29	Code	1	Based on 29 entities
			CI/CD Pipelines	1	Config	1	Based on one service
	Data Release UI	Web Application	Web Page	5	Code	2	Single Page App, calling Data Release Services
			Integration Tests	1	Code	1	Based on one application
			CI/CD Pipelines	1	Config	1	Based on one application

Work Package	Solution Feature	Type	Component	Quantity	Metric	Complexity Rating	Notes
	Data Release Rules Engine	Microservice	Microservice Endpoint	29	Code	2	Based on 29 entities, 1 endpoint per entity (release rules processing)
			Integration Tests	29	Code	1	Based on 29 entities, test all rules
			CI/CD Pipelines	1	Config	1	Based on one service
	Master Data Management (Data Release)	MDM	MDM Entity Coding and Configuration	29	MDM	1	Based on 29 entities, Master Database coding, config & integration for released data
	Data Delivery						
	Data Delivery Services	Microservice	Microservice Endpoint	29	Code	2	Based on 29 entities, 1 endpoint per entity (R)
			Integration Tests	29	Code	1	Based on 29 entities
			CI/CD Pipelines	1	Config	1	Based on one service
	Data Delivery UI	Web Application	Web Page	5	Code	2	Single Page App, calling Data Delivery Services
			Integration Tests	1	Code	1	Based on one application
			CI/CD Pipelines	1	Config	1	Based on one application
	Data Delivery Rules Engine	Microservice	Microservice Endpoint	29	Code	2	Based on 29 entities, 1 endpoint per entity (delivery rules processing)
			Integration Tests	29	Code	1	Based on 29 entities, test all rules
			CI/CD Pipelines	1	Config	1	Based on one service
PM6 **Work Package 6 - Management Portals**							

Work Package	Solution Feature	Type	Component	Quantity	Metric	Complexity Rating	Notes
	System & Data Management Services	Microservice	Microservice Endpoint	10	Code	2	Based on 5 management services, 2 endpoints per service (R/W)
			Integration Tests	10	Code	1	Based on 5 management services
			CI/CD Pipelines	1	Config	1	Based on one service
	System & Data Management UI	Web Application	Web Page	10	Code	2	Single Page App, calling Management Services
			Integration Tests	1	Code	1	Based on one application
			CI/CD Pipelines	1	Config	1	Based on one application

STARS Solution Features (New Scope)

This section lists the main STARS Solution features, representing the changed scope of the VDS Project. Please refer to section 10 'Software Architecture and Design' for details of each feature. Table 16: STARS Solution Components/Features and Assumptions (New), below lists all the solution features, including if it forms part of the 'core' STARS platform (Platform Y/N). Note that features, complexity and quantities listed below have been determined from the STARS Solution Requirements.

Table 16: STARS Solution Components/Features and Assumptions (New)

Solution Design Feature	Component	Type	Metric	Complexity	Qty	Release	Notes
Master Data Service	Functions - API	Microservice	Code	2	20	1	1 per VET IS DATA PACKET entity + query
Master Data Service	Databases - Core	Database	Schema	2	50	1	VET-IS physical schema, 1 National
Master Data Service	Databases - STA	Database	Schema	1	10	3	VET-IS physical schema, 4 x STA, STA extensions
Master Data Service	Integration Tests	Test	Code	1	50	1	1 per VET IS entity, national DB only
Master Data Service	Integration Tests	Test	Code	1	10	3	STA-specific entity tests
Master Data Service	CI/CD Pipelines	CI/CD	Config	2	3	1	Azure Functions, DBs - build/test/deploy
Reference Data Service	Workflows - Core	Workflow	Code	2	10	1	Reference data events, time-based workflows

Solution Design Feature	Component	Type	Metric	Complexity	Qty	Release	Notes
Reference Data Service	Workflows - CARMS Replacement	Workflow	Code	1	20	4	Reference data events, time-based workflows
Reference Data Service	Functions - Workflows - CARMS Replacement	Microservice	Code	1	40	4	Reference workflow functions
Reference Data Service	Functions - Workflows - Core	Microservice	Code	1	10	1	Reference workflow functions
Reference Data Service	Functions - API	Microservice	Code	1	15	1	Reference functions
Reference Data Service	ETL Pipeline - NCVER	Pipeline	Code	3	1	1	NCVER CARMS integration
Reference Data Service	ETL Pipeline - Direct	Pipeline	Code	3	12	4	Direct to source integration (e.g., TGA)
Reference Data Service	ETL Pipeline - STA	Pipeline	Code	3	4	3	STA Reference integration - 1 per STA
Reference Data Service	Database - Core	Database	Schema	3	10	1	87 Reference data tables (from CARMS)
Reference Data Service	Integration Tests - Core	Test	Code	2	15	1	APIs & workflows
Reference Data Service	Integration Tests - STA	Test	Code	2	15	3	STA APIs & workflows
Reference Data Service	CI/CD Pipelines - Core	CI/CD	Config	2	3	1	APIs, workflows & functions
Reference Data Service	CI/CD Pipelines - STA	CI/CD	Config	2	2	3	STA APIs, workflows & functions
Provider Service	Workflows - Core	Workflow	Workflow	1	10	1	10 event-driven workflows
Provider Service	Functions - Workflows - Core	Microservice	Code	2	10	1	Workflow functions
Provider Service	Functions - API	Microservice	Code	2	25	1	API functions
Provider Service	Databases - Core	Database	Schema	1	20	1	Provider data and event databases (Cosmos)
Provider Service	Integration Tests	Test	Code	1	40	1	Test workflows, functions, DBs, infrastructure
Provider Service	CI/CD Pipelines	CI/CD	Config	2	5	1	Logic App workflows, Azure Functions, DBs, Azure infrastructure - build/test/deploy
Submission Service	Workflows - Core	Workflow	Workflow	1	5	1	5 event-driven workflows
Submission Service	Workflows - STA	Workflow	Workflow	1	8	3	2 custom submission per STA
Submission Service	Functions - Workflows - Core	Microservice	Code	3	5	1	Core workflow functions
Submission Service	Functions - Workflows - STA	Microservice	Code	3	8	3	Core workflow functions

Solution Design Feature	Component	Type	Metric	Complexity	Qty	Release	Notes
Submission Service	Functions - API	Microservice	Code	2	2	1	API functions
Submission Service	Queue	Configuration	Config	1	1	1	Submission queue
Submission Service	Rules - Core	Rule	Code	1	5	1	Core submission workflow rules, 1 per core workflow (NB: not validation)
Submission Service	Rules - STA	Rule	Code	1	8	3	STA submission workflow rules, 1 per custom STA workflow (NB: not validation)
Submission Service	Databases - Core	Database	Schema	1	1	1	Submission data and event databases (Cosmos)
Submission Service	Integration Tests - Core	Test	Code	1	10	1	APIs, rules, workflows & functions
Submission Service	Integration Tests - STA	Test	Code	1	16	3	APIs, rules, workflows & functions
Submission Service	CI/CD Pipelines - Core	CI/CD	Config	1	1	1	Logic App workflows, Azure Functions, DBs, Azure infrastructure - build/test/deploy
Submission Service	CI/CD Pipelines - STA	CI/CD	Config	1	4	3	Logic App workflows, Azure Functions, DBs, Azure infrastructure - build/test/deploy
Staging Service	Workflows - Core	Workflow	Workflow	1	5	1	5 event-driven workflows
Staging Service	Workflows - STA	Workflow	Workflow	1	8	3	2 per STA - custom workflows
Staging Service	Functions - Workflows - Core	Microservice	Code	1	5	1	Workflow functions - 5 core events
Staging Service	Functions - Workflows - STA	Microservice	Code	2	8	3	Workflow functions - STA -8 custom
Staging Service	Functions - API	Microservice	Code	3	2	1	API functions
Staging Service	Queue	Configuration	Config	1	1	1	Submission queue
Staging Service	Rules - Core	Rule	Code	1	10	1	Staging workflow rules, 1 per core workflow (NB: not validation)
Staging Service	Rules - STA	Rule	Code	1	16	3	STA staging workflow rules, 1 per custom STA workflow (NB: not validation)
Staging Service	Databases - Core	Database	Schema	1	20	1	Staging event database (Cosmos)
Staging Service	Integration Tests - Core	Test	Code	2	4	1	APIs, rules, workflows & functions
Staging Service	Integration Tests - STA	Test	Code	2	7	3	APIs, rules, workflows & functions

Solution Design Feature	Component	Type	Metric	Complexity	Qty	Release	Notes
Staging Service	CI/CD Pipelines - Core	CI/CD	Config	2	2	1	Logic App workflows, Azure Functions, DBs, Azure infrastructure - build/test/deploy
Staging Service	CI/CD Pipelines - STA	CI/CD	Config	2	4	3	Logic App workflows, Azure Functions, DBs, Azure infrastructure - build/test/deploy
Release Service	Workflows - Core	Workflow	Code	1	5	1	5 event-driven workflows
Release Service	Workflows - STA	Workflow	Code	1	8	3	2 per STA
Release Service	Functions - Workflows - Core	Microservice	Code	1	5	1	Workflow functions - 10 events
Release Service	Functions - Workflows - STA	Microservice	Code	1	8	3	Workflow functions - 10 events
Release Service	Functions - API	Microservice	Code	1	2	1	API functions
Release Service	Queue	Configuration	Config	1	1	1	Release queue
Release Service	Rules - Core	Rule	Code	1	5	1	Core submission workflow rules, 1 per core workflow (NB: not validation)
Release Service	Rules - STA	Rule	Code	1	8	3	STA release workflow rules, 1 per custom STA workflow (NB: not validation)
Release Service	Databases - Core	Database	Schema	1	10	1	Release event database (Cosmos)
Release Service	Integration Tests - Core	Test	Code	1	7	1	APIs, rules, workflows & functions
Release Service	Integration Tests - STA	Test	Code	1	8	3	APIs, rules, workflows & functions
Release Service	CI/CD Pipelines - Core	CI/CD	Config	1	1	1	Logic App workflows, Azure Functions, DBs, Azure infrastructure - build/test/deploy
Release Service	CI/CD Pipelines - STA	CI/CD	Config	1	1	3	Logic App workflows, Azure Functions, DBs, Azure infrastructure - build/test/deploy
Mastering Service	Workflows - Core	Workflow	Code	2	5	1	5 event-driven workflows
Mastering Service	Functions - Workflows - Core	Microservice	Code	2	5	1	Workflow functions - 10 events
Mastering Service	Functions - API	Microservice	Code	2	2	1	1 per master data model entity
Mastering Service	Functions - MDM	Microservice	Code	2	60	1	MDM configuration & coding
Mastering Service	Queue	Configuration	Config	1	1	1	Mastering queue

Solution Design Feature	Component	Type	Metric	Complexity	Qty	Release	Notes
Mastering Service	Rules - Core	Rule	Code	1	10	1	Core mastering workflow rules, 1 per core workflow (NB: not validation)
Mastering Service	Databases - Core	Database	Schema	1	20	1	Mastering event database (Cosmos)
Mastering Service	Integration Tests	Test	Code	1	67	1	APIs, rules, workflows & functions
Mastering Service	CI/CD Pipelines	CI/CD	Config	2	6	1	Logic App workflows, Azure Functions, DBs, Azure infrastructure - build/test/deploy
R & A Provisioning Service	Functions - API	Microservice	Code	1	1	1	API functions
R & A Provisioning Service	Queue	Configuration	Config	1	1	1	R & A Provisioning queue
R & A Provisioning Service	ETL Pipeline	Pipeline	Code	1	2	1	1 per target reporting DB, 10 target DBs
R & A Provisioning Service	Databases - Core	Database	Schema	1	2	1	R & A Provisioning event database (Cosmos)
R & A Provisioning Service	Integration Tests	Test	Code	1	4	1	APIs, rules, workflows & functions
R & A Provisioning Service	CI/CD Pipelines	CI/CD	Config	1	2	1	Logic App/ADF workflows, Azure Functions, DBs, Azure infrastructure - build/test/deploy
Validation Service	Workflows - Core	Workflow	Workflow	1	5	1	10 event-driven workflows
Validation Service	Workflows - STA	Workflow	Workflow	1	8	3	4 per STA
Validation Service	Functions - Workflows - Core	Microservice	Code	2	5	1	different functions for funder variations to rules
Validation Service	Functions - Workflows - STA	Microservice	Code	2	8	3	get/search/post validation results + update status
Validation Service	Functions - API	Microservice	Code	2	2	1	1 per master data model entity
Validation Service	Rules - Core	Rule	Code	1	10	1	Core submission workflow rules, 1 per core workflow (NB: process oriented, not validation)
Validation Service	Rules - National	Rule	Rule	1	20	1	Initial submission validation rules
Validation Service	Rules - National	Rule	Rule	1	280	2	National submission validation rules
Validation Service	Rules - STA	Rule	Rule	1	80	3	STA submission validation rules, 20 per STA workflow
Validation Service	Integration Tests - Core	Test	Code	1	10	1	APIs & workflows

Solution Design Feature	Component	Type	Metric	Complexity	Qty	Release	Notes
Validation Service	Integration Tests - STA	Test	Code	1	16	3	APIs & workflows
Validation Service	CI/CD Pipelines - Core	CI/CD	Config	1	1	1	APIs, workflows & functions
Validation Service	CI/CD Pipelines - STA	CI/CD	Config	1	4	3	APIs, workflows & functions
Rules Service	Rules Engine	Microservice	Code	3	8	1	Execute/View/search/config/enable/disable/query rules + exemptions
Rules Service	Databases - Core	Database	Schema	1	20	1	Rules database (Cosmos)
Rules Service	Integration Tests	Test	Code	2	2	1	1 service + 1 DB
Rules Service	CI/CD Pipelines	CI/CD	Config	1	1	1	1 service + 1 DB
STA Bulk Submission Services	Functions - API	Microservice	Code	3	3	1	Bulk data submission, query, progress
STA Bulk Submission Services	Integration Tests	Test	Code	1	3	1	1 per data packet
STA Bulk Submission Services	CI/CD Pipelines	CI/CD	Config	2	1	1	1 service
Provider Portal	Portal	Web Page	Code	3	45	1	Registration, upload/entry per packet, nil returns, submission history, validation results, exemptions
Provider Portal	Integration Tests	Test	Code	1	1	1	1 portal
Provider Portal	CI/CD Pipelines	CI/CD	Config	1	1	1	1 portal
STA Gateway Service	Functions - API	Microservice	Code	2	15	3	1 per data packet + Submission query
STA Gateway Service	Integration Tests	Test	Code	1	15	3	1 per function
STA Gateway Service	CI/CD Pipelines	CI/CD	Config	1	1	3	1 service
Rules Service Management Portal	Portal	Web Page	Code	2	30	1	List/Configure/Export/Copy/Exemptions
Rules Service Management Portal	Integration Tests	Test	Code	1	1	1	1 portal
Rules Service Management Portal	CI/CD Pipelines	Web Page	Config	1	1	1	1 portal
Data Governance Infrastructure	Data Governance	Configuration	Config	2	20	1	Purview config
Data Governance Infrastructure	CI/CD Pipelines	CI/CD	Config	1	1	1	Purview environments
Data Quality Framework	Portal	Web Page	Report	1	15	1	Web delivered data quality report dashboards (Power BI)

Solution Design Feature	Component	Type	Metric	Complexity	Qty	Release	Notes
Data Quality Framework	Integration Tests	Test	Code	1	1	1	1 Portal
Data Quality Framework	CI/CD Pipelines	CI/CD	Config	1	1	1	1 Portal
SMS SDK Service	SDK API - .Net	Code Library	Code	2	20	1	Entity and event orchestration for submission & validation .Net
SMS SDK Service	SDK API - Java	Code Library	Code	2	20	1	Entity and event orchestration for submission & validation Java
SMS SDK Service	Integration Tests	Code Library	Code	1	40	1	SDK operations tests
Release Service Management Portal	Portal	Web Page	Code	2	20	3	View/approve/reject submissions/packets + Configure auto release
Release Service Management Portal	Integration Tests	Test	Code	1	1	3	1 portal
Release Service Management Portal	CI/CD Pipelines	CI/CD	Config	1	1	3	1 portal
Downstream Consumer ETL Pipeline Services	ETL Pipeline	Pipeline	Code	2	28	2	2 per AVETMISS 8.0 NAT file + 1 per data packet + 1 per VET-IS entity
Downstream Consumer ETL Pipeline Services	Integration Tests	Test	Code	3	4	2	1 per ETL pipeline
Downstream Consumer ETL Pipeline Services	CI/CD Pipelines	CI/CD	Config	1	4	2	Pipelines
STA Bulk Data Consumption Service	Data Share - Core	Configuration	Config	2	3	3	Cosmos, Azure SQL, Synapse
STA Bulk Data Consumption Service	Data Share - STA	Configuration	Config	3	4	3	1 per STA
STA API Service	Functions - API	Microservice	Code	2	15	3	1 per data packet + Submission query
STA API Service	Integration Tests	Test	Code	1	15	3	1 per function
STA API Service	CI/CD Pipelines	CI/CD	Config	1	1	3	1 service
USI Transcript Service	Functions - API	Microservice	Code	3	4	4	generate USI transcripts & student snapshot data, 12 attributes

Solution Design Feature	Component	Type	Metric	Complexity	Qty	Release	Notes
USI Transcript Service	Integration Tests	Test	Code	2	4	4	2 endpoints
USI Transcript Service	CI/CD Pipelines	CI/CD	Config	1	1	4	1 service
Event, Logging & Audit Services	Configuration	CI/CD	Config	1	20	1	Event, logging & auditing operations
Event, Logging & Audit Services	Integration Tests	Test	Code	1	20	1	per operation
Event, Logging & Audit Services	CI/CD Pipelines	CI/CD	Config	1	2	1	per service
Notification Service	Workflows - Core	Workflow	Workflow	1	5	1	Time-based and Event-based notifications
Notification Service	Functions - Workflows - Core	Microservice	Code	2	5	1	specific notification processing logic
Notification Service	Functions - API	Microservice	Code	1	10	1	Manage notification configuration & templates
Notification Service	Integration Tests	Test	Code	1	25	1	APIs & workflows
Notification Service	CI/CD Pipelines	CI/CD	Config	1	2	1	2 functions + workflows
Reference Data Service Management Portal	Portal	Web Page	Code	1	70	2	View curated reference data, Curate reference data for STAs
Reference Data Service Management Portal	Portal	Web Page	Code	2	15	4	Full curation of all refeference data
Reference Data Service Management Portal	Integration Tests	Test	Code	1	1	2	1 portal
Reference Data Service Management Portal	CI/CD Pipelines	CI/CD	Config	1	1	2	1 portal
Notification Service Management Portal	Portal	Web Page	Code	3	10	2	Manage notification configuration & templates
Notification Service Management Portal	Integration Tests	Test	Code	2	1	2	1 portal
Notification Service Management Portal	CI/CD Pipelines	CI/CD	Config	1	1	2	1 portal
Master Data Service Management Portal	Portal	Web Page	Code	3	10	1	Manage master data

Solution Design Feature	Component	Type	Metric	Complexity	Qty	Release	Notes
Master Data Service Management Portal	Integration Tests	Test	Code	2	1	1	1 portal
Master Data Service Management Portal	CI/CD Pipelines	CI/CD	Config	1	1	1	1 portal
Submission Management Portal	Portal	Web Page	Code	3	10	1	View data submission pipeline progress & events
Submission Management Portal	Integration Tests	Test	Code	2	1	1	1 portal
Submission Management Portal	CI/CD Pipelines	CI/CD	Config	1	1	1	1 portal
Downstream Consumer Reporting and Analytics	R & A	Web Page	Report	2	12	3	Web delivered data report dashboards (Power BI)
Downstream Consumer Reporting and Analytics	Data Share - Core	CI/CD	Config	2	5	3	Non-STA data share

Australian Government
Department of Employment and Workplace Relations

Detailed Design Document

Data Governance Infrastructure

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 276 of 1295

1 Document Management

1.1 Document Control

Version	Change Description	Date
0.1	Initial draft version	10/02/2023

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This is a living document and will be updated as required.

1.2 Document Author

Role	Name
Data Governance Lead	s 47F(1)

1.3 Document Reviewers

Role	Name

1.4 Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

1.5 Related Documents

Reference	Document
A	STARS Solution Requirements Document v1.3.docx
B	STARS Solution Design v1.0.docx
C	Governance Framework for Vocational Educational and Training Student and Training Activity Data – Draft - 2023
D	VET Data Access and Privacy Protocol – Draft - 2023

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 277 of 1295

Contents

1 Document Management 2

1.1 Document Control 2

1.2 Document Author 2

1.3 Document Reviewers 2

1.4 Document Approval 2

1.5 Related Documents 2

Contents 3

2 Introduction 4

2.1 Purpose 4

2.2 Intended Audience 5

3 Scope and Exclusions 6

3.1 In Scope 6

3.2 Out of Scope 6

3.3 Assumptions 6

4 Design Decisions 7

5 Framework Detail 8

5.1 Roles, Personas and Responsibilities 8

5.2 Data Governance Roles – Descriptions 8

5.3 Data Governance Structure (Personas) 9

1.1 Data Governance Structure 10

5.4 Data Security Classification 10

5.5 Control and Access 12

5.6 Security 15

5.7 Data Literacy Programme 16

5.8 Organisational Change Management 17

6 Data Governance Infrastructure – Microsoft Purview 18

7 Requirements 20

7.1 Functional 20

7.2 Non-Functional 22

Appendix - Acronyms 23

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 278 of 1295

2 Introduction

The Data Governance Infrastructure detailed design document describes how Data Governance Infrastructure will be managed to ensure an appropriate level of Data Governance for all data related to STARS.

Note: This is a living document that will be updated as designs for various other STARS services are developed and evolve. Accordingly, this document will evolve to reflect and document notification components as the STARS solution build progresses.

2.1 Purpose

The purpose of the Data Governance Infrastructure Detailed Design document is to ensure that:

1. Proper and functional Data Governance is enabled through a Governance Framework and;
2. Check and confirm that the proposed framework is not lacking and has all bases covered for delivering a holistic Data Governance program.

This document is intended to cover the conceptual aspect of Data Governance, rather than the physical technical aspect (tooling) and outlines the People and Process aspects of Data Governance.

The physical and technology aspect of Data Governance will be covered in a separate document – as DXC awaits the confirmation of the following documents from DEWR/NCVER:

1. Data Management Framework
2. Metadata Management

A Governance Framework is conventionally enabled through the three-thronged approach of:

1. People
2. Process
3. Technology

SYNERGY through a People, Process and Technology approach

This approach guarantees synergy through a holistic approach ensuring all aspects of Data Governance are covered through People, Process and Technology tools.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 279 of 1295

2.2 Intended Audience

The intended audience for STARS is all those that interact with the STARS Reporting System. All these Persona interactions are deemed as an intended audience.

Thus, the primary audience for this document includes:

- The Policy and Compliance Teams – who are very close to creating Data Governance Police and Frameworks
- The DEWR and NCVER Enterprise Architects, Data Architects and Solutions Architects
- All those Personas who are part of the Data Governance Council(s)
- The Executive Team – who need to agree to and endorse this policy
- Any Team that is responsible for creating the Data Management Policies as these policies are closely tied

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 280 of 1295

3 Scope and Exclusions

3.1 In Scope

The following are in-scope of this detailed design solution:

1. Installation of the Data Intelligence Tool:
    - MS Purview
2. Ensuring that the Data Governance and Data Management Frameworks are complete and cover all aspects of data – and that there are no missing gaps which can leave data vulnerable.

Also:

The MS Purview tool – once installed – will have the following aspects of Data Governance covered:

- Data Map (Business Glossary)
- Data Catalogue/Data Lineage
- Data Estate Insights
- Metadata reports (Assets, Classifications, Sensitivity Labels)

3.2 Out of Scope

The following are out of scope:

Creating Policy documents and maintenance of these documents:

Some but not all are listed below:

- Data Governance Policy, Data Governance Framework
- Data Management Policy, Data management Framework
- Metadata Management Documentation
- Data Literacy Programme – to onboard Data Governance Framework Personas
- Data Governance Council(s) Framework Establishment
- Data Governance/Data Persona establishment (Champion, Adept, Citizen).

3.3 Assumptions

The assumptions are:

1. The **Access and Privacy Protocol** document is final, and no more changes are made.
2. The **Data Governance Framework** document is final, and no more changes are made.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 281 of 1295

4 Design Decisions

The intention of this document is to cover the people and process aspect of Data Governance.

The technology aspect of Data Governance will be covered in a separate document which will be based on the **Metadata Management document** when completed by DEWR.

As at 10 Feb 2023, DXC are awaiting completion of that document in the very near future.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 282 of 1295

5 Framework Detail

The Data Governance Framework – as per the STARS Reporting system – ensures the accuracy, security, and relevancy of data for an Organisation and all Customers/Members concerned.

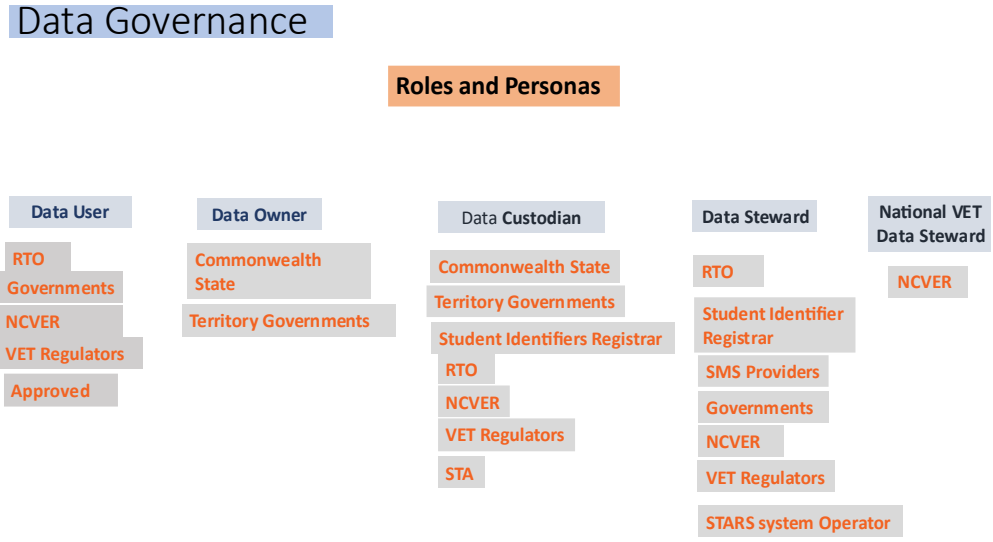
The content that follows has been taken from the following two documents.

1. Access and Privacy Protocol
2. VDS Governance Framework

The DEWR Policy Team have confirmed that these two documents (as at 10 Feb 2023 currently in Draft mode) will have no more changes.

5.1 Roles, Personas and Responsibilities

Data Governance involves several people-based activities to be effective and help enforce the policies, standards and procedures as well as provide oversight to guidelines. This requires several roles, described below, which can be carried out by one or more people.



5.2 Data Governance Roles – Descriptions

National VET Data Custodian

NCVER is the National VET Data Custodian responsible for collecting, managing, analysing and communicating research and statistics on the Australian VET sector. NCVER is contracted by the Commonwealth Government to perform its core duties for all governments and for the sector more broadly.

NCVER is responsible for maintaining the VET Information Standard and has responsibility for the day-to-day management of the VET data it holds. NCVER has subject matter experts who understand the meaning and use of the data. It is responsible for working with other Data Custodians to ensure that the protection, use and quality of the data is in line with the decisions and policies agreed by the Data Owners. It is in the National VET Data Custodian's interest to ensure that this framework is

being followed and that VET data is of high quality and fit for purpose, including for regulatory purposes, the publication of national statistics and to support research and analysis.

Data Steward

Data Stewards are responsible for the technical environment that support the transmission of VET data. Data Stewards do not have effective control over the data. They cannot access, use, delete or change VET data unless expressly instructed to do so. They must take all reasonable steps to protect data from misuse, interference and loss, and from unauthorised access, modification and disclosure. They support the effective and efficient flow of VET data in line with established business rules of Data Owners and directions of Data Custodians. *Generally, within the VET data environment, Data Stewards may be RTOs, the Student Identifiers Registrar, SMS providers, governments, NCVER, VET regulators, the STARS system operator, and any third-party IT vendors.*

National VET Data Steward

The National VET Data Steward is responsible for the technical environment supporting the STARS ICT system, including the central data repository. The National VET Data Steward maintains data integrity and security and support the effective and efficient flow of VET data, in line with established business rules. Following handover from the Commonwealth, *NCVER will perform the role of National VET Data Steward in conjunction with third-party ICT vendors.*

Data User

A Data User is an entity that receives data outputs in accordance with the NVETR Act and when they meet the conditions of use and disclosure which have been provided by Data Owners (and implemented by Data Custodians). Data Users are responsible for the appropriate storage and use of data they receive. *Generally, within the VET data environment, RTOs, governments, NCVER, VET regulators, and approved research bodies are Data Users.*

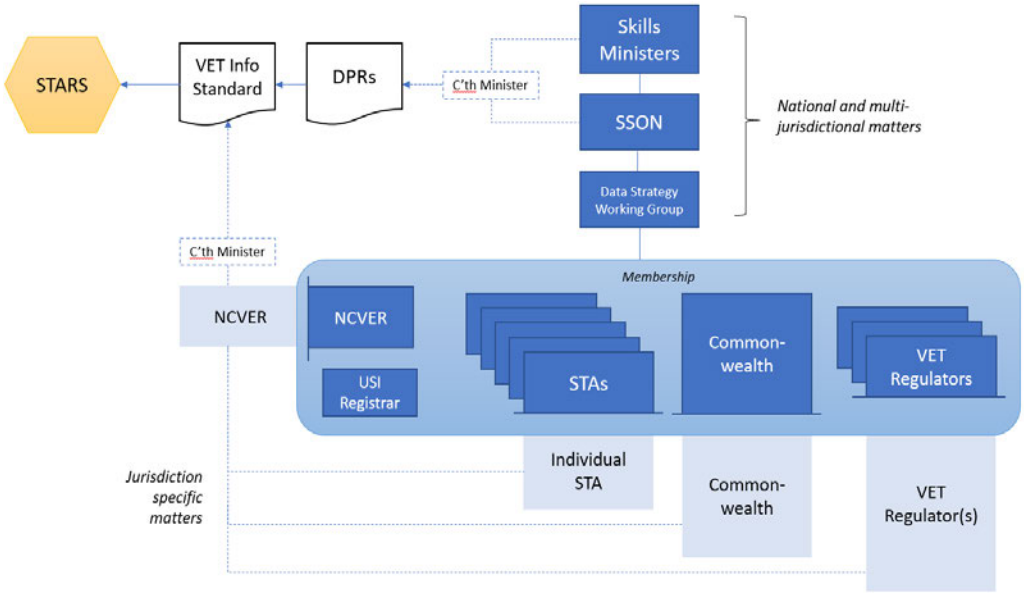
5.3 Data Governance Structure (Personas)

The proposed Data Governance structure (Personas) – as proposed:

- Skills Ministers have ultimate responsibility for VET data, access and use.

However, *proposed* amendments to the NVETR Act will permit the Ministerial Council to agree to Skills Senior Officials or other bodies having some responsibilities for agreeing the Data Provision Requirements. Given the Data Provision Requirements are a Commonwealth instrument, the commonwealth minister, or a delegate thereof, will always *be required to make any* amendments.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 284 of 1295



Data Governance Structure (As Proposed)

5.4 Data Security Classification

This protocol, as per the DEWR Data Governance Framework, distinguishes between *identifiable data, de-identified data and confidentialised data*. This ensures that access to data balances the needs of data users while protecting the privacy of individuals and the commercially sensitive information of RTOs.

Identified data - data which relates to personal an identified individual or organisation, such as an RTO.

De-identified or Identifiable data - data which, in some circumstances, could enable the identification of an individual or organisation.

Confidentialised data - data provided in a manner that is unlikely to enable the identification of an individual or organisation.

All training providers that collect student formation need to ensure they comply with all relevant Commonwealth and state privacy legislation. On submission of VET data, all training providers must

provide a declaration that acknowledges their compliance with relevant privacy legislation and policies. Where possible, this process will be undertaken through electronic means.

NCVER *may, in accordance with section 210A of the NVETR Act, disclose identified student data to the Commonwealth and to state and territory authorities* (including regulators) that deal with matters relating to VET, for the purposes of that body, or where the disclosure is otherwise permitted under the *Privacy Act 1988* (the Privacy Act).

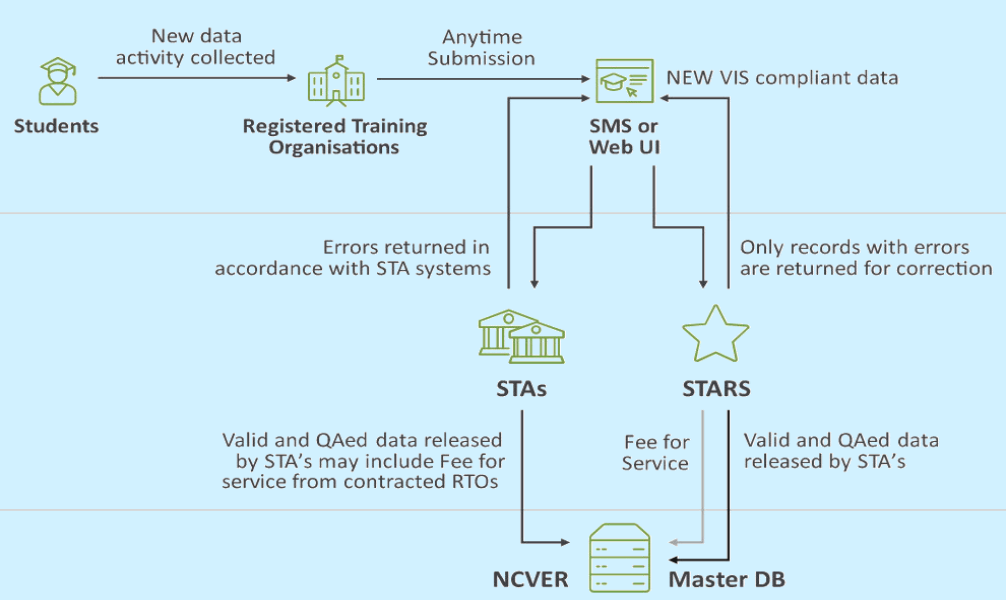
In STARS:

Access will be codified in STARS using role-based permissions, supporting the retrieval of relevant data at any time. Access permissions will be allocated on an individual basis, with individuals each supplied a unique log-in and password.

STARS will have two primary data partitions:

- Data that has been submitted through STARS by an RTO and is awaiting quality assurance and compliance checks by a jurisdiction. Data held in this temporary holding place is known as staging data or data in the staging database. (Note: data that requires quality assurance does not have to be submitted through STARS - it can alternatively be provided to a jurisdiction via other VET data systems).
- Data that has either been validated by jurisdiction, or does not require validation by jurisdiction. This is known as mastered data or data in the master database.

Figure below shows how VET data will flow from RTOs into STARS (including NCVER master database)



Government subsidised training data in STARS staging would only be available to the funding jurisdiction and would be controlled by the jurisdiction for the purposes of quality assurance and to support payment approvals to training providers.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 286 of 1295

5.5 Control and Access

Data Control

The two core central roles for accessing data from STARS are that of the Data Owner and the Data Custodian, summarised below:

Data Owner

A Data Owner has authority for establishing rules surrounding the quality, protection and appropriate access to data. They individually and collectively have a responsibility to protect the personal and sensitive information provided by Data Providers. Data can be jointly or co-owned.

Data Custodian

Data Custodians receive authorised data from Data Providers or from other Data Custodians. Data Custodians are authorised to collect and handle that data to perform their functions. They are responsible for ensuring that the personal information collected and stored is reasonably necessary for them to conduct their functions and activities in relation to the VET sector.

VET data stored in STARS will be controlled based on the data's characteristics, namely:

- Whether the training was government funded or delivered on a fee for service basis
- Whether the data is part of the National Minimum Data Set (NMDS) or is a jurisdiction specific data item
- Whether the data is in the STARS staging database or the STARS master database.

Fee for service data

Fee for service data contained in STARS (reflecting the NMDS) will be owned by VET regulators. Once in the master database, NCVER will be the sole data custodian.

Government funded data

Government funded training data in the STARS staging database will be controlled by the responsible jurisdiction for quality assurance and to support payment approvals to training providers.

- Government funding bodies will have responsibility for the quality of data provided by training providers and will control the release of any data from STARS staging to the STARS master database, following quality assurance against program/contract reference data held by the funding body.

  The national VET data set (which excludes jurisdiction specific data items) will be available from the STARS master database, under the control of the National VET Data Custodian. The NVETR Act, the SI Act and the SI regulation collectively set out a range of authorised users and their authorised uses of VET data. NCVER has a key role in collecting VET data on behalf of governments and disclosing the data to authorised users for them to fulfil their legislated functions and purposes. Such purposes include regulation, administrating the VET sector, consumer information, market and policy analysis, and generating public reports and student transcripts.

- The table below sets out the entities that will fill key roles for VET data contained in STARS databases.

STARS Data Ownership table

Data Type	Staging Data		Mastered Data	
	Owner	Custodian(s)	Owner	Custodian(s)
Fee for service provider data (Direct)	Australian governments	NCVER	Australian governments	NCVER
Fee for service provider data[1] (via STA)	Australian governments	Submitting jurisdiction & NCVER	Australian governments	NCVER
Government funded training – NMDS elements (Direct)	Australian governments	Responsible jurisdiction & NCVER	Australian governments	NCVER
Government funded training – NMDS elements (via STA)	Australian governments	Submitting jurisdiction & NCVER	Australian governments	NCVER
Government funded training - non-NMDS elements (Direct)	Responsible jurisdiction	Responsible jurisdiction & NCVER	Responsible jurisdiction	Responsible jurisdiction & NCVER

- *Shaded cells indicate data may never exist in the relevant database.*
- All other entities are considered users of that data until such time as they extract the data out of STARS. If an entity extracts data out of STARS, they immediately take on the role of data custodian for that data.

Data Access Levels

- The table below sets out the levels of access each jurisdiction and VET regulators may be granted by NCVER to data stored in STARS. As National VET Data Custodian, NCVER will have access to all data in STARS.
- STAs may be granted access to unpublished identified student and training activity data in the STARS master database for training undertaken nationally (NMDS only), including for training delivered on a fee for service basis. The Commonwealth may also be granted access to data from the master database for purposes related to the

[1] Includes fee for service training delivered by TAFEs or other government bodies.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 288 of 1295

administration of the VET sector and/or the collection and preparation of statistics relating to VET.

- Unique Student Identifiers (USIs) will be accessible for all VET students as authorised under the Student Identifiers Act.

The table below sets out the access arrangements for VET data contained in STARS.

STARS Data Access Arrangements*

Data location - type	Jurisdictional scope of access	Regulator Access
Staging database - Government funded training (NMDS)	Identified unit record data (own jurisdiction)	Identified unit record data
Staging database - Government funded training (non-NMDS)	Identified unit record data (own jurisdiction)	Nil
Master DB – Government funded training	Identified unit record data (NMDS for other jurisdictions)	Identified unit record data (NMDS only)
Master DB – Fee for service training	Identified unit record data (NMDS for other jurisdictions)	Identified unit record data (NMDS only)

**Access to data is limited to a range of established enabling purposes set out in both the NVETR Act 2011 and the Student Identifiers Regulation 2014, as well as to restrictions set out in state and territory privacy legislation.*

Data Custodians are responsible for ensuring that this requirement is met. This will also be agreed by users as an ongoing requirement of STARS access. Privacy Notices issued to students by RTOs must acknowledge all identified and identifiable data sharing arrangements in the above table.
Parties will not exercise any rights under a contract with ICT vendors to access data in a manner that is inconsistent with the access arrangements set out in this protocol.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 289 of 1295

5.6 Security

STARS Security Protocols

STARS will be built to meet existing Australian Government policies, frameworks, standards and design principles, including:

- Australian Government API Design Standards
- Digital Service Standard
- Protective Security Policy Framework (PSPF)
- Information Security Registered Assessors Program (IRAP)
- Australian Government Information Security Manual (ISM)
- the Open Web Application Security Project® (OWASP)
- Hosting Certification Framework.

All data within STARS will be handled in accordance with relevant privacy laws and record keeping requirements, including the Privacy Act, the *Archives Act 1983* (Cth) and any other applicable legislation (e.g. applicable State and Territory privacy legislation or archiving requirements).
For each STARS major release, the contracted system operator must:

- provide a security compliance statement
- confirm all services are being delivered in accordance with the relevant requirements of the PSPF and ISM
- complete relevant security documentation for each major release and then annually thereafter, including:
    - Security Risk Management Plan (SRMP),
    - System Security Plan (SSP),
    - Incident Response Plan (IRP) and
    - Standard Operating Procedures (SOP).

STARS will have a continuous security monitoring plan which will necessarily include event logging and auditing policies, processes and procedures.
Contractual arrangements with the STARS systems operator(s), as well as any third-party vendors, must impose appropriate privacy and security obligations to ensure the protection of VET data.

Data Breaches

A personal information data breach occurs when personal information, including that of a student or RTO, is lost or subjected to unauthorised access, modification, use or disclosure, or other misuse.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 290 of 1295

Data breaches can be caused by or exacerbated by a variety of factors, such as human error, malicious or criminal attack, or system fault etc. They can affect different types of personal information and give rise to a range of actual or potential harms to affected individuals.

Notification of certain data breaches involving personal information to the individuals concerned and to the Office of the Australian Information Commissioner (OAIC) is a legal requirement under the Privacy Act.

Data Breach Principles

1. All data breaches are dealt with as a serious matter, and, where relevant, in accordance with the OAIC's Notifiable Data Breaches Scheme (NDB Scheme).
2. All parties to this framework will have a data breach policy in place that covers VET data collected in accordance with the Data Provision Requirements and sets out:
    a. Internal and external reporting pathways; and
    b. Steps required to be taken in responding to an actual or suspected data breach.
3. Each policy must apply to all VET data breaches, or suspected data breaches.
4. Each policy must require notification to the National VET Data Custodian, NCVER. Notification can be made via privacy@ncver.edu.au.

5.7 Data Literacy Programme

Data Literacy is critical in on-boarding Data Personas into the Data Governance Program, as it helps educate the Data Personas and simplifies the transition into a Data Governance Program.

Without a Data Literacy Program – transitioning into a Data Governance Framework is virtually impossible – as none of the on-boarded personas of Data Owners, Data Stewards and Data Custodians will be unable to perform their tasks in their specific roles.

And last but not least, the Personas that can access the Gold, Silver and Bronze layers of data – do not have a training program – allowing them to safely access these layers – without breaching compliance.

Note: As of at 10 Feb 2023 (first draft) DXC has not received a Data Literacy Program or Framework document at DEWR.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 291 of 1295

5.8 Organisational Change Management

Organisational Change Management is critical to implement changes in an Organisation – to minimize impact on People and how it affects their way of working and make a seamlessly transition from current to future state.

This is all part of the People – Process and Technology triage, as it related to people's behaviour and how it can be changed to fit into the new framework in a seamless manner.

Change will only happen only if people behave differently – It has to be a cultural shift through a shift in people's behaviour – and it must be seamless.

Note: This needs to be discussed with DEWR Team and Org. change management team. A DevOps Item has been raised and assigned for DXC data team to follow-up (DevOps #1943).

6 Data Governance Infrastructure – Microsoft Purview

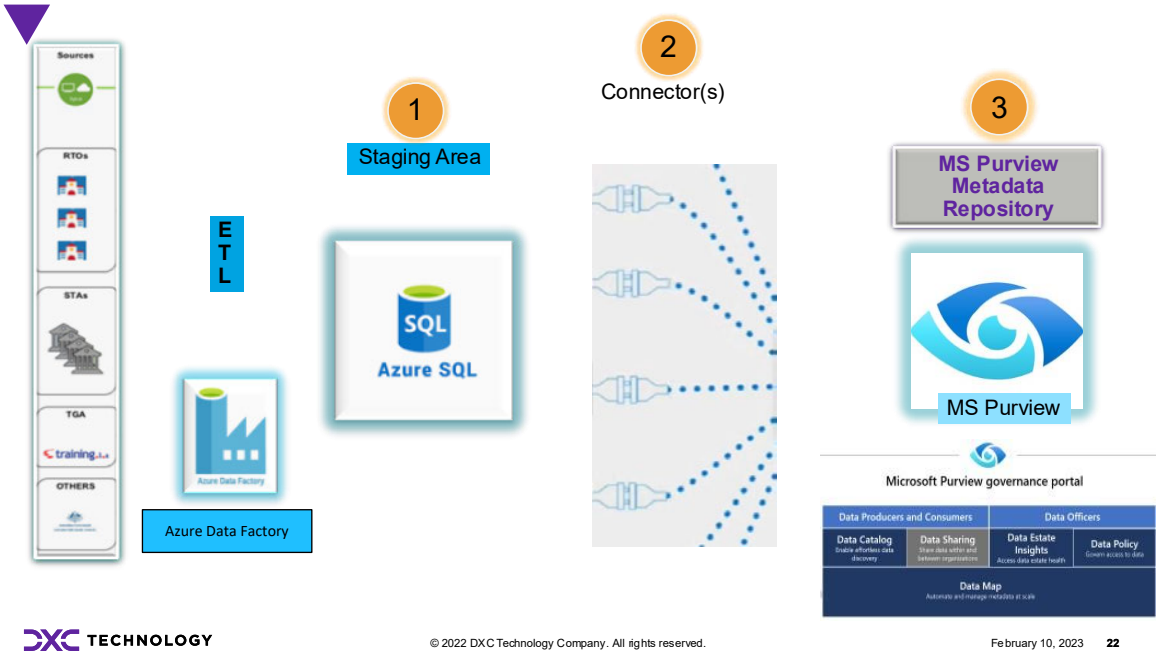
MS Purview will help consolidate all metadata from all data sources.

1 –The Metadata 'crawlers' from purview – will initially connect to the Azure Data Factory Staging Area through MS Purview Connector.

2 – MS Purview connector(s) – ingest all the metadata from the Azure Data Factory Data Models

3 – MS Purview performs its own A1/ML on the metadata and creating a

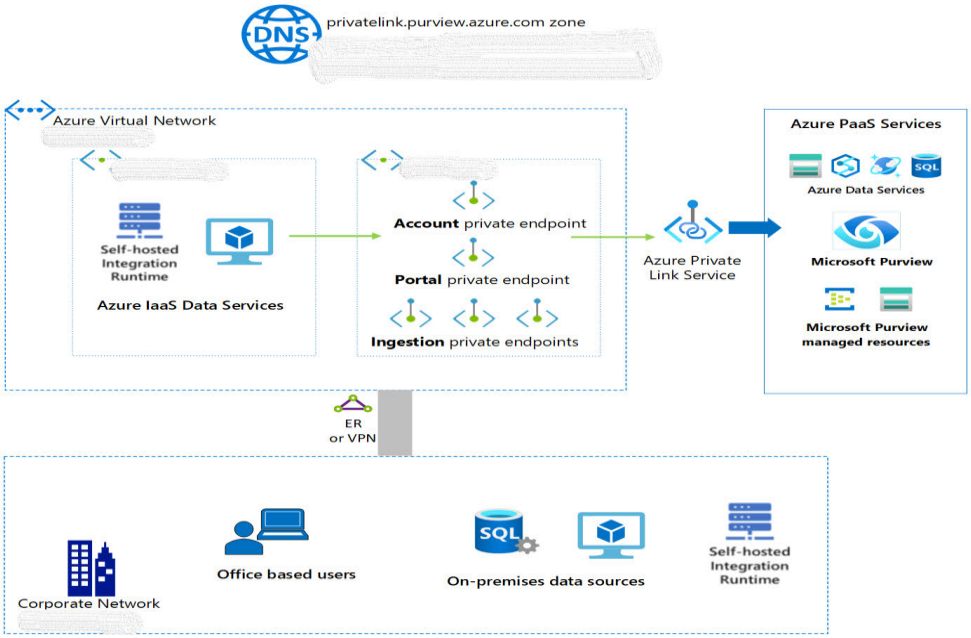
- Data Catalogue – which is a business Glossary and a Data Lineage Map – ensuring all entities have a proper business definition and can be traced back from Consumption to Capture stages
- Data Estate Insights – which gives an overview of the assets that have been scanned into the data map



Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 293 of 1295

Connecting to MS Purview

Access to Microsoft Purview account will be restricted via a private endpoint, including access to the Microsoft Purview governance portal, Atlas APIs and scan data sources in on-premises and Azure behind a virtual network using self-hosted integration runtime ensuring end to end network isolation.



MS Purview Connectivity through endpoints

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 294 of 1295

7 Requirements

7.1 Functional

FR No	Requirement Name	Description	Acceptance Criteria	Traceability
FR026	Test environment	The System Operator must be able to test a data classification, business or validation rule in a test environment.	Given a test environment When the system operator adds or modifies a data classification Then the new data classification can be tested in that environment Given a test environment When the system operator adds or modifies a business rule Then the new business rule can be tested in that environment Given a test environment When the system operator adds or modifies a validation rule Then the new validation rule can be tested in that environment	As can been in section 6, MS Purview has real-time connectors – which will update data classification as it is changed on the go.
FR029	Compliance Metrics	The System Operator must be able to configure compliance metrics in the VDS system on the behalf of the VET Regulator. E.g. thresholds for data integrity metrics to be defined in business rules	Given business rules with configured compliance metrics When the System Operator updates the configuration on behalf of the VET Regulator Then STARS will apply the configuration changes	MS Purview has its own reporting capability and Data Custodians will have the ability to use that capability.
FR038	Data lineage	The VDS system must provide capability for data lineage so that consumers and System	Given submitted training data When users query the changes in data between source and	As shown is section 6

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Description	Acceptance Criteria	Traceability
		Operator know how data has changed in its lifecycle from source to consumption.	consumption Then an audit log of changes can be viewed	(diagram), MS Purview will have the ability to perform lineage – from target to source – performing reverse trace capability through API connectors.
FR039	Metadata	The VDS system must enable metadata management. DEWR is currently using Aristotle and Azure Purview for metadata management. The compatibility of these systems needs to be ensured.	Given the ability to maintain data about data (i.e. metadata) When data is edited and captured Then metadata will be compatible with DEWR's systems - Aristotle and Azure Purview	All Purview metadata will be stored in its own repository and metadata is Upserted on the fly as it changes through live connectors.
FR059	Data labels and Identifiers	The VDS system must assign data labels and allow for data tagging. The VDS system must be able to export labelled data and provide identifiers to allow linkage with data from other systems like SEE, ADMS, NESM and "Skills, Education and Employment Data integration platform".	Given training activity data When data labels are assigned Then the data labels may be used for reporting And the data labels may be used for integrations Given training activity data When records are tagged Then the record tags may be used for reporting And the record tags may be used for integrations	This will be described in detail in another document – which entails MS Purview and its technical aspect of DEWR. This document will be created after the

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Description	Acceptance Criteria	Traceability
				completion and sharing of the Metadata management document from DEWR.
FR060	Record Connecting	The VDS system must provide capability to connect all records for a particular student	Given student training data When a user queries a student record Then all records for the student can be returned	MS Purview will have the ability to perform data lineage and record traceability

7.2 Non-Functional

The Non-Functional requirements are more applicable to a Data Management Framework and will be addressed in a separate document after the Data Management Framework document(s) are available.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

Appendix - Acronyms

Acronym/ Abbreviation	Definition
API	Application Programming Interfaces
AVETMISS	Australian Vocational Education and Training Management Information Statistical Standard
DEWR	Department of Employment and Workplace Relations
DXC	STARS Solution Provider and all round rockstars
FFS	Fee For Service
FR	Functional Requirement
MDM	Master Data Management
NCVER	National Centre for Vocational and Education Research
NFR	Non-Functional Requirement
Error! Reference source not found.	Organisational Change Management
RTO	Registered Training Organisation
SMS	Student Management System
STA	State Training Authority
STARS	Student Training & Activity Reporting System
TGA	Training.gov.au
TO	Training Organisation
VDS	VET Data Streamlining
VET	Vocational Education & Training
MS Purview	Microsoft Data Intelligence Tool
SRMP	Security Risk Management Plan
SSP	System Security Plan
IRP	Incident Response Plan
SOP	Standard Operating Procedures

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 298 of 1295

Australian Government
Department of Employment and Workplace Relations

Style Definition: TOC 1

Style Definition: TOC 2: Tab stops: 1.55 cm, Left + 15.98 cm, Right,Leader: ...

Style Definition: TOC 3: Tab stops: 2.33 cm, Left + 15.98 cm, Right,Leader: ...

Detailed Design Document

Master Data Service

1 Document Management

1.1 Document Control

Version	Change Description	Date
0.1	Draft Version	2023-02-10
0.2	Client Feedback	2023-03-15
1.0		

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This is a living document and will be updated as required.

1.2 Document Author

Role	Name
DXC Data Architect	s 47F(1)

1.3 Document Reviewers

Role	Name
xxx	xxxxx

1.4 Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

1.5 Related Documents

Reference	Document
A	STARS Solution Requirements Document – 1.3.docx
B	STARS Solution Design – 1.0.docx
C	STARS Detailed Design - Master DB Physical ERD - V0.1.pdf
D	DRAFT VET Information Standard Logical Interchange Specification V2 R1_V1.0.xlsx
E	Concordance Mapping Design Verification 24 Nov 22 - DEWR feedback R1_V1.0.xlsx
F	IS Prod Technical Dictionary_01092022_R1_V1.0.xlsx

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 300 of 1295

Reference	Document
G	Data Architecture Naming Standards.docx
H	IS Prod Data Model Stakeholder_Core_Concepts_01092022_R1_V1.0
I	IS Prod Data Model Responsibility_Core_Concepts_01092022_R1_V1.0
J	IS Prod Data Model Location_Core_Concepts_01092022_R1_V1.0
K	IS Prod Data Model Product_Core_Concept_01092022_R1_V1.0
L	IS Prod Data Model Offering_Core_Concepts_01092022_R1_V1.0
M	STARS Detailed Design - Master DB Physical ERD – Ref Data Subject Areas V0.1.pdf

Formatted: Caption, Space Before: 0 pt, After: 0 pt

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 301 of 1295

2 Open Design Items

This section describes items arising from the design process that remain unresolved pending further work that will take place later during the project.

ID	Open Item	Owner	Due Date	Work Item Ref.

Formatted Table

Contents

1 Document Management 2

1.1 Document Control 2

1.2 Document Author 2

1.3 Document Reviewers 2

1.4 Document Approval 2

1.5 Related Documents 2

2 Open Design Items 4

Contents 5

3 Introduction 108

3.1 Purpose 108

3.2 Intended Audience 108

4 Scope and Exclusions 119

4.1 In Scope 119

4.2 Out of Scope 119

4.2.1 State Specific Fields (SSF) 119

4.2.2 Referential Integrity relating to Reference Data sets 119

4.2.3 External Identifiers 119

4.2.4 Physical Database Naming Conventions 119

4.2.5 Partitioning Scheme 119

5 Design Decisions 1210

5.1 Assumptions 1210

5.1.1 Auditing 1210

5.1.2 Soft Deletes 1210

5.1.3 Below the line fields 1210

5.2 Issues 1210

5.3 Changes to Design 1210

5.3.1 Physical Database Data Type Changes 1210

5.4 Versioning 1310

5.5 Auditing 1311

5.6 Deletion of an attribute 1412

5.7 Soft Deletes of an entity 1512

5.8 Surrogate Key Generation 1613

5.9 Referential Integrity of dependent entities 1614

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 303 of 1295

5.9.1 Referential integrity - Location - Student example 1715
5.10 Nullability 2018
5.11 Data Retention 2118
5.12 Database Indexing Schemes 2118
5.12.1 STARS Master Database Surrogate Keys 2119
5.12.2 Retrieval of STARS Surrogate Key 2119
5.13 Database Architecture 2522
5.14 Database Backup 2623
5.14.1 Short-term retention 2623
5.14.2 Long-term retention 2724
5.14.3 Backup storage redundancy 2724
5.14.4 Restore capabilities and features 2825
5.15 Database Archiving 3027
5.15.1 Hot, cool, and archive access tiers 3027
5.15.2 Updating of Historical Records 3027
5.15.3 Table Partitioning 3128
5.15.4 Cool data storage 3229
5.15.5 Disk Types 3229
5.15.6 Removal of Historical Data 3330
5.16 Database Availability 3330
6 Requirements 3532
6.1 Functional 3532
6.2 Non-Functional 3633
7 Feature Overview 4137
7.1 Logical Data Model 4137
7.2 External Identifiers 4137
7.3 Relationship Entities 4541
7.3.1 Stakeholder 4541
7.3.2 Responsibility 4642
7.3.3 Location 4844
7.4 Navigation of the Data Model 4945
7.5 Physical Data Model 4945
7.6 Physical Data Types 5549
8 API Design 5650

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 304 of 1295

8.1 Master Data Service API Overview 5650
8.2 API Consumers 5650
8.3 API Security and Data Segregation 5751
8.4 API Service Limits 5751
8.5 Requesting Data (examples) 5852
8.6 Code Sample 6054
8.7 VET-IS Logical Packet Entities 6256
9 Data 6357
9.1 Schemas 6357
10 Access Management 7766
11 Appendix 7867
11.1 VET-IS Data Packet Logical Entity Relationship Diagram 8367
12 Acronyms 8468
~~**1 Document Management ..... 2**~~
~~1.1 Document Control ..... 2~~
~~1.2 Document Author ..... 2~~
~~1.3 Document Reviewers ..... 2~~
~~1.4 Document Approval ..... 2~~
~~1.5 Related Documents ..... 2~~
~~**Contents ..... 3**~~
~~**2 Introduction ..... 6**~~
~~2.1 Purpose ..... 6~~
~~2.2 Intended Audience ..... 6~~
~~**3 Scope and Exclusions ..... 7**~~
~~3.1 In Scope ..... 7~~
~~3.2 Out of Scope ..... 7~~
~~3.2.1 State Specific Fields (SSF) ..... 7~~
~~3.2.2 Referential Integrity relating to Reference Data sets ..... 7~~
~~3.2.3 External Identifiers ..... 7~~
~~3.2.4 Physical Database Naming Conventions ..... 7~~
~~**4 Design Decisions ..... 8**~~
~~4.1 Assumptions ..... 8~~
~~4.1.1 Auditing ..... 8~~
~~4.1.2 Soft Deletes ..... 8~~

Formatted: Default Paragraph Font, Font: Bold, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Font: Bold, Check spelling and grammar
Formatted: Default Paragraph Font, Font: Bold, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Font: Bold, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Font: Bold, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar

~~4.1.3 Below the line fields ..... 8~~
~~4.2 Issues ..... 8~~
~~4.2.1 Data Retention ..... 8~~
~~4.2.2 Database Availability ..... 8~~
~~4.3 Versioning ..... 8~~
~~4.4 Auditing ..... 9~~
~~4.5 Deletion of an attribute ..... 10~~
~~4.6 Soft Deletes of an entity ..... 10~~
~~4.7 Surrogate Key Generation ..... 11~~
~~4.8 Referential Integrity of dependent entities ..... 12~~
~~4.9 Nullability ..... 13~~
~~4.10 Data Retention ..... 13~~
~~4.11 Database Indexing Schemes ..... 13~~
~~4.11.1 STARS Master Database Surrogate Keys ..... 13~~
~~4.11.2 Retrieval of STARS Surrogate Key ..... 13~~
~~4.12 Database Architecture ..... 17~~
~~**5 Requirements ..... 19**~~
~~5.1 Functional ..... 19~~
~~5.2 Non-Functional ..... 20~~
~~**6 Feature Overview ..... 21**~~
~~6.1 Logical Data Model ..... 21~~
~~6.2 External Identifiers ..... 21~~
~~6.3 Relationship Entities ..... 25~~
~~6.3.1 Stakeholder ..... 25~~
~~6.3.2 Responsibility ..... 26~~
~~6.3.3 Location ..... 28~~
~~6.4 Navigation of the Data Model ..... 29~~
~~6.5 Physical Data Model ..... 29~~
~~6.6 Physical Data Types ..... 33~~
~~**7 API Design ..... 34**~~
~~7.1 Master Data Service API Overview ..... 34~~
~~7.2 API Consumers ..... 34~~
~~7.3 API Security and Data Segregation ..... 35~~
~~7.4 API Service Limits ..... 35~~

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 306 of 1295

7.5 Requesting Data (examples) 36
7.6 Code Sample 38
7.7 VET-IS Logical Packet Entities 40
8 Data 41
8.1 Schemas 41
9 Access Management 50
10 Appendix 51
10.1 VET Information Standard Logical Data Model 51
10.2 VET-IS Data Packet Logical Entity Relationship Diagram 56
11 Acronyms 57

Figure 1 - Azure SQL Ledger 10
Figure 2 - Jurisdictional Databases 17
Figure 3 - Master Database Instance Architecture 17
Figure 4 - VET IS Logical Data Model high level 21
Figure 5 - STARS Master DB Physical ERD 32
Figure 6 : Master Data Service Overview 34
Figure 7 : Data Segregation and Access Security 35
Figure 8 - IS Prod Data Model Stakeholder_Core_Concepts_01092022_R1_V1.0 51
Figure 9 - IS Prod Data Model Responsibility_Core_Concepts_01092022_R1_V1.0 52
Figure 10 - IS Prod Data Model Location_Core_Concepts_01092022_R1_V1.0 53
Figure 11 - IS Prod Data Model Product_Core_Concept_01092022_R1_V1.0 54
Figure 12 - IS Prod Data Model Offering_Core_Concepts_01092022_R1_V1.0 55
Figure 13 - VET-IS Data Packet Logical Entity Relationship Diagram 56

Figure 1 : Azure SQL Ledger ..... 14~~12~~
Figure 2 : Jurisdictional Databases ..... 25~~22~~
Figure 3 : Master Database Instance Architecture ..... 26~~23~~
Figure 4: Geo-redundant storage (GRS) ..... 28~~25~~
Figure 5: Data Archiving Tiers ..... 31~~28~~
Figure 6 : VET IS Logical Data Model high level ..... 41~~37~~
Figure 7 : STARS Master DB Physical ERD ..... 54~~48~~
Figure 8 : Master Data Service Overview ..... 56~~50~~
Figure 9 : Data Segregation and Access Security ..... 57~~51~~
Figure 8 : VET-IS Data Packet Logical Entity Relationship Diagram ..... 83~~67~~

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Font: Bold, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Font: Bold, Check spelling and grammar

Formatted: Default Paragraph Font, Font: Bold, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Font: Bold, Check spelling and grammar

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 307 of 1295

~~2~~3 Introduction

The purpose of this document is to capture the detailed design requirements for the STARS Master Data service solution.

The Master Data service will provide the storage of mastered VET training activity data as required for national reporting purposes.

Note: This is a living document that will be updated as designs for various other STARS services are developed and evolve. Accordingly, this document will evolve to reflect and document master database components as the STARS solution build progresses.

~~2.1~~3.1 Purpose

The purpose of this detailed design document is to provide a technical description of the Master database component of the STARS solution. The document is intended for technical stakeholders, including architects, software engineers, administrators, and operations teams, who are responsible for designing, developing, deploying, and operating the STARS solution in Azure.

The document is intended to be a living document, updated as the STARS solution evolves and changes over time.

~~2.2~~3.2 Intended Audience

This detailed design document provides technical information and guidance to help technical stakeholders understand the Master database component of the STARS solution. The document is expected to be used by technical stakeholders who have a good understanding of database modelling concepts and are experienced in relational database platforms.

This document is not intended for non-technical readers, nor for readers who are not experienced in the design and configuration of relational databases.

The primary audience for this document includes:

- Program technical staff: This includes architects, software engineers, administrators, and operations teams within the program responsible for the STARS solution.
- External technical stakeholders: This includes technical staff from other federal, state, and territory governments agencies, including VDS embedded team members.
- ICT Technical Staff: This includes a variety of technical roles involved during the VDS program and post-go live, including network engineers, systems administrators, database administrators, security engineers, and DevOps engineers.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 308 of 1295

~~3~~4 Scope and Exclusions

~~3.1~~4.1 In Scope

The following are in-scope of this detail design solution:

The physical design of the STARS Master relational database and associated API Service.

~~3.2~~4.2 Out of Scope

The following are out of scope:

~~3.2.1~~4.2.1 State Specific Fields (SSF)

The VET Information Standard (VET IS) data model will not be extended to include any further SSF other than those already defined as part of the VET IS.

~~3.2.2~~4.2.2 Referential Integrity relating to Reference Data sets

The checking of whether a Code, Reference Number or Name is a valid member of an associated reference data set will not be enforced at the database level. This will be handled by the Validation API Service.

However, in order to support harvesting of these relationships by BI tools referential integrity constraints against Reference Data sets will be defined with the STARS Master database but not enabled.

~~3.2.3~~4.2.3 External Identifiers

This document does not address the management of external identifiers or their content but rather only where they are stored in relation to an entity within the VET IS data model.

~~3.2.4~~4.2.4 Physical Database Naming Conventions

~~The naming of physical database objects within this document is indicative only and associated naming conventions are subject to change.~~ Please see the related document 'Data Architecture Naming Standards.docx'~~Documentation.~~ pertaining to physical database naming conventions~~ will be supplied subsequently prior to the associated implementation phase.~~

4.2.5 Partitioning Scheme

The Partitioning Scheme for Master database tables has yet to be defined and will be done subsequently in consulation with the client.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 309 of 1295

45 Design Decisions

4.15.1 Assumptions

4.1.15.1.1 Auditing

This document assumes the following in the absence of information to the contrary:

The user/role that was responsible for submitting data at source (i.e., Training Organisation's Student Management System) which in turn then caused the insert/update in the STARS Master database is required to be recorded in the associated 'ModifiedBy' column of the related row in the database table.

4.1.25.1.2 Soft Deletes

The scope of 'Soft Delete' functionality has been limited to that of a particular Stakeholder or Responsibility as the scope of cascading 'Soft Delete' relating to other entities would have too larger impact with potentially undesirable results (i.e., 'Soft Deleting' a Product would require 'Soft Deleting' all the enrolments that have a dependency upon it).

4.1.35.1.3 Below the line fields

All "below the line fields" as stated in the DRAFT VET Information Standard Logical Interchange Specification V2 R1_V1.0.xlsx document have been included as part of the VET Information Standard (VET IS) and as such fit for National reporting consumption.

5.2 Issues

5.3 Changes to Design

This section lists changes to the design that have been made independently of feedback from the client and aren't captured in the previous version of the document.

4.25.3.1 Physical Database Data Type Changes

The following changes have been made in relation to best database practice design. ~~Data Retention~~

- BIGINT will be used instead of UNIQUEIDENTIFIER as the data type for STARS surrogate keys.
- NVARCHAR will be used instead of VARCHAR as the data type to represent strings.

~~STARS Master Database data retention periods are yet to be defined by the client. The absence of an archiving strategy is not practical and may cause future capacity and performance issues in the future. A pragmatic data archiving strategy must be defined by the client.~~

Formatted: Heading 3,Level 1 - 1,Numbered Heading 3,hd3,h3,h3 sub heading,Heading 3 - old,H3,Head 3,C Sub-Sub/Italic,Head 31,Head 32,C Sub-Sub/Italic1,Head 33,C Sub-Sub/Italic2,Head 311,Head 321,C Sub-Sub/Italic11,h31,3,not in TOC,Heading 3 Char1,Para3,H31,sub-sub

Formatted: List Paragraph,NFP GP Bulleted List,TOC style,lp1,Bullet OSM,Proposal Bullet List,List Paragraph1,Recommendation,List Paragraph11,List Paragraph111,L,F5 List Paragraph,Dot pt,CV text,Medium Grid 1 - Accent 21,Numbered Paragraph,Bullets,NAST Quote,lp11,n, Bulleted + Level: 1 + Aligned at: 0.63 cm + Indent at: 1.27 cm

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 310 of 1295

~~4.2.1 Database Availability~~

~~STARS Master Database service level agreements are yet to be defined.~~

~~4.3~~5.4 Versioning

Master Data will be held in temporal tables, for audit and time-based analysis.

Temporal tables (SYSTEM_VERSIONING) will be used to retain history on applicable objects in the Azure SQL Master database. Changes to data will be tracked using this mechanism.

Temporal tables are a programmability feature of Azure SQL Database that allows the tracking and analysis of the full history of changes in data, without the need for custom coding. Temporal tables keep data closely related to time context so that stored facts can be interpreted as valid only within the specific period. This property of temporal tables allows for efficient time-based analysis and getting insights from data evolution.

Below is a sample code for creating temporal tables in Azure SQL database.

```
CREATE TABLE Stakeholder
(
    [StakeholderKey] int NOT NULL PRIMARY KEY CLUSTERED
  , [StakeholderTypeCode] nvarcharNVARCHAR(100) NOT NULL
  , [ValidFrom] datetime2 (0) GENERATED ALWAYS AS ROW START
  , [ValidTo] datetime2 (0) GENERATED ALWAYS AS ROW END
  , PERIOD FOR SYSTEM_TIME (ValidFrom, ValidTo)
 )
 WITH (SYSTEM_VERSIONING = ON (HISTORY_TABLE = dbo.StakeholderHistory));
```

~~4.4~~5.5 Auditing

A ‘ModifiedBy’ column will record the user/role which last modified an attribute within each row of every database table within the STARS Master database. The associated API process of the Master Data Service will supply the appropriate user/role to be inserted.

The user/role that was responsible for submitting data at source (i.e., Training Organisation’s Student Management System) which in turn then caused the insert/update in the STARS Master database is required to be recorded in the associated ‘ModifiedBy’ column of the related row in the database table.

The reason that the insert/update was initiated will also be recorded in a ‘ModifiedReason’ column.

Microsoft Azure SQL Ledger is an encrypted reporting layer which sits above the Temporal Tables described above as part of Versioning. It will be employed to provide cryptographic proof of data integrity to auditors. This proof can help streamline the auditing process. It also provides nonrepudiation regarding the integrity of the system's data.

Azure SQL Ledger Overview:

Establishing trust around the integrity of data stored in database systems has been a longstanding problem for all organizations that manage financial, medical, or other sensitive data. The ledger

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 311 of 1295

feature provides tamper-evidence capabilities in your database. You can cryptographically attest to other parties, such as auditors or other business parties, that your data hasn't been tampered with.

Ledger helps protect data from any attacker or high-privileged user, including database administrators (DBAs), system administrators, and cloud administrators. As with a traditional ledger, the feature preserves historical data. If a row is updated in the database, its previous value is maintained and protected in a history table. Ledger provides a chronicle of all changes made to the database over time.

Ledger and the historical data are managed transparently, offering protection without any application changes. The feature maintains historical data in a relational form to support SQL queries for auditing, forensics, and other purposes. It provides guarantees of cryptographic data integrity while maintaining the power, flexibility, and performance of the SQL database.

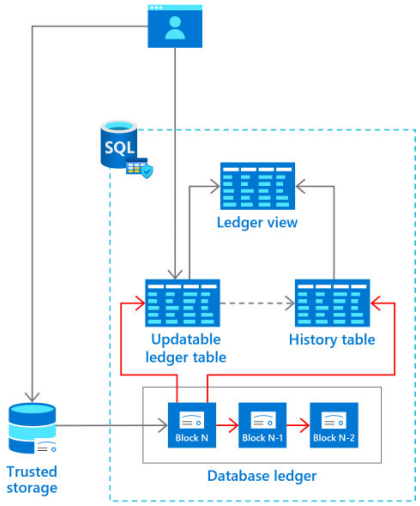


Figure 111 :– Azure SQL Ledger

4.55.6 Deletion of an attribute

An entity can have multiple values for the same attribute (i.e., a Person can have multiple Educational Achievements) stored as a one-to-many table relationship in the Master database. In the event that a subsequent data submission related to an entity contains less values than what is stored in the Master database then a delete will be performed for the missing value. The historic presence of the deleted record will be retained as explained in the Versioning section. A comparison of an entities' set of attributes will need to be performed prior to any update operation within the master database.

This concept does not apply to Parent entities such as Stakeholder, Responsibility, Product, Location and Offering as the full set of all these entities will not form part of a submission so it is not possible

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 312 of 1295

to determine if the entity has been deleted at source. To delete a Parent entity 'Soft Delete' functionality must be used.

~~4.6~~5.7 Soft Deletes of an entity

A 'SoftDelete' column of a Boolean type will be added to every table within the STARS Master database to enable a STARS operator to hide an entity (i.e., an entity refers to a particular unique Person or a particular unique Subject Enrolment and not all Stakeholders or Responsibilities etc… in general) and its' related entities within the STARS Master database without actually deleting any data. A cascading data update will be ~~enabled~~ enabled ~~related~~ that will hide an entities' related attributes as well as related entities ~~to also be hidden~~. Hiding an entity and its' related entities will prevent them from being reported, ~~extracted or~~extracted or receiving any submission updates.

The scope of the cascade will depend upon the entity being 'Soft Deleted' as documented in the below table.

Table 11~~1~~ - Soft Delete Cascade Scope

Entity	Cascading Update Required
Stakeholder	• All associated Stakeholder attributes • All relationships held within the Stakeholder Relationships table but not the related entities defined as part of the relationship • All associated Stakeholder Locations • All related relationships held within the Stakeholder Responsibility table and the associated Responsibility entities • Any related relationships in the Responsibility Relationship table but not the related entities defined as part of the relationship
Responsibility	• All associated Responsibility attributes • All related relationships held within the Stakeholder Responsibility table and the associated Responsibility entities

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 313 of 1295

	• Any related relationships in the Responsibility Relationship table but not the related entities defined as part of the relationship

~~4.7~~5.8 Surrogate Key Generation

All Surrogate Keys (SK) used to uniquely identify entities and their associated attributes at the record level within the VET IS data model will be generated internally within the STARS Master database upon insertion of a new entity or attribute.

SKs are internal STARS/SQL keys, used for row-level identification and associated Foreign Keys (FK), that have no meaning or purpose outside of Azure SQL and logging/audit/queries. The SK will form an atomic primary key (PK) of each table consisting of a GUID generated using the newid () function.

The identification of whether a new SK needs to be generated (or whether the entity already exists) will be defined by an associated Business Key (BK) for that entity. This may be a composite key comprised of more than a single column.

~~4.8~~5.9 Referential Integrity of dependent entities

Referential integrity of dependent VET IS data model entities will be enforced ~~via database triggers~~within the Master database as per the following hierarchy:

Table 22~~2~~ - Referential integrity of dependent VET IS data model entities

Entity	Dependency	Relationship Enabler
Stakeholder - Student	None	N/A
Stakeholder – Training Organisation	None	N/A
Stakeholder - Government	None	N/A
Product - Program	None	N/A
Product - Subject	None	N/A
Location - Student	Stakeholder - Student	Stakeholder Location table
Location – Training Organisation	Stakeholder – Training Organisation	Stakeholder Location table
Location – Delivery Location	Stakeholder – Training Organisation	Stakeholder Location table
Offering - Program	Product - Program	

Formatted: Space After: Auto

Formatted: Space After: Auto

Formatted: Space After: Auto

Formatted: Space After: Auto

Formatted: Space After: Auto

Formatted: Space After: Auto

Formatted: Space After: Auto

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 314 of 1295

Offering - Program	Stakeholder – Training Organisation	
Offering - Subject	Product - Subject	
Offering - Subject	Stakeholder – Training Organisation	
Responsibility – Program Enrolment	Stakeholder - Student	Stakeholder Responsibility table
Responsibility – Program Enrolment	Product - Program	Program Enrolment. Associated Program Reference Number
Responsibility – Program Enrolment	Stakeholder - Training Organisation	Stakeholder Responsibility table
Responsibility – Program Enrolment (funded programs only)	Stakeholder - Government	Stakeholder Responsibility table
Responsibility – Subject Enrolment	Stakeholder - Student	Stakeholder Responsibility table
Responsibility – Subject Enrolment	Product - Program	
Responsibility – Subject Enrolment	Product - Subject	
Responsibility – Subject Enrolment	Stakeholder - Training Organisation	Stakeholder Responsibility table
Responsibility – Subject Enrolment (funded subjects only)	Stakeholder - Government	Stakeholder Responsibility table
Responsibility – Subject Enrolment	Location – Delivery Location	Responsibility Location and Location Identifier tables

As such, any submission packages containing multiple entities (i.e., a Stakeholder – Training Organisation with an associated Location – Delivery Location) which have not been mastered before will need to be inserted as per the above order.

5.9.1 Referential integrity - Location - Student example

Below is a pseudo code example of the implementation of a database code used to check the referential integrity between a Student's Location and the Student. In this example the Student must

Formatted: Heading 3,Level 1 - 1,Numbered Heading 3,hd3,h3,h3 sub heading,Heading 3 - old,H3,Head 3,C Sub-Sub/Italic,Head 31,Head 32,C Sub-Sub/Italic1,Head 33,C Sub-Sub/Italic2,Head 311,Head 321,C Sub-Sub/Italic11,h31,3,not in TOC,Heading 3 Char1,Para3,H31,sub-sub

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 315 of 1295

exist within the Master database prior to an insertion of a record representing the student's email address into the ElectronicAddress table.

The below table shows the row that the example is based upon from the full list of dependencies shown in the Referential integrity of dependent VET IS data model entities table.

Table 333 - Referential integrity - Location - Student example

Entity	Dependency	Relationship Enabler
Location - Student	Stakeholder - Student	Stakeholder Location table

The following code has been written for illustrative purposes only:

5.9.1.1 Insert/Update Operation that would execute database code

The following data values (i.e., 'Address','AUS','Electronic address','Email address','student@email.com.au') would be included in a submitted Json data package which would also include (but not be limited to) the associated Student's external identifier and the external identifier of the Training Organisation that the Student is registered with.

```
--Before the following insert (or update) can be performed, check that the Student exists

INSERT INTO ElectronicAddress (LocationTypeCode, CountryCode, AddressTypeCode, ElectronicAddressTypeCode,
ElectronicAddressName)

VALUES ('Address','AUS','Electronic address','Email address','student@email.com.au');
```

5.9.1.2 Get Student's STARS surrogate key

```
--If a row is returned from the below then Student exists (should be only 1 row returned), so continue with Student
Location Insert/Update

--Get unique Student using Composite Key of Student and Training Organisation

SELECT a. StakeholderSKey AS StudentSKey, a. RelatedStakeholderSKey AS TrainingOrgSKey,
FROM StakeholderRelationship a
WHERE a.StakeholderRoleTypeCode = 'Student'
AND a.RelatedStakeholderRoleTypeCode = 'Primary organisation'
AND a.StakeholderRelationshipTypeCode = 'Enrolled at Training Organisation'  --NOTE: This is Classification does not
currently exist as part of VET IS R1 V0.1 documentation set.
AND a.StakeholderSKey = (

    --Get Training Organisation SK

    SELECT StakeholderSkey

    FROM StakeholderIdentifier

    WHERE  StakeholderIdentifierTypeCode = 'Training Organisation external identifier'

    AND StakeholderIdentifierTypeReferenceNumber = '<supplied external identifier for Training Org>' --(provided as
    part of data package)
```

Formatted: Font: 11 pt

Formatted: Caption, Keep with next

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted: Font: 13 pt, Font color: Auto

Formatted: Heading 4,Level 2 - a,h4,Heading 4 Char Char Char,1.1 Heading,Fourth Level,sub-sub-sub para,h4 sub sub heading,h41,h42,Para4,H4,Sub Sub section,4,(Alt+4),H41,(Alt+4)1,H42,(Alt+4)2,H43,(Alt+4)3,H44,(Alt+4)4,H45,(Alt+4)5,H411,(Alt+4)11,H421,(Alt+4)21,

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Commented [s 47F1]: This information needs to be fed back to NCVER Data Standards Team.

Formatted ...

Formatted: Font: 9 pt, Font color: Green

Formatted ...

Formatted: Font: 9 pt, Font color: Green

Formatted: Indent: Left: 1.27 cm

Formatted ...

Formatted ...

Formatted ...

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 316 of 1295

```
)
AND a.RelatedStakeholderSKey = (
        --Get Student SK
        SELECT StakeholderSkey
        FROM StakeholderIdentifier
        WHERE  StakeholderIdentifierTypeCode = 'Student external identifier'
        AND StakeholderIdentifierTypeReferenceNumber = '<supplied external identifier for Student>' --(provided as part of data package)
)
```

5.9.1.3 Check if Student already has an email address

```
--Determine if INSERT (no rows returned) or UPDATE (1 row is returned) is required
--Does an Electronic Location exist for the given Student?
--NOTE: This code assumes only 1 email address per student!
SELECT LocationSkey
FROM StakeholderLocation a
INNER JOIN
ElectronicAddress b
ON a.LocationSkey = b.LocationSkey
WHERE a.StakeholderSKey = StudentSKey --(from above)
AND a.StakeholderLocationUsageTypeCode = 'Electronic'
AND a.StakeholderRoleTypeCode = 'Student'
AND b. LocationTypeCode = 'Address'
AND b. CountryCode= 'AUS'
AND b. AddressTypeCode='Email address'
```

5.9.1.4 Perform Insert of ElectronicAddress if the Student exists and the email address does not exist

```
--Then either INSERT
INSERT INTO ElectronicAddress (LocationTypeCode, CountryCode, AddressTypeCode, ElectronicAddressTypeCode, ElectronicAddressName)
VALUES ('Address','AUS','Electronic address','Email address','student@email.com.au');
```

Formatted: Normal
Formatted: Font: (Default) Consolas, Font color: Custom Color(RGB(1,1,253))
Formatted: Font: 9 pt
Formatted: Font: 9 pt, Font color: Green
Formatted: Indent: Left: 1.27 cm
Formatted: Font: (Default) Consolas, 9 pt, Font color: Custom Color(RGB(1,1,253))
Formatted: Font: 9 pt
Formatted: Font: (Default) Consolas, 9 pt, Font color: Custom Color(RGB(1,1,253))
Formatted: Font: 9 pt
Formatted: Font: (Default) Consolas, 9 pt, Font color: Custom Color(RGB(1,1,253))
Formatted: Font: 9 pt
Formatted: Font: 9 pt, Italic, Highlight
Formatted: Highlight
Formatted: Font: 9 pt, Italic, Highlight
Formatted: Font: 9 pt, Highlight
Formatted: Font: 9 pt
Formatted: Heading 4,Level 2 - a,h4,Heading 4 Char Char Char,1.1 Heading,Fourth Level,sub-sub-sub para,h4 sub sub heading,h41,h42,Para4,H4,Sub Sub section,4,(Alt+4),H41,(Alt+4)1,H42,(Alt+4)2,H43,(Alt+4)3,H44,(Alt+4)4,H45,(Alt+4)5,H411,(Alt+4)11,H421,(Alt+4)21,
Formatted: Highlight
Formatted: Font: 9 pt
Formatted: Font color: Green
Formatted: Font: Italic, Font color: Green
Formatted: Heading 4,Level 2 - a,h4,Heading 4 Char Char Char,1.1 Heading,Fourth Level,sub-sub-sub para,h4 sub sub heading,h41,h42,Para4,H4,Sub Sub section,4,(Alt+4),H41,(Alt+4)1,H42,(Alt+4)2,H43,(Alt+4)3,H44,(Alt+4)4,H45,(Alt+4)5,H411,(Alt+4)11,H421,(Alt+4)21,H431, Space After: 0 pt, Line spacing: single
Formatted: Font color: Green

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 317 of 1295

5.9.1.5 Perform Update of ElectronicAddress. ElectronicAddressName if the Student exists and the email address does exist

```
--OR UPDATE
UPDATE ElectronicAddress
SET ElectronicAddressName = 'student@email.com.au'
WHERE LocationTypeCode = 'Address'
AND CountryCode = 'AUS'
AND AddressTypeCode = 'Electronic address'
AND ElectronicAddressTypeCode = 'Email address'
AND LocationSKey = LocationSkey;      --(from above)
```

5.9.1.6 Perform Insert of both Student and ElectronicAddress if the Student does not exist

The following pseudo code steps must be performed in order and all required data elements are assumed to have been provided as part of a submitted Json data package.

1. Insert new Student into Person table and associated attribute tables.
2. Insert associated Training Organisation into Organisation table and associated attribute table (if it does not already exist)
3. Insert the relationship between the Student and the Training Organisation into the StakeholderRelationship table.
4. Perform the following insert and use the newly generated LocationSkey as part of the insert for the next step:

```
INSERT INTO ElectronicAddress (LocationTypeCode, CountryCode, AddressTypeCode, ElectronicAddressTypeCode, ElectronicAddressName)
VALUES ('Address','AUS','Electronic address','Email address','student@email.com.au');
```

5. Insert a relationship between the student and the email address into the StakeholderLocation table.

~~4.9~~5.10 Nullability

All columns within the STARS Master database will be defined as Nullable (i.e., can contain a NULL value) except for columns which belong to the primary key of a table. Ensuring that a column has a mandatory value is the responsibility of the Validation Service.

Formatted: Font: 13 pt

Formatted: Font color: Green

Formatted: Font: (Default) Consolas, 9 pt, Font color: Custom Color(RGB(1,1,253))

Formatted: Font: (Default) Consolas, Font color: Custom Color(RGB(1,1,253))

Formatted: Font: (Default) Consolas, Font color: Custom Color(RGB(1,1,253))

Formatted: Font: (Default) Consolas, Font color: Custom Color(RGB(1,1,253))

Formatted: Font: (Default) Consolas, Font color: Custom Color(RGB(1,1,253))

Formatted: Font: (Default) Consolas, Font color: Custom Color(RGB(1,1,253))

Formatted: Font: 9 pt

Formatted: Normal, No bullets or numbering

Formatted: No bullets or numbering

Formatted: List Paragraph,NFP GP Bulleted List,TOC style,lp1,Bullet OSM,Proposal Bullet List,List Paragraph1,Recommendation,List Paragraph11,List Paragraph111,L,F5 List Paragraph,Dot pt,CV text,Medium Grid 1 - Accent 21,Numbered Paragraph,Bullets,NAST Quote,lp11,n, Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.63 cm + Indent at: 1.27

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 318 of 1295

~~4.10~~5.11 Data Retention

All data that has been stored withing the STARS Master database will be accessible and kept in perpetuity within the STARS Data Landscape as part of either the STARS Master database, STARS Data Warehouse, STARS Data Lake or STARS Data Lakehouse.

A data archiving strategy for the Master database will ~~need to be advised by the client~~be devised which takes into consideration ~~legislative requirements and~~ the functional need for data to remain in the STARS Master database once it has been pushed to the STARS Datawarehouse ~~to suffice reporting purposes~~. One of the key considerations for keeping data stored in the Master database being at what point data can be deemed static from a pragmatic perspective to balance performance, capacity issues with ongoing rising costs.

~~4.11~~5.12 Database Indexing Schemes

~~4.11.1~~5.12.1 STARS Master Database Surrogate Keys

Surrogate keys (SK) of type BIGINT~~(GUID)~~ will form the Primary Keys of all tables within the STARS Master database conforming to the VET IS Logical Data Model. As such these SK will define the physical ordering of a table and will provide an efficient linking mechanism to navigate the physical data model and join data entities together. In order to use these SK they must be retrieved using external identifiers.

An example of a STARS Master database primary key has been provided here for clarity and brevity.

```
CREATE CLUSTERED INDEX PK_ PersonStakeholderSK ON Person (StakeholderSK);
```

All other primary keys will be based solely on the surrogate key of the table and will follow the above naming convention as detailed in the related document 'Data Architecture Naming Standards.docx'.

~~4.11.2~~5.12.2 Retrieval of STARS Surrogate Key

~~4.11.2.1~~5.12.2.1 Stakeholder and Responsibility

Both the Stakeholder and Responsibility entities contain an 'Identifier' table (Stakeholder Identifier and Responsibility Identifier) where all related external identifiers will be stored in the columns Stakeholder Identifier Type Reference Number and Responsibility Identifier Type Reference Number respectively.

These tables are logically partitioned by an 'Identifier Type Code' which gives context to the type of identifier stored (not all Reference Numbers stored relate to identifiers classified as 'external identifiers'). As such a subset of fully covered filtered indexes will be created to support efficient retrieval of an entities' associated STARS surrogate key. These indexes will be restricted to Stakeholder/Responsibility Identifier Type Code classifications relating to external identifiers.

Formatted: Default Paragraph Font, Font: (Default) +Body (Calibri), 11 pt, Font color: Auto, Pattern: Clear

Formatted: Pattern: Clear

Formatted: Default Paragraph Font, Font: (Default) +Body (Calibri), 11 pt, Font color: Auto, Pattern: Clear

Formatted: Pattern: Clear

Formatted: Pattern: Clear

Formatted: Pattern: Clear

Formatted: Default Paragraph Font, Font: (Default) +Body (Calibri), 11 pt, Font color: Auto, Pattern: Clear

Formatted: Pattern: Clear

Formatted: Pattern: Clear

Formatted: Pattern: Clear

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 319 of 1295

These indexes cannot be unique as the composite key of Person and Organisation is required to uniquely identify a Person's external identifier (i.e., the same value for an external id of '123' may be used across multiple Training Organisations to represent two different Students).

~~4.11.2.2~~5.12.2.2 Stakeholder

~~4.11.2.2.1~~5.12.2.2.1 Student

The use of the following index will be specifically limited to the retrieval of 'Stakeholder Skey' for Stakeholder Identifier Type Code = 'Student external identifier':

```
CREATE NONCLUSTERED INDEX
IX_NIX_StakeholderIdentifier_StakeholderIdentifierTypeCode_StakeholderIdentifierTypeReferenceNumberGet_StakeholderSkey_Student
ON StakeholderIdentifier (StakeholderIdentifierTypeReferenceNumber) INCLUDE
(StakeholderSkey)
WHERE StakeholderIdentifierTypeCode = 'Student external identifier';
```

~~4.11.2.2.2~~5.12.2.2.2 Training Organisation

The use of the following index will be specifically limited to the retrieval of 'Stakeholder Skey' for Stakeholder Identifier Type Code = 'Training Organisation external identifier':

```
CREATE NONCLUSTERED INDEX
IX_NIX_StakeholderIdentifier_StakeholderIdentifierTypeCode_StakeholderIdentifierTypeReferenceNumberGet_StakeholderSkey_TrainingOrganisation
ON StakeholderIdentifier (StakeholderIdentifierTypeReferenceNumber) INCLUDE
(StakeholderSkey)
WHERE StakeholderIdentifierTypeCode = 'Training Organisation external identifier';
```

~~4.11.2.2.3~~5.12.2.2.3 School

The use of the following index will be specifically limited to the retrieval of 'Stakeholder Skey' for Stakeholder Identifier Type Code = 'School external identifier':

```
CREATE NONCLUSTERED INDEX
IX_NIX_StakeholderIdentifier_GetStakeholderIdentifierTypeCode_StakeholderIdentifierTypeReferenceNumber_StakeholderSkey_School
ON StakeholderIdentifier (StakeholderIdentifierTypeReferenceNumber) INCLUDE
(StakeholderSkey)
WHERE StakeholderIdentifierTypeCode = 'School external identifier';
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 320 of 1295

~~4.11.2.3~~5.12.2.3 Responsibility

~~4.11.2.3.1~~5.12.2.3.1 Program Enrolment

The use of the following index will be specifically limited to the retrieval of 'Responsibility Skey' for Responsibility Identifier Type Code = 'Program enrolment external identifier':

```
CREATE NONCLUSTERED INDEX
IX_NIX_ResponsibilityIdentifier_GetResponsibilityIdentifierTypeCode_ResponsibilityIdentifierTypeReferenceNumber_ResponsibilitySkey_ProgramEnrolment

ON ResponsibilityIdentifier (ResponsibilityIdentifierTypeReferenceNumber) INCLUDE
(ResponsibilitySkey)
WHERE ResponsibilityIdentifierTypeCode = 'Program enrolment external identifier';
```

~~4.11.2.3.2~~5.12.2.3.2 Subject Enrolment

The use of the following index will be specifically limited to the retrieval of 'Responsibility Skey' for Responsibility Identifier Type Code = 'Subject enrolment external identifier':

```
CREATE NONCLUSTERED INDEX
IX_NIX_ResponsibilityIdentifier_GetResponsibilityIdentifierTypeCode_ResponsibilityIdentifierTypeReferenceNumber_ResponsibilitySkey_SubjectEnrolment

ON ResponsibilityIdentifier (ResponsibilityIdentifierTypeReferenceNumber) INCLUDE
(ResponsibilitySkey)
WHERE ResponsibilityIdentifierTypeCode = 'Subject enrolment external identifier';
```

~~4.11.2.3.3~~5.12.2.3.3 Work Placement

The use of the following index will be specifically limited to the retrieval of 'Responsibility Skey' for Responsibility Identifier Type Code = 'Funding Arrangement external identifier':

```
CREATE NONCLUSTERED INDEX
IX_NIX_ResponsibilityIdentifier_ResponsibilityIdentifierTypeCode_ResponsibilityIdentifierTypeReferenceNumber_GetResponsibilitySkey_FundingArrangement

ON ResponsibilityIdentifier (ResponsibilityIdentifierTypeReferenceNumber) INCLUDE
(ResponsibilitySkey)
WHERE ResponsibilityIdentifierTypeCode = 'Work Placement external identifier';
```

~~4.11.2.3.4~~5.12.2.3.4 Subsidy Funding Arrangement

The use of the following index will be specifically limited to the retrieval of 'Responsibility Skey' for Responsibility Identifier Type Code = 'Subsidy Funding Arrangement external identifier':

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 321 of 1295

```
CREATE NONCLUSTERED INDEX
IX_NIX_ResponsibilityIdentifier_ResponsibilityIdentifierTypeCode_ResponsibilityIdentifierTypeReferenceNumber_GetResponsibilitySkey_SubsidyFundingArrangement

ON ResponsibilityIdentifier (ResponsibilityIdentifierTypeReferenceNumber) INCLUDE
(ResponsibilitySkey)
WHERE ResponsibilityIdentifierTypeCode = 'Subsidy Funding Arrangement external identifier';
```

4.11.2.45.12.2.4 Product

4.11.2.4.15.12.2.4.1 Program

The use of the following index will be specifically limited to the retrieval of 'Product Skey' for Program entities:

```
CREATE NONCLUSTERED INDEX
IX_NIX_Program_ProductReferenceNumber_GetProductSKeyProgram

ON Program (ProductReferenceNumber) INCLUDE (ProductSKey)
WHERE ProductTypeCode = 'Program';
```

4.11.2.4.25.12.2.4.2 Subject

The use of the following index will be specifically limited to the retrieval of 'Product Skey' for Subject entities:

```
CREATE NONCLUSTERED INDEX IX_NIX_Subject_ProductReferenceNumber_GetProductSKeySubject

ON Subject (ProductReferenceNumber) INCLUDE (ProductSKey)
WHERE ProductTypeCode = 'Subject';
```

4.11.2.55.12.2.5 Location

4.11.2.5.15.12.2.5.1 Responsibility Location

The use of the following index will be specifically limited to the retrieval of 'Responsibility Location Skey' for Responsibility Location entities:

```
CREATE NONCLUSTERED INDEX IX_NIX_ ResponsibilityLocation_
ResponsibilityLocationReferenceNumber_GetResponsibilityLocationSKey

ON Subject (ResponsibilityLocationReferenceNumber) INCLUDE (ResponsibilityLocationSKey)
WHERE ResponsibilityLocationUsageTypeCode = '<list to be provided by client>';
```

Formatted: List Paragraph,NFP GP Bulleted List,TOC style,lp1,Bullet OSM,Proposal Bullet List,List Paragraph1,Recommendation,List Paragraph11,List Paragraph111,L,F5 List Paragraph,Dot pt,CV text,Medium Grid 1 - Accent 21,Numbered Paragraph,Bullets,NAST Quote,lp11,n

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 322 of 1295

~~4.12~~5.13 Database Architecture

Distinct separate Azure SQL databases will be created for each state jurisdiction to enable:

- secure segregation of jurisdictional data
- efficient management of database schema updates
- jurisdictional performance management

Figure ~~22~~ ~~:-~~ Jurisdictional Databases

Dedicated SQL Database Managed Instances will be used to suffice the heavy loads expected from larger jurisdictions with Azure SQL Database Elastic Pools employed to suffice the smaller system loads expected from smaller jurisdictions as per the following:

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 323 of 1295

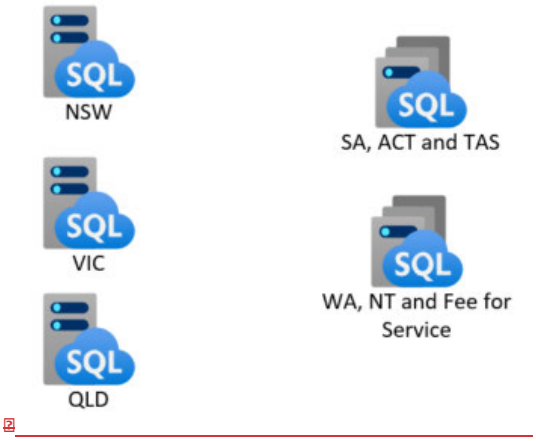


Figure 333 :- Master Database Instance Architecture

SQL Database Managed Instance offers a simplified instance-scoped programming model that is like an on-premises SQL Server instance. The databases in Managed Instance share the resources allocated to the Managed Instance, and the Managed Instance also represents the management grouping for these databases. The emphasis of this offering is on high compatibility with the programming model of on-premises SQL Server and out-of-box support for the large majority of SQL Server features and accompanying tools/services.

Azure SQL Database Elastic Pool is a shared resource model for Single Azure SQL PaaS databases to achieve higher resource utilization efficiency, and all the databases within an elastic pool share predefined resources within the same elastic pool. The emphasis of this offering is on a simplified database-scoped programming model for multi-tenant SaaS apps where the workload pattern is well defined and delivers high cost-effectiveness when serving many tenants.

5.14 Database Backup

An Azure SQL Database provides the following database backup mechanisms used for both disaster recovery and data retention purposes.

5.14.1 Short-term retention

For all new, restored, and copied databases, Azure SQL Database retains sufficient backups to allow PITR within the last 7 days by default. The service takes regular full, differential, and log backups to ensure that databases are restorable to any point in time within the retention period that's defined for the database.

Formatted: Heading 3,Level 1 - 1,Numbered Heading 3,hd3,h3,h3 sub heading,Heading 3 - old,H3,Head 3,C Sub-Sub/Italic,Head 31,Head 32,C Sub-Sub/Italic1,Head 33,C Sub-Sub/Italic2,Head 311,Head 321,C Sub-Sub/Italic11,h31,3,not in TOC,Heading 3 Char1,Para3,H31,sub-sub, Pattern: Clear

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 324 of 1295

5.14.2 Long-term retention

For SQL Database, you can configure full LTR backups for up to 10 years in Azure Blob Storage. After the LTR policy is configured, full backups are automatically copied to a different storage container weekly.

To meet various compliance requirements, you can select different retention periods for weekly, monthly, and/or yearly full backups. The frequency depends on the policy. For example, setting W=0, M=1 would create an LTR copy monthly. For more information about LTR, see Long-term retention.

You can configure long-term backup retention using the Azure portal and PowerShell for Azure SQL Database and Azure SQL Managed Instance. To restore a database from the LTR storage, you can select a specific backup based on its timestamp. The database can be restored to any existing server or managed instance under the same subscription as the original database.

5.14.3 Backup storage redundancy

By default, Azure SQL Managed Instance stores data in geo-redundant storage blobs that are replicated to a paired region. Geo-redundancy helps protect against outages that affect backup storage in the primary region. It also allows you to restore your instance to a different region in the event of a disaster.

The storage redundancy mechanism stores multiple copies of your data so that it's protected from planned and unplanned events. Those events might include transient hardware failures, network or power outages, or massive natural disasters.

Both Geo-redundant and Geo-zone-redundant storage options are available.

5.14.3.1 Geo-redundant storage (GRS)

Copies your backups synchronously three times within a single physical location in the primary region by using LRS. Then it copies your data asynchronously three times to a single physical location in the paired secondary region.

The result is:

- Three synchronous copies in the primary region.
- Three synchronous copies in the paired region that were copied over from the primary region to the secondary region asynchronously.

Formatted: Heading 3,Level 1 - 1,Numbered Heading 3,hd3,h3,h3 sub heading,Heading 3 - old,H3,Head 3,C Sub-Sub/Italic,Head 31,Head 32,C Sub-Sub/Italic1,Head 33,C Sub-Sub/Italic2,Head 311,Head 321,C Sub-Sub/Italic11,h31,3,not in TOC,Heading 3 Char1,Para3,H31,sub-sub, Pattern: Clear

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Formatted: Default Paragraph Font, Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto, Pattern: Clear

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Formatted: Default Paragraph Font, Font: (Default) Calibri, Font color: Custom Color(RGB(95,99,105))

Formatted ...

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Formatted: No bullets or numbering

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Formatted: Indent: Left: 0 cm

Formatted: Font: (Default) +Body (Calibri), Font color: Auto

Formatted

Formatted: Indent: Left: 1.27 cm, No bullets or numbering

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 325 of 1295

Figure 444: Geo-redundant storage (GRS)

5.14.4 Restore capabilities and features

You can use Azure SQL backups to:

- Restore an existing database to a point in time in the past within the retention period by using the Azure portal, Azure PowerShell, the Azure CLI, or the REST API. This operation creates a new database on either the same instance as the original database or a different instance in the same subscription and region. It uses a different name to avoid overwriting the original database. You can also use the Azure portal to restore your point-in-time database backup to an instance in a different subscription from your source instance. After the restore finishes, you can delete the original database. Alternatively, you can both rename the original database and rename the restored database to the original database name.
- Restore a deleted database to a point in time within the retention period, including the time of deletion. You can restore the deleted database to the same managed instance where the backup was taken, or another instance in the same, or different subscription to the source instance. Before you delete a database, the service takes a final transaction log backup to prevent any data loss.
- Restore a database to another geographic region. Geo-restore allows you to recover from a geographic disaster when you can't access your database or backups in the primary region. It creates a new database on any existing managed instance in any Azure region.
- Restore a database from a specific long-term backup of a database, if the database has been configured with an LTR policy. LTR allows you to restore an old version of the database by using the Azure portal, the Azure CLI, or Azure PowerShell to satisfy a compliance request or to run an old version of the application. For more information, see Long-term retention.

This table summarizes the capabilities and features of point-in-time restore, geo-restore, and long-term retention for an Azure SQL database.

Table 444 - Restore capabilities and features

Backup properties	PITR	Geo-restore	LTR
Types of SQL backup	Full, differential, log.	Replicated copies of PITR backups.	Only the full backups.

Formatted: Keep with next
Formatted: Caption, No bullets or numbering, Pattern: Clear
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Indent: Left: 0 cm, Outline numbered + Level: 1 + Numbering Style: Bullet + Aligned at: 0.63 cm + Tab after: 1.27 cm + Indent at: 1.27 cm, Tab stops: -0.37 cm, List tab + Not at 1.27 cm
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted
Formatted: Normal (Web), Pattern: Clear (White)
Formatted
Formatted: Caption, Keep with next
Formatted
Formatted Table
Formatted
Formatted: Font: +Body (Calibri), Bold
Formatted
Formatted: Font: +Body (Calibri), Bold
Formatted
Formatted: Font: +Body (Calibri), Bold
Formatted: Font: +Body (Calibri), Bold
Formatted
Formatted
Formatted
Formatted
Formatted: Font: Bold

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 326 of 1295

Recovery point objective (RPO)	10 minutes, based on compute size and amount of database activity.	Up to 1 hour, based on geo-replication.*	One week (or user's policy).
Recovery time objective (RTO)	Restore usually takes less than 12 hours but could take longer, depending on size and activity. See Recovery.	Restore usually takes less than 12 hours but could take longer, depending on size and activity. See Recovery.	Restore usually takes less than 12 hours but could take longer, depending on size and activity. See Recovery.
Retention	1 to 35 days.	Enabled by default, same as source.**	Not enabled by default. Retention is up to 10 years.
Azure storage	Geo-redundant by default. You can optionally configure zone-redundant or locally redundant storage.	Available when PITR backup storage redundancy is set to geo-redundant. Not available when PITR backup storage is zone-redundant or locally redundant.	Geo-redundant by default. You can configure zone-redundant or locally redundant storage.
Restoring a new database in the same region	Supported	Supported	Supported
Restoring a new database in another region	Not supported	Supported in any Azure region	Supported in any Azure region
Restoring a new database in another subscription	Supported	Not supported***	Not supported***
Restoring via Azure portal	Yes	Yes	Yes
Restoring via PowerShell	Yes	Yes	Yes

Formatted ...

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 327 of 1295

Restoring via Azure CLI	Yes	No	Yes
Backup properties	PITR	Geo-restore	LTR

* For business-critical applications that require large databases and must ensure business continuity, use auto-failover groups.

** All PITR backups are stored on geo-redundant storage by default, so geo-restore is enabled by default.

*** The workaround is to restore to a new server and use Resource Move to move the server to another subscription.

5.15 Database Archiving

As a database experiences continued growth throughout its lifetime an increase in costs and a decrease in performance can be expected. A data archiving strategy is required to maintain data volume at an acceptable level in the operational datastore (i.e., the Master database).

Historical data that has been archived will still be available however, the time required to access the data will be increased.

5.15.1 Hot, cool, and archive access tiers

Archive access tiers can be broadly defined as Hot, Cool and Archived data.

5.15.1.1 Hot data access

Hot data has the quickest access times possible as per the system's specifications and is the most expensive. Hot data is needed for high demand day to day business-as-usual activities.

5.15.1.2 Cool data access

Cool data has acceptable data access times but not as performant as 'Hot' storage. Cool data access times are acceptable for less frequent or scheduled activities.

5.15.1.3 Archived data access

Archived data is used for long term storage with lengthy data access times.

5.15.2 Updating of Historical Records

Historical data will need to be updated within the STARS Master Database. Therefore, any data that has been archived out of the Master database will require the ability to be re-inserted into back into the Master database to support any update operations that need to be made upon it due to a resubmission of data from a Training Organisation or the like after an extended period has passed.

Formatted: Font: (Default) +Body (Calibri), Font color: Auto
Formatted: Font: (Default) +Body (Calibri), Font color: Auto
Formatted: Font: (Default) +Body (Calibri), Font color: Auto
Formatted: Font: (Default) +Body (Calibri), Font color: Auto
Formatted: Font: Bold
Formatted: Font: (Default) +Body (Calibri), Font color: Auto
Formatted: Font: (Default) +Body (Calibri), Not Bold, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), Not Bold, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), Not Bold, Font color: Auto
Formatted: Font: Bold
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Normal
Formatted: Heading 3,Level 1 - 1,Numbered Heading 3,hd3,h3,h3 sub heading,Heading 3 - old,H3,Head 3,C Sub-Sub/Italic,Head 31,Head 32,C Sub-Sub/Italic1,Head 33,C Sub-Sub/Italic2,Head 311,Head 321,C Sub-Sub/Italic11,h31,3,not in TOC,Heading 3 Char1,Para3,H31,sub-sub
Formatted: Heading 4,Level 2 - a,h4,Heading 4 Char Char Char,1.1 Heading,Fourth Level,sub-sub-sub para,h4 sub sub heading,h41,h42,Para4,H4,Sub Sub section,4,(Alt+4),H41,(Alt+4)1,H42,(Alt+4)2,H43,(Alt+4)3,H44,(Alt+4)4,H45,(Alt+4)5,H411,(Alt+4)11,H421,(Alt+4)21,

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 328 of 1295

This will be achieved by retrieving the most recent historical record from the 'Main' table's (this is where STARS functionality manipulates data) associated Temporal Table (historical data capture version of the table) and then inserting the historical record back into the 'Main' table so it can be updated with the re-submission data.

The 'Main' tables and their associated Temporal Tables can be independently horizontally partitioned and indexed. As such, partitions of historical data (defined as per a sliding window of time) can be deleted from the 'Main' tables while all historical data (and its associated history of change) will be retained in the Temporal Tables.

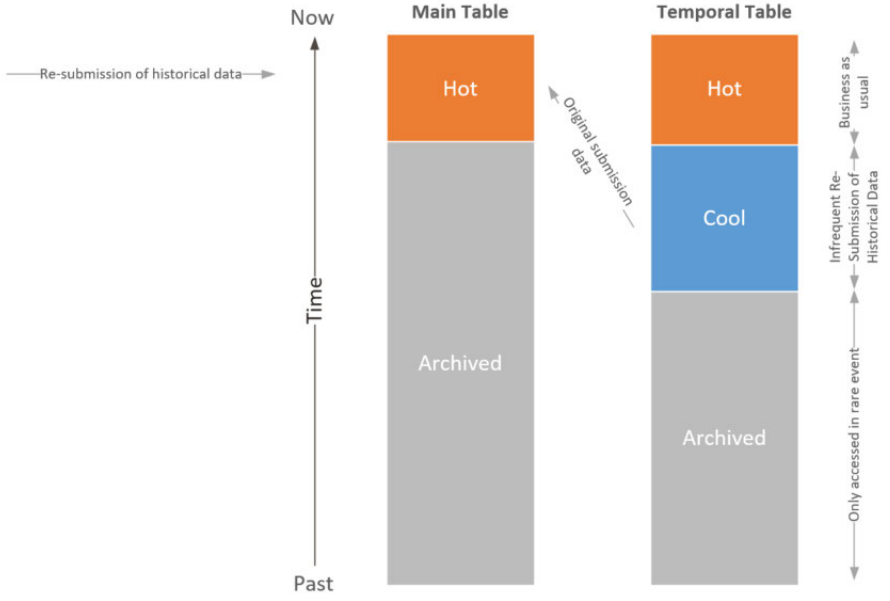


Figure 555: Data Archiving Tiers

Historical data stored in Temporal Tables will also be archived out of the Master database into an 'Archived' state once it is no longer considered active.

5.15.3 Table Partitioning

Table partitioning can make large tables more manageable and scalable. Using the table partitioning approach, you can use table partitions to implement custom data clean up or offline archival based on a time condition. Table partitioning will also give you performance benefits when querying temporal tables on a subset of data by using partition elimination.

Formatted: Keep with next

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 329 of 1295

With table partitioning, you can implement a sliding window approach to move out the oldest portion of historical data from the table and keep the size of the retained part constant in terms of age - maintaining data in the table equal to required retention period.

You can create a partitioned table or index in SQL Server horizontally divided into units that can be spread across more than one filegroup in a database, or stored in a single filegroup. Partitioning can make large tables and indexes more manageable and scalable.

Creating a partitioned table or index typically happens in three or four parts:

1. Optionally create a filegroup or filegroups and corresponding data files that will hold the partitions specified by the partition scheme. The main reason to place partitions on multiple filegroups is to ensure you can independently perform backup and restore operations on filegroups. If this is not required, you may choose to assign all partitions to a single filegroup, using either an existing filegroup, such as PRIMARY, or a new filegroup with related data file(s). In nearly all scenarios, you will achieve all benefits of partitioning whether or not you use multiple filegroups.
2. Create a partition function that maps the rows of a table or index into partitions based on the values of a specified column. You can use a single partition function to partition multiple objects.
3. Create a partition scheme that maps the partitions of a partitioned table or index to one filegroup or to multiple filegroups. You can use a single partition scheme to partition multiple objects.
4. Create or alter a table or index and specify the partition scheme as the storage location, along with the column that will serve as the partitioning column.

5.15.4 Cool data storage

The filegroups created as part of the partitioning scheme for a Temporal Table will be stored separately on different data access tiers (see Disk Types). Data belonging to partitions which are no longer actively being updated as part of business-as-usual activities will be moved to the filegroup that resides on 'Cool' tier storage as the data becomes older as per the associated partitioning scheme.

This 'Cool' dataset of historical records stored in Temporal Tables will be used to source the original version of the re-submission that needs updating but no longer resides in the 'Main' tables' 'Hot' storage (see Updating of Historical Records).

5.15.5 Disk Types

Azure managed disks currently offers five disk types, each intended to address a specific customer scenario.

Data belonging to the 'Hot' tier will be contained in filegroups stored on 'Ultra disk' where as 'Cool' tier filegroups will be stored on 'Standard SSD' and 'Archived' filegroups stored on 'Standard HDD'.

Table 555 - Database Archiving - Disk Types

	Ultra disk	Premium SSD v2	Premium SSD	Standard SSD	Standard HDD

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Default Paragraph Font, Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), Font color: Auto, Pattern: Clear
Formatted: Caption, Keep with next
Formatted: Font: (Default) +Body (Calibri), Not Bold, Font color: Auto
Formatted: Font: +Body (Calibri)
Formatted: Font: (Default) +Body (Calibri), Not Bold, Font color: Auto
Formatted: Font: (Default) +Body (Calibri), Not Bold, Font color: Auto
Formatted: Font: +Body (Calibri)
Formatted ...
Formatted: Font: +Body (Calibri)
Formatted ...
Formatted: Font: +Body (Calibri)
Formatted: Font: +Body (Calibri), Font color: Auto
Formatted: Font: +Body (Calibri)
Formatted Table

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 330 of 1295

Disk type	SSD	SSD	SSD	SSD	HDD
Scenario	IO-intensive workloads such as SAP HANA, top tier databases (for example, SQL, Oracle), and other transaction-heavy workloads.	Production and performance-sensitive workloads that consistently require low latency and high IOPS and throughput	Production and performance sensitive workloads	Web servers, lightly used enterprise applications and dev/test	Backup, non-critical, infrequent access
Max disk size	65,536 gigabytes (GiB)	65,536 GiB	32,767 GiB	32,767 GiB	32,767 GiB
Max throughput	4,000 MB/s	1,200 MB/s	900 MB/s	750 MB/s	500 MB/s
Max IOPS	160,000	80,000	20,000	6,000	2,000, 3,000*
Usable as OS Disk?	No	No	Yes	Yes	Yes

5.15.6 Removal of Historical Data

Where data residing the 'Archived' tier has been deemed static, data can be physically deleted from the 'Main' tables. This data will be archived into a full database back up prior to any deletion activity occurring and stored on an 'Archived' storage tier. This database back up will be able to be restored as a copy of the original database where any required historical data can be retrieved.

It is worth noting that historical data required for reporting purposes will be available in the 'R & A Data Warehouse'.

5.16 Database Availability

Azure SQL Database now offers a 99.995% availability SLA for zone redundant databases in its business-critical tier. This is the highest SLA in the industry among all relational database services. It is also backed by up to a 100% monthly cost credit for when the SLA is not maintained. Secondly, it offers a business continuity SLA for databases in the business-critical tier that are geo-replicated between two different Azure regions. That SLA comes with very strong guarantees of a five second recovery point objective (RPO) and a 30 second recovery time objective (RTO), including a 100% monthly cost credit when the SLA is not maintained. Azure SQL Database is the only relational database service in the industry offering a business continuity SLA.

Formatted …

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 331 of 1295

Table 666 - Azure SQL Availability

Platform	Availability		Business continuity			
-	Uptime	Max Credit	RTO	Max Credit	RPO	Max Credit
Azure SQL Database	99.995%	100%	30 seconds	100%	5 seconds	100%

Formatted: Caption, Keep with next
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted Table
Formatted: Font: Bold

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 332 of 1295

~~5~~6 Requirements

~~5.1~~6.1 Functional

Table ~~77~~73 - Functional Requirements

FR No	Requirement Name	Description	~~Acceptance Criteria~~ Traceability
FR015	Store data in Master DB	The VDS system must store successfully validated and quality assured Data submitted by a training organisation in the Master Database.	See Feature Overview section.
FR032b		The VDS system must ensure that the appropriate data access controls (Role and Attribute Based Access Control) are implemented in the Master Database. This includes the ability for a STA to manage access via the Systems Operator: • Based on data set type (i.e. aggregated and de-identified data sets, financial data, national core data, state based program data, and student and individual enrolment data) • Based on assigned user groups	See Access Management section.

FR No	Requirement Name	Description	~~Acceptance Criteria~~ Traceability

~~5.~~26.2 Non-Functional

Table ~~88~~84 - Non-Functional Requirements

FR No	Requirement Name	Description	Traceability
NFR002a	Minimum Uptime	The VDS System will support continual operations by being available and functioning with minimal downtime (High Availability). • The VDS System will support a minimum proposed uptime for the users, excluding scheduled downtime or outage	See Database Availability section.
NFR002d	Production environment uptime	The VDS System, including all user interfaces, in the Production environment is available 99.86% of the scheduled hours of availability, measured over a month. The system is scheduled to be available for all hours, except those in approved change windows (i.e., planned outages). This equates to 60 minutes of unavailability in a 30-day month with no planned outages.	See Database Availability section.

Formatted: Font: Bold

Formatted Table

FR No	Requirement Name	Description	Traceability
NFR002e	Non-Production environments uptime	The VDS System, including all user interfaces, in environments other than Production (per NFR11), is available 99.79% of the scheduled hours of availability, measured over a month. The system is scheduled to be available for all hours, except those in approved change windows (i.e., planned outages). This equates to 90 minutes of unavailability in a 30-day month with no planned outages.	See Database Availability section.
NFR002i	Overnight extraction	The VDS system will be available for data extraction overnight	See Database Availability section.
NFR015a	Secondary site operation	The VDS System will support business continuity, such that if there is a disaster at the primary location, the VDS System can continue operations from a secondary site for business continuity purposes.	See Database Availability section.
NFR016a	Recovery Mechanisms	The VDS System will support disaster recovery for all hosting services and systems. Three major recovery mechanisms must include Geo-Replication, Geo-Recovery, and Geo-Restore, to warrant that fail over of primary services and/or databases to a secondary	See Database Backup section.

Formatted: Font: Bold

Formatted Table

Formatted: Font: Bold

Formatted: Font: Bold

Formatted: Font: Bold

Formatted: Font: Bold

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Description	Traceability
		counterpart in the event of a datacentre or regions downtime: • During a failover process, primary services and databases must fail over to secondary services and databases in different regions. • Geo-Restore must be enabled to recover a database from automatically maintained geo-redundant backups. • Repatriation and Geo-Replication must be used to repatriate updated databases to the primary database in their original regions.	
NFR016b	Defined tier levels for disaster recovery	The VDS System will support off-site recoverability based on a defined tier level for each of the VDS System's core functions, e.g. to support transaction integrity and data loss within the target RPO (per NFR17).	See Database Backup section.
NFR016c	Maximum allowable outage (MAO)	The maximum allowable outage (MAO) in the event of a disaster for all system components is **five (5)** hours. Within this the execution of the recovery for the IT systems should take no longer than **two (2)** hours to execute, which allows **three (3)** hours for the declaration of a disaster	See Database Availability section.
NFR017a	Offline Backups	The VDS System will support data recovery and retention by storing secure offline backups in a remote location in	See Database Backup section.

Formatted: Font: Bold

Formatted Table

Formatted: Font: Bold

Formatted: Font: Bold

Formatted: Font: Bold

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Description	Traceability
		accordance with the data classification 'UNCLASSIFIED DLM: SENSITIVE-PERSONAL'.	
NFR017b	Recovery time objective (training organisation)	In the event of a system crash, the Recovery Time Objective will be within 4 hrs from the time of the system crash.	See Database Backup section.
NFR017c	Recovery point objective (RPO)	In the event of a system crash, the Recovery Point Objective will be the most recent 5-minute checkpoint before the system crash.	See Database Backup section.
NFR017d	Daily backups	All server images and system data will be backed up on at least a daily basis.	See Database Backup section.
NFR017e	Backup retention period	A full monthly backup must be retained securely for a period of seven (7) years.	See Database Backup section.
NFR017f	Audit log retention	The retention of audit logs must comply with the National Archives Australia (NAA) Administrative Functions Disposal Authority of a minimum of seven (7) years.	See Database Backup section.
NFR018a	Data storage	The VET data stored in the VDS System will be retained indefinitely.	See Database Backup section.
NFR018b	Access to data retained	The **seven (7)** number of years data is retained in the VDS system can be modifed, in order to reflect industry policy and subject to the data amemdment policy.	See Database Backup section.

Formatted: Font: Bold
Formatted Table
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

~~6~~7 Feature Overview

~~6.1~~7.1 Logical Data Model

The master data model in the Master database is based on the VET IS Logical Schema as shown at a high level below. For a detailed version of the below model ~~see~~ please reference related the ~~VET Information Standard Logical Data Model~~VET Information Standard Logical Data Model ~~in the appendix of this document.~~ listed in the Related Documents section of this document.

Formatted: Default Paragraph Font

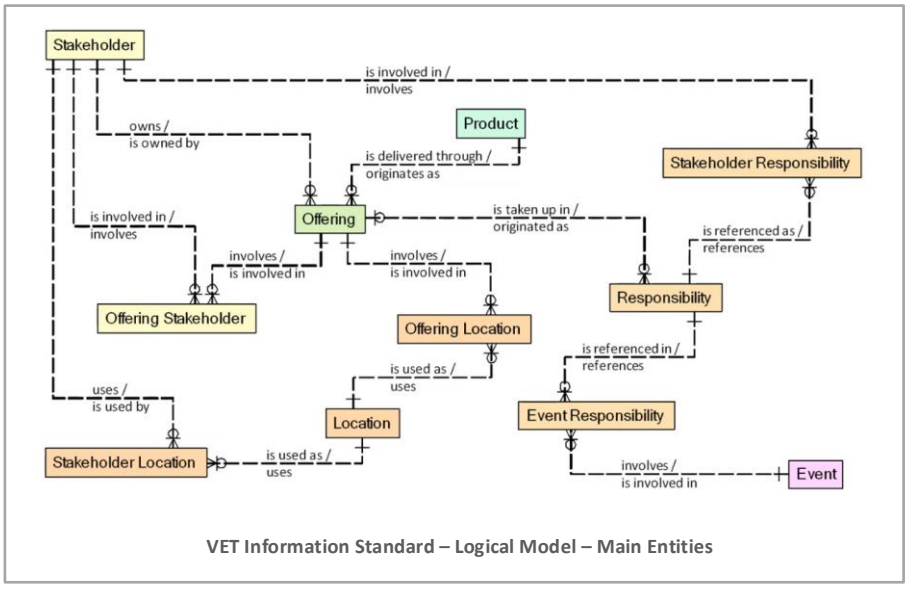


Figure ~~66~~64 ~~:-~~ VET IS Logical Data Model high level

The master data model in STARS includes the following subject areas/logical entities that's mastered in the Azure SQL Database.

- Stakeholder (representing people, organisations. NB: employers will be covered in future releases)
- Location (representing addresses, virtual spaces)
- Product (representing programs, subjects)
- Offering (representing a product offered by a training organisation)
- Responsibility (representing ~~many things, including~~ enrolments, arrangements, placements~~enrolments, training contracts, financial arrangements~~)
- Event (representing a business event, but not lifecycle events – currently not in R1)

~~6.2~~7.2 External Identifiers

The VET IS data model makes provision for the linkage of external identifiers through columns that have a logical 'Domain Name Type' of "Reference Number" scattered throughout the logical (and

later physical) schema. For example, the Stakeholder Identifier table. This table will hold items such as USI, SMS student id, STA (national) client ID etc.

These external identifiers will be required to determine whether an entity does not exist in the Master database and therefore requires an initial insert operation as opposed to performing an update on an existing mastered entities' associated records~~ty~~. Once an entity has been identified via an external identifier then its' surrogate key (SK) (a unique identifier defined within the STARS Master database) can be retrieved and used for more efficient data retrieval.

The below tables do not describe any intermediary navigation of relationship tables required to link external identifiers where a composite key needs to be employed.

Table 9995 - Stakeholder External Identifiers

Entity	Table	Column
Person	Stakeholder Identifier	Stakeholder Identifier Reference Number Where Stakeholder identifier type code='Student external identifier'
Person	Stakeholder Identifier	Stakeholder Identifier Reference Number Where Stakeholder identifier type code='Training organisation external identifier'
Organisation	Stakeholder Identifier	Stakeholder Identifier Reference Number Where Stakeholder identifier type code='Training organisation external identifier'
Government	Stakeholder Identifier	Stakeholder Identifier Reference Number Where Stakeholder identifier type code='Government external identifier'

Table 1010106 - Responsibility External Identifiers

Entity	Table	Column
Contract	Responsibility Identifier	Responsibility Identifier Type Reference Number Where Responsibility identifier type code=' Training contract external identifier'
Contract	Stakeholder Identifier	Stakeholder Identifier Reference Number

		Where Stakeholder identifier type code='Training organisation external identifier'
Program enrolment	Program Enrolment	Responsibility Identifier Type Reference Number Where Responsibility identifier type code='Program Enrolment external identifier'
Program enrolment	Stakeholder Identifier	Stakeholder Identifier Reference Number Where Stakeholder identifier type code='Training organisation external identifier'
Subject enrolment	Subject Enrolment	Responsibility Identifier Type Reference Number Where Responsibility identifier type code='Subject Enrolment external identifier'
Subject enrolment	Stakeholder Identifier	Stakeholder Identifier Reference Number Where Stakeholder identifier type code='Training organisation external identifier'
Work Placement	Responsibility Identifier	Responsibility Identifier Type Reference Number Where Responsibility identifier type code='Work placement external identifier'
Work Placement	Stakeholder Identifier	Stakeholder Identifier Reference Number Where Stakeholder identifier type code='Training organisation external identifier'
Subsidy Funding Arrangement	Responsibility Identifier	Responsibility Identifier Type Reference Number Where Responsibility identifier type code='Subsidy funding arrangement external identifier'
Subsidy Funding Arrangement	Stakeholder Identifier	Stakeholder Identifier Reference Number Where Stakeholder identifier type code='Training organisation external identifier'

Table 1111117 - Product External Identifiers

Entity	Table	Column

Product	Program	Product Reference Number
Product	Subject	Product Reference Number

Table 1212~~128~~ - Responsibility Location External Identifiers

Entity	Table	Column
Subject Delivery Location	Responsibility Location	Responsibility Location Reference Number
Subject Delivery Location	Stakeholder Identifier	Stakeholder Identifier Reference Number Where Stakeholder identifier type code='Training organisation external identifier'
Student learning	Responsibility Location	Responsibility Location Reference Number
Student learning	Stakeholder Identifier	Stakeholder Identifier Reference Number Where Stakeholder identifier type code='Training organisation external identifier'

The content of Stakeholder Locations of the following types constitutes an external business identifier (i.e., an email address or street address should be unique in its' own right). Navigation of Stakeholder Locations can be performed via the parent Stakeholder entity and uniqueness is defined by the Stakeholder and the associated 'Stakeholder location usage type code' as per the following table:

Table 1313~~139~~: Stakeholder location usage type code

Type Code	Value
Stakeholder location usage type code	Permanent
Stakeholder location usage type code	Postal
Stakeholder location usage type code	Term address
Stakeholder location usage type code	Electronic

Stakeholder location usage type code	Telecommunication
Stakeholder location usage type code	Training organisation head office
Stakeholder location usage type code	Employer head office
Stakeholder location usage type code	Employer workplace

A Subject Offering does not have an external identifier associated with it as part of the VET IS so the following composite key of the related 'Training Organisation' and 'Subject' is required.

Table 14141410 - ~~Product~~ Offering External Identifiers

Dependencies	Table	External Identifier Attribute	Filter
Organisation	Stakeholder Identifier	Stakeholder Identifier Reference Number	Where Stakeholder identifier type code='Training organisation external identifier' or 'School external identifier'
Subject	Subject	Product Reference Number	

~~6.3~~7.3 Relationship Entities

Relationship tables are use throughout the VET IS to associate Stakeholder, Responsibility and Location entities with each other. They are flexible in design, data driven and given meaning by the set of classification codes used to define them. When future releases of the VET IS are implemented this set of classification codes will be expanded to enable new relationships between new entity sub types.

~~6.3.1~~7.3.1 Stakeholder

~~6.3.1.1~~7.3.1.1 Stakeholder relationship

The Stakeholder relationship table is used to link associated Stakeholder entity types (Stakeholder type code) with each other via their Surrogate Keys (SK) (i.e., Stakeholder SKey to Related Stakeholder Skey). One row in this table represents a relationship between two Stakeholder entities (i.e., Person to an Organisation).

Table 15151511 - Stakeholder Relationship - Stakeholder type code

Type Code	Value
Stakeholder type code	Person
Stakeholder type code	Organisation

Stakeholder type code	Government

Each Stakeholder is assigned a role as part of the relationship using Stakeholder Role Type Code and Related Stakeholder Role Type Code (i.e., Student of Enrolled school). In this example this gives both the Person and Organisation entities further meaning in the context of the relationship.

Table 16161612 - Stakeholder Relationship - Stakeholder role type code

Type Code	Value
Stakeholder role type code	Employer
Stakeholder role type code	Student
Stakeholder role type code	Delivery provider
Stakeholder role type code	Primary organisation
Stakeholder role type code	Enrolled school
Stakeholder role type code	Provider of funding

The relationship itself is assigned meaning using the Stakeholder relationship type code attribute.

Table 17171713 - Stakeholder Relationship - Stakeholder relationship type code

Type Code	Value
Stakeholder relationship type code	Most recent school

Therefore, this example defines “the most recent school that a student is enrolled in”.

6.3.27.3.2 Responsibility

6.3.2.17.3.2.1 Responsibility relationship

The Responsibility Relationship table defines relationships between Responsibility entities of type:

Table 18181814 - Responsibility Relationship - Responsibility type code

Type Code	Value
Responsibility type code	Enrolment
Responsibility type code	Contract
Responsibility type code	Program enrolment
Responsibility type code	Subject enrolment

Responsibility type code	Funding Arrangement
Responsibility type code	Subsidy Funding Arrangement

Having relationships of type:

Table 19191~~915~~ - Responsibility Relationship - Responsibility relationship type code

Type Code	Value
Responsibility relationship type code	Subject in a program
Responsibility relationship type code	Superseded by
Responsibility relationship type code	Transferred into
Responsibility relationship type code	LLN Support for

~~6.3.2.2~~7.3.2.2 Stakeholder responsibility

The Stakeholder Responsibility table defines relationships between Responsibility and Stakeholder entities of type:

Table 2020~~2016~~ - Stakeholder Responsibility - Stakeholder role type code

Type Code	Value
Stakeholder role type code	Employer
Stakeholder role type code	Student
Stakeholder role type code	Delivery provider
Stakeholder role type code	Primary organisation
Stakeholder role type code	Enrolled school
Stakeholder role type code	Provider of funding

Having relationships of type:

Table 2121~~2117~~ - Stakeholder Responsibility - Stakeholder responsibility type code

Type Code	Value
Stakeholder responsibility type code	Student enrolment
Stakeholder responsibility type code	Provider of training
Stakeholder responsibility type code	Is employed by

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 345 of 1295

Stakeholder responsibility type code	Provider of assessment

For instance, Stakeholder 'Student' has a 'Student enrolment' relationship with Responsibility 'Subject enrolment'.

~~6.3.3~~7.3.3 Location

~~6.3.3.1~~7.3.3.1 Location Relationship

The Location ~~Responsibility~~ Relationship table defines relationships between Locations. Location Relationship Type Codes have yet to be defined by the client.

Table 222~~22~~ - Location Relationship - Location Relationship Type Code

Formatted: Caption, Keep with next

Type Code	Value
Location Relationship Type Code	Physical address of venue - Campus

~~6.3.3.2~~7.3.3.2 Stakeholder Location

The Stakeholder Location table defines relationships between Stakeholder and Location entities of type:

Table 232~~32~~318 - Stakeholder Location - Stakeholder location usage type code

Type Code	Value
Stakeholder location usage type code	Permanent
Stakeholder location usage type code	Postal
Stakeholder location usage type code	Term address
Stakeholder location usage type code	Electronic
Stakeholder location usage type code	Telecommunication
Stakeholder location usage type code	Training organisation head office
Stakeholder location usage type code	Employer head office
Stakeholder location usage type code	Employer workplace

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 346 of 1295

~~6.3.3.3~~7.3.3.3 Responsibility Location

The Responsibility Location table defines relationships between Locations and Responsibilities. Responsibility Location Usage Type Codes have yet to be defined by the client.

Table 242~~424~~ - Responsibility Location - Responsibility Location Usage Type Code

Type Code	Value
Responsibility Location Usage Type Code	Student learning
Responsibility Location Usage Type Code	Trainer delivery

Formatted: Caption, Keep with next

Formatted Table

~~6.4~~7.4 Navigation of the Data Model

Navigation of the VET IS Data Model is performed by:

- Step 1 - Retrieving the driving entities' STARS Surrogate Key (SK) by using its' associated External Identifier.
- Step 1a – Where a composite key defines the uniqueness of an entity (i.e., for a Student both the related Student and associated Training Organisation surrogate keys need to be retrieved) this step will need to be repeated for each key.
- Step 2 - Using the surrogate keys retrieved information relating to an entity can be retrieved as per the 'Concordance Mapping Design Verification 24 Nov 22 - DEWR feedback R1_V1.0' and 'DRAFT VET Information Standard Logical Interchange Specification V2 (1) R1_V1.0' documents.
- Step 2a - Where a composite key defines the uniqueness of an entity then a 'Relationship' table (i.e., Stakeholder Relationship, Responsibility Relationship, Stakeholder Responsibility, Stakeholder Location and Responsibility Location) will need to be referenced to retrieve relationships that exist between surrogate keys retrieved as part of Step 1a. Rows may need to be reduced as part of this step as the uniqueness of external identifiers can only be assumed within a particular scope (i.e., the same external identifier of '123' may be in use across multiple Training Organisations).

~~6.5~~7.5 Physical Data Model

The materialisation VET IS Logical Data Model is defined in the STARS Detailed Design - Master DB Physical ERD - V0.1.pdf document and can be summarised as follows:

The Stakeholder subject area has been materialised as per the following Stakeholder types with satellite attribute tables associated where relevant to that type.

Table 25252519 - Physical DB - Stakeholder Type Code

Type Code	Value
Stakeholder type code	Person
Stakeholder type code	Organisation
Stakeholder type code	Government

The Responsibility subject area has been materialised as per the following Responsibility types with satellite attribute tables associated where relevant to that type.

Table 26262620 - Physical DB - Responsibility Type Code

Type Code	Value
Responsibility type code	Program enrolment
Responsibility type code	Subject enrolment
Responsibility type code	Subsidy Funding Arrangement
Responsibility type code	Work Placement

The Product subject area has been materialised as per the following Product types with satellite attribute tables associated where relevant to that type.

Table 27272721 - Physical DB - Product Type Code

Type Code	Value
Product type code	Program
Product type code	Subject

The Offering subject area has been materialised as per the following Offering types with satellite attribute tables associated where relevant to that type.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 348 of 1295

Table 28282822 - Physical DB - Offering Subject Area

Subject Area	Entity
Offering	Subject Offering

The Location subject area has been materialised as per the following Location types and associated Address sub-types with satellite attribute tables associated where relevant to that type.

Table 29292923 - Physical DB - Location Type Code

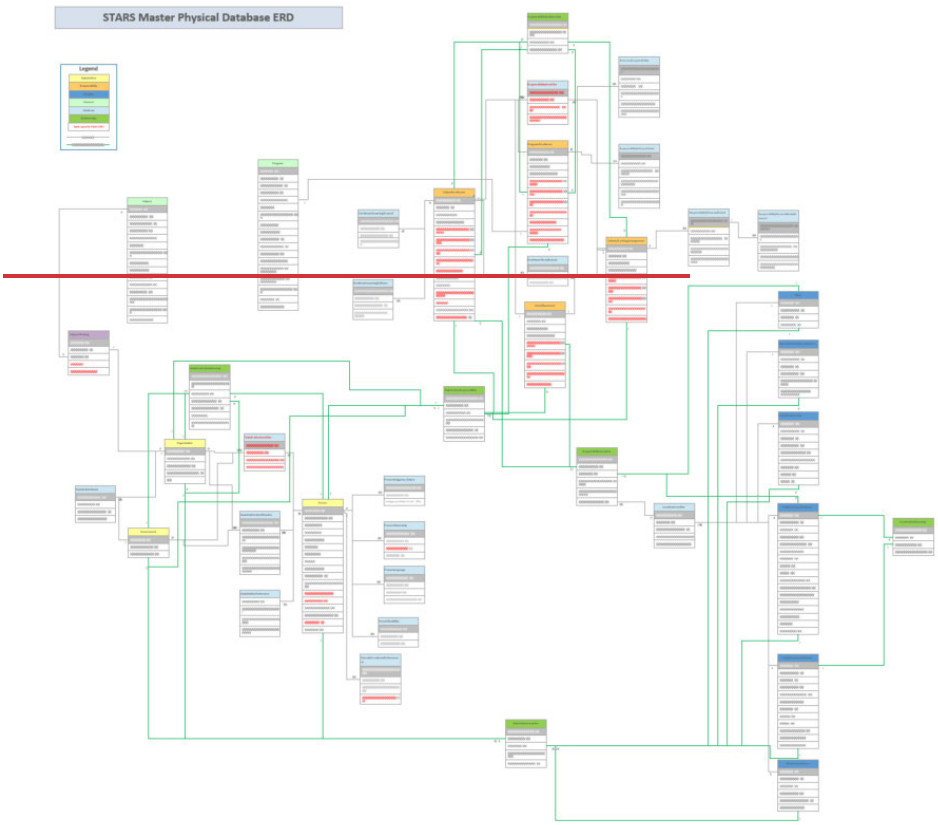
Type Code	Value
Location type code	Address
Location type code	Place

Table 30303024 - Physical DB - Address Type Code

Type Code	Value
Address type code	Electronic address
Address type code	Post office address
Address type code	Fully structured address
Address type code	Semi structured address
Address type code	Telecommunications address

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 349 of 1295

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 350 of 1295



Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 351 of 1295

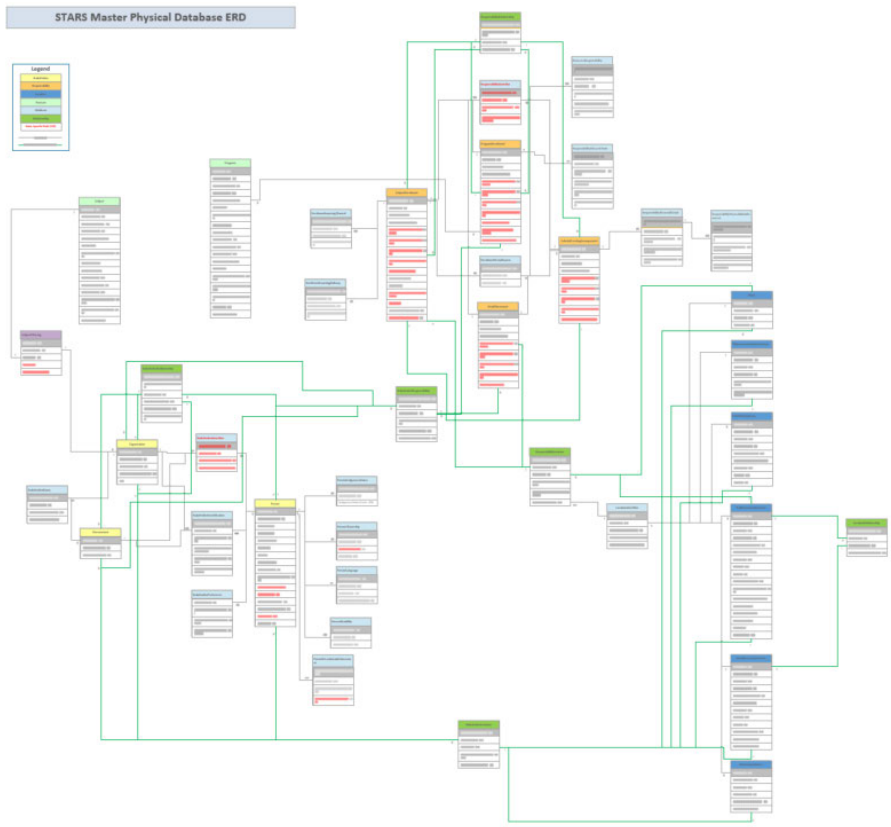


Figure 7775 :- STARS Master DB Physical ERD
Please the STARS Detailed Design - Master DB Physical ERD - V0.1.pdf document for a full-sized scalable version.

Please reference the related document, STARS Detailed Design - Master DB Physical ERD – Ref Data Subject Areas V0.1.pdf for a full-sized scalable version.

~~6.6~~7.6 Physical Data Types

The following logical Domain Names have been defined along with their associated data types including the maximum data length required.

All physical data types will conform to the below types and a mapping between logical attributes and physical column names can be made by referencing the IS Prod Technical Dictionary_01092022_R1_V1.0.xlsx document.

Table 31313~~125~~ - Physical Data Types

DOMAIN NAME	DATA TYPE
ACount	INTEGER
AFinancialAmount	~~VARCHAR~~NVARCHAR(50)
AnIndicator	CHAR(1)
ASurrogateKey	~~UNIQUEIDENTIFIER~~BIGINT
Code	~~VARCHAR~~NVARCHAR(25)
Date	DATE
Description	~~VARCHAR~~NVARCHAR(500)
JustDate	DATE
Name	~~VARCHAR~~NVARCHAR(100)
Number	INTEGER
Quantity	INTEGER
ReferenceNumber	~~VARCHAR~~NVARCHAR(50)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 353 of 1295

~~7~~8 API Design

Master Data API service will utilize Open Data Protocol (OData). This is a data access protocol built on core protocols like HTTP and commonly accepted methodologies like REST for the web.

For more detailed basic and extended features please refer to the OData documentation.
https://www.odata.org/documentation/

~~7.1~~8.1 Master Data Service API Overview

The Master Data Service is the core data repository for STARS. It manages and provides VET-IS data, held in VET-IS Logical Schema format within the Master Database, hosted in Azure SQL. It is the source of truth for all Stakeholders, Products, Offerings, Responsibilities, along with data lineage. All master data will be traceable back through release, staging and submission.

8

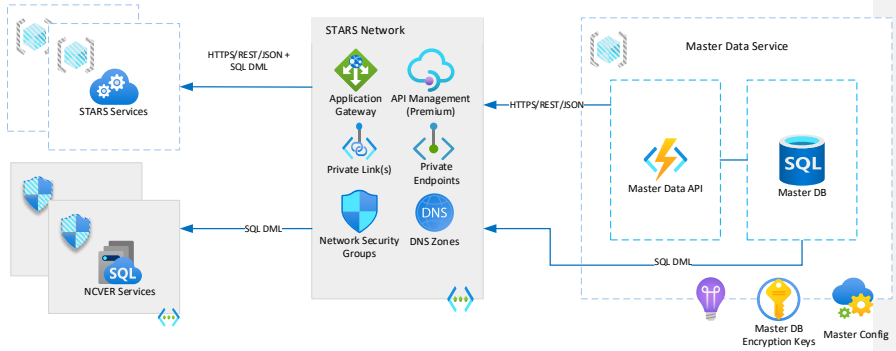


Figure ~~8~~8~~8~~6 : Master Data Service Overview

The master data service api will provide a read only interface for the VET-IS Data Packet entities. The entities shall be exposed via typed classes that would be accessible to all the Star's services via standardized code libraries.

~~7.2~~8.2 API Consumers

The following services have been identified as the consumers for the Master Data Service Api's.

- STAR's Core Services
- Master Data Management Portal

This service can be extended to other external systems in future depending on the specific use case scenarios. Any extension of usage shall need to be re-evaluated depending upon platform usage metrics. This may require any infrastructure design changes or upgrades depending on the demand.

~~7.3~~8.3 API Security and Data Segregation

The data repositories for the States and National datasets will be segregated using separate databases. The data sources shall be secured applying necessary database controls. The API endpoints for the states and national data sources shall be separated via separate function apps. These function apps will have unique managed identities using which their access to the data sources will be controlled. This model is being used to address the requirement of fine-grained control to different states data.

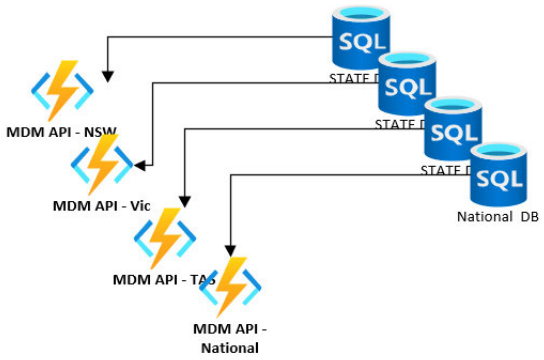


Figure ~~9997~~ : Data Segregation and Access Security

~~7.4~~8.4 API Service Limits

This section will be updated once the system once the activities related to capacity planning are over.

~~7.5~~8.5 Requesting Data (examples)

OData services support requests for data via HTTP GET requests.

Requesting Entity Collections

The request below returns the collection of Person People.

GET serviceRoot/Student

Response Payload

```
{
    "@odata.context": "serviceRoot/$metadata#Student",
    "@odata.nextLink": "serviceRoot/Student?%24skiptoken=8",
    "value": [
        {
            "@odata.id": "serviceRoot/Student('russellwhyte')",
            "@odata.etag": "W/"08D1694BD49A0F11"",
            "@odata.editLink": "serviceRoot/Student('russellwhyte')",
            "UserName": "russellwhyte",
            "FirstName": "Russell",
            "LastName": "Whyte",
            "Emails": [
                "Russell@example.com",
                "Russell@contoso.com"
            ],
            "AddressInfo": [
                {
                    "Address": "187 Suffolk Ln.",
                    "City": {
                        "CountryRegion": "United States",
                        "Name": "Boise",
                        "Region": "ID"
                    }
                }
            ],
            "Gender": "Male",
            "Concurrency": 635404796846280400
        },
        ......
        ,
        {
            "@odata.id": "serviceRoot/Student('keithpinckney')",
            "@odata.etag": "W/"08D1694BD49A0F11"",
            "@odata.editLink": "serviceRoot/Student('keithpinckney')",
            "UserName": "keithpinckney",
            "FirstName": "Keith",
            "LastName": "Pinckney",
            "Emails": [
                "Keith@example.com",
                "Keith@contoso.com"
            ],
            "AddressInfo": [],
            "Gender": "Male",
            "Concurrency": 635404796846280400
        }
    ]
}
```

Requesting an Individual Entity by ID

The request below returns an individual entity of type Person by the given UserName "russellwhyte"

GET serviceRoot/Student('russellwhyte')

Response Payload


```
{
    "@odata.context": "serviceRoot/$metadata#Student/$entity",
    "@odata.id": "serviceRoot/Student('russellwhyte')",
    "@odata.etag": "W/"08D1694BF26D2BC9"",
    "@odata.editLink": "serviceRoot/Student('russellwhyte')",
    "UserName": "russellwhyte",
    "FirstName": "Russell",
    "LastName": "Whyte",
    "Emails": [
        "Russell@example.com",
        "Russell@contoso.com"
    ],
    "AddressInfo": [
        {
            "Address": "187 Suffolk Ln.",
            "City": {
                "CountryRegion": "United States",
                "Name": "Boise",
                "Region": "ID"
            }
        }
    ],
    "Gender": "Male",
    "Concurrency": 635404797346655200
}
```


Requesting an Individual Property

To address an entity property clients append a path segment containing property name to the URL of the entity. If the property has a complex type, properties of that value can be addressed by further property name composition.

GET serviceRoot/School('JCU')/Name

Response Payload


```
{
    "@odata.context": "serviceRoot/$metadata#Schools('JCU')/Name",
    "value": "James Cook University"
}
```


To get a property value of a complex type. The request below returns the Address of the complex type Location in a School.

Response Payload


```
{
"@odata.context": "serviceRoot/$metadata#School('JCU')/Location/Address",
"value": "1/14-88 McGregor Rd, Smithfield QLD 4878"
}
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 357 of 1295

Requesting an Individual Property Raw Value

To address the raw value of a primitive property, clients append a path segment containing the string $value to the property URL. The request below returns the raw value of property *Name* of a School.

GET serviceRoot/School('JCU')/Name/$value

Response Payload

James Cook University

~~7.6~~8.6 Code Sample

Example of how to retrieve the response from an OData request to retrieve information about students in C#:

```
// Import required libraries
using System.Net.Http;
using System.Threading.Tasks;

// Set the OData service root URL and the entity set name
string serviceRoot = "https://example.com/odata/";
string entitySet = "Student";

// Construct the URL for the OData request
string url = serviceRoot + entitySet;

// Set any desired query options (optional)
// For example, to filter the results to only include students with a GPA greater than 3.5:
string queryOptions = "$filter=GPA gt 3.5";

// Create a new HttpClient to send the request
HttpClient client = new HttpClient();

// Send the OData request and retrieve the response
HttpResponseMessage response = await client.GetAsync(url + "?" + queryOptions);

// Check the response status code
if (response.IsSuccessStatusCode)
{
    // If the response was successful, extract the data from the response JSON
    string responseBody = await response.Content.ReadAsStringAsync();
    // Do something with the data, such as deserialize it into a Student object
    // For example, using the Newtonsoft.Json library:
    List<Student> students = JsonConvert.DeserializeObject<List<Student>>(responseBody);
}
else
{
    // If the response was not successful, throw an exception or handle the error in some other way
    throw new Exception("Error: OData request failed with status code " + response.StatusCode.ToString());
}
```

In this example, we're using the HttpClient class in C# to make the OData request. We first set the service root URL and the entity set name, and then construct the URL for the request by concatenating them together with the desired query options. We then create a new HttpClient object and use its GetAsync method to send the request and retrieve the response.

If the response status code is 200 (indicating success), we extract the data from the response JSON using the ReadAsStringAsync method of the HttpResponseMessage object. We then do something with the data, such as deserializing it into a list of Student objects using the JsonConvert.DeserializeObject method of the Newtonsoft.Json library.

If the response status code is not 200, we throw an exception or handle the error in some other way.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 358 of 1295

OFFICIAL

This is an example of an OData response for a "Student" using a query in a request in C#:

If you have a "Student" entity in your OData service with the following properties:

```
public class Student
{
    public int Id { get; set; }
    public string Name { get; set; }
    public int Age { get; set; }
    public string Grade { get; set; }
}
```

And you have an OData endpoint for students at https://example.com/odatawwwwwwwwwwwwwwwwwwwwwwwwwwwwww/Students, you can use the following code to query for students with ages between 18 and 25:

```
using System.Net.Http;
using Microsoft.OData.Client;
using System.Linq;

var context = new DataServiceContext(new Uri("https://example.com/odata/"));
var query = context.CreateQuery<Student>("Students").Where(s => s.Age >= 18 && s.Age <= 25);

var response = context.Execute(query) as QueryOperationResponse<Student>;

foreach (var student in response)
{
    Console.WriteLine($"Name: {student.Name}, Age: {student.Age}, Grade: {student.Grade}");
}
```

This code sends a GET request to https://example.com/odata/Students?$filter=Age ge 18 and Age le 25 to retrieve a collection of Student entities that match the filter criteria. The response is then parsed as a QueryOperationResponse<Student> object, which allows you to iterate over the collection of Student entities and access their properties. The Console.WriteLine statement prints out the name, age, and grade of each student in the collection.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 359 of 1295

~~7.7~~8.7 VET-IS Logical Packet Entities

List of entity collections (and their combinations) exposed by this API are:

- Student
- Student Location
- School
- Program Enrolment
- Delivery Location
- Training Organisation
- Program
- Program Offering
- ~~Government~~~~Delivery Location~~
-
- Subject Offering
- Subject Enrolment
- Subject
- Training Organisation Location

Please refer to **Appendix 12.2 VET-IS Data Packet Logical Entity Relationship Diagram**

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 360 of 1295

~~8~~9 Data

~~8.1~~9.1 Schemas

Table ~~323233~~226 - STARS Master Database Physical Table Schemas

SUBJECTAREA	TABLE	COLUMN	DATATYPE
Location	ElectronicAddress	LocationSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Location	ElectronicAddress	LocationTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	ElectronicAddress	CountryCode	~~VARCHAR~~NVARCHAR(25)
Location	ElectronicAddress	AddressTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	ElectronicAddress	ElectronicAddressTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	ElectronicAddress	ElectronicAddressName	~~VARCHAR~~NVARCHAR(100)
Location	LocationIdentifier	LocationIdentifierReferenceName	~~VARCHAR~~NVARCHAR(100)
Location	LocationIdentifier	LocationIdentifierReferenceNumber	~~VARCHAR~~NVARCHAR(25)
Location	LocationIdentifier	LocationIdentifierSkey	~~UNIQUEIDENTIFIER~~ BIGINT
Location	LocationIdentifier	LocationIdentifierTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	LocationIdentifier	LocationSkey	~~UNIQUEIDENTIFIER~~ BIGINT
Location	LocationRelationship	LocationRelationshipSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Location	LocationRelationship	LocationSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Location	LocationRelationship	RelatedLocationSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Location	LocationRelationship	LocationRelationshipTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	FullyStructuredAddress	LocationSKey	~~UNIQUEIDENTIFIER~~ BIGINT

Location	FullyStructuredAddress	LocationTypeCode	~~VARCHAR~~NVARCHAR(50)
Location	FullyStructuredAddress	CountryCode	~~VARCHAR~~NVARCHAR(25)
Location	FullyStructuredAddress	AddressTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	FullyStructuredAddress	PhysicalAddressTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	FullyStructuredAddress	BuildingPropertyName	~~VARCHAR~~NVARCHAR(100)
Location	FullyStructuredAddress	SuburbName	~~VARCHAR~~NVARCHAR(100)
Location	FullyStructuredAddress	StateCode	~~VARCHAR~~NVARCHAR(25)
Location	FullyStructuredAddress	Postcode	~~VARCHAR~~NVARCHAR(25)
Location	FullyStructuredAddress	StreetAddressTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	FullyStructuredAddress	BuildingComplexSubUnitTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	FullyStructuredAddress	BuildingComplexSubUnitNumber	~~VARCHAR~~NVARCHAR(25)
Location	FullyStructuredAddress	FloorLevelNumber	~~VARCHAR~~NVARCHAR(25)
Location	FullyStructuredAddress	StreetPropertyNumber	~~VARCHAR~~NVARCHAR(25)
Location	FullyStructuredAddress	LotSectionNumber	~~VARCHAR~~NVARCHAR(25)
Location	FullyStructuredAddress	StreetName	~~VARCHAR~~NVARCHAR(100)
Location	FullyStructuredAddress	StreetTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	Place	LocationSKey	~~UNIQUEIDENTIFIER~~BIGINT
Location	Place	LocationTypeCode	~~VARCHAR~~NVARCHAR(50)
Location	Place	CountryCode	~~VARCHAR~~NVARCHAR(25)
Location	Place	PlaceTypeCode	~~VARCHAR~~NVARCHAR(25)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 362 of 1295

Location	PostOfficeAddress	LocationSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Location	PostOfficeAddress	LocationTypeCode	~~VARCHAR~~NVARCHAR(50)
Location	PostOfficeAddress	CountryCode	~~VARCHAR~~NVARCHAR(25)
Location	PostOfficeAddress	AddressTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	PostOfficeAddress	PostalDeliveryTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	PostOfficeAddress	PostalDeliveryReferenceNumber	~~VARCHAR~~NVARCHAR(25)
Location	PostOfficeAddress	SuburbName	~~VARCHAR~~NVARCHAR(100)
Location	PostOfficeAddress	StateCode	~~VARCHAR~~NVARCHAR(25)
Location	PostOfficeAddress	Postcode	~~VARCHAR~~NVARCHAR(25)
Location	ResponsibilityLocation	ResponsibilityLocationSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Location	ResponsibilityLocation	ResponsibilitySKey	~~UNIQUEIDENTIFIER~~ BIGINT
Location	ResponsibilityLocation	LocationSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Location	ResponsibilityLocation	ResponsibilityLocationUsageTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	ResponsibilityLocation	PredominantResponsibilityLocationIndicator	CHAR(1)
Location	ResponsibilityLocation	LocationIdentifierSkey	~~UNIQUEIDENTIFIER~~ BIGINT
Location	SemiStructuredAddress	LocationSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Location	SemiStructuredAddress	LocationTypeCode	~~VARCHAR~~NVARCHAR(50)
Location	SemiStructuredAddress	CountryCode	~~VARCHAR~~NVARCHAR(25)
Location	SemiStructuredAddress	AddressTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	SemiStructuredAddress	BuildingPropertyName	~~VARCHAR~~NVARCHAR(100)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 363 of 1295

Location	SemiStructuredAddress	PhysicalAddressTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	SemiStructuredAddress	Postcode	~~VARCHAR~~NVARCHAR(25)
Location	SemiStructuredAddress	StateCode	~~VARCHAR~~NVARCHAR(25)
Location	SemiStructuredAddress	SuburbName	~~VARCHAR~~NVARCHAR(100)
Location	SemiStructuredAddress	StreetAddressTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	SemiStructuredAddress	AddressLine1Description	~~VARCHAR~~NVARCHAR(500)
Location	SemiStructuredAddress	AddressLine2Description	~~VARCHAR~~NVARCHAR(500)
Location	StakeholderLocation	StakeholderLocationSKey	~~UNIQUEIDENTIFIER~~BIGINT
Location	StakeholderLocation	StakeholderSKey	~~UNIQUEIDENTIFIER~~BIGINT
Location	StakeholderLocation	LocationSKey	~~UNIQUEIDENTIFIER~~BIGINT
Location	StakeholderLocation	StakeholderLocationUsageTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	StakeholderLocation	StakeholderRoleTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	TelecommunicationsAddress	LocationSKey	~~UNIQUEIDENTIFIER~~BIGINT
Location	TelecommunicationsAddress	LocationTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	TelecommunicationsAddress	CountryCode	~~VARCHAR~~NVARCHAR(25)
Location	TelecommunicationsAddress	AddressTypeCode	~~VARCHAR~~NVARCHAR(25)
Location	TelecommunicationsAddress	TelecommunicationsAddressReferenceNumber	~~VARCHAR~~NVARCHAR(25)
Location	TelecommunicationsAddress	TelecommunicationsAddressTypeCode	~~VARCHAR~~NVARCHAR(25)
Offering	SubjectOffering	OfferingSKey	~~UNIQUEIDENTIFIER~~BIGINT
Offering	SubjectOffering	StakeholderSKey	~~UNIQUEIDENTIFIER~~BIGINT

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 364 of 1295

Offering	SubjectOffering	ProductSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Offering	SubjectOffering	CensusDate	DATE
Offering	SubjectOffering	ScheduledHoursQuantity	INTEGER
Product	Program	ProductSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Product	Program	ProductClassCode	~~VARCHAR~~NVARCHAR(25)
Product	Program	ProductCategoryCode	~~VARCHAR~~NVARCHAR(25)
Product	Program	ProductTypeCode	~~VARCHAR~~NVARCHAR(25)
Product	Program	ProductReferenceNumber	~~VARCHAR~~NVARCHAR(25)
Product	Program	ProductName	~~VARCHAR~~NVARCHAR(100)
Product	Program	RecognisedAchievementTypeCode	~~VARCHAR~~NVARCHAR(25)
Product	Program	ProductClosedDate	DATE
Product	Program	ProductOpenDate	DATE
Product	Program	ProgramClassCode	~~VARCHAR~~NVARCHAR(25)
Product	Program	ProgramCategoryCode	~~VARCHAR~~NVARCHAR(25)
Product	Program	ProgramTypeCode	~~VARCHAR~~NVARCHAR(25)
Product	Program	StreamReferenceNumber	~~VARCHAR~~NVARCHAR(25)
Product	Program	StreamName	~~VARCHAR~~NVARCHAR(100)
Product	Program	ASCEDNarrowFieldofEducationCode	~~VARCHAR~~NVARCHAR(25)
Product	Program	ASCEDDetailedLevelofEducationCode	~~VARCHAR~~NVARCHAR(25)
Product	Program	VocationalEducationPurposeCode	~~VARCHAR~~NVARCHAR(25)
Product	Program	ANZSCOCode	~~VARCHAR~~NVARCHAR(25)
Product	Program	NominalHoursQuantity	INTEGER

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 365 of 1295

Product	Subject	ProductSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Product	Subject	ProductClassCode	~~VARCHAR~~NVARCHAR(25)
Product	Subject	ProductCategoryCode	~~VARCHAR~~NVARCHAR(25)
Product	Subject	ProductTypeCode	~~VARCHAR~~NVARCHAR(25)
Product	Subject	ProductReferenceNumber	~~VARCHAR~~NVARCHAR(25)
Product	Subject	ProductName	~~VARCHAR~~NVARCHAR(100)
Product	Subject	RecognisedAchievementTypeCode	~~VARCHAR~~NVARCHAR(25)
Product	Subject	ProductClosedDate	DATE
Product	Subject	ProductOpenDate	DATE
Product	Subject	SubjectClassCode	~~VARCHAR~~NVARCHAR(25)
Product	Subject	SubjectCategoryCode	~~VARCHAR~~NVARCHAR(25)
Product	Subject	SubjectTypeCode	~~VARCHAR~~NVARCHAR(25)
Product	Subject	ASCEDDetailedFieldofEducationCode	~~VARCHAR~~NVARCHAR(25)
Product	Subject	VocationalEducationPurposeCode	~~VARCHAR~~NVARCHAR(25)
Product	Subject	NominalHoursQuantity	INTEGER
Responsibility	EnrolmentLearningChannel	EnrolmentLearningChannelSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	EnrolmentLearningChannel	ResponsibilitySKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	EnrolmentLearningChannel	LearningChannelTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	EnrolmentLearningChannel	PredominantLearningChannelIndicator	CHAR(1)
Responsibility	EnrolmentLearningDelivery	EnrolmentLearningDeliverySKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	EnrolmentLearningDelivery	ResponsibilitySKey	~~UNIQUEIDENTIFIER~~ BIGINT

Responsibility	EnrolmentLearningDelivery	LearningDeliveryTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	EnrolmentLearningDelivery	PredominantLearningDeliveryIndicator	CHAR(1)
Responsibility	EnrolmentStudyReason	EnrolmentStudyReasonSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	EnrolmentStudyReason	ResponsibilitySKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	EnrolmentStudyReason	StudyReasonCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	ResourceResponsibility	ResourceResponsibilityIssueDate	DATE
Responsibility	ResourceResponsibility	ResourceResponsibilityReferenceNumber	~~VARCHAR~~NVARCHAR(50)
Responsibility	ResourceResponsibility	ResourceResponsibilitySKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ResourceResponsibility	ResourceResponsibilityTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	ResourceResponsibility	ResourceSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ResourceResponsibility	ResponsibilitySKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ProgramEnrolment	ResponsibilitySKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ProgramEnrolment	ResponsibilityTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	ProgramEnrolment	OfferingSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ProgramEnrolment	ScheduledStartDate	DATE
Responsibility	ProgramEnrolment	ScheduledCompletionDate	DATE
Responsibility	ProgramEnrolment	ResponsibilityEngagementMethodCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	ProgramEnrolment	EmploymentExperienceANZSCOCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	ProgramEnrolment	EmploymentExperienceANZSICCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	ProgramEnrolment	IncomeContingentLoanLiabilityAmount	~~VARCHAR~~NVARCHAR(50)
Responsibility	ProgramEnrolment	FeeMaintenanceIndicator	CHAR(1)
Responsibility	ProgramEnrolment	AssociatedProgramReferenceNumber	~~VARCHAR~~NVARCHAR(25)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 367 of 1295

Responsibility	ProgramEnrolment	AdministrativeProgramIndicator	CHAR(1)
Responsibility	ResponsibilityFinancialDetail	ResponsibilityFinancialDetailSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ResponsibilityFinancialDetail	ResponsibilitySKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ResponsibilityFinancialDetail	ResponsibilityFinancialDetailClassCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	ResponsibilityFinancialDetail	ResponsibilityFinancialDetailCategoryCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	ResponsibilityFinancialDetail	ResponsibilityFinancialDetailTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	ResponsibilityFinancialDetailAmount	ResponsibilityFinancialDetailAmountSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ResponsibilityFinancialDetailAmount	ResponsibilityFinancialDetailSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ResponsibilityFinancialDetailAmount	ResponsibilityFinancialDetailAmountTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	ResponsibilityFinancialDetailAmount	ResponsibilityFinancialDetailAmount	~~VARCHAR~~NVARCHAR(50)
Responsibility	ResponsibilityFinancialDetailAmount	FinancialDetailPaymentContributionTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	ResponsibilityIdentifier	ResponsibilityIdentifierSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ResponsibilityIdentifier	ResponsibilitySKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ResponsibilityIdentifier	ResponsibilityIdentifierTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	ResponsibilityIdentifier	ResponsibilityIdentifierTypeReferenceNumber	~~VARCHAR~~NVARCHAR(25)
Responsibility	ResponsibilityLifecycleState	ResponsibilityStateSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ResponsibilityLifecycleState	ResponsibilitySKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ResponsibilityLifecycleState	ResponsibilityLifecycleStateTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	ResponsibilityLifecycleState	ResponsibilityLifecycleStateReasonTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	ResponsibilityLifecycleState	ResponsibilityLifecycleStateEffectiveDate	DATE
Responsibility	ResponsibilityLifecycleState	ResponsibilityLifecycleStateExpiryDate	DATE

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 368 of 1295

Responsibility	ResponsibilityRelationship	ResponsibilityRelationshipSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ResponsibilityRelationship	ResponsibilityRelationshipTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	ResponsibilityRelationship	ResponsibilitySKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ResponsibilityRelationship	RelatedResponsibilitySKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ResourceResponsibility	ResourceResponsibilitySKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ResourceResponsibility	ResourceSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ResourceResponsibility	ResponsibilitySKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	ResourceResponsibility	ResourceResponsibilityTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	ResourceResponsibility	ResourceResponsibilityIssueDate	DATE
Responsibility	ResourceResponsibility	ResourceResponsibilityReferenceNumber	~~VARCHAR~~NVARCHAR(50)
Responsibility	StakeholderResponsibility	StakeholderResponsibilitySKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	StakeholderResponsibility	StakeholderSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	StakeholderResponsibility	ResponsibilitySKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	StakeholderResponsibility	StakeholderResponsibilityTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	StakeholderResponsibility	StakeholderRoleTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	StakeholderResponsibility	InternationalStudentTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	SubjectEnrolment	ResponsibilitySKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	SubjectEnrolment	ResponsibilityTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	SubjectEnrolment	OfferingSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	SubjectEnrolment	ScheduledStartDate	DATE
Responsibility	SubjectEnrolment	ScheduledCompletionDate	DATE

Responsibility	SubjectEnrolment	ResponsibilityEngagementMethodCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	SubjectEnrolment	EmploymentExperienceANZSCOCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	SubjectEnrolment	EmploymentExperienceANZSICCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	SubjectEnrolment	IncomeContingentLoanLiabilityAmount	~~VARCHAR~~NVARCHAR(50)
Responsibility	SubjectEnrolment	FeeMaintenanceIndicator	CHAR(1)
Responsibility	SubjectEnrolment	VETInSchoolIndicator	CHAR(1)
Responsibility	SubjectEnrolment	RCCIndicator	CHAR(1)
Responsibility	SubjectEnrolment	UnitAttendanceHoursAtWithdrawQuantity	INTEGER
Responsibility	SubjectEnrolment	CensusDate	DATE
Responsibility	SubjectEnrolment	SubjectAssessmentTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	SubjectEnrolment	EnrolmentLiabilityTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	SubsidyFundingArrangement	ResponsibilitySKey	~~UNIQUEIDENTIFIER~~BIGINT
Responsibility	SubsidyFundingArrangement	ResponsibilityTypeCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	SubsidyFundingArrangement	OfferingSKey	~~UNIQUEIDENTIFIER~~BIGINT
Responsibility	SubsidyFundingArrangement	ScheduledStartDate	DATE
Responsibility	SubsidyFundingArrangement	ScheduledCompletionDate	DATE
Responsibility	SubsidyFundingArrangement	ResponsibilityEngagementMethodCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	SubsidyFundingArrangement	EmploymentExperienceANZSCOCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	SubsidyFundingArrangement	EmploymentExperienceANZSICCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	SubsidyFundingArrangement	IncomeContingentLoanLiabilityAmount	~~VARCHAR~~NVARCHAR(50)
Responsibility	SubsidyFundingArrangement	SubsidyEligibilityOverrideIndicator	CHAR(1)
Responsibility	WorkPlacement	ResponsibilitySKey	~~UNIQUEIDENTIFIER~~BIGINT
Responsibility	WorkPlacement	ResponsibilityTypeCode	~~VARCHAR~~NVARCHAR(25)

Responsibility	WorkPlacement	OfferingSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Responsibility	WorkPlacement	ScheduledStartDate	DATE
Responsibility	WorkPlacement	ScheduledCompletionDate	DATE
Responsibility	WorkPlacement	ResponsibilityEngagementMethodCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	WorkPlacement	EmploymentExperienceANZSCOCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	WorkPlacement	EmploymentExperienceANZSICCode	~~VARCHAR~~NVARCHAR(25)
Responsibility	WorkPlacement	IncomeContingentLoanLiabilityAmount	~~VARCHAR~~NVARCHAR(50)
Responsibility	WorkPlacement	SupervisedHoursCount	INTEGER
Stakeholder	Stakeholder	StakeholderSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Stakeholder	Stakeholder	StakeholderTypeCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	Government	GovernmentTypeCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	Organisation	OrganisationTypeCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	Organisation	TrainingOrganisationTypeCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	Organisation	ABN	~~VARCHAR~~NVARCHAR(25)
Stakeholder	Person	PersonTitleCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	Person	FirstGivenName	~~VARCHAR~~NVARCHAR(100)
Stakeholder	Person	OtherGivenNames	~~VARCHAR~~NVARCHAR(100)
Stakeholder	Person	FamilyName	~~VARCHAR~~NVARCHAR(100)
Stakeholder	Person	BirthDate	DATE
Stakeholder	Person	DeceasedIndicator	CHAR(1)
Stakeholder	Person	GenderTypeCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	Person	HighestSchoolLevelCompletedCode	~~VARCHAR~~NVARCHAR(25)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 371 of 1295

Stakeholder	Person	HighSchoolCompletionYear	INTEGER
Stakeholder	Person	SchoolLevelCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	Person	LabourForceStatusCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	Person	SecondarySchoolTypeCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	Person	VisaTypeCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	Person	CountryCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	PersonCitizenship	PersonCitizenshipSKey	~~UNIQUEIDENTIFIER~~BIGINT
Stakeholder	PersonCitizenship	StakeholderSKey	~~UNIQUEIDENTIFIER~~BIGINT
Stakeholder	PersonCitizenship	CitizenshipTypeCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	PersonDisability	PersonDisabilitySKey	~~UNIQUEIDENTIFIER~~BIGINT
Stakeholder	PersonDisability	StakeholderSKey	~~UNIQUEIDENTIFIER~~BIGINT
Stakeholder	PersonDisability	DisabilityTypeCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	PersonEducationalAchievement	PersonEducationalAchievementSKey	~~UNIQUEIDENTIFIER~~BIGINT
Stakeholder	PersonEducationalAchievement	StakeholderSKey	~~UNIQUEIDENTIFIER~~BIGINT
Stakeholder	PersonEducationalAchievement	EducationalAchievementTypeCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	PersonEducationalAchievement	RecognisedAchievementTypeCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	PersonIndigenousStatus	PersonIndigenousStatusSKey	~~UNIQUEIDENTIFIER~~BIGINT
Stakeholder	PersonIndigenousStatus	StakeholderSKey	~~UNIQUEIDENTIFIER~~BIGINT
Stakeholder	PersonIndigenousStatus	IndigenousStatusCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	PersonLanguage	PersonLanguageSK	~~VARCHAR~~NVARCHAR(50)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 372 of 1295

Stakeholder	PersonLanguage	StakeholderSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Stakeholder	PersonLanguage	LanguageCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	PersonLanguage	PersonLanguageContextCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	StakeholderIdentification	StakeholderIdentificationSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Stakeholder	StakeholderIdentification	StakeholderSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Stakeholder	StakeholderIdentification	StakeholderIdentificationTypeCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	StakeholderIdentification	StakeholderIdentificationTypeReferenceNumber	~~VARCHAR~~NVARCHAR(25)
Stakeholder	StakeholderIdentification	StakeholderIdentificationVerificationStatusCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	StakeholderIdentification	StakeholderIdentificationTypeVerificationDate	DATE
Stakeholder	StakeholderIdentifier	StakeholderIdentifierSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Stakeholder	StakeholderIdentifier	StakeholderSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Stakeholder	StakeholderIdentifier	StakeholderIdentifierTypeCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	StakeholderIdentifier	StakeholderIdentifierTypeReferenceNumber	~~VARCHAR~~NVARCHAR(25)
Stakeholder	StakeholderName	StakeholderNameSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Stakeholder	StakeholderName	StakeholderSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Stakeholder	StakeholderName	StakeholderNameTypeCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	StakeholderName	StakeholderNameTypeValue	~~VARCHAR~~NVARCHAR(100)
Stakeholder	StakeholderPreference	StakeholderPreferenceSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Stakeholder	StakeholderPreference	StakeholderSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Stakeholder	StakeholderPreference	StakeholderPreferenceTypeCode	~~VARCHAR~~NVARCHAR(25)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 373 of 1295

Stakeholder	StakeholderPreference	StakeholderPreferenceTypeValueCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	StakeholderPreference	StakeholderPreferenceReasonTypeCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	StakeholderRelationship	StakeholderRelationshipSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Stakeholder	StakeholderRelationship	StakeholderRelationshipTypeCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	StakeholderRelationship	StakeholderSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Stakeholder	StakeholderRelationship	RelatedStakeholderSKey	~~UNIQUEIDENTIFIER~~ BIGINT
Stakeholder	StakeholderRelationship	StakeholderRoleTypeCode	~~VARCHAR~~NVARCHAR(25)
Stakeholder	StakeholderRelationship	RelatedStakeholderRoleTypeCode	~~VARCHAR~~NVARCHAR(25)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 374 of 1295

910 Access Management

The detailed requirements relating to Master Database data access controls (Role and Attribute Based Access Control) will be advised by detailed design decisions documented as part of other STARS system services that will directly access data stored within the Master Database.

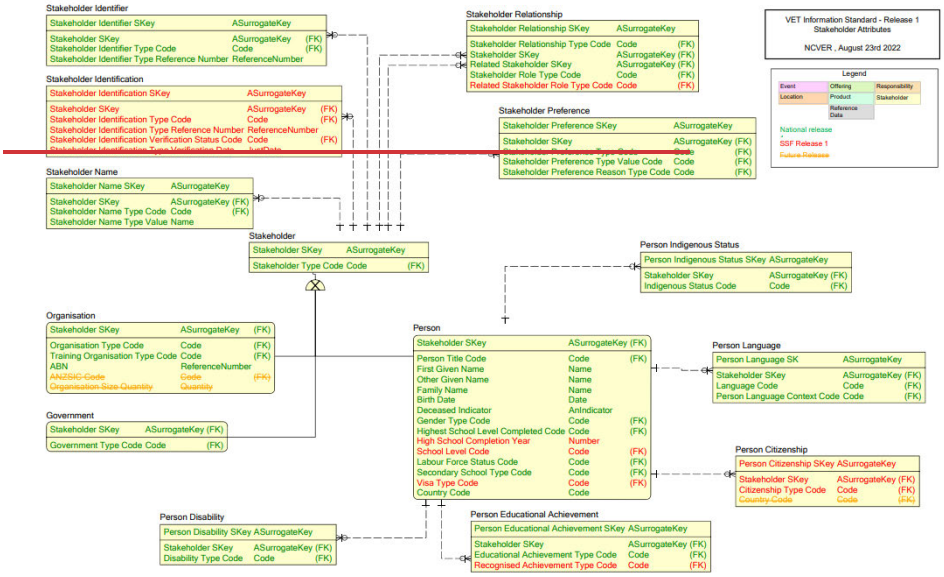
State jurisdictional data sets will be segregated and stored within separate databases as will National datasets. Optionally row-based database access can be implemented to restrict access to data set types (i.e., aggregated, and de-identified data sets, financial data, national core data, state-based program data, and student and individual enrolment data) within each database based on assigned user groups.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 375 of 1295

OFFICIAL

~~10~~11Appendix

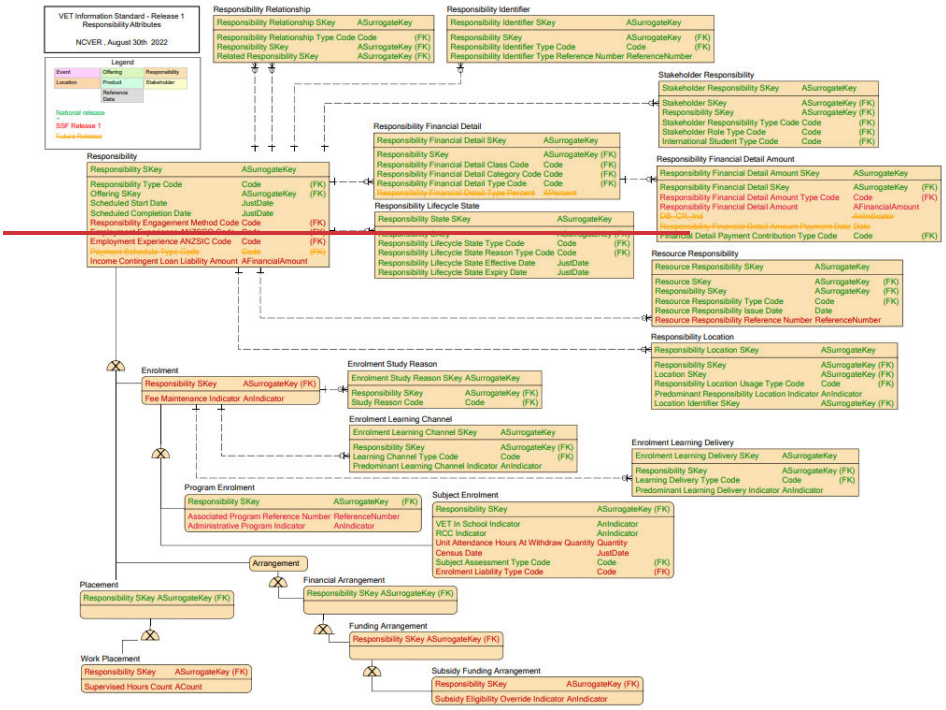
~~10.1 VET Information Standard Logical Data Model~~



~~Figure 8 - IS Prod Data Model Stakeholder_Core_Concepts_01092022_R1_V1.0~~

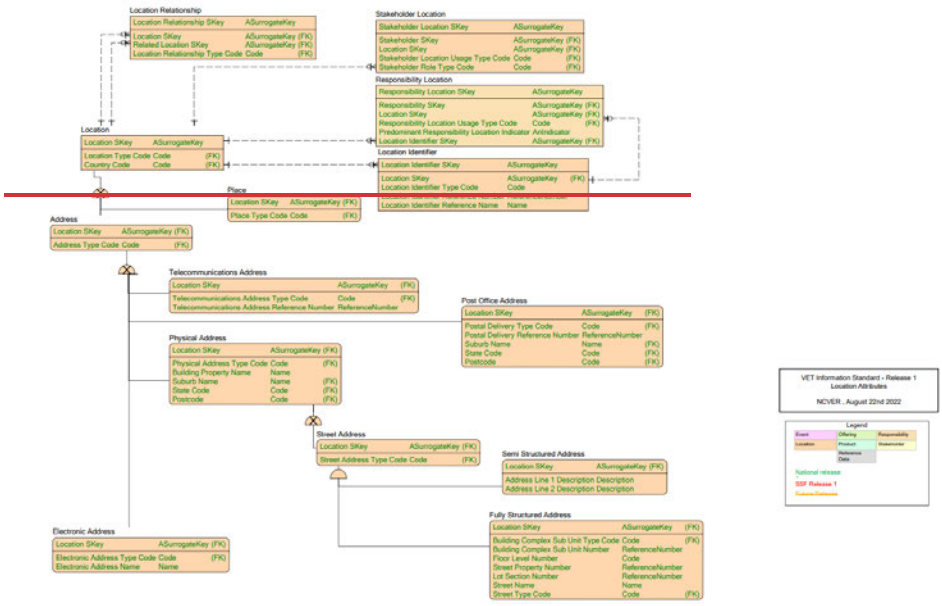
Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 376 of 1295

OFFICIAL



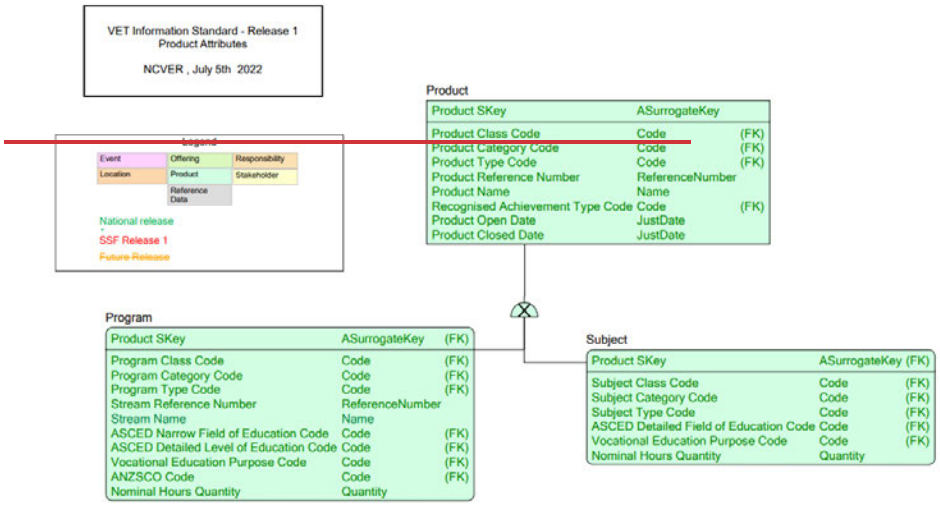
~~Figure 9 - IS Prod Data Model Responsibility_Core_Concepts_01092022_R1_V1.0~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 377 of 1295



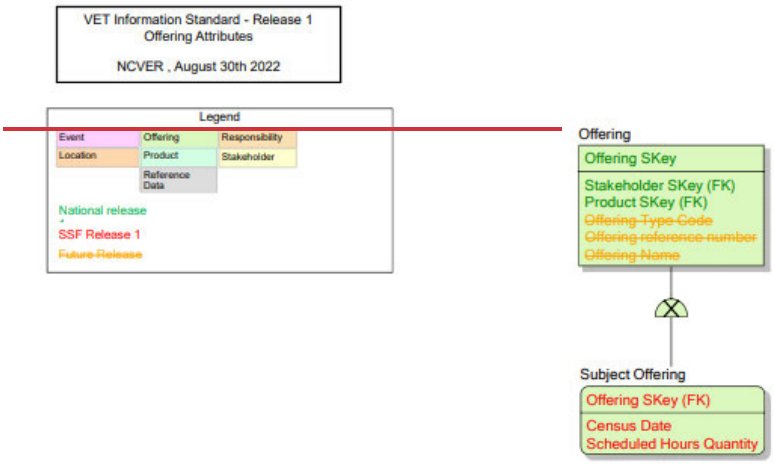
~~Figure 10 - IS Prod Data Model Location_Core_Concepts_01092022_R1_V1.0~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 378 of 1295



~~Figure 11 - IS Prod Data Model Product_Core_Concept_01092022_R1_V1.0~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 379 of 1295



~~Figure 12 - IS Prod Data Model Offering_Core_Concepts_01092022_R1_V1.0~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 380 of 1295

~~10.2~~11.1 VET-IS Data Packet Logical Entity Relationship Diagram

The following high level logical ERD diagram show the structure and relationship between the VET-IS data packets.

Note that green fields contain repeating elements. These will be represented in the associated JSON schemas as JSON lists.

Please refer to the Visio file Attachment 4 – Data Packet ERD – D.0.2.vsdx for a full-size representation.

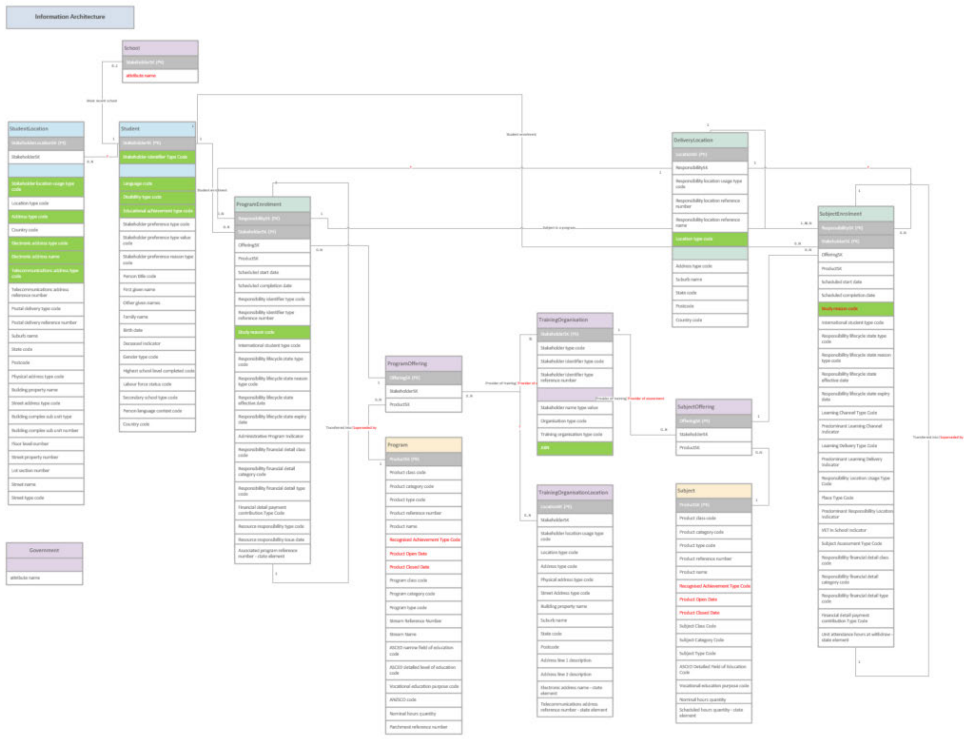


Figure ~~10~~10~~8~~ : VET-IS Data Packet Logical Entity Relationship Diagram

Formatted: Caption, Don't keep with next

~~Figure 13 - VET-IS Data Packet Logical Entity Relationship Diagram~~

~~11~~12Acronyms

Acronym	Definition
API	Application Programming Interface
CRUD	Create Read Update Delete
REST	Representational State Transfer
FR	Functional Requirement
NFR	Non-functional Requirement
STARS	Student Training & Activity Reporting System
VET	Vocational Education Training
VDS	VET Data Streamlining
IVETS	Integrated Vocational Education and Training System
DB	Database
DDL	Data Definition Language
SQL	Structured Query Language
TP	Training Provider
STA	State Training Authority
SMS	Student Management System

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 382 of 1295

Formatted: Normal, Space After: 8 pt, Line spacing: Multiple 1.08 li

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 383 of 1295

Australian Government
Department of Employment and Workplace Relations

Detailed Design Document

Reference Data Service

1 Document Management

1.1 Document Control

Version	Change Description	Date
0.1	Draft Version	24 Feb 2023
0.2	Updated as per feedback	08 Mar 2023
0.3	Updated after design workshop review	17 Mar 2023
1.0		

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This is a living document and will be updated as required.

Document Author

Role	Name
Data Architect/Tech Lead	s 47F(1)
Technical Architect	s 47F(1)
Design Lead	s 47F(1)

Document Reviewers

Role	Name

Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

Related Documents

Reference	Document
A	STARS Solution Requirements Document – V.1.3.docx
B	STARS Solution Design – V.1.0.docx
C	IS Prod Technical Dictionary_01092022_R1_V1.0
D	STARS Solution Design - Attachment 3 - API Design - V.1.0.docx
E	Detailed Design Document – IDAM.docx
F	VET Data Access and Privacy Protocol.docx
G	Reference Data Service Diagrams - Apps - D.0.21

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 385 of 1295

Reference	Document
I	Reference Data Service Diagrams - Data - D.0.231
J	Detailed Design Document - Notification Service D.0.4
K	STARS API Design Document

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 386 of 1295

1 Document Management 2
1.1 Document Control 2
Document Author 2
Document Reviewers 2
Document Approval 2
Related Documents 2
2 Introduction ..... 1~~4~~8
2.1 Purpose ..... 1~~4~~8
2.2 Intended Audience ..... 1~~4~~8
3 Scope and Exclusions ..... 1~~5~~9
3.1 In Scope ..... 1~~5~~9
3.2 Out of Scope ..... 16~~10~~
4 Assumptions ..... 16~~10~~
5 Design Decisions ..... 16~~10~~
6 Feature Overview ..... 17~~11~~
6.1 Service Components ..... 19~~13~~
6.1.1 Reference Data Workflow ..... 19~~13~~
6.1.2 Reference Data Functions ..... 19~~13~~
6.1.3 Reference Data Ingest ETL Pipelines ..... 19~~13~~
6.1.4 Reference Data API ..... 19~~13~~
6.1.5 Reference Data Database ..... 20~~14~~
6.1.6 Reference Data Event Database ..... 20~~14~~
6.2 Overview of Reference Data Service ..... 20~~14~~
6.2.1 Reference Data API ..... 21~~15~~
6.2.2 Reference Data Logic App ..... 21~~15~~
6.2.3 Process Active Failed Validation Function ..... 21~~15~~
6.2.4 Reference Data database ..... 21~~15~~
6.2.5 Reference Data Key Vault ..... 21~~15~~
6.2.6 Reference Data Config ..... 21~~15~~
7 Business Process ..... 21~~15~~
8 Reference Data Service Workflow ..... 22~~16~~
8.1 Basic Flow of Reference Data Update via Portal ..... 22~~16~~
8.2 Revalidation from Reference Data Update ..... 22~~16~~
8.3 Basic Flow of Reference Data Inserts via Portal ..... 23~~18~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 387 of 1295

9 Reference Data Service Events 2418
10 Technical Design Patterns 2419
11 API Design 2419
11.1 Reference Data API 2519
11.2 Reference Data database 2519
11.3 Reference Data Event database 2519
11.4 Reference Data Config 2519
11.5 Reference Data API Endpoints 2520
11.5.1 Reference Data Retrievals 2520
11.5.2 Reference Data Amendments 2721
12 Reference Data Sync Solution Components 2722
12.1 Data Source and Environment 2822
12.2 Target Databases 2924
12.3 Reference Data Entities 3024
12.3.1 NCVER (CARMS) 3024
12.3.2 States 4541
12.4 Reference Data Auditing and Versioning 5246
12.5 Reference Data Integration 5347
12.5.1 Data transfer from NCVER (CARMS) into STARS Reference Database (**Stage 1 release**) 5549
12.5.2 Data transfer from NCVER (CARMS) into STARS Reference Database (**Stage 2 release**) 5549
12.5.3 South Australia – Manual Files to STARS Reference Database (**Stage 1 release**) 5650
12.5.4 South Australia – Azure Data Share to STARS Reference Database (**Stage 3 release**) 5650
12.5.5 USI Verification 5651
12.5.6 Reference Data Validation 5651
12.5.7 Data Refresh Interval 5751
12.5.8 Full/ Incremental 5751
12.5.9 Data Volume 5751
12.6 Data Load – NCVER to STARS Azure Data Lake (Stage 2 release) 5752
12.6.1 ADF Pipeline – data load 5752
12.6.2 Raw Data Storage 5953
12.6.3 Hierarchical Namespace enabled ADLS Gen2 5953
12.6.4 Folder Structure and File Name Convention 5953

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 388 of 1295

12.6.5 File Format for Storage 6054
12.7 Data Load – Azure Data Lake into Azure SQL Reference database (Stage 2 release) 6155
12.7.1 Reference Data Cleansing and Transformation 6155
12.7.2 ADF Pipeline – data load 6155
12.8 Reference Data Updated Events (Stage 2 release) 6256
12.9 ADF Logging (Stage 2 release) 6256
12.9.1 Logs 6256
12.9.2 Alerting and Notification 6256
12.9.3 Exception Handling 6357
12.9.4 Custom Logging 6357
12.9.5 Metadata Extraction and Cataloguing 6357
13 Access Management 6357
13.1 Control of data 6458
13.1.1 Data Owner 6458
13.1.2 Data Custodian 6458
13.2 Security 6458
14 Dependency on Other Services 6559
14.1 Event Service 6559
14.2 Rules Service 6559
14.3 Validation Service 6559
14.4 Notification Service 6559
15 Functional Requirements 6660
16 Non-Functional Requirements 7165
17 Appendix 7266
17.1 Supporting documents 7266
17.2 Sample Data 7266
18 Acronyms 7367

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 389 of 1295

Formatted: Tab stops: Not at 1.16 cm

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 390 of 1295

Formatted: Tab stops: Not at 1.16 cm

Formatted: Tab stops: Not at 1.16 cm

Formatted: Tab stops: Not at 1.16 cm

Formatted: Tab stops: Not at 1.16 cm

Formatted: Tab stops: Not at 1.16 cm

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 391 of 1295

Formatted: Tab stops: Not at 1.16 cm

Formatted: Tab stops: Not at 1.16 cm

Formatted: Tab stops: Not at 1.16 cm

Formatted: Tab stops: Not at 1.16 cm

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

~~**Contents**~~

~~**1 Document Management ..... 2**~~

~~1.1 Document Control ..... 2~~

~~Document Author ..... 2~~

~~Document Reviewers ..... 2~~

~~Document Approval ..... 2~~

~~Related Documents ..... 2~~

~~**2 Introduction ..... 7**~~

~~2.1 Purpose ..... 8~~

~~2.2 Intended Audience ..... 8~~

~~**3 Scope and Exclusions ..... 8**~~

~~3.1 In Scope ..... 8~~

~~3.2 Out of Scope ..... 9~~

~~**4 Assumptions ..... 9**~~

~~**5 Design Decisions ..... 10**~~

~~**6 Feature Overview ..... 10**~~

~~6.1 Service Components ..... 12~~

~~6.1.1 Reference Data Workflow ..... 12~~

~~6.1.2 Reference Data Functions ..... 12~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 392 of 1295

~~6.1.3 Reference Data Ingest ETL Pipelines ..... 12~~
~~6.1.4 Reference Data API ..... 12~~
~~6.1.5 Reference Data Database ..... 13~~
~~6.1.6 Reference Data Event Database ..... 13~~
~~6.2 Overview of Reference Data Service ..... 13~~
~~6.2.1 Reference Data API ..... 14~~
~~6.2.2 Reference Data Logic App ..... 14~~
~~6.2.3 Process Active Failed Validation Function ..... 14~~
~~6.2.4 Reference Data database ..... 14~~
~~6.2.5 Reference Data Key Vault ..... 14~~
~~6.2.6 Reference Data Config ..... 14~~
~~**7 Business Process ..... 14**~~
~~**8 Reference Data Service Workflow ..... 14**~~
~~8.1 Basic Flow of Reference Data Update via Portal ..... 15~~
~~8.2 Revalidation from Reference Data Update ..... 15~~
~~8.3 Basic Flow of Reference Data Inserts via Portal ..... 16~~
~~**9 Reference Data Service Events ..... 17**~~
~~**10 Technical Design Patterns ..... 17**~~
~~**11 API Design ..... 17**~~
~~11.1 Reference Data API ..... 18~~
~~11.2 Reference Data database ..... 18~~
~~11.3 Reference Data Event database ..... 18~~
~~11.4 Reference Data Config ..... 18~~
~~11.5 Reference Data API Endpoints ..... 18~~
~~**12 Reference Data Sync Solution Components ..... 19**~~
~~12.1 Data Source and Environment ..... 20~~
~~12.2 Target Databases ..... 21~~
~~12.3 Reference Data Entities ..... 22~~

Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar

12.3.1 NCVER (CARMS) 22
12.3.2 States 37
12.4 Reference Data Auditing and Versioning 42
12.5 Reference Data Integration 42
12.5.1 Data transfer from NCVER (CARMS) into STARS Reference Database (Stage 1 release) 44
12.5.2 Data transfer from NCVER (CARMS) into STARS Reference Database (Stage 2 release) 44
12.5.3 South Australia – Azure Data Share to STARS Reference Database (Stage 3 release) .. 45
12.5.4 USI Verification 45
12.5.5 Data Refresh Interval 45
12.5.6 Full/ Incremental 45
12.5.7 Data Volume 46
12.6 Data Load – NCVER to STARS Azure Data Lake (Stage 2 release) 46
12.6.1 ADF Pipeline – data load 46
12.6.2 Raw Data Storage 47
12.6.3 Hierarchical Namespace enabled ADLS Gen2 48
12.6.4 Folder Structure and File Name Convention 48
12.6.5 File Format for Storage 49
12.7 Data Load – Azure Data Lake into Azure SQL Reference database (Stage 2 release) 49
12.7.1 Reference Data Cleansing and Transformation 49
12.7.2 ADF Pipeline – data load 49
12.8 Reference Data Updated Events (Stage 2 release) 50
12.9 ADF Logging (Stage 2 release) 50
12.9.1 Logs 50
12.9.2 Alerting and Notification 50
12.9.3 Exception Handling 51
12.9.4 Custom Logging 51
12.9.5 Metadata Extraction and Cataloguing 51
13 Access Management 51

Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Tab stops: Not at 1.16 cm
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Tab stops: Not at 1.16 cm
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted: Tab stops: Not at 1.16 cm
Formatted ...
Formatted ...
Formatted ...
Formatted: Tab stops: Not at 1.16 cm
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 394 of 1295

~~13.1 Control of data ..... 52~~
~~13.1.1 Data Owner ..... 52~~
~~13.1.2 Data Custodian ..... 52~~
~~13.2 Security ..... 52~~
~~**14 Dependency on Other Services ..... 53**~~
~~14.1 Event Service ..... 53~~
~~14.2 Rules Service ..... 53~~
~~14.3 Validation Service ..... 53~~
~~**15 Functional Requirements ..... 54**~~
~~**16 Non Functional Requirements ..... 59**~~
~~**17 Appendix ..... 60**~~
~~17.1 Supporting documents ..... 60~~
~~SA Reference Data ..... 60~~
~~17.2 Sample Data ..... 60~~
~~STARS Reference Data – SA Specific ..... 60~~
~~**18 Acronyms ..... 61**~~

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Tab stops: Not at 1.16 cm

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Tab stops: Not at 1.16 cm

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Font: Check spelling and grammar

Formatted: Tab stops: Not at 1.16 cm

Formatted: Default Paragraph Font, Font: Check spelling and grammar

Formatted: Default Paragraph Font, Font: Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Tab stops: Not at 1.16 cm

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Tab stops: Not at 1.16 cm

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 395 of 1295

Figure 1: STARS Support Service Context - Reference Data Service (I1) 1812
Figure 2: Reference Data Service Overview (G1) 1913
Figure 3 : Reference Data Service Overview (G1) 2014
Figure 4: Internal Reference Data Update VIA Portal (G2) 2216
Figure 5: Basic Flow to Process Active Validation Failures (G3) 2317
Figure 6: Basic Flow of Internal Reference Data Inserts VIA Portal (G4) 2418
Figure 7: STARS Azure Data Estate (I2) 2822
Figure 8: Reference Data Integration Technical Design (I3) 5448
Figure 9: NCVER (CARMS) Data Transfer pipeline to Reference database (I4) 5550
Figure 10: SA Azure Data Share to STARS Reference database (I5) 5650
Figure 11: ADF Pipeline steps to load into Azure Data Lake (I6) 5852
Figure 12: Data Lake Folder Structure (I7) 6054
Figure 13: ADF Pipeline Data Load (I8) 6155
Figure 14: ADF To Service Bus Event Trigger (I9) 6256
~~Figure 1: STARS Support Service Context - Reference Data Service (I1) ..... 11~~
~~Figure 2: Reference Data Service Overview ..... 12~~
~~Figure 3 : Reference Data Service Overview (G1) ..... 13~~
~~Figure 4: Reference Data Update VIA Portal (G2) ..... 15~~
~~Figure 5: Basic Flow to Process Active Validation Failures (G3) ..... 16~~
~~Figure 6: Basic Flow of Reference Data Inserts VIA Portal (G4) ..... 17~~
~~Figure 7: STARS Azure Data Estate (I2) ..... 20~~
~~Figure 8: Reference Data Integration Technical Design (I3) ..... 43~~
~~Figure 9: NCVER (CARMS) Data Transfer pipeline to Reference database (I4) ..... 44~~
~~Figure 10: SA Azure Data Share to STARS Reference database (I5) ..... 45~~
~~Figure 11: ADF Pipeline steps to load into Azure Data Lake (I6) ..... 46~~
~~Figure 12: Data Lake Folder Structure (I7) ..... 48~~
~~Figure 13: ADF Pipeline Data Load (I8) ..... 49~~
~~Figure 14: ADF To Service Bus Event Trigger (I9) ..... 50~~

Formatted: Default Paragraph Font, Check spelling and grammar

2 Introduction

This document covers the design of the STARS Reference Data Service.

Note: This is a living document that will be updated as designs for various other STARS services are developed and evolve. Accordingly, this document will evolve to reflect and document reference data components as the STARS solution build progresses.

The design is informed by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework, which provide a comprehensive set of best practices and guidance for designing, deploying, and operating secure, high-performing, and resilient cloud infrastructure and applications.

The Reference Data Service manages and provides reference data to core services like the Rules Service, the Validation Service in the STARS. This reference data is utilized by core services to verify and validate the data submitted to STARS.

The Reference Data Service has two primary roles:

1. Manage the workflows and processes involved with sourcing reference data from external sources, such as TGA.
2. Managing and providing reference data through a consistent schema, for use by other STARS solution services.

The Reference Data Service manages and provides reference data for many data quality and business rules, used by the Provider Service, the Rules Service, and workflows throughout STARS. The Reference Data Service is used by most services to source reference or lookup data.

2.1 Purpose

The purpose of this detailed design document is to provide a technical description of the Reference Data Service. The document is intended for technical stakeholders, including architects, software engineers, administrators, and operations teams, who are responsible for designing, developing, deploying, and operating the STARS solution in Azure.

The document also serves as a reference for ongoing management and evolution of the STARS solution and to ensure that the STARS solution is deployed and operated in accordance with the best practices and guidance provided by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework.

The document is intended to be a living document, updated as the STARS solution evolves and changes over time. It provides a clear and concise understanding of the Azure infrastructure and networking components and their configurations, helping to ensure the STARS solution is deployed, operated, and managed in a consistent and efficient manner.

2.2 Intended Audience

This detailed design document provides technical information and guidance to help technical stakeholders understand the Reference Data Service components of the STARS solution and how they

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 397 of 1295

integrate with each other. The document is expected to be used by technical stakeholders who have a good understanding of cloud computing concepts and are experienced in the Azure platform.

This document is not intended for non-technical readers, nor for readers who are not experienced in the design and configuration of Azure-based solutions; informational and educational resources regarding Azure are readily available at https://Azure.microsoft.com

The primary audience for this document includes:

- Program technical staff: This includes architects, software engineers, administrators, and operations teams within the program responsible for the STARS solution.
- External technical stakeholders: This includes technical staff from other federal, state, and territory governments agencies, including VDS embedded team members.
- ICT Technical Staff: This includes a variety of technical roles involved during the VDS program and post-go live, including network engineers, systems administrators, database administrators, security engineers, and DevOps engineers.

3 Scope and Exclusions

3.1 In Scope

The following are in-scope of this detail design solution:

- CARMS and State specific reference data entities in STARS
- Data Ingestion from NCVER and State environments into STARS platform (as per Stages defined below)
- Reference data load into reference data tables in STARS Azure SQL Reference database
- Reference Data Curation activities/processes
- Reference Data Consumption

Reference Data in STARS will be implemented in stages as below:

Stage 1:

1. Trigger refresh (manual data load) of CARMS and SA specific reference data.
2. All API operations will be implemented. Not expected to be used in stage 1 but the capability will exist.
3. No curation of CARMS reference or SA specific reference data.

Stage 2:

1. Production level high availability integration between STARS & NCVER including CARMS.
2. No curation of CARMS reference data in STARS. Data will be curated in CARMS.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 398 of 1295

Stage 3:

1. Curation of states reference data.
2. State reference data integrations.

Stage 4:

3. Direct integration with reference data sources (external & internal sources), this is dependent on the data source~~.~~
4. Full curation of all reference data.

3.2 Out of Scope

The following are out of scope for this document:

- The Reference Data Management Portal detail design will be covered in separate workshop starting 28 April 2023. This is where the reference data management process will be documented.
- ~~Data Ingestion from STAs environments into STARS platform~~
- Curation of ingested reference data before loading into Azure SQL Reference database (this will be implemented in Stage 3 & 4)

4 Assumptions

Id #	Assumption
A001	Data sources will include on-premises components in NCVER.
A002	Assume Azure Data Factory to be used as a tool for data ingestion and movement.
A003	Assume Azure Data Lake Gen 2 to be used as a location to store ingested ~~raw~~ reference data
A004	Assume STARS ADF ~~can~~ connects to NCVER and other external sources via SHIR (VM)
~~A005~~	~~Assume STARS to not be a reference data source for other external systems~~

5 Design Decisions

The STARS solution will utilise Service Oriented Architecture (SOA) principles for design and implementation, coupled with Master Data Management (MDM) principles for population of the Master Database. The solution will be hosted in Microsoft Azure and will conform to the Microsoft Well-Architected Framework[1], utilising recognised Azure Reference Architectures[2].

[1] https://docs.microsoft.com/en-us/azure/architecture/framework/
[2] https://docs.microsoft.com/en-us/azure/architecture/

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 399 of 1295

The architecture and design principles are described in detail in the STARS Solution Design document.

Of note is that the entire STARS solution will be implemented within the Microsoft Azure cloud platform and will heavily utilise Platform as a Service (PaaS) offerings. Accordingly, the Solution Design is oriented around utilising Azure features and resources that support the design and implementation of loosely coupled services. These include, for example, having services implemented in separate Resource Groups, using features such as Azure Functions, Azure SQL, and service-specific Azure App configurations.

Loose coupling will be achieved using Message- and Event-Driven approaches for communication between services, using features such as Azure Service Bus and Azure Event Grid. Services that present HTTPS/REST/JSON APIs will have their URLs controlled through Azure Application Gateway. This will enable a consistent URL scheme that also facilitates loose coupling.

Table 11~~1~~: Design Decisions

Decision #	Design Decision	Status
DD01	CARMS Data Integration with STARS. Dependent on DEWR Stage 2 Data Integration design.	Open
DD02	USI data integration for STARS to perform USI verification. Options for verifying USI is to be confirmed: • API Calls • Get Data into Reference DB • Decision Outcome: USI verification will be done through API services	~~Open~~Closed
DD03	Reference data need to be VIS version specific and support multiple concurrent VIS versions just like training activity submissions. This needs to be worked in with consultation with NCVER, required for Stage 2. Action item 3579.	Open
DD04	Validation process and rules for validating incoming reference data to be worked in with consultation with NCVER, required for Stage 2.	Open

Formatted: Normal, No bullets or numbering

6 Feature Overview

The Reference Data Service has two primary roles:

1. Manage the workflows and processes involved with sourcing reference data from external sources, such as TGA, ~~and~~ CARMS and Australia Post.
2. Managing and providing reference data through a consistent schema, for use by other STARS solution services.

The Reference Data Service manages and provides reference data for many data quality and business rules, used by the Provider Service, the Rules Service, and workflows throughout STARS. The Reference

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 400 of 1295

Data Service is used by most services to source reference or lookup data, as shown in the diagram below. The Reference Data Service also records a full history of reference data management events.

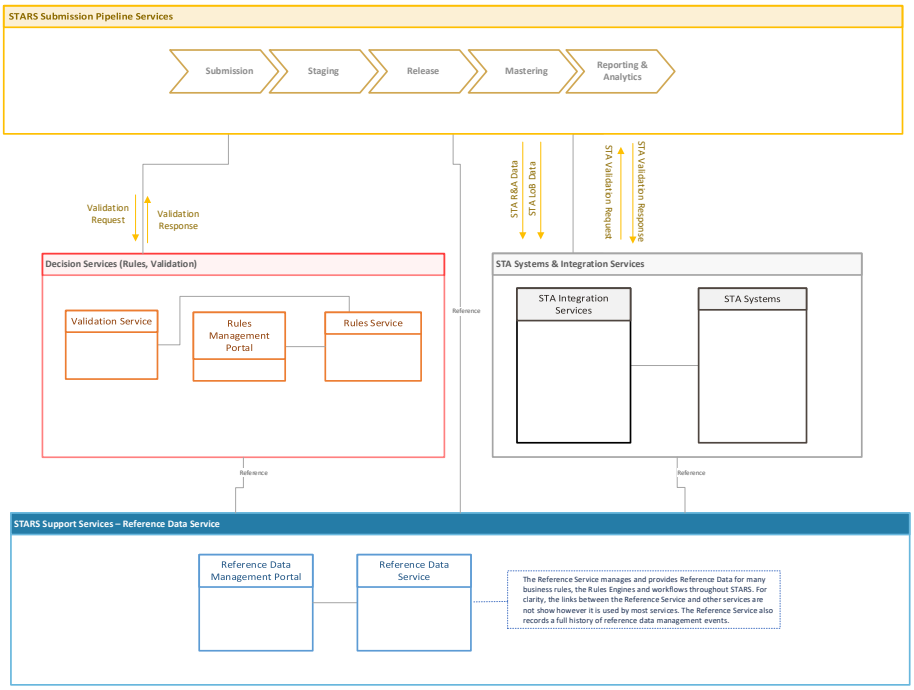


Figure 11~~1~~: STARS Support Service Context - Reference Data Service (I1)

The Reference Data Management Portal provides the STARS management interface for reference data, including the ability to amend or add internal reference data manually in case of errors. This is important, as source systems are occasionally missing information or contain errors; the portal will allow STARS system operators to amend information so that STARS processing can continue. The Reference Data Management Portal detail design will be covered in separate workshop. The workshop will cover the reference data management process and will be documented in the detail design.

Formatted: Font: Bold

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 401 of 1295

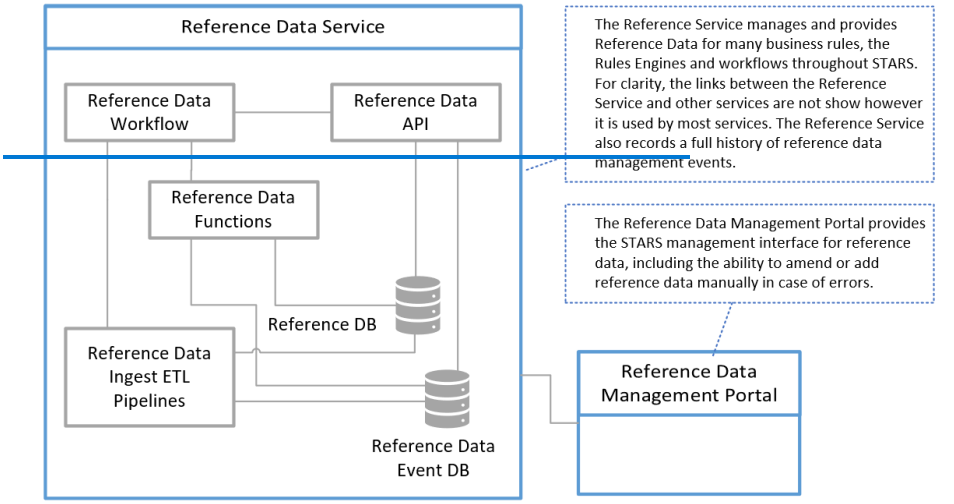


Figure 22: Reference Data Service Overview (G1)

6.1 Service Components

6.1.1 Reference Data Workflow

The Reference Data Service internal processes will be controlled by one or more Azure Logic App workflows. These workflows will manage, for example, the process for handling alerts raised by other services for invalid reference data. The workflows will also manage the processes for ingesting source data from external services, including handling exceptions.

6.1.2 Reference Data Functions

The Reference Data Service code-based functionality will be implemented as Azure Function Apps. These functions will support the operation of the service's workflows as well as functionality provided through the Reference Data Management Portal. For example, the portal will have functions to amend reference data with the actual code to implement this being at the database level will be within the Function Apps.

6.1.3 Reference Data Ingest ETL Pipelines

The Reference Data Service will maintain several Azure Data Factory based ETL pipelines for data ingestion. These will be configured to connect to upstream reference data sources, using the appropriate transport and protocols. The pipelines will extract, transform and then load reference data into the Reference DB, using a consistent schema.

6.1.4 Reference Data API

The Reference Data API will be implemented as an Azure Function App, utilising HTTP endpoints.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 402 of 1295

Care will be taken to ensure that reference data can be presented via the API to workflows and the rules engine(s), in a format to allow rapid processing of rules.

The API will conform to the STARS Design Principles and the STARS API Design Document.

6.1.5 Reference Data Database

The Reference Database will be implemented in Azure SQL. It will utilise a standardised schema to ensure consistency in the management and presentation of reference data. Care will be taken to ensure reference data schemas can be presented (via the Reference Data API) to workflows and the rules engine(s) in a format to allow rapid processing of rules.

6.1.6 Reference Data Event Database

All service events relating to the lifecycle and management of reference data will be recorded, in the STARS Event schema, in Cosmos DB.

6.2 Overview of Reference Data Service

Following are different Azure Services will be utilized in Reference Data Service:

- Azure Logic App
- Azure Functions
- SQL Server Database
- Key Vault
- Application Insight
- App Config

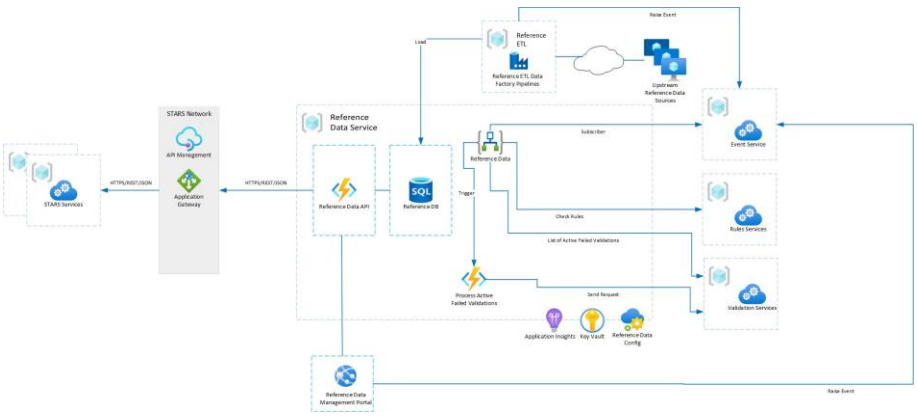


Figure ~~33~~ : Reference Data Service Overview (G1)

Following are different services in the Reference Data Service:

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 403 of 1295

6.2.1 Reference Data API

The Reference Data API is an HTTP-triggered azure function app. Reference data management and query operations will be mapped to relevant function app endpoints.

6.2.2 Reference Data Logic App

The Reference data logic app will subscribe to the Event Service azure service bus, it will be triggered when there is an event raised in the azure service bus. The Reference data logic app will trigger the workflow in the Reference data service to check the rules and validate the active failed validations.

6.2.3 Process Active Failed Validation Function

The process active failed validation azure function will be triggered by the Reference Data Logic App and it will send the request to the Validation Service to validate the active failed validation records.

6.2.4 Reference Data database

The Reference Data Database will be SQL Server relational database.

6.2.5 Reference Data Key Vault

The Reference data will be encrypted at rest, encryption keys to be held in Azure key vault.

6.2.6 Reference Data Config

The Reference Data Config will have configurations relevant to any RG-specific configurations, including for function apps, to be held in a common Azure app config.

7 Business Process

The Reference Data Service provides reference data management with the full history of reference data related events.

The Reference Data Management Portal will provide a user interface to manage reference data.

The Reference Data Management Portal will allow for viewing reference data for all external and internal entities, but it will only support changes for additions or amendments to internal reference data entities.

Formatted: Indent: Left: 0 cm

~~These reference data management functionalities include adding new data record, amending existing data record, mainly to validate, correct invalid or missing reference data to ensure data accuracy.~~

All updates to internal reference data will require an approval process. This process can be automated, for regular, automated updates, but will be manual for any manual changes through the Reference Data Management Portal. The details of such curation activities are pending, and as such, the current design does not include details of this approval process yet and will be updated when such details are provided (See W34 – Action #04).

When reference data update occurs, it should automatically revalidate all active validation failures related to the reference data changed recently.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 404 of 1295

STARS will have regular sync to ingest or integrate data from multiple other sources to ensure values are consistent and match across all systems.

STARS will expose the external reference data via reference data retrieval APIs to external systems.

8 Reference Data Service Workflow

Reference data service has the following workflows:

8.1 Basic Flow of Reference Data Update via Portal

The Reference Data Management Portal will allow for viewing reference data for all external and internal entities, but it will only support changes to internal reference data entities.

System admin will update missing, out of date or incorrect reference data from reference data management portal. The Reference data management portal will send update request to the reference data API.

This Reference Data API will trigger reference data update function to apply requested change in reference database. The request data update function will validate and save updated reference data and publish **Reference Data Updated** event message via Event Service.

Details of the validations required for Reference Data Updates are not known at this time but will be updated as responses to outstanding actions are received (See W34 – Action #04).

Figure 444: Internal Reference Data Update VIA Portal (G2)

~~8.1~~8.2 Revalidation from Reference Data Update

If a Submission is validated, and some records return error responses, it's possible that some of these failures could be due to incorrect, or out of date Reference Data. As such, it would be of benefit to Training Providers if they do not need to re-submit their data with no changes, just to revalidate against updated Reference Data. Therefore, STARS will implement a workflow to revalidate submitted records that have failed against rules that depend on Reference Data that has just been updated.

The following diagram describes the internal process within the Reference Data Service to make this happen. The Rules Service will store relationships between the Validation Rules, and any Reference Data entities that it depends on. The Reference Data service can query the Rules Service API to retrieve the specific Rules that depend on data that has just been updated.

The Validation Service stores all Validation Results which will include the validation outcome, and Rule Identifier for the rules that produce that outcome. The Reference Data Service will call the Validation Service API to gather all Validation Results with a Failed/Error outcome, where the failing rule is one that depends on Reference Data that has just been updated.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 405 of 1295

The Reference Data service will collate these validation results, and initiate revalidation of the Records that have failed against the full set of rulesets that those Records were validated against initially.

The Validation Service will provide a response that will include any validation results that have had outcomes immediately. It's possible some outcomes will still be pending. If any Records have now Passed validation, the Reference Data service will raise an event to indicate that the Record has passed validation.

The Reference Data Service will then poll the Validation Service to check for updated results, for any that were Pending, until all results are completed. As outcomes are returned, any Records that pass validation will have events raised to signify this and allow the Records to continue along the Submission pipeline.

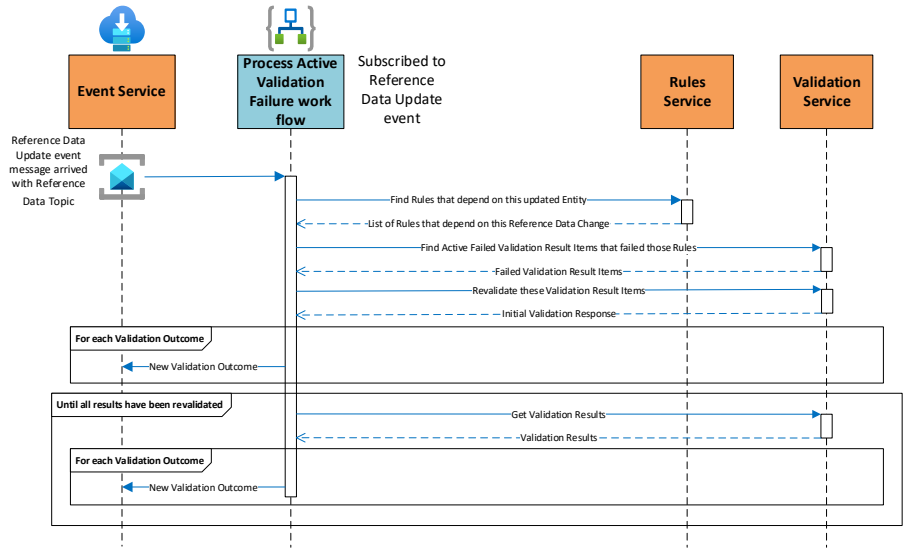


Figure 555: Basic Flow to Process Active Validation Failures (G3)

8.28.3 Basic Flow of Reference Data Inserts via Portal

System admin will add new internal reference data from reference data management portal. The Reference data management portal will send request to add new reference data to the Reference Data API.

This Reference Data API will trigger reference data insert function to add new reference data in reference database. This reference data insert function will save new reference data and will publish Reference Data Added event VIA Event Service.

Details of the validations required for Reference Data Inserts are not known at this time but will be updated as responses to outstanding actions are received (See W34 – Action #04).

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 406 of 1295

Figure 666: Basic Flow of Internal Reference Data Inserts VIA Portal (G4)

9 Reference Data Service Events

The Reference Data Service will raise events via the Events Service. The Reference Data ETL Pipeline and the Reference management portal will raise an event via the Events Service. The Reference Data Service Logic app will be subscribed to the Events Service which will get triggered when the event will be raised in the Azure Service Bused of the Events Service.

Following are the list of the events raised by the Reference Data Service via the Events Service:

Name	Purpose
Reference Data Updated	Message to the Reference Data Service to trigger the workflow when the Reference data is updated in the reference data database.
Reference Data Inserted	Message to the Reference Data Service to trigger the workflow when the Reference data is inserted in the reference data database.

10 Technical Design Patterns

The Reference Data Service will follow SOA (Service Oriented Architecture) design pattern like other STARS services. Loose coupling will be achieved using message- and event-driven architectures, primarily using Azure Service Bus. Message and event payloads conforming to several standardised information JSON-based schemas.

Many services present Application Programming Interfaces (API), typically as HTTP/REST/JSON based endpoints. API designs are presented in a separate document, please refer to the API Design Document (H). API-based services (both internal and external) will be managed through Azure API Management. For external-facing services STARS will utilise Azure Application Gateway as an intermediary between the APIs and their clients.

11 API Design

Reference Data API will be OData REST API using HTTP requests for GET, POST and PUT operations. The service will follow standard ODATA filter query and REST standards for requests and operations. The Reference Data API will receive and send data in JSON format conforming to JSON Schema standards.

The API will conform to the STARS Design Principles and the STARS API Design Document. The endpoints described below will be standard REST endpoints with OData filter query. See the STARS API Design Document for success / failure status codes and expected response headers.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 407 of 1295

11.1 Reference Data API

The Reference Data API is an HTTP-triggered function app. Reference Data management and query operations will be mapped to relevant function app endpoints. The API will be factored around reference Data entities (e.g., Address Type, ANZSCO, Anzsic etc.).

STARS will expose the external reference data via reference data retrieval APIs to external systems.

11.2 Reference Data database

The Reference Database is an Azure SQL DB database that stores Reference data entities.

11.3 Reference Data Event database

The Reference Data Event Database is an Azure Cosmos DB database instance that stores reference data entities lifecycle related events.

11.4 Reference Data Config

Any RG-specific config, including for function apps, to be held in a common Azure app config.

11.5 Reference Data API Endpoints

API endpoints for reference data entities will be consistent for all Reference Data entities, in line with the OData specification.

For each Reference entity the following operations will be available:

11.5.1 Reference Data Retrievals

Formatted: Font: Not Bold

The Reference Data Retrieval APIs supports requesting a specific version of VET-IS reference data by providing the release number/version number as an input parameter with the request. This functionality will be available for all data retrieval endpoints related to version specific reference data entities.

[Section to be updated to retrieve default VET-IS version after the clarification in stage 2 based on DD03]

The reference data retrievals for state specific reference data entities will only be accessible by state users.

Formatted: Space After: 0 pt, Line spacing: single

Query: The query endpoint is implemented using OData. It is a HTTP GET endpoint. This endpoint will be used to query the reference data entity based on OData filter parameters. This will provide a list of reference data entities based on filters on the OData request URL.

~~**Add**: The add endpoint will be the same as the query endpoint with HTTP POST verb and the body of the request will contain the JSON representation of the entity to be created.~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 408 of 1295

~~**Update**: This update endpoint will be the same endpoint as the query with HTTP PUT/PATCH verb. The body of the request will contain a JSON PUT/PATCH representing the changes that need to be made to the reference data entity for a PUT/PATCH request.~~

As an example, to use an OData query with a filter to query for AddressTypes with a Code of 'O', the query would look like:~~As an example, if querying for AddressTypes with Code of 'O', the query would look like:~~

<baseUrl>/referenceData/AddressTypes?$filter=Code eq 'O'

The result would be a JSON array with the response content containing information about the reference data, similar to the following:~~The result would be a JSON array with the similar to the following:~~


```
[{
        "Id":"535464567",
        "Code":"O",
        "Name":"Office Address",
        "Description":"Specifies Address Type for a Business Location",
        "EffectiveDateFrom":"2001-01-01",
        "EffectiveDateTo":"2040-12-31",
        "CreatedBy":"UserCreated",
        "ModifiedBy":"UserUpdated",
        "CreatedDate":"2001-01-01T12:00:00.000000+00:00",
        "ModifiedDate":"2022-12-10T12:00:00.000000+00:00",
        "ReleaseNumber": "1.0"
        "VersionNumber": "1.0",
        "IsCurrent": true,
        "IsDeleted": false

}]
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 409 of 1295

This style of API interactions applies to all Reference Data entities. The payload can be different for each entity, and the fields used for filtering, selecting, expanding etc will be specific to each Reference Data Entity, too.

Each reference data entity, however, will have ~~some~~ certain common attributes as specified below, to allow consistent handling of reference data by consuming services.

~~These are~~**Common Attributes**: Id, Code, Name, Description, EffectiveDateFrom, EffectiveDate~~T~~to, CreatedBy, ModifiedBy, CreatedDate~~, and~~ ModifiedDate, ReleaseNumber, VersionNumber, IsCurrent and IsDeleted.

Formatted: Font: Bold

For details on OData query endpoints and parameters refer to OData in ASP.NET Web API - ASP.NET Core 2.x

For details on modifying entities through OData refer to Data Modification- POST - OData and Data Modification - UPDATE - OData

For details of OData Actions and functions refer to Actions and Functions – OData

11.5.2 Reference Data Amendments

Formatted: Font: (Default) Calibri, 14 pt, Not Bold, Font color: Text 1

Formatted: Indent: Left: 0 cm

The Reference Data Management Portal will allow for viewing reference data for all external and internal entities, however it will only support changes for additions or amendments to internal reference data entities.

The changes to reference data for add and update endpoints will be limited to certain internal entities based on functional requirements.

Add: The add endpoint will be the same as the query endpoint with HTTP POST verb and the body of the request will contain the JSON representation of the entity to be created.

Update: This update endpoint will be the same endpoint as the query with HTTP PUT/PATCH verb. The body of the request will contain a JSON PUT/PATCH representing the changes that need to be made to the reference data entity for a PUT/PATCH request.

12 Reference Data Sync Solution Components

The below picture shows overall high-level STARS target state data architecture, section 12 in this document will cover the Reference Data ~~s~~Sync ~~Services~~ solution components (numbered 1, 2 & 3).

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 410 of 1295

Figure 7~~7~~: STARS Azure Data Estate (I2)

Table 2~~2~~: Reference Data Sync Steps

Step No	Description	Technologies
1	Extract/Share reference data from external sources like ~~NCVER,~~ States, Australia Post and ASQA into central repository	Azure Data Factory, Azure Data Share, Azure Data Lake
2	Reference data is stored in the central repository in Azure Data Lake	Azure Data Lake, Azure Synapse
3	Load reference data from Azure Data Lake into Reference Database	Azure Data Factory, Azure SQL Database

12.1 Data Source and Environment

This section covers the source environment details that will be required in Stage 2 and Stage 3 release.

Table 3~~3~~: Reference Data Source DEV Environment

Source	Data Integration	Dev Details
NCVER (CARMS)	ADF – NCVER Data Source	~~To be provided~~Details will be added after NCVER Integration with STARS is finalised
State – SA	Azure Data Share	~~To be provided~~Details will be available in Stage 3 release

Table 4~~4~~: Reference Data Source TEST Environment

Source	Data Integration	Test Details
NCVER (CARMS)	ADF – NCVER Data Source	Details will be added after NCVER Integration with STARS is finalised~~To be provided~~
State – SA	Azure Data Share	Details will be available in Stage 3 release~~To be provided~~

Table 5~~5~~: Reference Data Source SIT Environment

Source	Data Integration	SIT Details
NCVER (CARMS)	ADF – NCVER Data Source	Details will be added after NCVER Integration with STARS is finalised~~To be provided~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 411 of 1295

State – SA	Azure Data Share	Details will be available in Stage 3 release~~To be provided~~

Table 6~~6~~: Reference Data Source UAT Environment

Source	Data Integration	UAT Details
NCVER (CARMS)	ADF – NCVER Data Source	Details will be added after NCVER Integration with STARS is finalised~~To be provided~~
State – SA	Azure Data Share	Details will be available in Stage 3 release~~To be provided~~

Table 7~~7~~: Reference Data Source PROD Environment

Source	Data Integration	Prod Details
NCVER (CARMS)	ADF – NCVER Data Source	Details will be added after NCVER Integration with STARS is finalised~~To be provided~~
State – SA	Azure Data Share	Details will be available in Stage 3 release~~To be provided~~

12.2 Target Databases

STARS Reference database will be implemented in Azure SQL Database. Server and database details for each environment as below.

Table 8~~8~~: STARS Reference Database Environment

Environment	Details
Dev	Server Name: vds-stars-aue-sqls-ref-dev Database Name: vds-stars-sqldb-ref-dev
Test	Server Name: vds-stars-aue-sqls-ref-test Database Name: vds-stars-sqldb-ref-test
SIT	Server Name: vds-stars-aue-sqls-ref-sit Database Name: vds-stars-sqldb-ref-sit
UAT	Server Name: vds-stars-aue-sqls-ref-uat Database Name: vds-stars-sqldb-ref-uat
Production	Server Name: vds-stars-aue-sqls-ref-prd Database Name: vds-stars-sqldb-ref-prd

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 412 of 1295

12.3 Reference Data Entities

Below are the reference data entities that will be extracted from NCVER (CARMS) and States and stored in the STARS Reference Database.

12.3.1 NCVER (CARMS)

Below are the data entities that will be maintained in STARS. These entities will be created under schema name ~~<ref>~~<internal> or <external> based on the source column below.

Sources such as TGA and ABS are external and their entities will be stored in *external* schema in the database.

Sources such as ~~NCVER~~CARMS are internal and their entities will be stored in *internal* schema in the database.

TGA reference entities needs to be confirmed with NCVER and added to the list below, open action item 2856.

Table 999: CARMS Reference Entities

Entity	Attribute	Data Type	Domain Name	Source
Address Type	Address Type Name	NVARCHAR(100)	Name	Internal
Address Type	Address Type Description	NVARCHAR(500)	Description	Internal
Address Type	Address Type Code	NVARCHAR(256)	Code	Internal
ANZSCO	ANZSCO Unit Group Code	NVARCHAR(256)	Code	External
ANZSCO	ANZSCO Sub Major Group Code	NVARCHAR(256)	Code	External
ANZSCO	ANZSCO Occupation Title Name	NVARCHAR(100)	Name	External
ANZSCO	ANZSCO Minor Group Code	NVARCHAR(256)	Code	External
ANZSCO	ANZSCO Major Group Code	NVARCHAR(256)	Code	External
ANZSCO	ANZSCO Code	NVARCHAR(256)	Code	External
ANZSCO Major Group	ANZSCO Major Group Title Name	NVARCHAR(100)	Name	External
ANZSCO Major Group	ANZSCO Major Group Code	NVARCHAR(256)	Code	External

Formatted: Font: Italic

Formatted: Font: Italic

Formatted Table

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 413 of 1295

ANZSCO Minor Group	ANZSCO Minor Group Title Name	NVARCHAR(100)	Name	External
ANZSCO Minor Group	ANZSCO Minor Group Code	NVARCHAR(256)	Code	External
ANZSCO Sub Major Group	ANZSCO Sub Major Group Code	NVARCHAR(256)	Code	External
ANZSCO Sub Major Group	ANZSCO Sub Major Group Title Name	NVARCHAR(~~100~~256)	Name	External
ANZSCO Unit Group	ANZSCO Unit Group Code	NVARCHAR(256)	Code	External
ANZSCO Unit Group	ANZSCO Unit Group Title Name	NVARCHAR(~~100~~256)	Name	External
ANZSIC	ANZSIC Description	NVARCHAR(500)	Description	External
ANZSIC	ANZSIC Division Code	NVARCHAR(256)	Code	External
ANZSIC	ANZSIC Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	External
ANZSIC Division	ANZSIC Division Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	External
ANZSIC Division	ANZSIC Division Title Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	External
ASCED Broad Field of Education	ASCED Broad Field of Education Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	External
ASCED Broad Field of Education	ASCED Broad Field of Education Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	External
ASCED Broad Level of Education	ASCED Broad Level of Education Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	External
ASCED Broad Level of Education	ASCED Broad Level of Education Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	External
ASCED Detailed Field of Education	ASCED Narrow Field of Education Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	External
ASCED Detailed Field of Education	ASCED Detailed Field of Education Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	External
ASCED Detailed Field of Education	ASCED Broad Field of Education Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	External

ASCED Detailed Field of Education	ASCED Detailed Field of Education Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	External
ASCED Detailed Level of Education	ASCED Detailed Level of Education Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	External
ASCED Detailed Level of Education	ASCED Narrow Level of Education Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	External
ASCED Detailed Level of Education	ASCED Detailed Level of Education Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	External
ASCED Detailed Level of Education	ASCED Broad Level of Education Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	External
ASCED Narrow Field of Education	ASCED Narrow Field of Education Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	External
ASCED Narrow Field of Education	ASCED Broad Field of Education Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	External
ASCED Narrow Field of Education	ASCED Narrow Field of Education Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	External
ASCED Narrow Level of Education	ASCED Narrow Level of Education Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	External
ASCED Narrow Level of Education	ASCED Narrow Level of Education Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	External
ASCED Narrow Level of Education	ASCED Broad Level of Education Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	External
Building Complex Sub Unit Type	Building Complex Sub Unit Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Building Complex Sub Unit Type	Building Complex Sub Unit Type Description	NVARCHAR(500)	Description	Internal
Building Complex Sub Unit Type	Building Complex Sub Unit Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Citizenship Type	Citizenship Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Citizenship Type	Citizenship Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 415 of 1295

Country	Country Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	External
Country	Country Abbreviation	NVARCHAR(256)~~NVARCHAR(25)~~	Code	External
Country	Country Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	External
Disability Type	Disability Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Disability Type	Disability Description	NVARCHAR(500)	Description	Internal
Disability Type	Disability Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Educational Achievement Type	Educational Achievement Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Educational Achievement Type	Educational Achievement Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Educational Achievement Type	Educational Achievement Type Description	NVARCHAR(500)	Description	Internal
Electronic Address Type	Electronic Address Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Electronic Address Type	Electronic Address Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Electronic Address Type	Electronic Address Type Description	NVARCHAR(500)	Description	Internal
Enrolment Liability Type	Enrolment Liability Type Name	NVARCHAR(256)	Name	Internal
Enrolment Liability Type	Enrolment Liability Type Code	NVARCHAR(256)	Code	Internal
Enrolment Liability Type	Enrolment Liability Type Description	NVARCHAR(500)	Description	Internal
Financial Detail Payment Contribution Type	Financial Detail Payment Contribution Type Name	NVARCHAR(~~100~~256)	Name	Internal

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 416 of 1295

Financial Detail Payment Contribution Type	Financial Detail Payment Contribution Type Description	NVARCHAR(500)	Description	Internal
Financial Detail Payment Contribution Type	Financial Detail Payment Contribution Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Gender Type	Gender Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Gender Type	Gender Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Gender Type	Gender Type Description	NVARCHAR(500)	Description	Internal
Government Type	Government Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Government Type	Government Type Description	NVARCHAR(500)	Description	Internal
Government Type	Government Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Highest School Level Completed	Highest School Level Completed Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Highest School Level Completed	Highest School Level Completed Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Highest School Level Completed	Highest School Level Completed Description	NVARCHAR(500)	Description	Internal
Indigenous Status	Indigenous Status Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Indigenous Status	Indigenous Status Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Indigenous Status	Indigenous Status Description	NVARCHAR(500)	Description	Internal
International Student Type	International Student Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
International Student Type	International Student Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 417 of 1295

International Student Type	International Student Type Description	NVARCHAR(500)	Description	Internal
Labour Force Status	Labour Force Status Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Labour Force Status	Labour Force Status Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Labour Force Status	Labour Force Status Description	NVARCHAR(500)	Description	Internal
Language	Language Code	NVARCHAR(256)	Code	Internal
Language	Language Description	NVARCHAR(500)	Description	Internal
Language	Language Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Learning Channel Type	Learning Channel Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Learning Channel Type	Learning Channel Type Description	NVARCHAR(500)	Description	Internal
Learning Channel Type	Learning Channel Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Learning Delivery Type	Learning Delivery Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Learning Delivery Type	Learning Delivery Type Description	NVARCHAR(500)	Description	Internal
Learning Delivery Type	Learning Delivery Type Code	NVARCHAR(256)	Code	Internal
Location Relationship Type	Location Relationship Type Description	NVARCHAR(500)	Description	Internal
Location Relationship Type	Location Relationship Type Code	NVARCHAR(256)	Code	Internal
Location Relationship Type	Location Relationship Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Location Type	Location Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal

Location Type	Location Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Location Type	Location Type Description	NVARCHAR(500)	Description	Internal
Organisation Type	Organisation Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Organisation Type	Organisation Type Description	NVARCHAR(500)	Description	Internal
Organisation Type	Organisation Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Person Language Context	Person Language Context Code	NVARCHAR(256)	Code	Internal
Person Language Context	Person Language Context Description	NVARCHAR(500)	Description	Internal
Person Language Context	Person Language Context Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Person Title	Person Title Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Person Title	Person Title Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Physical Address Type	Physical Address Type Description	NVARCHAR(500)	Description	Internal
Physical Address Type	Physical Address Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Physical Address Type	Physical Address Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Place Type	Place Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Place Type	Place Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Place Type	Place Type Description	NVARCHAR(500)	Description	Internal
Postal Delivery Type	Postal Delivery Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 419 of 1295

Postal Delivery Type	Postal Delivery Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Postcode	Postcode	NVARCHAR(256)~~NVARCHAR(25)~~	Code	External
Product Category	Product Category Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Product Category	Product Category Description	NVARCHAR(500)	Description	Internal
Product Category	Product Category Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Product Class	Product Class Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Product Class	Product Class Description	NVARCHAR(500)	Description	Internal
Product Class	Product Class Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Product Type	Product Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Product Type	Product Type Description	NVARCHAR(500)	Description	Internal
Product Type	Product Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Program Category	Program Category Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Program Category	Program Category Description	NVARCHAR(500)	Description	Internal
Program Category	Program Category Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Program Class	Program Class Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Program Class	Program Class Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Program Class	Program Class Description	NVARCHAR(500)	Description	Internal

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 420 of 1295

Program Type	Program Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Program Type	Program Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Program Type	Program Type Description	NVARCHAR(500)	Description	Internal
Recognised Achievement Type	Recognised Achievement Type Code	NVARCHAR(256)	Code	Internal
Recognised Achievement Type	Recognised Achievement Type Description	NVARCHAR(500)	Description	Internal
Recognised Achievement Type	Recognised Achievement Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Resource Responsibility Type	Resource Responsibility Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Resource Responsibility Type	Resource Responsibility Type Description	NVARCHAR(500)	Description	Internal
Resource Responsibility Type	Resource Responsibility Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Responsibility Engagement Method	Responsibility Engagement Method Code	NVARCHAR(256)	Code	Internal
Responsibility Engagement Method	Responsibility Engagement Method Name	NVARCHAR(500)	Description	Internal
Responsibility Engagement Method	Responsibility Engagement Method Description	NVARCHAR(256)	Name	Internal
Responsibility Financial Detail Amount Type	Responsibility Financial Detail Amount Description	NVARCHAR(500)	Description	Internal
Responsibility Financial Detail Amount Type	Responsibility Financial Detail Amount Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Responsibility Financial Detail Amount Type	Responsibility Financial Detail Amount Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 421 of 1295

Responsibility Financial Detail Category	Responsibility Financial Detail Class Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Responsibility Financial Detail Category	Responsibility Financial Detail Category Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Responsibility Financial Detail Category	Responsibility Financial Detail Category Description	NVARCHAR(500)	Description	Internal
Responsibility Financial Detail Category	Responsibility Financial Detail Category Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Responsibility Financial Detail Class	Responsibility Financial Detail Class Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Responsibility Financial Detail Class	Responsibility Financial Detail Class Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Responsibility Financial Detail Class	Responsibility Financial Detail Class Description	NVARCHAR(500)	Description	Internal
Responsibility Financial Detail Type	Responsibility Financial Detail Category Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Responsibility Financial Detail Type	Responsibility Financial Detail Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Responsibility Financial Detail Type	Responsibility Financial Detail Type Description	NVARCHAR(500)	Description	Internal
Responsibility Financial Detail Type	Responsibility Financial Detail Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Responsibility Identifier Type	Responsibility Identifier Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Responsibility Identifier Type	Responsibility Identifier Type Description	NVARCHAR(500)	Description	Internal
Responsibility Identifier Type	Responsibility Identifier Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Responsibility Lifecycle Reason Type	Responsibility Lifecycle State Reason Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Responsibility Lifecycle Reason Type	Responsibility Lifecycle State Reason Type Description	NVARCHAR(500)	Description	Internal

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 422 of 1295

Responsibility Lifecycle Reason Type	Responsibility Lifecycle State Reason Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Responsibility Lifecycle State Type	Responsibility Lifecycle State Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Responsibility Lifecycle State Type	Responsibility Lifecycle State Type Description	NVARCHAR(500)	Description	Internal
Responsibility Lifecycle State Type	Responsibility Lifecycle State Type Code	NVARCHAR(256)	Code	Internal
Responsibility Location Usage Type	Responsibility Location Usage Type Description	NVARCHAR(500)	Description	Internal
Responsibility Location Usage Type	Responsibility Location Usage Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Responsibility Location Usage Type	Responsibility Location Usage Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Responsibility Relationship Type	Responsibility Relationship Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Responsibility Relationship Type	Responsibility Relationship Type Description	NVARCHAR(500)	Description	Internal
Responsibility Relationship Type	Responsibility Relationship Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Responsibility Type	Responsibility Category Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Responsibility Type	Responsibility Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Responsibility Type	Responsibility Type Description	NVARCHAR(500)	Description	Internal
Responsibility Type	Responsibility Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
School Level	School Level Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
School Level	School Level Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 423 of 1295

School Level	School Level Description	NVARCHAR(500)	Description	Internal
Secondary School Type	Secondary School Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Secondary School Type	Secondary School Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Secondary School Type	Secondary School Type Description	NVARCHAR(500)	Description	Internal
Stakeholder Identification Type	Stakeholder Identification Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Stakeholder Identification Verification Status	Stakeholder Identification Verification Status Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Stakeholder Identification Verification Status	Stakeholder Identification Verification Status Description	NVARCHAR(500)	Description	Internal
Stakeholder Identification Verification Status	Stakeholder Identification Verification Status Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Stakeholder Identifier Type	Stakeholder Identifier Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Stakeholder Identifier Type	Stakeholder Identifier Type Description	NVARCHAR(500)	Description	Internal
Stakeholder Identifier Type	Stakeholder Identifier Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Stakeholder Lifecycle State	Stakeholder Lifecycle State Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Stakeholder Lifecycle State Reason	Stakeholder Lifecycle State Reason Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Stakeholder Location Usage Type	Stakeholder Location Usage Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Stakeholder Location Usage Type	Stakeholder Location Usage Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Stakeholder Location Usage Type	Stakeholder Location Usage Type Description	NVARCHAR(500)	Description	Internal

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 424 of 1295

Stakeholder Name Type	Stakeholder Name Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Stakeholder Name Type	Stakeholder Name Type Description	NVARCHAR(500)	Description	Internal
Stakeholder Name Type	Stakeholder Name Type Code	NVARCHAR(256)	Code	Internal
Stakeholder Preference Reason Type	Stakeholder Preference Reason Type Code	NVARCHAR(256)	Code	Internal
Stakeholder Preference Reason Type	Stakeholder Preference Reason Type Description	NVARCHAR(500)	Description	Internal
Stakeholder Preference Reason Type	Stakeholder Preference Reason Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Stakeholder Preference Type	Stakeholder Preference Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Stakeholder Preference Type	Stakeholder Preference Type Description	NVARCHAR(500)	Description	Internal
Stakeholder Preference Type	Stakeholder Preference Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Stakeholder Preference Type Value	Stakeholder Preference Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Stakeholder Preference Type Value	Stakeholder Preference Type Value Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Stakeholder Preference Type Value	Stakeholder Preference Type Value Description	NVARCHAR(500)	Description	Internal
Stakeholder Preference Type Value	Stakeholder Preference Type Value Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Stakeholder Relationship Type	Stakeholder Relationship Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Stakeholder Relationship Type	Stakeholder Relationship Type Description	NVARCHAR(500)	Description	Internal
Stakeholder Relationship Type	Stakeholder Relationship Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 425 of 1295

Stakeholder Responsibility Type	Stakeholder Responsibility Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Stakeholder Responsibility Type	Stakeholder Responsibility Type Description	NVARCHAR(500)	Description	Internal
Stakeholder Responsibility Type	Stakeholder Responsibility Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Stakeholder Role Type	Stakeholder Role Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Stakeholder Role Type	Stakeholder Role Type Description	NVARCHAR(500)	Description	Internal
Stakeholder Role Type	Stakeholder Role Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Stakeholder Type	Stakeholder Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Stakeholder Type	Stakeholder Type Description	NVARCHAR(500)	Description	Internal
Stakeholder Type	Stakeholder Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
State	State Abbreviation	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
State	State Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
State	State Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Street Address Type	Street Address Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Street Address Type	Street Address Type Description	NVARCHAR(500)	Description	Internal
Street Address Type	Street Address Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Street Type	Street Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 426 of 1295

Street Type	Street Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Study Reason	Study Reason Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Study Reason	Study Reason Description	NVARCHAR(500)	Description	Internal
Study Reason	Study Reason Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Subject Assessment Type	Subject Assessment Type Description	NVARCHAR(500)	Description	Internal
Subject Assessment Type	Subject Assessment Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Subject Assessment Type	Subject Assessment Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Subject Category	Subject Category Description	NVARCHAR(500)	Description	Internal
Subject Category	Subject Category Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Subject Category	Subject Category Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Subject Class	Subject Class Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Subject Class	Subject Class Description	NVARCHAR(500)	Description	Internal
Subject Class	Subject Class Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Subject Type	Subject Type Description	NVARCHAR(500)	Description	Internal
Subject Type	Subject Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Subject Type	Subject Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Suburb	Suburb Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Telecommunications Address Type	Telecommunications Address Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 427 of 1295

Telecommunications Address Type	Telecommunications Address Type Description	NVARCHAR(500)	Description	Internal
Telecommunications Address Type	Telecommunications Address Type Code	NVARCHAR(256)	Code	Internal
Training Organisation Type	Training Organisation Type Description	NVARCHAR(500)	Description	Internal
Training Organisation Type	Training Organisation Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Training Organisation Type	Training Organisation Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Visa Type	Visa Type Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Visa Type	Visa Type Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal
Vocational Education Purpose	Vocational Education Purpose Code	NVARCHAR(256)~~NVARCHAR(25)~~	Code	Internal
Vocational Education Purpose	Vocational Education Purpose Name	NVARCHAR(256)~~NVARCHAR(100)~~	Name	Internal

Table 101~~0~~: CARMS Attribute Nullability

Attribute	Nullability
Code	NOT NULL
Name	NULL
Description	NULL

Formatted: Caption
Formatted: Font color: Background 1
Formatted: Space Before: Auto, After: Auto
Formatted Table

12.3.2 States

12.3.2.1 ACT

ACT does not currently require any reference data in STARS outside that which is required by NCVER and for national compliance with the standard (covered in section 12.3.1). ~~There is a requirement to load Funding source reference data as below into STARS to validate.~~

~~As per discussion with NCVER & DEWR, Funding source codes will be added into the classifications (CARMS), which will eventually flow into STARS. Funding source is mapped to "Responsibility~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 428 of 1295

~~Financial Detail Type Code" in CARMS (Entity: Responsibility Financial Detail Type), refer section 12.3.1.~~

Formatted: Normal

12.3.2.2Queensland

QLD does not currently require any reference data in STARS outside that which is required by NCVER and for national compliance with the standard (covered in section 12.3.1). There will be requirements in the future and when the requirements are finalised for QLD, the integration and reference data load with NT will follow the same process as described in section 12.5 Data Integration.

12.3.2.3New South Wales

NSW does not currently require any additional reference data in STARS outside that which is required by NCVER and for national compliance with the standard (covered in section 12.3.1).

~~Below Reference data entity requirements for NSW are already covered in section 12.3.1 CARMS.~~

~~Table 10: NSW Required STARS Reference Entities~~

~~Entity~~
~~Responsibility Lifecycle Reason Type~~
~~Responsibility Lifecycle State Type~~
~~Subject Assessment Type~~
~~Responsibility Financial Detail Type~~
~~Responsibility Identifier Type~~

~~Below are the NSW state specific fields that are required for state validations:~~

1. ~~Purchasing Contract Identifier~~
2. ~~Outcome Identifier- Training Organisation~~
3. ~~Funding source state training authority~~
4. ~~Parchment Issue date~~
5. ~~Parchment number~~
6. ~~Predominant Delivery Mode~~
7. ~~Commitment ID (Outstanding action item W34 – Action #01)~~

12.3.2.4 Northern Territory

The NT does not currently require any reference data in STARS outside that which is required by NCVER and for national compliance with the standard (covered in section 12.3.1). There will be requirements in the future after the publication of release 1 of the STARS and after NTs boarding in late 2024/early 2025 as development and fine tuning of validation rules are complete from their end.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 429 of 1295

In future when the requirements are finalised for NT, the integration and reference data load with NT will follow the same process as described in section 12.5 Data Integration.

12.3.2.5 South Australia

Below Reference data will be ingested into STARS for internal validation and sharing with SMS's via API. These entities will be created under schema name <sa>.

1. School locations
2. Funding Source ~~– STA (mapped to Responsibility financial detail type code)~~
3. Non-registered training organisations
4. Non-Accredited Programs/Skillsets
5. Non-Accredited Modules/subjects

12.3.2.5.1 School Location

Table 1111~~1011~~: SA Reference Entity - School Location

Entity	Attribute	Data Type	Nullability
School Location ID	Sace School Code	NVARCHAR(256)	NOT NULL
School Location ID	School Name	NVARCHAR(500)	NOT NULL
School Location ID	Suburb	NVARCHAR(256)~~NVARCHAR(100)~~	NOT NULL
School Location ID	Postcode	NVARCHAR(256)~~NVARCHAR(25)~~	NOT NULL
School Location ID	State	NVARCHAR(256)~~NVARCHAR(25)~~	NOT NULL
School Location ID	Active From	DATETIME	NOT NULL
School Location ID	Active To	DATETIME	NOT NULL
School Location ID	School Location ID	NVARCHAR(25)	NOT NULL

Formatted Table

12.3.2.5.2 Funding Source

~~Mapped to "Responsibility Financial Detail Type Code" in CARMS (Entity: Responsibility Financial Detail Type), refer section 12.3.1.~~

Formatted: Caption, Keep with next

Table 1212~~11~~: SA Reference Entity – Funding Source

Formatted: Caption

Entity	Attribute	Data Type	Nullability
Funding Source	State	NVARCHAR(256)	NOT NULL
Funding Source	Fund Source State Id	NVARCHAR(256)	NOT NULL

Formatted Table

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 430 of 1295

Funding Source	Fund Source State Description	NVARCHAR(500)	NOT NULL
Funding Source	Fund Source National Id	SMALLINMT	NOT NULL
Funding Source	Effective From	DATETIME	NOT NULL
Funding Source	Effective To	DATETIME	NOT NULL
Funding Source	Contract Validation RegEx	NVARCHAR(256)	NULL
Funding Source	Contract Validation Error Message	NVARCHAR(1000)	NULL
Funding Source	Specific Funding Id	SMALLINMT	NULL
Funding Source	Create Date Time	DATETIME	NOT NULL
Funding Source	Update Date Time	DATETIME	NOT NULL
Funding Source	Delete Date Time	DATETIME	NULL

12.3.2.5.3 Non-registered Training Organisation

Table 131~~312~~: SA Reference Entity - Non-registered Training Organisations

Entity	Attribute	Data Type	Nullability
Non-registered Training Organisation	State	NVARCHAR(256)	NOT NULL
Non-registered Training Organisation	Legal Name	NVARCHAR(500)	NOT NULL
Non-registered Training Organisation	Training Org ID	NVARCHAR(~~100~~256)	NOT NULL
Non-registered Training Organisation	Trading Name	NVARCHAR(500)	NOT NULL
Non-registered Training Organisation	ABN	NVARCHAR(~~50~~256)	NULL
Non-registered Training Organisation	ACN	NVARCHAR(~~50~~256)	NULL
Non-registered Training Organisation	Organisation Type	NVARCHAR(~~100~~256)	NULL

Formatted Table

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 431 of 1295

Non-registered Training Organisation	Address Line 1	NVARCHAR(1000)	NULL
Non-registered Training Organisation	Address Line 2	NVARCHAR(1000)	NULL
Non-registered Training Organisation	City Suburb	NVARCHAR(~~100~~256)	NULL
Non-registered Training Organisation	Post Code	NVARCHAR(~~25~~256)	NULL
Non-registered Training Organisation	Telephone	NVARCHAR(~~100~~256)	NULL
Non-registered Training Organisation	Email	NVARCHAR(~~100~~256)	NULL
Non-registered Training Organisation	Primary Contact	NVARCHAR(500)	NULL
Non-registered Training Organisation	Created On	DATETIME	NOT NULL
Non-registered Training Organisation	Modified On	DATETIME	NOT NULL
Non-registered Training Organisation	Status Code	NVARCHAR(~~100~~256)	NULL
Non-registered Training Organisation	Effective From	DATETIME	NOT NULL
Non-registered Training Organisation	Effective To	DATETIME	NOT NULL

12.3.2.5.4 Non-Accredited Training Module

Table 141~~413~~: SA Reference Entity - Non-Accredited Training Module

Entity	Attribute	Data Type	
Non-Accredited Training Module	State	NVARCHAR(~~25~~256)	NOT NULL
Non-Accredited Training Module	State Subject Identifier	NVARCHAR(~~100~~256)	NOT NULL

Formatted Table

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 432 of 1295

Non-Accredited Training Module	Subject Name	NVARCHAR(500)	NOT NULL
Non-Accredited Training Module	Unit Code	NVARCHAR(~~55~~256)	NOT NULL
Non-Accredited Training Module	Nominal Hours	NVARCHAR(~~25~~256)	NULL
Non-Accredited Training Module	Accredited Indicator	NVARCHAR(~~100~~256)	NULL
Non-Accredited Training Module	Sector	NVARCHAR(256)	NULL
Non-Accredited Training Module	Effective From	DATETIME	NOT NULL
Non-Accredited Training Module	Effective To	DATETIME	NOT NULL
Non-Accredited Training Module	Created On	DATETIME	NOT NULL
Non-Accredited Training Module	Modified On	DATETIME	NOT NULL
Non-Accredited Training Module	ANZSCO	NVARCHAR(256)	NULL
Non-Accredited Training Module	Allow Academic Pass	NVARCHAR(256)	NULL
Non-Accredited Training Module	Supersedes Code	NVARCHAR(256)	NULL
Non-Accredited Training Module	Product Type	NVARCHAR(256~~100~~)	NULL
Non-Accredited Training Module	Status	NVARCHAR(256~~100~~)	NULL

12.3.2.5.5 Non-Accredited Program

Table 151~~14~~: SA Reference Entity - Non-Accredited Program

Entity	Attribute	Data Type

Formatted Table

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 433 of 1295

Non-Accredited Program	Program Type	NVARCHAR(~~50~~256)	NOT NULL
Non-Accredited Program	State	NVARCHAR(~~10~~256)	NOT NULL
Non-Accredited Program	Program Identifier	NVARCHAR(~~100~~256)	NOT NULL
Non-Accredited Program	Program Name	NVARCHAR(500)	NOT NULL
Non-Accredited Program	Maximum Nominal Hour	NVARCHAR(256)	NULL
Non-Accredited Program	Minimum Nominal Hour	NVARCHAR(256)	NULL
Non-Accredited Program	Program Recognition Identifier	NVARCHAR(~~100~~256)	NULL
Non-Accredited Program	Program Level Of Education Identifier	NVARCHAR(256)	NULL
Non-Accredited Program	Program Field Of Education Identifier	NVARCHAR(256)	NULL
Non-Accredited Program	ANZSCO Identifier	NVARCHAR(256)	NULL
Non-Accredited Program	VET Flag	NVARCHAR(256)	NULL
Non-Accredited Program	Effective From	DATETIME	NOT NULL
Non-Accredited Program	Effective To	DATETIME	NOT NULL

Formatted: Left

12.3.2.6 Tasmania

No additional reference data required for TAS as of now, assuming the NCVER (CARMS) reference data will suffice TAS data validations.

12.3.2.7 Victoria

No additional reference data required for Victoria, the NCVER (CARMS) reference data will suffice national level validations.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 434 of 1295

Victoria will do its own validations and hold the needed state specific reference data.

12.3.2.8 Western Australia

No additional reference data required for WA, the NCVER (CARMS) reference data will suffice national level validations.

WA will manage all SS (State Specific) validations and maintain their SS reference data (purchasing contract IDs, local curriculum).

12.4 Reference Data Auditing and Versioning

Reference Data will maintain version history for both the external and internal source data ~~be held in temporal tables, for audit and time-based analysis~~.

~~Temporal tables (SYSTEM_VERSIONING) will be used to retain history on applicable objects in the Azure SQL Reference database. Changes to or audit on data will be tracked using this mechanism. Temporal tables are a programmability feature of Azure SQL Database that allows the tracking and analysis of the full history of changes in data, without the need for custom coding.~~

In addition to the above, all reference tables will include the below attributes for auditing purposes.

- Effective-Date-From
- Effective-Date-To
- Created-Date
- Created-By
- Modified-Date
- Modified-By
- ETLBatchId
- ReleaseNumber~
- VersionNumber~
- IsDeleted*
- IsCurrent

**IsDeleted flag added to allow (soft) deletes if entered in error. TGA do this and will represent a significant portion of STARS reference data.*

~Release and Version number added to accommodate VIS versioning. This might change after consultation with NCVER in stage2.

Table 161~~6~~: Audit Attributes

Audit Attribute	Data Type and Size	Nullability
EffectiveDateFrom	DATETIME2	NOT NULL
EffectiveDateTo	DATETIME2	NULL

Formatted: Font: Not Bold, Italic

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Normal

Formatted: Font color: Background 1

Formatted: Font color: Background 1

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 435 of 1295

CreatedDate	DATETIME2	NOT NULL
CreatedBy	NVARCHAR(256)	NOT NULL
ModifiedDate	DATETIME2	NOT NULL
ModifiedBy	NVARCHAR(256)	NOT NULL
ETLBatchId	UNIQUEIDENTIFIER	NULL
ReleaseNumber	NVARCHAR(25)	NULL
VersionNumber	NVARCHAR(25)	NULL
IsDeleted	TINYINT	NOT NULL , Default 0 1- Deleted 0- Not Deleted
IsCurrent	TINYINT	NOT NULL (Default 1) 1- Active 0- In Active

Formatted: Normal, No bullets or numbering

12.5 Reference Data Integration

At the time of writing this document reference data will be sourced from

- CARMS*
- State – SA Only

** The sourcing of reference data from CARMS is only a transition state. In the final state (Stage 4 release), STARS will manage its own reference data directly from source.*

Formatted: Font: Italic

Formatted: Font: Italic

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 436 of 1295

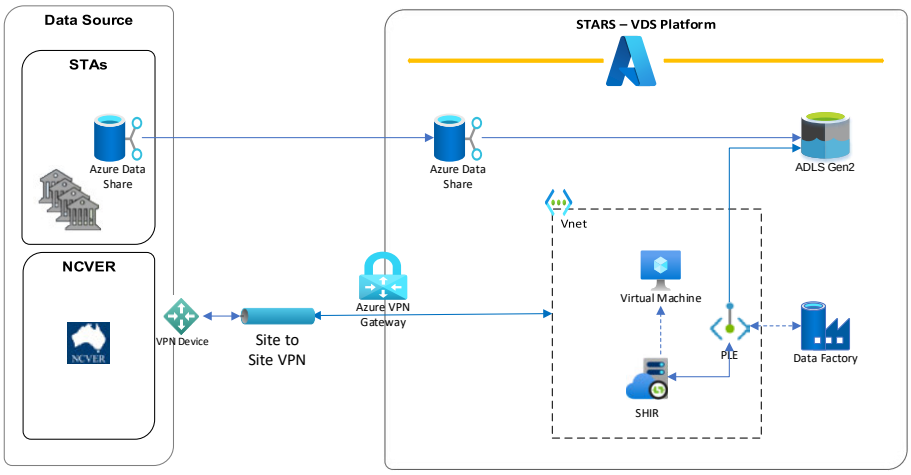


Figure 8~~8~~: Reference Data Integration Technical Design (I3)

NCVER(CARMS) Data Integration:

This integration is to be confirmed with NCVER for stage 2, more details on the staged approach in the following section.

Azure Data Factory is the logical and secure service to use for ETL operations in STARS. Azure Data Factory will connect to external data sources through a feature called Self-Hosted Integrated Runtime (SHIR). SHIR is a fully managed and serverless Azure-based integration runtime that will move data and process data activities to be performed between NCVER/State on-premises and STARS.

To connect to an NCVER data source using SHIR, an integration runtime (IR) will be created in Azure Data Factory (as shown in the above diagram, "Reference Data Integration Technical Design"). The IR will act as a bridge between Azure data factory and the NCVER data source, and it's responsible for performing the data movement and processing activities.

Once the SHIR is created, it will be configured to connect to the NCVER data source. Security settings will be configured, such as enabling SSL encryption and setting up a firewall to restrict access to the external data source and STARS public IP range address will be whitelisted on NCVER/State side.

Once the SHIR is configured, SHIR will be used to transfer data between the NCVER data source and Azure Data Factory by creating pipelines and data flows. The SHIR will handle the secure transfer of data and will provide monitoring and logging capabilities so that the progress of the data movement and processing activities be monitored.

South Australia (SA) Reference data Integration:

An option for data sharing includes the use of the Azure Data Share feature (as shown in the above diagram, "Reference Data Integration Technical Design"). This feature will provide secure, automated

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 437 of 1295

data replication to authorised consumers (e.g., STARS) via the Azure Data share capability. South Australia has Azure Data Share configured in their environment, and therefore it is agreed to share SA specific reference data with STARS using Azure Data Share. **In Stage 1 we will leverage this option of getting on demand SA reference data into STARS environment.** If this option is not available, STARS will rely on the manual files shared by SA.

12.5.1 Data transfer from NCVER (CARMS) into STARS Reference Database (**Stage 1 release**)

Please note in Stage 1 release, reference data in STARS will manual refresh (on demand) from NCVER.

The agreed process is: NCVER will place reference data in CSV files on a shared drive (e.g. VDS SharePoint) where Azure Data Factory can connect and load the data into STARS Reference Database. Please note that this is an interim solution until Stage 2 release and it won't be a fully automated data flow between NCVER and STARS.

12.5.2 Data transfer from NCVER (CARMS) into STARS Reference Database (**Stage 2 release**)

Placeholder for Data Integration design work between NCVER and STARS.

This data integration design is part of Stage 2 that hasn't started yet and is with NCVER & DERW. It will not start until March 2023 and could take 6 to 9 months for a final design which will include the technology roadmap and the data integration.

The below design could be an option for STARS integration with NCVER using Site to Site VPN. A site-to-site Virtual Private Network (VPN) will provide an encrypted link between VPN gateways located at NCVER and STARS.

The Reference Data Service will maintain Azure Data Factory based data ingest ETL pipelines. The pipelines are configured to connect to upstream reference data source using the appropriate transport and protocols. The pipelines extract, and then load reference data into the Data Lake and then to the Azure SQL Reference Database.

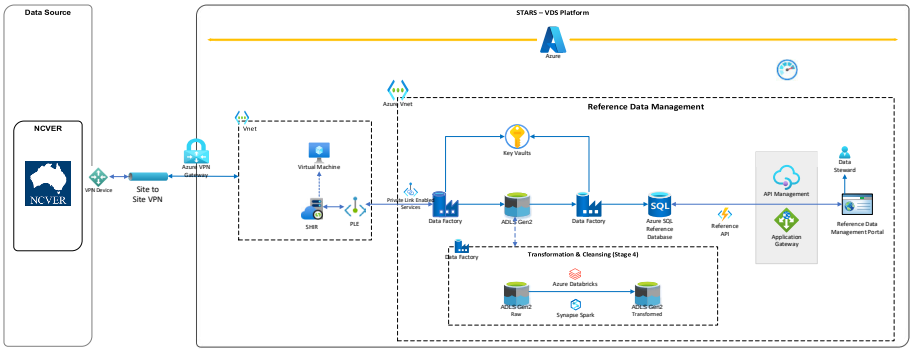


Figure 999: NCVER (CARMS) Data Transfer pipeline to Reference database (I4)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 438 of 1295

12.5.3 South Australia – Manual Files to STARS Reference Database (**Stage 1 release**)

In Stage 1 release, reference data in STARS will be refreshed manually as files provided by SA. Please note that this is an interim solution until Stage 3 release and it won't be a fully automated data flow between SA and STARS.

~~12.5.3~~12.5.4 South Australia – Azure Data Share to STARS Reference Database (**Stage 3 release**)

~~Please note i~~In Stage ~~3~~1 release, reference data in STARS will be refreshed ~~manually (on demand)~~automatically and on demand if required from SA using Azure Data Share.

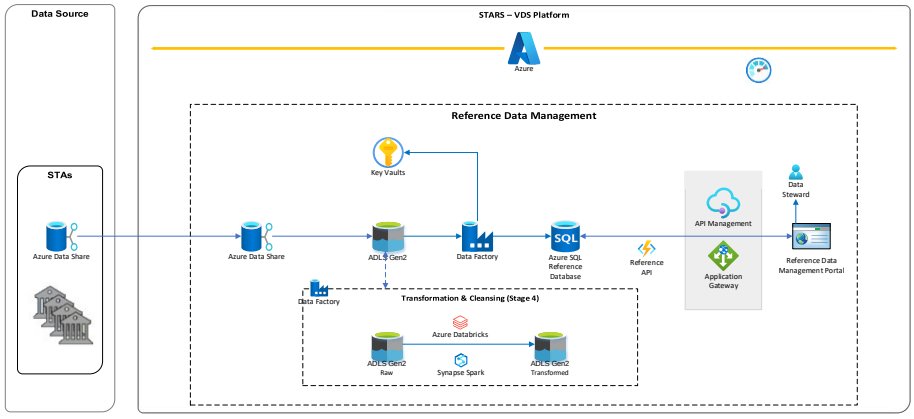


Figure ~~101010~~: SA Azure Data Share to STARS Reference database (I5)

~~12.5.4~~12.5.5 USI Verification

Direction has been given that USI verification is required in STARS. USI is required to be verified via API services exposed by USI will be covered in Validation services ~~Further engagement with OSIR is required to discover options for ingesting this reference data to STARS~~. (Action item 1959~~See W18 – Action #10~~)

Formatted: Font: (Default) +Body (Calibri), 11 pt, Pattern: Clear

12.5.6Reference Data Validation

Reference data will be validated and verified before the data is loaded into the reference tables. Some of the validations that will be applied but not limited to:

- Validation of file format
- Validation of file size

Formatted: List Paragraph,NFP GP Bulleted List,TOC style,lp1,Bullet OSM,Proposal Bullet List,List Paragraph1,Recommendation,List Paragraph11,List Paragraph111,L,F5 List Paragraph,Dot pt,CV text,Medium Grid 1 - Accent 21,Numbered Paragraph,Bullets,NAST Quote,lp11,n, Bulleted + Level: 1 + Aligned at: 0.63 cm + Indent at: 1.27 cm

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 439 of 1295

Details of validation process and rules will be worked out in consultation with NCVER in Stage 2 (Open design decision DD04).

~~12.5.5~~12.5.7 Data Refresh Interval

The reference data will be refreshed in STARs as below.

Table 1717~~15~~: STARS Reference Data Refresh Interval

Data Source	Frequency
NCVER – CARMS (Stage 2 release)	Daily (Stage 2)/ On Demand (Stage 1)
State – SA only (Stage 3 release)	Daily (Stage 3)/ Hourly (Stage 3)/ On demand (Stage 1)

~~12.5.6~~12.5.8 Full/ Incremental

The initial Reference data ingestion from NCVER and States will be full load and then incremental.

Integration with TGA data in Stage 4 will require to handle deletions by merge with DELETE when not in source. This will not be a problem when relying on CARMS in stage 1/2/3.

Formatted: Normal

~~12.5.7~~12.5.9 Data Volume

Table 1818~~16~~: STARS Reference Data Volume

Data Source	Data Volume
NCVER – CARMS	Outstanding action item 1948
State – SA only	200 KB

12.6 Data Load – NCVER to STARS Azure Data Lake (Stage 2 release)

This section covers the data transfer from sources into Azure Data Lake in the STARS Data Platform.

12.6.1 ADF Pipeline – data load

Below is the high-level solution diagram to load delta data from NCVER to the STARS Azure Data Lake by using Azure Data Factory pipeline.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 440 of 1295

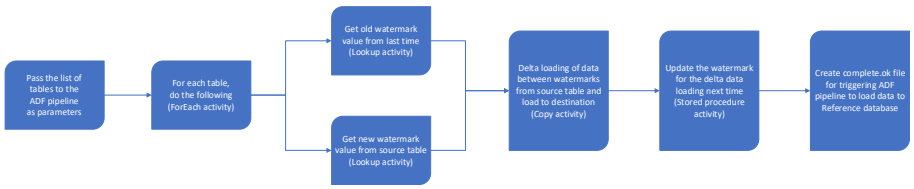


Figure 111~~11~~: ADF Pipeline steps to load into Azure Data Lake (I6)

12.6.1.1 Steps to create this solution:

1. Select the watermark column: Last Modified Time will be selected in the source tables, which will be used to identify the new or updated records for every run.
2. Store the Watermark value: This value will be stored in the Azure SQL Reference database.
3. Create ADF pipeline as per the above solution shown in the diagram.
    a. Create a ForEach activity that will iterate through a list of source table names that is passed as a parameter to the pipeline. For each source table, it will invoke the following activities to perform delta loading for that table.
    b. Create two lookup activities. Use the first Lookup activity to retrieve the last watermark value. Use the second Lookup activity to retrieve the new watermark value. These watermark values will be passed to the Copy activity.
    c. Create a Copy activity that will copy rows from the source data store with the value of the watermark column greater than the old watermark value and less than the new watermark value. Then, it will copy the delta data from the source data store to Azure Data Lake as a new file. Please note if the process is ran twice in a day, then for each successful run a new file will be created in Azure data lake.
    d. Create a StoredProcedure activity that will update the watermark value for the pipeline that runs next time.
    e. Create complete.ok file to kick off the next ADF pipeline to load data from Data Lake to Azure SQL Reference database.

12.6.1.2 Watermark table structure:

Table 191~~917~~: Table name - etl.Watermark

Field Name	Data type
WatermarkKey	Integer, IDENTITY (1, 1) NOT NULL
TableName	NVARCHAR(255)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 441 of 1295

WatermarkValue	DATETIME
LastStartTime	DATETIME
LastEndtime	DATETIME
LastProcessStatus	NVARCHAR(25)

Sample SQL script:

```
CREATE TABLE etl.watermark
(
    WatermarkKey integer IDENTITY (1, 1) NOT NULL,
    TableName nvarchar(255),
    WatermarkValue datetime,
    LastStartTime datetime,
    LastEndTime datetime,
    LastProcessStatus nvarchar(255),
);
```

12.6.2 Raw Data Storage

The ingested NCVER(CARMS) and State data will be stored in Azure Data Lake Gen2 as it has a Windows Azure Storage Blob over Secured Sockets layer (SSL) providing connectors to most of the windows application in case requirements arises in future for transformation and analytics (at the time of writing this document we don't have such requirement). WASBS will provide connectors to Synapse Analytics, Power BI and other windows compatible applications so that data exploration and reports generation will be easier and quicker.

12.6.3 Hierarchical Namespace enabled ADLS Gen2

ADLS Gen2 will be enabled with hierarchical namespace over blob data for atomic directory manipulation.

12.6.4 Folder Structure and File Name Convention

The folder structure for raw data layer will have 7 hierarchical levels as shown in the diagram. After the main folder (Raw Data), the highest level will have subject area in it to automatically organise the data into enterprise level subject areas based on business units such as Stakeholder, Responsibility, Training, etc.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 442 of 1295

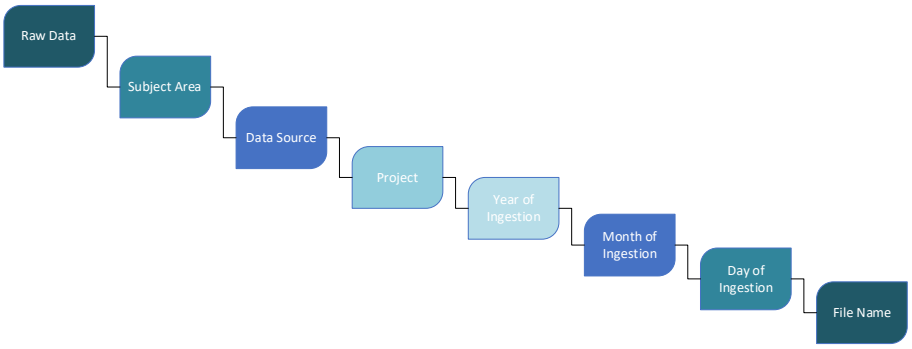


Figure 121212: Data Lake Folder Structure (I7)

1. **Root Folder for Layer Identification:** The root folder identifies the logical layer or the data tier.
   E.g., RDL (Raw Data Layer)/DIL (data Integration Layer)/DEL (data Enrichment Layer)/DAL (Data Access Layer) or Bronze/Silver/Gold
2. **Subject Area Identification:** The second level folder i.e. the subject area will be based on the business function such as Stakeholder, Responsibility, Training, Reference etc.
3. **Data Source:** This is to identify the system of records and/or the authenticated source system. For Data Platform, it will be NCVER, ASQA, etc.
4. **Project:** This is to identify the project that has ingested the data, it will be VDS.
5. **Data Partition Folders:** Based on the frequency of data transfer Year, Month and Day are the required folders for intraday ingestion files.
6. **File Name Convention**: It is best practice to prefix/postfix the context to the files that are ingested and created. Example: for raw files, if the source table is "Responsibility Financial Detail Category", naming will be:

   RDL_Reference_NCVER_VDS_ResponsibilityFinancialDetailCategory_20230214

7. **The full folder path will be as follows**:

   <data-layer>/<subject-area>/<Source System>/<project-name (Optional)>/<year>/<month>/<day>/ <file_name>

   Example:

   RDL/Reference/NCVER/VDS/2022/02/14/RDL_Reference_NCVER_VDS_ResponsibilityFinancialDetailCategory_20230214

12.6.5 File Format for Storage

Files will be stored in parquet file formats. Parquet supports read heavy workloads and retrieve record subset information easily.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 443 of 1295

12.7 Data Load – Azure Data Lake into Azure SQL Reference database (Stage 2 release)

The following section covers data load from Azure Data Lake into the Azure SQL reference database by using Azure Data factory pipeline.

12.7.1 Reference Data Cleansing and Transformation

Reference data sourced from NCVER (CARMS), and States will be ingested as is into STARS and no cleansing and transformation of data is required within STARS in Stage 1 and 2 release.

Outstanding Action #2859 relates to any known manipulations/transformations required as data arrives and the response to this will inform further details here.

The curation of data will be covered in Stage 3 release.

12.7.2 ADF Pipeline – data load

ADF Pipeline will be created to load the CARMS data from Azure Data Lake into Azure SQL Reference database. This pipeline will be triggered when the complete.ok file is created as part of section 12.6.1.

ADF pipeline will loop through all the files that got created on the latest run (Watermark table) in Azure Data Lake and will copy the data (insert/update) into specific reference data entities in Azure SQL Reference database via staging tables. If there are multiple files in a day for a reference entity, the ADF will process the files in ascending sequence of file creation date and time.

Reference data will be first loaded into staging tables and then transformed into Reference data tables in a transaction using a stored procedure.

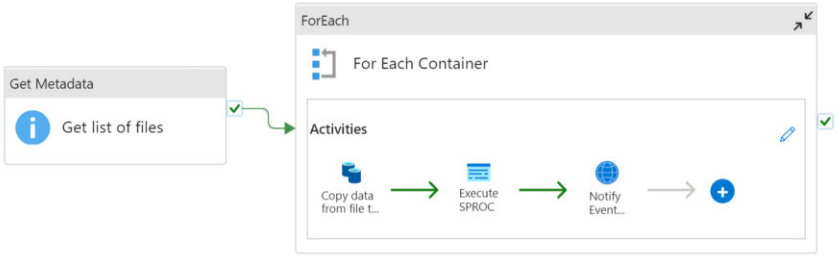


Figure 131313: ADF Pipeline Data Load (I8)

The For each container will have the copy task (at first the data will be loaded into staging tables and then to the Reference data tables using Stored procedures.) and a Web activity task to notify the STARS Event service if the reference data entity was updated (refer section 12.8 got more details).

For loading State (SA) data from Azure Data Lake, separate Azure Data Factory pipeline will be created to load SA specific reference data into the Azure SQL Reference database.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 444 of 1295

12.8 Reference Data Updated Events (Stage 2 release)

At the end of each reference entity data load into Azure SQL Reference database via Azure Data Factory pipeline, ADF pipeline will use Service Bus REST API and ADF's Web activity to notify the Event service only when the reference data is updated.

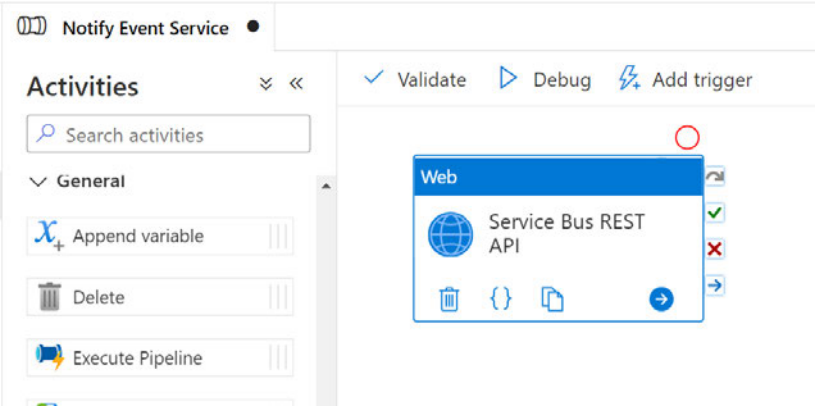


Figure 141414: ADF To Service Bus Event Trigger (I9)

"Azure Service Bus Data Sender role" will be assigned to the data factory's managed identity to grant permission for the data factory to send messages to the Service Bus queue or topic.

Select Managed Identity authentication from the advanced options in the web activity for authentication.

12.9 ADF Logging (Stage 2 release)

12.9.1Logs

Azure Data Factory (ADF) keeps the pipeline-run-data for only 45 days by default. Azure Monitor (AM) will be integrated with ADF to keep the logs data for a longer time and route diagnostic logs for further analysis. AM will provide higher visibility on the performance and health metrics of the workloads which will; be used for monitoring and trouble shooting.

Examples: Count of – pipeline failures, succeeded activities, failed trigger runs, IR failures, failed activities due to insufficient or unavailable memory for IR, etc.,

12.9.2 Alerting and Notification

Alerts and notifications will be generated for STARS and Stakeholders by calling Notification Service API (UserNotification) from ADF pipeline to add a new notification to the Notification Service. The notification will be sent via email (depending on the user's preferences) and can be retrieved via the Notification API.

For more details refer Notification Services detail design document.~~section 8.3.5 Detailed Design Document - Notification Service D.0.4~~.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 445 of 1295

12.9.3 Exception Handling

ADF Pipelines fail for variety of reasons. While this is expected, it is important that the system safely recover from errors, they will be logged appropriately and that the system administrators will be notified immediately for remediation actions (by calling Nitification Service API as defined in section above).

The activities in ADF have 'activity dependency statuses' such as success, failure, completion and skipped (condition based skipped) built-in which will be used for exception handling scenarios.

The Azure Data Factory will also also retry the pipeline execution in case of any failures. Retry will be set to 3.

12.9.4 Custom Logging

STARS Reference database will maintain a custom pipeline log table for storing the ADF pipeline activity details as records in the log table for each run.

Sample SQL script to create log table in Azure SQL Reference database:

```
CREATE TABLE [etl].[PipelineLog](
   PipelineLogKey bigint IDENTITY (1, 1) NOT NULL,
   DataFactoryName nvarchar(500) NULL,
   PipelineName nvarchar(500) NULL,
   RunId nvarchar(500) NULL,
   Source nvarchar(500) NULL,
   Destination nvarchar(500) NULL,
   TriggerType nvarchar(500) NULL,
   TriggerId nvarchar(500) NULL,
   TriggerName nvarchar(500) NULL,
   TriggerTime nvarchar(500) NULL,
   RowsCopied nvarchar(500) NULL,
   RowsRead int NULL,
   NumberOfParallelCopies int NULL,
   CopyDurationInSeconds nvarchar(500) NULL,
   EffectiveIntegrationRuntime nvarchar(500) NULL,
   SourceType nvarchar(500) NULL,
   SinkType nvarchar(500) NULL,
   ExecutionStatus nvarchar(500) NULL,
   CopyActivityStartTime datetime NULL,
   CopyActivityEndTime datetime NULL,
   CopyActivityQueuingDurationInSeconds nvarchar(500) NULL,
   CopyActivityTimeToFirstByteInSeconds nvarchar(500) NULL,
   CopyActivityTransferDurationInSeconds nvarchar(500) NULL,
) ;
```

12.9.5 Metadata Extraction and Cataloguing

Microsoft Purview will be connected to the ADF for capturing the metadata lineage, and to discover and explore data assets across the STARS data platform. This will be covered in a separate document Purview – technical Detailed Design.

13 Access Management

The Reference Data Service will conform to the Data Governance Access and Privacy Protocol document and the IDAM detail design. Access to the Reference Data Service will be controlled by Azure

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 446 of 1295

RBAC, with all calling services identified and granted access through Azure AD Managed Identities. For detail about accessing the Reference Data Service please refer to IDAM detail design document.

13.1 Control of data

The Governance Framework sets out the key data governance roles surrounding VET Data Streamlining. The two core central roles for accessing data from STARS are that of the Data Owner and the Data Custodian, summarised below:

13.1.1 Data Owner

The Data Owner has authority for establishing rules surrounding the quality, protection and appropriate access to data. They individually and collectively have a responsibility to protect the personal and sensitive information provided by Data Providers. Data can be jointly or co-owned.

13.1.2 Data Custodian

The Data Custodians receive authorised data from Data Providers or from other Data Custodians. Data Custodians are authorised to collect and handle that data to perform their functions. They are responsible for ensuring that the personal information collected and stored is reasonably necessary for them to conduct their functions and activities in relation to the VET sector.

For detail about the data access and control please refer to the Data governance Access and Privacy protocol document.

13.2 Security

The STARS solution will be built to meet an existing Australian Government policies, frameworks, standards and design principles, including:

- Australian Government API Design Standards
- Digital Service Standard
- Protective Security Policy Framework (PSPF)
- Information Security Registered Assessors Program (IRAP)
- Australian Government Information Security Manual (ISM)
- the Open Web Application Security Project® (OWASP)
- Hosting Certification Framework.

All data within STARS will be handled in accordance with relevant privacy laws and record keeping requirements, including the Privacy Act, the *Archives Act 1983* (Cth) and any other applicable legislation (e.g. applicable State and Territory privacy legislation or archiving requirements).

The Reference Data Service will conform to the Security Architecture defined for STARS in Data Governance Access and Privacy Control document and IDAM detail design document. Access to the Reference Data Service will be controlled by Azure RBAC, with all calling services identified and granted access through Azure AD Managed Identities.

For detail about security please refer to the following

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 447 of 1295

- Data Governance Access and Privacy Control document
- IDAM detail design document

14 Dependency on Other Services

Following are STARS services upon which Reference Data Service has dependencies.

14.1 Event Service

The Reference Data Service will depend upon the Event service for raising events, information regarding these Reference Data Service event database. Please refer to Event service detail design document for details about the Events Service.

14.2 Rules Service

The Reference Data Service depends upon the Rules service which is used to validate business rules of the Reference Data Service. Please refer to the Rules Service detail design document for the details about the Rules Service.

14.3 Validation Service

The Reference Data Service depends upon the Validation Service to revalidate the active failed validations. Please refer to the Validation Service detail design document for the details about the Validation Service.

14.4 Notification Service

The Reference Data Service will depend upon the Notification service for raising notifications. Please refer to Notification service detail design document for details about the Notification Service.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 448 of 1295

15 Functional Requirements

Note: The description talks about management which I assume is part of the Reference Management portal scope but will drive what entities are extracted into Reference DB from CARMS and STAs.

Table 202018: Related Functional Requirements

FR Id	Category	Requirement Name	Description	Acceptance Criteria	Traceability
FR054	Validation and Data Quality	Reference data sets	The VDS system must validate the Data submitted by a training organisation against reference data sets, for example: • TGA • ANZSCO reference data • Program field of education • Subject field of education • Level of education • Language spoken at home • **USI** • Training Contract ID and Client ID • **CARMS (NCVER)** • address and postcode This includes state-based training programs that are not recognised by TGA.		Section 12.3.1, Section 12.5.4, Section 11
FR064	Validation and Data Quality	Reference data curation and enrichment	The VDS system must leverage best practice industry standards for reference data curation and management.		Section 12.7.1, Section 7

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR Id	Category	Requirement Name	Description	Acceptance Criteria	Traceability
			If the reference data is imported into the VDS system, it must be updated whenever the source data is updated.		
FR088	Validation and Data Quality	Exceptional Data Handling (Auto reprocessing of latest records when any lookup data changes)	As a STA User, I want the system to auto reprocess the list of failed records (those without a newer record as a result of resubmission) where any lookup data changes could potentially resolve the errors (e.g **reference data fixed** or data package received out of order) so that those errors can now be auto resolved rather than relying on manual human intervention.	**Able to trigger the auto reprocessing upon detecting lookup data changes.** Only picking up those failed records where there are no subsequent resubmissions (as a resubmission might have resolved that error, and also we only want to process the latest record so that any previous records are treated as historical records) If the validation still failed, update the validation timestamp and maintain status as failed If the validation passes now, then update the status and validation timestamp, and trigger the notification and release process (whether it is manual or auto)	Section 8.2, section 12.8
FR096	Validation and Data Quality	Training Org Reference Data	Management of Training Organisation Delivery Location via UI for non-Registered Training Org without SMS	Able to add a new training location via UI and system will assign a training location ID for my organisation. Able to update an existing training location and the training	Section 8.1, 8.3

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR Id	Category	Requirement Name	Description	Acceptance Criteria	Traceability
				location ID shall remain the same in this situation	
FR097	Validation and Data Quality	Training Org Reference Data	Management of Training Organisation Delivery Location for Registered Training Org without SMS	Only need to add/update the training delivery location once and the info will be synced between systems	Section 12.5.1, Section 12.5.2, Section 8.1, 8.3
FR098	Validation and Data Quality	Training Org Reference Data	Management of Training Organisation Delivery Location for Training Org using SMS	Any newly added training delivery location in SMS for my org will be able to be submitted/synced to VDS system easily. Any update of existing training location in SMS for my org will be able to be submitted/synced to VDS system easily. If my org is a Registered Training Org, the delivery location shall be in sync with TGA and only one source of truth	Section 12.5.1, Section 12.5.2, Section 8.1, 8.3
FR099	Validation and Data Quality	National Reference Data	Management of Registered Training Organisation D~~s~~ata	Newly added registered training organisation data in TGA will appear in VDS system without human intervention (real-time or at defined interval TBC) Any update of registered training organisation in TGA will appear in VDS system without human intervention (real-time or at defined interval TBC)	Section 12.5.1, Section 12.5.2, Section 8.1, 8.3
FR100	Validation and Data Quality	National Reference Data	Management of accredited Courses/Qualifications/Skills/Units and training organisation Scope	Newly added accredited Courses/Qualifications/Skill Sets/Units in TGA will appear in VDS system without human intervention Any update of accredited Courses/Qualifications/Skill Sets/Units in	Section 12.5.1, Section 12.5.2, Section 8.1, 8.3

FR Id	Category	Requirement Name	Description	Acceptance Criteria	Traceability
				TGA will appear in VDS system without human intervention Any changes to training organisation scope of accredited courses/qualifications/skills/units in TGA will appear in VDS system without human intervention	
FR101	Validation and Data Quality	State Reference Data	Management of Non-Registered Training Organisation Data by STA	Any newly created record will only need to be added in one system the info will be synced to other systems without human intervention. Any changes to the non-registered training org only need to be done in one system and the info will be synced to other systems without human intervention.	Section 12.3.2.5, Section 8.1, 8.3
FR102	Validation and Data Quality	State Reference Data	Management of Non-Registered Training Organisation Data by non-Registered Training Organisation Admin user	Able to make changes to my organisation data (e.g address, contact info etc). The information is synced to state specific system without a need for me to raise request to STA or update in state specific system	Section 8.1, 8.3
FR103	Validation and Data Quality	State Reference Data	Management of Non-Accredited (State specific) Courses/Qualifications/Skill Sets/Modules/Units	New non-Accredited Courses/Qualifications/Skills/Units will only need to be added to one system and the info will be synced between VDS system and other state specific systems without human intervention. Any update of non-Accredited	Section 12.5.3, Section 8.1, 8.3

FR Id	Category	Requirement Name	Description	Acceptance Criteria	Traceability
				Courses/Qualifications/Skills/Units will only be done on one system and the info will be synced between VDS system and other state specific systems without human intervention.	
FR104	Validation and Data Quality	State Reference Data	Management of Funding Source - STA (Current NAT00120 Funding Source - State Training Authority Field)	Any new funding source -STA only need to be added to one system and the info will be synced to other systems without human intervention Any changes of Funding Source - STA only need to be updated in one system (same system as creation) and the info will be synced to other systems without human intervention	Section 12.5.3
FR105	Validation and Data Quality	State Reference Data	Management of School Location ID (Used by SA when the training activity is for VET in School)	Newly added school locations and ID is only required to be added to one system and the info will be synced to other systems without human intervention Any change of school locations and ID is only required to be updated in one system and the info will be synced to other systems without human intervention School locations include School Code (SACE ID), School Location ID, School Name, School Sector, Suburb, State, Postcode, School Type ID. Capability of import of school location list is supported	Section 12.5.3

16 Non-Functional Requirements

Table 212~~19~~: Related Non-Functional Requirements

NFR Id	Category	NFR Category	Requirement Name	Requirement Description	Traceability
NFR003d	Data Validation and Classification	Performance	Concurrency of reference data	The VDS System provides a mechanism to enable concurrency of reference data from state-managed systems into VDS.	Section 12.5, ADF will connect to State data sources and get data into VDS simultaneously. Section 8 – Events are raised when data is updated, allowing comsuming services to refresh caches
NFR013a	VDS System and Data Governance	Audit	Generate Audit information	The VDS System will be able to log activities and provide audit information as needed. The VDS System will be able to generate all audit information supporting all historical logging for a specified user function.	Section 6.1.6, Section 12.4

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

17 Appendix

17.1 Supporting documents

Reference	Document
Reference Data Entity ERD	Reference Data Service Diagrams - V.0.1 (ERD: CARMS Reference Data Entity)
SA Reference Data ERD	Reference Data Service Diagrams - V.0.1 (ERD: South Australia Reference Data)

17.2 Sample Data

Reference	Document
CARMS	Outstanding action item 1948
STARS Reference Data – SA Specific	STARS Reference Data-SASpecific

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 455 of 1295

18 Acronyms

Acronym	Definition
ADF	Azure Data Factory
ADLS	Azure Data Lake Storage
ETL	Extract, Transform & Load
FR	Functional Requirement
MDM	Master Data Management
NFR	Non-functional Requirement
STARS	Student Training & Activity Reporting System
VET	Vocational Education Training
VDS	VET Data Streamlining
IVETS	Integrated Vocational Education and Training System
DB	Database
~~DDS~~DDL	Data Definition Language
SQL	Structured Query Language
SHIR	Self-Hosted Integration Runtime
SIT	System Integration testing
STA	State Training Authority
SMS	Student Management
TP	Training Provider
UAT	User Acceptance Testing
DTO	Data Transfer Object

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 456 of 1295

Document 6

Australian Government
Department of Employment and Workplace Relations

Detailed Design Document

STA Batch Submission Service

DXC TECHNOLOGY

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 457 of 1295

1 Document Management

1.1 Document Control

Version	Change Description	Date
0.1	Draft Version	2023-03-21
0.2		
1.0		

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This is a living document and will be updated as required.

1.2 Document Author

Role	Name
Senior Technical Consultant	s 47F(1)
Lead Architect	s 47F(1)

1.3 Document Reviewers

Role	Name

1.4 Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

1.5 Related Documents

Reference	Document
A	STARS Solution Requirements Document V.1.3.docx
B	STARS Solution Design – V1.0.docx
C	STARS Solution Design - Attachment 3 - API Design - V.1.0.docx
D	STARS Solution Design - Attachment 1 - Collaboration Diagrams - V.1.0.vsdx
E	STARS Solution Design - Attachment 5 - Technical Architecture - V.1.0.vsdx
F	STA Batch Submission Detail Design Diagrams - D.0.1.vsdx
G	Detailed Design Document – Notification Service – D.0.5.docx
H	Detailed Design Document – Provider Service – D.0.2.docx
I	Detailed Design Document – IDAM – D.0.4.docx
J	Detailed Design Document – Data Governance Infrastructure – D.0.1.docx

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 458 of 1295

1 Document Management 2
1.1 Document Control 2
1.2 Document Author 2
1.3 Document Reviewers 2
1.4 Document Approval 2
1.5 Related Documents 2
2 Introduction 5
2.1 Purpose 6
2.2 Intended Audience 6
2.3 Scope and Exclusions 6
2.4 Acronyms 7
2.5 Definitions 8
3 Design Decisions 8
3.1 Assumptions 8
4 Feature Overview 8
4.1 Batch Submission Service Functions 10
4.2 Batch Service File Transfer Storage 10
5 Technical Design 10
5.1 STA Batch Service Storage 10
5.2 STA Batch Service Submission Process 12
6 Error Management 14
7 Data 14
7.1 VET-IS Data Packet Schema 14
7.2 Submission File Schema 15
7.3 Reconciliation File Schema 15
7.4 Data Flow 15
8 Other Service Dependencies 15
8.1 Provider Service 15
8.2 Notification Service 16
9 Access Management 16
9.1 Control of Data ..... **Error! Bookmark not defined.**
9.2 Security 16
10 Requirements 18
10.1 Functional Requirements 18
10.2 Non-functional Requirements 19

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 459 of 1295

11 Appendices..... 20

Figure 1 : Provider Service Context 9
Figure 2 : Batch Submission Service Design Overview (F:1)..... 10
Figure 3 : STA Batch Interface Storage Design (F:2)..... 11

Table 1: Batch Submission File Naming Patterns 11
Table 2: Reconciliation File Schema 15
Table 3: Function Requirements relevant to the STA Batch Submission Service..... 18
Table 4: Non-functional Requirements related to the STA Batch Submission Service 19

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 460 of 1295

2 Introduction

The purpose of this document is to capture the detailed design requirements for the STARS STA Batch Submission Service.

The STA Batch Submission Service provides the STAs with the following capabilities.

- Provides an indirect path for the RTO Submission Data to be interfaced to STARS
- Act as bulk data Submission Service for the STAs to bundle multiple RTO submissions.

The STA Batch Submission Service leverages the capabilities already built as part of the Provider Service to manage Submissions. The Batch Submission Service does not intend to re-build those capabilities.

When the STA Batch Submission Service receives the Submission training activity data from STAs in the form of JSON files, it extracts the activity data from the submission files and initiates submission of the training activity data via the Provider Service API.

These VET Information Standard Schema Entities are stored as their associated VET Information Standard Data Packet types, i.e.:

- Government Packet
- Training Organisation Packet
- Training Organisation Location Packet
- Program Packet
- Subject Packet
- Offering Packet
- Delivery Location Packet
- Student Packet
- Student Location Packet
- Program Enrolment Packet
- Subject Enrolment Packet

The Provider Service creates Surrogate Keys automatically for new data as it is received from the Batch Submission Service. It is assumed the data will be pre-validated using the Provider Service capabilities.

Note: This is a living document that will be updated as designs for various other STARS services are developed and evolve. Accordingly, this document will evolve to reflect and document notification components as the STARS solution build progresses.

The design is informed by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework, which provide a comprehensive set of best practices and guidance for designing, deploying, and operating secure, high-performing, and resilient cloud infrastructure and applications.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 461 of 1295

2.1 Purpose

The purpose of this detailed design document is to provide a technical description of the Azure components of the STARS solution. The document is intended for technical stakeholders, including architects, software engineers, administrators, and operations teams, who are responsible for designing, developing, deploying, and operating the STARS solution in Azure.

The document also serves as a reference for ongoing management and evolution of the STARS solution and to ensure that the STARS solution is deployed and operated in accordance with the best practices and guidance provided by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework.

The document is intended to be a living document, updated as the STARS solution evolves and changes over time. It provides a clear and concise understanding of the Azure provider components and their configurations, helping to ensure the STARS solution is deployed, operated, and managed in a consistent and efficient manner.

2.2 Intended Audience

This detailed design document provides technical information and guidance to help technical stakeholders understand the Azure notification components of the STARS solution and how they integrate with each other. The document is expected to be used by technical stakeholders who have a good understanding of cloud computing concepts and are experienced in the Azure platform.

This document is not intended for non-technical readers, nor for readers who are not experienced in the design and configuration of Azure-based solutions; informational and educational resources regarding Azure are readily available at https://azure.microsoft.com

The primary audience for this document includes:

- Program technical staff: This includes architects, software engineers, administrators, and operations teams within the program responsible for the STARS solution.
- External technical stakeholders: This includes technical staff from other federal, state, and territory governments agencies, including VDS embedded team members.
- ICT Technical Staff: This includes a variety of technical roles involved during the VDS program and post-go live, including network engineers, systems administrators, database administrators, security engineers, and DevOps engineers.

2.3 Scope and Exclusions

2.3.1 In Scope

This document elaborates the process of collecting Submission data files from STAs who have been identified for collecting and submitting bulk RTO submission data.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 462 of 1295

The process allows STAs to submit data from all of their RTOs in an asynchronous pattern, that handles orchestration of interactions for submitting the data from the STA and ensures large volumes of data can be ingested at scale.

This service also includes a feedback mechanism to communicate errors that would prevent data from being ingested to STARS (e.g., corrupted files, invalid JSON structure).

2.3.2 Out of Scope

This document does not cover the downstream processes for managing the process of training activity records submission. The details of the same are covered under the detailed design documents for Provider Service and Submission Service.

The process of data collection by the STAs from various providers is also beyond the scope of this document.

The Batch Submission service scope is limited to receiving submission batches and forwarding them to Provider Service. This service does not provide direct means to track the progress of the submissions.

2.4 Acronyms

Acronym	Definition
API	Application Programming Interface
ADF	Azure Data Factory
CRUD	Create Read Update Delete
REST	Representational State Transfer
FR	Functional Requirement
NFR	Non-functional Requirement
STARS	Student Training & Activity Reporting System
VET	Vocational Education Training
VDS	VET Data Streamlining
IVETS	Integrated Vocational Education and Training System
DB	Database
DDS	Data Definition Language
SQL	Structured Query Language
TP	Training Provider
STA	State Training Authority

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 463 of 1295

SMS	Student Management System
DET	Data Entry Tool
RAM	Relationship Authorisation Manager
PWA	Progressive Web App
AD	Active Directory
RBAC	Role Based Access Control
Micro-TP	A training organisation that does not use a Student Management System

2.5 Definitions

Term	Definition
Micro-TP	A training organisation that does not use a Student Management System
STARS Collection Model	The physical representation of the VET-IS data packets
Activity Record	A training activity data packet

3 Design Decisions

As per the design decision captured in the Provider Service detailed design document the “Indirect Path” training providers will be supported through the Provider Service API, via their STA systems. This decision has a direct consequence in the design of the STA Batch Submission Service. The STA batch Submission Service will leverage the capabilities provided by the Provider Service to make the submissions.

3.1 Assumptions

- It has been assumed that STAs will pre-validate the submission data before interfacing with the Batch Submission Service. The data will still be validated upon Submission, too.
- Victorian STA has indicated that may directly interface with Provider Service. However, we are currently assuming they will use the Batch Submission Service.
- It is assumed that STAs will track the Submission progress (validation results etc) directly using the Provider Service API once the Submission has been processed.
- It is assumed that STAs will provide the training activity records batched as one file per RTO.

4 Feature Overview

The STA Batch Submission Service requires the STAs to collate the Training Activity records to be collated for multiple training providers and send them as batches to be processed and ingested by STARS. This may be on a predefined schedule such as daily, weekly, or monthly based on their respective requirements. The Submission Service expects the Submission records to be provided in a

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 464 of 1295

structured JSON format, which will conform with the expectations of the Provider Service API, for operations that load training activity data.

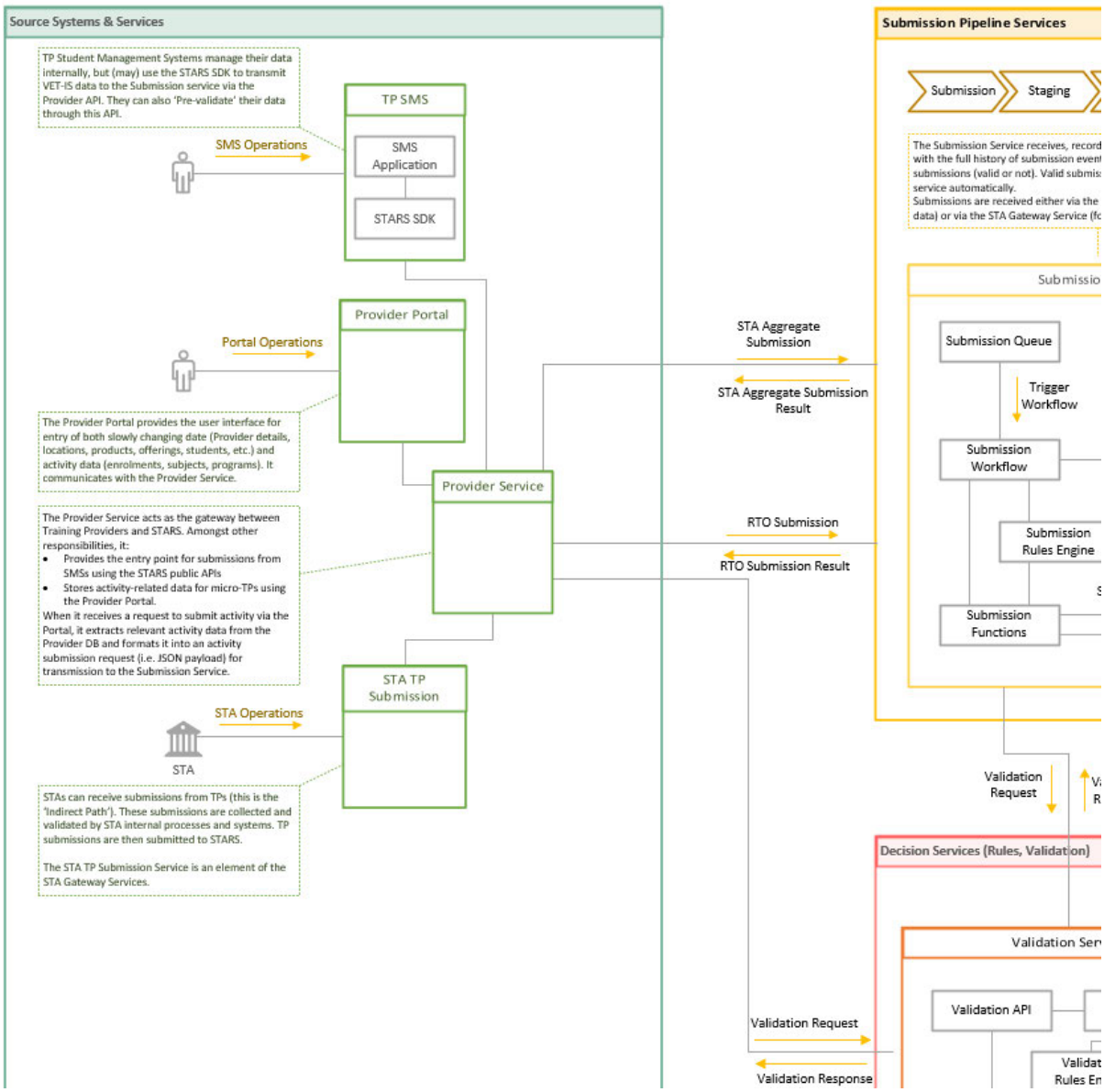


Figure 1 : Provider Service Context

The above diagram describes how the batch Submission Service referred as STA TP Submission service sits in the STARS eco system. The above is a high-level overview of the service to establish the overall context. We will not delve into the details of Provider Service in order to maintain objectivity. The batch Submission Service relies on capability of Provider Service to submit bulk activity records and then allow the submission.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 465 of 1295

4.1 Batch Submission Service Functions

The Batch Submission Service functions deal with the process of retrieval of the submission files from the blob storage containers and interfacing the data with the Provider Service. The function apps will be triggered using blob storage trigger to start processing the submissions.

4.2 Batch Service File Transfer Storage

The Batch Submission Service will provide an Azure Storage Account per STA for the purpose of transferring submission files containing the training activity records.

5 Technical Design

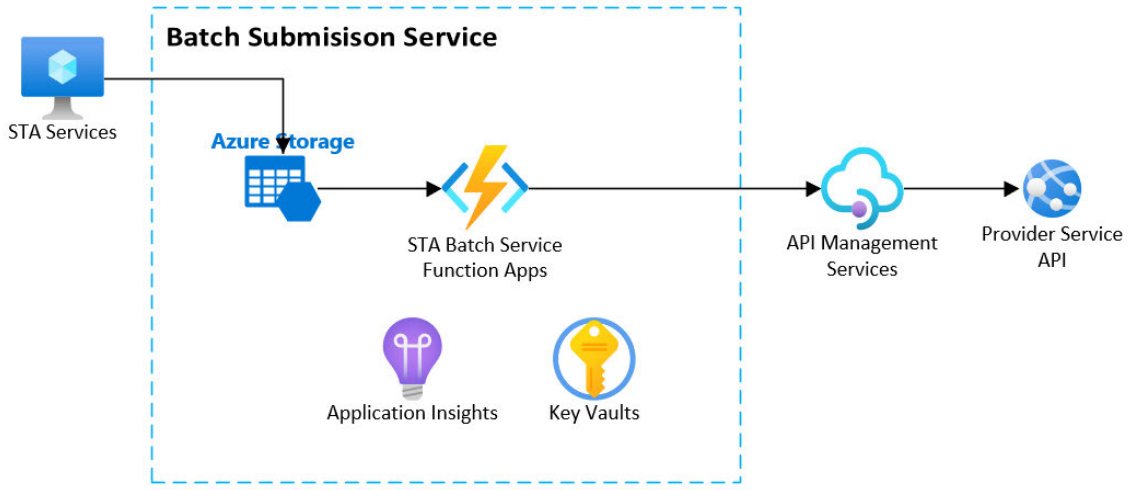


Figure 2 : Batch Submission Service Design Overview (F:1)

5.1 STA Batch Service Storage

The STAs will send the batch submission files to Azure Storage accounts. The storage accounts will be dedicated to each STA. The STAs will load the submission files along with a reconciliation file. The files will need to be submitted **in order** i.e., the submission files will be loaded first, and the reconciliation file will be loaded last. The reason behind loading the reconciliation file last is because it is going to act as a trigger for processing to start. Hence, to ensure the system does not start processing until all the files in the batch have been received.

Once the files are loaded they will then be moved to the processing folder. Once the files are loaded in the processing folder, the ADF pipelines will validate the files for structural consistency and cross validate the files against the data in the reconciliation file. In case of any errors processing the file, the respective files will be moved to an error folder which will have chronologically named subfolders. Along with the file a processing log will also be placed in the folder describing the details of the errors. A Notification will be raised if a batch contains any errors.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 466 of 1295

In case of successful acceptance of the submission file by the Provider Service a processing log file will be stored along with the file in the archive folder which will also be structured based on the date of processing.

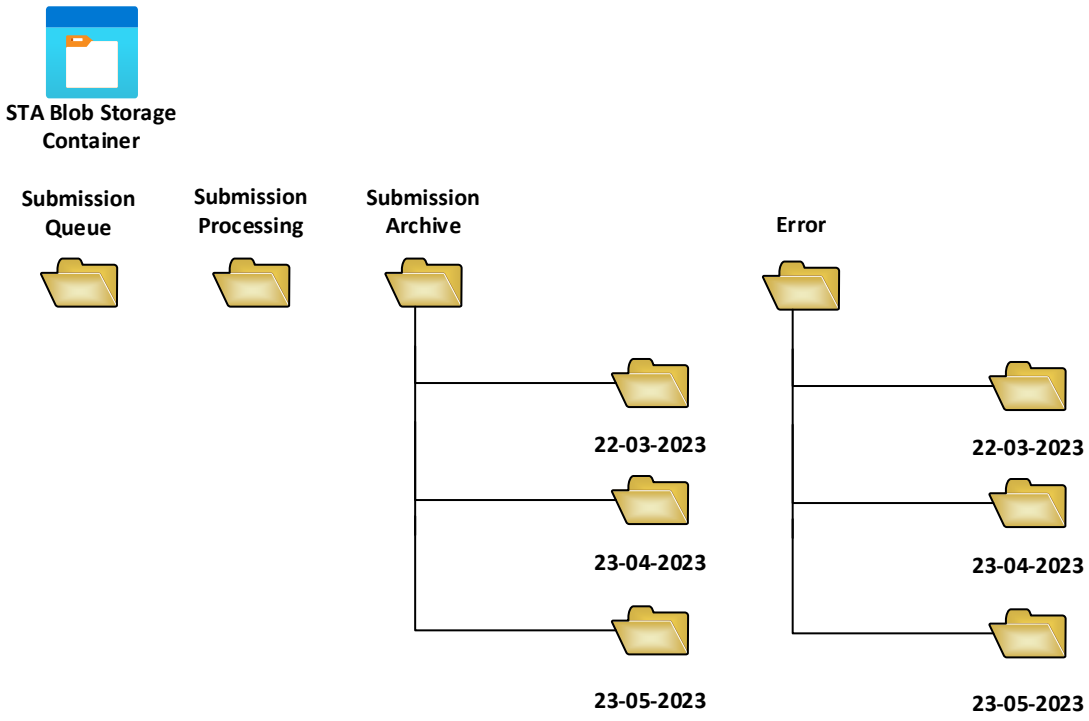


Figure 3 : STA Batch Interface Storage Design (F:2)

5.1.1 File Naming Patterns

Table 1: Batch Submission File Naming Patterns

File Type	Naming Pattern	Example
Submission File	[RTO Name]-[State]-[BatchID]-[Sub]-[Date].json	ABC-QLD-1-SUB-21032023.json
Reconciliation File	[REC]-[State]-[BatchID]-[Date].csv	REC-QLD-1-21032023.csv
Processing Log File	[RTO Name]-[BatchID]-[Log].txt	ABC-1-Log.txt

In the table above we have used ABC as the name of the RTO.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 467 of 1295

5.2 STA Batch Service Submission Process

The STA Batch Submission Service will primarily be responsible for four functions as part of the overall submission lifecycle.

- Check the files against the reconciliation file to ensure all the files supposed to be in the batch have been received.
- Validation of the submission files to ensure they are valid JSON files and not corrupted.
- Extraction of training activity records and submission of the same to the Provider Service.
- Initiation of the Submission process.

5.2.1 Submission Batch Reconciliation

Once the submission files, along with the reconciliation file, are received in the submission queue folder, it will trigger the Azure Function App. The Blob Storage trigger for the Function App will be configured based on the reconciliation filename pattern.

Once the files are copied to the processing folder the function app will check the files against the reconciliation file to ensure all the files listed in the reconciliation file have been received.

If some files are missing, a log of the error will be written to the Error folder, and a Notification will be sent to the STA to advise of the missing file. The files that have been received will still be processed, though.

5.2.2 Bulk Activity Records Upload

For each file received a Function App will check the content to ensure it's a valid JSON document, and then the Bulk Upsert operation in the Provider Service will be called, with the file content as the payload. The API will ingest the records and return a draft Submission ID.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 468 of 1295

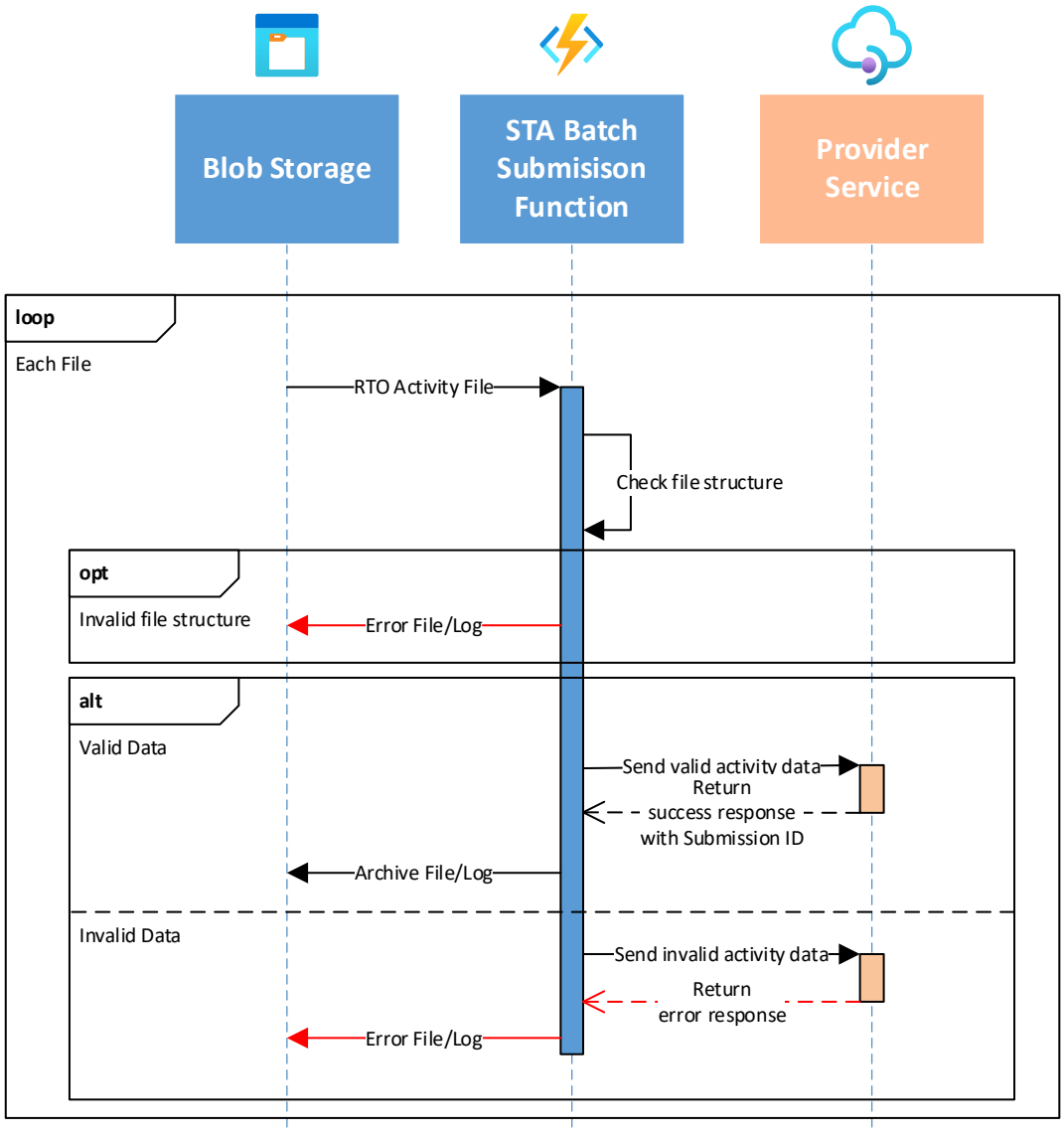


Figure 4: Activity Records Extraction and Bulk Upload (F:4)

5.2.3 Confirm Submission

Once the bulk activity records are uploaded to STARS, the draft Submission ID will be returned along with the response. The subsequent step will be invoking the Submit Data operation exposed by the Provider Service with the draft Submission ID.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 469 of 1295

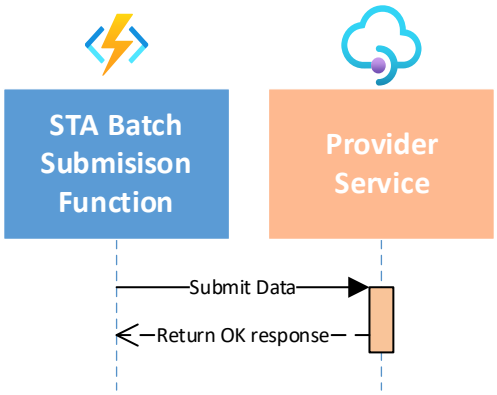


Figure 5: Submission Confirmation (F:3)

6 Error Management

The scope of the STA Batch Submission Service error management capability is providing the STAs with any error that happens up until the point where the submission is made. Any downstream issues with the submission process shall be handled by the core STARS services. Any error that happens with a training provider submission file will be logged with a processing log and saved to the error folder in the STA storage container. An event message and/or Notification will also be sent raised to alert the STA that their attention is required.

The Provider Service provides API end points to monitor the progress of a given submission. Please refer to the Provider Service Detailed Design document (I) for further details.

7 Data

The core data entities involved with the STA Batch Submission Service are the Submission Data Packets prepared and submitted by Training Providers, which are based on the VET-IS data standard.

All STARS services will be using the STARS Collection data model, based the VET-IS Collection specification.

7.1 VET-IS Data Packet Schema

The VET-IS Data Packet Schema forms both a logical information model and the basis for implementing data interchange protocols between STARS and external systems, including STA systems. The Data Packet Model informs system-to-system data presentation formats and, for example, Application Programming Interfaces (API).

The STARS Collection Model is the physical representation of this logical information model, that will be used for communicating training activity data within STARS, and between STARS and external systems. It will be represented in JSON format and conform to the specification in the logical model.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 470 of 1295

7.2 Submission File Schema

The File Schema used to define the content of Submission Data supplied in the files loaded to the STA Batch Submission Service will be consistent with the API Specification for the Bulk Upsert operation of the Provider Service API.

[As of time of writing this document the submission file schema has not been finalized. This will be provided by the Provider Service and the section will be updated.]

7.3 Reconciliation File Schema

The reconciliation file will be a CSV file having the following structure as described in the table below.

Table 2: Reconciliation File Schema

Field Name	Purpose
FileName	FileName of the submission file. The filenames will be checked against the actual files received.
FileSize	File Size of the submission file. The size of the actual files will be checked against this field.

7.4 Data Flow

The following Data Flow Diagram covers the high-level processes/features of the STA Batch Submission Services.

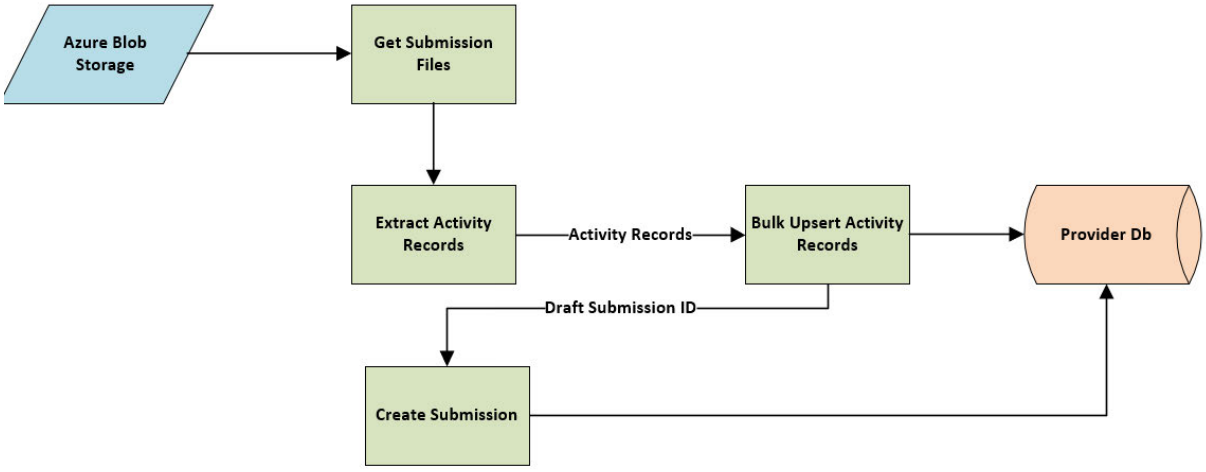


Figure 6: STA Batch Submission Service Data Flow (K:5)

8 Other Service Dependencies

8.1 Provider Service

The STA Batch Submission Service will depend on the Provider Service API for the following capabilities

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 471 of 1295

8.1.1 Bulk Activity Record Upsert

This operation is responsible for loading the submission data into the Provider DB.

8.1.2 Confirm Submission

Once data has been loaded, the formal Submission can be initiated. This is the operation that initiates formal Submission with the Submission Service

8.2 Notification Service

The STA Batch Submission Service will depend on the Notification Service API for the following capabilities.

8.2.1 Error Notification

If an error is encountered prior to confirming submission. Any downstream errors or notifications will be handled the respective services. This particular Notification Type is currently not documented in the Notification Service design and will be updated in an upcoming revision of the Notification Service Detailed Design document.

8.2.2 Completion Notification

When all files in the batch have completed processing, within the Batch Submission service responsibilities, a Notification is to be raised to communicate to the STA system that the information is submission information can be viewed through the Provider Service API.

9 Access Management

The STA Batch Submission Service will conform to the Data Governance Access and Policy Protocol documents and IDAM Detailed Design (I). Access to the STA Batch Submission Service Storage will be controlled by Azure RBAC, with all calling services identified and granted access through Azure AD Managed Identities and B2B. For further details please refer to the IDAM Detailed Design document (I).

9.1 Security

STARS will be built to meet existing Australian Government policies, frameworks, standards and design principles, including:

- Australian Government API Design Standards
- Digital Service Standard
- Protective Security Policy Framework (PSPF)
- Information Security Registered Assessors Program (IRAP)
- Australian Government Information Security Manual (ISM)
- the Open Web Application Security Project® (OWASP)
- Hosting Certification Framework.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 472 of 1295

All data within STARS will be handled in accordance with relevant privacy laws and record keeping requirements, including the Privacy Act, the *Archives Act 1983* (Cth) and any other applicable legislation (e.g. applicable State and Territory privacy legislation or archiving requirements).

For detail about security please refer to the following

- Data Governance Access and Privacy Control document (J)
- IDAM detailed design document (I)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 473 of 1295

10 Requirements

The following section lists the relevant Functional and Non-functional requirements relevant to this service, with traceability to the individual sections of the document that speak to each requirement.

This service provides a submission pathway for high volume data load by STAs, for data provided by multiple training organisations. Where submission data from Training Providers is not batched, the standard pathway is through the Provider Service directly. As such, there are not many requirements specifically addressing the functionality of this service.

10.1 Functional Requirements

Table 3: Function Requirements relevant to the STA Batch Submission Service

FR No	Requirement Name	Comments/Description	Acceptance Criteria
FR076	Other Collection agencies are able to provide bulk event data to VDS system using any version of the VET Information standard that is still accepted nationally.	The VDS system must allow another collection agency to upload data that does not exist in VDS system and the collection agency will receive validation responses on behalf of training organisation. This can be a partnering or non-partnering jurisdiction.	Given access for data entry or upload And access for data submission And a recorded agreement between the submitter and a training organisation When data is submitted on behalf of the training organisation Then STARS will record the submitter of the data And process the submission as if it was submitted by the training organisation

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

10.2 Non-functional Requirements

Table 4: Non-functional Requirements related to the STA Batch Submission Service

NFR No	Requirement Name	Description	Comment/Traceability
NFR021k	Data Submission Methods	The VDS system supports different forms of event - driven data submission in parallel such as single record entry and batch upload	Data Submission is supported from multiple STAs in parallel, via File Upload/Transfer through this service.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

11 Appendices

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 476 of 1295

Australian Government
Department of Employment and Workplace Relations

Detailed Design Document

Provider Service

DXC TECHNOLOGY

1 Document Management

1.1 Document Control

Version	Change Description	Date
0.1	Draft Version	2023-02-21
0.2	Updated following feedback	2023-03-07
0.3	Updated following late feedback	2023-03-23
0.4	Split API definition to separate document	2023-03-27
1.0		

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This is a living document and will be updated as required.

1.2 Document Author

Role	Name
Senior Technical Consultant	s 47F(1)
Lead Architect	s 47F(1)

1.3 Document Reviewers

Role	Name

1.4 Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

1.5 Related Documents

Reference	Document
A	STARS Solution Requirements Document V.1.3.docx
B	STARS Solution Design – V1.0.docx
C	STARS Solution Design - Attachment 3 - API Design - V.1.0.docx
D	STARS Solution Design - Attachment 1 - Collaboration Diagrams - V.1.0.vsdx
E	STARS Solution Design - Attachment 5 - Technical Architecture - V.1.0.vsdx
F	Provider Service Diagrams – D.0.~~1~~2.vsdx
G	Detailed Design Document – Notification Service – D.0.5.docx
H	Detailed Design Document – Validation Service – D.0.2.docx
I	API Design Document – Provider Service – D.0.1.docx

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 478 of 1295

1 Document Management 2
1.1 Document Control 2
1.2 Document Author 2
1.3 Document Reviewers 2
1.4 Document Approval 2
1.5 Related Documents 2
2 Introduction 86
2.1 Purpose 97
2.2 Intended Audience 97
2.3 Scope and Exclusions 107
2.4 Acronyms 108
2.5 Definitions 119
3 Design Decisions 119
3.1 Submission Pathways ..... **Error! Bookmark not defined.**9
4 Feature Overview 1310
4.1 Provider Workflow 1411
4.2 Provider Functions 1411
4.3 Provider API 1411
4.4 Provider Service Internal Processes 1512
5 Technical Design 1613
5.1 Provider Service Config 1613
5.2 Provider Event DB 1814
5.3 Provider DB 1814
5.4 Provider API 2318
5.5 Provider API Operations 2420
5.6 Provider Service Events 5547
5.7 Data Quality Reporting 5849
6 Technical Design Patterns 5849
7 Data 6252
7.1 VET-IS Data Packet Schema 6252
7.2 Data Flow 6354
8 Other Service Dependencies 6556
8.1 Validation Service 6556
8.2 Submission Service 6556
8.3 Notification Service 6556

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 479 of 1295

8.4 Rules Service 6556
8.5 Reference Data Service 6556
9 Dependent Services 6556
9.1 Provider Portal 6556
9.2 Provider Management Portal 6556
10 Access Management 6656
10.1 Control of Data 6656
10.2 Security 6657
11 Requirements 6858
11.1 Functional Requirements 6858
11.2 Non-functional Requirements 8777
12 Appendices 9383
12.1 STARS Solution Design Diagrams 9383
~~**1 Document Management ..... 2**~~
~~1.1 Document Control ..... 2~~
~~1.2 Document Author ..... 2~~
~~1.3 Document Reviewers ..... 2~~
~~1.4 Document Approval ..... 2~~
~~1.5 Related Documents ..... 2~~
~~**2 Introduction ..... 6**~~
~~2.1 Purpose ..... 7~~
~~2.2 Intended Audience ..... 7~~
~~2.3 Scope and Exclusions ..... 8~~
~~**3 Design Decisions ..... 8**~~
~~**4 Feature Overview ..... 9**~~
~~4.1 Provider Workflow ..... 10~~
~~4.2 Provider Functions ..... 10~~
~~4.3 Provider API ..... 10~~
~~4.4 Provider Service Internal Processes ..... 11~~
~~**5 Technical Design ..... 12**~~
~~5.1 Provider Service Config ..... 12~~
~~5.2 Provider Event DB ..... 14~~
~~5.3 Provider DB ..... 14~~
~~5.4 Provider API ..... 18~~
~~5.5 Provider API Operations ..... 19~~

Formatted: Default Paragraph Font, Check spelling and grammar

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 480 of 1295

5.6 Provider Service Events 48
5.7 Data Quality Reporting 51
6 Technical Design Patterns 51
7 Data 55
7.1 VET-IS Data Packet Schema 55
7.2 Data Flow 56
8 Other Service Dependencies 58
8.1 Validation Service 58
8.2 Submission Service 58
8.3 Notification Service 58
8.4 Rules Service 58
8.5 Reference Data Service 58
9 Dependent Services 58
9.1 Provider Portal 58
9.2 Provider Management Portal 58
10 Access Management 59
10.1 Control of Data 59
10.2 Security 59
11 Requirements 61
11.1 Functional Requirements 61
11.2 Non-functional Requirements 80
12 Appendices 86
12.1 STARS Solution Design Diagrams 86
13 Acronyms 87

Figure 1: Provider Service Context (F:12) 1310
Figure 2: Provider Service Components (F:12) 1411
Figure 3: API Design Overview (F:1) 1613
Figure 4: Sequence - Get Terms and Conditions (F:33) 2520
Figure 5: Sequence - Register Training Provider (F:25) 2621
Figure 6: Sequence - Register User (F:26) 2621
Figure 7: Sequence – Upload, List and Download Files (F:4) 2823
Figure 8: Sequence – Validate Data (F:5) 3025
Figure 9: Sequence - Get Validation (F:7) 3126
Figure 10: Sequence – Create Activity Record (General) (F:12) 3327
Figure 11: Sequence – Update Activity Record (General) (F:13) 3528
Figure 12: Sequence – Bulk Edit Activity Records (General) (F:14) 3629
Figure 13: Sequence – Delete Activity Record (General) (F:15) 3730

Formatted: Default Paragraph Font, Check spelling and grammar

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 481 of 1295

Figure 14: Sequence – Get List of Activity Records (General) (F:16) 3831
Figure 15: Sequence – Get Activity Record by ID (General) (F:17) 3932
Figure 16: Sequence - Purge Activity Records (F:24) 4033
Figure 17: Sequence – Get Local Validation Request History (F:20) 4033
Figure 18: Sequence – Create Local Submission (F:35) 4134
Figure 19: Sequence - Update Local Submission (F:36) 4135
Figure 20: Sequence – Submit Data (F:6) 4236
Figure 21: Sequence – Get Validation Results (F:8) 4336
Figure 22: Sequence – Get Submissions (F:9) 4437
Figure 23: Sequence – Get Local Submission History (F:21) 4537
Figure 24: Sequence – New Validation Outcome Review (F:18) 4638
Figure 25: Sequence – Get Validation Reviews (F:11) 4739
Figure 26: Sequence - Get Training Organisation (F:29) 4840
Figure 27: Sequence - Get Subjects (F:30) 4840
Figure 28: Sequence - Get Programs (F:31) 4941
Figure 29: Sequence - Get Delivery Locations (F:32) 5042
Figure 30: Sequence - Get Notification Preferences (F:27) 5143
Figure 31: Sequence - Update Notification Preferences (F:28) 5244
Figure 32: Sequence – Get Notifications (F:10) 5345
Figure 33: Sequence – Get Business Glossary (F:19) 5446
Figure 34: Sequence – Get Form by Record Type (F:23) 5446
Figure 35: VET-IS Data Packet Schema 6353
Figure 36: Collection Packet Dependencies (F34) ..... **Error! Bookmark not defined.**54
Figure 37: Provider Service Data Flow 6455
Figure 38: Appendix: Provider Service (D:12) 9383
Figure 39: Appendix: Provider Service (E:4) 9383
~~Figure 1: Provider Service Context (F:12) ..... 9~~
~~Figure 2: Provider Service Components (F:12) ..... 10~~
~~Figure 3: API Design Overview (F:1) ..... 12~~
~~Figure 4: Sequence - Get Terms and Conditions (F:33) ..... 20~~
~~Figure 5: Sequence - Register Training Provider (F:25) ..... 21~~
~~Figure 6: Sequence - Register User (F:26) ..... 21~~
~~Figure 7: Sequence – Upload Data (F:4) ..... 23~~
~~Figure 8: Sequence – Validate Data (F:5) ..... 25~~
~~Figure 9: Sequence - Get Validation (F:7) ..... 26~~
~~Figure 10: Sequence – Create Activity Record (General) (F:12) ..... 27~~
~~Figure 11: Sequence – Update Activity Record (General) (F:13) ..... 29~~
~~Figure 12: Sequence – Bulk Edit Activity Records (General) (F:14) ..... 30~~
~~Figure 13: Sequence – Delete Activity Record (General) (F:15) ..... 31~~
~~Figure 14: Sequence – Get List of Activity Records (General) (F:16) ..... 32~~
~~Figure 15: Sequence – Get Activity Record by ID (General) (F:17) ..... 33~~
~~Figure 21: Sequence - Purge Activity Records (F:24) ..... 34~~
~~Figure 16: Sequence – Get Local Validation Request History (F:20) ..... 35~~
~~Figure 17: Sequence – Submit Data (F:6) ..... 36~~
~~Figure 18: Sequence – Get Validation Results (F:8) ..... 37~~

Formatted: Default Paragraph Font, Check spelling and grammar

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 482 of 1295

Figure 19: Sequence – Get Submissions (F:9) 38
Figure 20: Sequence – Get Local Submission History (F:21) 39
Figure 22: Sequence – New Validation Outcome Review (F:18) 40
Figure 23: Sequence – Get Validation Reviews (F:11) 41
Figure 24: Sequence - Get Training Organisation (F:29) 42
Figure 25: Sequence - Get Subjects (F:30) 42
Figure 26: Sequence - Get Programs (F:31) 43
Figure 27: Sequence - Get Delivery Locations (F:32) 44
Figure 28: Sequence - Get Notification Preferences (F:27) 44
Figure 29: Sequence - Update Notification Preferences (F:28) 45
Figure 30: Sequence – Get Notifications (F:10) 46
Figure 31: Sequence – Get Business Glossary (F:19) 47
Figure 32: Sequence – Get Form by Record Type (F:23) 47
Figure 33: VET IS Data Packet Schema 56
Figure 34: Provider Service Data Flow 57
Figure 35: Appendix: Provider Service (D:12) 86
Figure 36: Appendix: Provider Service (E:4) 86

Table 1: Provider Service internal processes 1512
Table 2: Provider DB Configuration 1613
Table 3: Provider API Configuration 1613
Table 4: Provider Blob Storage Configuration 1714
Table 5: Provider Logic App Configuration 1714
Table 6: Provider Service User Configuration Fields 1814
Table 7: Validation Request Fields 1815
Table 8: Submission Fields 1815
Table 9: Dynamic Form Fields 2016
Table 10: Dynamic Form Field Metadata 2016
Table 11: Dynamic Form Field Validation Checks 2116
Table 12: Dynamic Form Control Types 2117
Table 13: Provider Service Endpoints ..... **Error! Bookmark not defined.**18
Table 14: Events raised by the Provider Service 5547
Table 15: Events consumed by the Provider Service 5748
Table 16: Notifications raised by the Provider Service 5748
Table 17: List Endpoint Request Parameters ..... **Error! Bookmark not defined.**49
Table 18: List Endpoint Response Body Fields ..... **Error! Bookmark not defined.**49
Table 19: Read Endpoint Query Parameters ..... **Error! Bookmark not defined.**50
Table 20: Function Requirements relevant to the Provider Service 6858
Table 21: Non-functional Requirements related to the Provider Service 8777

Formatted: Default Paragraph Font, Check spelling and grammar

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 483 of 1295

2 Introduction

The purpose of this document is to capture the detailed design requirements for the STARS Provider Service solution.

The Provider Service acts as the gateway between Training Providers and STARS. Amongst other responsibilities, it:

- Provides the entry point for submissions from SMSs using the STARS public APIs
- Stores activity-related data for micro-RTOs using the Provider Portal.

The Provider Service maintains and presents RTO data and the Stakeholders, Products, Locations, Offerings and Responsibilities associated with the training provider.

When the Provider Service receives a request to submit activity via the Portal, it extracts relevant activity data from the Provider DB and formats it into a submission request for transmission to the Submission Service via the API.

These VET Information Standard Schema Entities are stored as their associated VET Information Standard Data Packet types, i.e.:

- Government Packet
- Training Organisation Packet
- Training Organisation Location Packet
- Program Packet
- Subject Packet
- Offering Packet
- Delivery Location Packet
- Student Packet
- Student Location Packet
- Program Enrolment Packet
- Subject Enrolment Packet

The Provider Service creates Surrogate Keys automatically for new data as it is received, both from SMSs and from the Provider Portal. New and updated Packets are candidates for Validation and Submission; these processes are initiated by Provider Portal users manually or via API call for SMSs.

> Note: This is a living document that will be updated as designs for various other STARS services are developed and evolve. Accordingly, this document will evolve to reflect and document notification components as the STARS solution build progresses.

The design is informed by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework, which provide a comprehensive set of best practices and guidance for designing, deploying, and operating secure, high-performing, and resilient cloud infrastructure and applications.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 484 of 1295

2.1 Purpose

The purpose of this detailed design document is to provide a technical description of the Azure provider components of the STARS solution. The document is intended for technical stakeholders, including architects, software engineers, administrators, and operations teams, who are responsible for designing, developing, deploying, and operating the STARS solution in Azure.

The document also serves as a reference for ongoing management and evolution of the STARS solution and to ensure that the STARS solution is deployed and operated in accordance with the best practices and guidance provided by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework.

The document is intended to be a living document, updated as the STARS solution evolves and changes over time. It provides a clear and concise understanding of the Azure provider components and their configurations, helping to ensure the STARS solution is deployed, operated, and managed in a consistent and efficient manner.

2.2 Intended Audience

This detailed design document provides technical information and guidance to help technical stakeholders understand the Azure notification components of the STARS solution and how they integrate with each other. The document is expected to be used by technical stakeholders who have a good understanding of cloud computing concepts and are experienced in the Azure platform.

This document is not intended for non-technical readers, nor for readers who are not experienced in the design and configuration of Azure-based solutions; informational and educational resources regarding Azure are readily available at https://azure.microsoft.com

The primary audience for this document includes:

- Program technical staff: This includes architects, software engineers, administrators, and operations teams within the program responsible for the STARS solution.
- External technical stakeholders: This includes technical staff from other federal, state, and territory governments agencies, including VDS embedded team members.
- ICT Technical Staff: This includes a variety of technical roles involved during the VDS program and post-go live, including network engineers, systems administrators, database administrators, security engineers, and DevOps engineers.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 485 of 1295

2.3 Scope and Exclusions

2.3.1 In Scope

This service is the common interface to support training activity reporting regardless of the source. It presents APIs for consumption by SMS applications, the STARS Provider Portal, and STA systems.

Some operations are designed specifically to support interactions with the Provider Portal and Provider Management Portal.

This document describes the processes supported by the Provider Service, with sufficient detail to build the components of the service, in conjunction with the related documents, and VET-IS documentation.

The service will receive data from training providers and store the raw files without modification. It will be able to transform these files into VET-IS JSON schema entities for validation and submission.

The service will provide API endpoints to support external SMSs, the Provider Portal, and the Provider Management Portal.

The service will keep track of user acceptance of Terms and Conditions.

2.3.2 Out of Scope

This document will not describe the SMS SDK in detail. These details will be in the SMS SDK Service design.

This document does not describe the UI design of the Provider Portal nor the Provider Management Portal.

This document does not provide details of the data structures or data elements used for the API operations, as these will be based on the VET-IS documentation, and common documentation in related documents.

2.4 Acronyms

Acronym	Definition
API	Application Programming Interface
CRUD	Create Read Update Delete
REST	Representational State Transfer
FR	Functional Requirement
NFR	Non-functional Requirement
STARS	Student Training & Activity Reporting System
VET	Vocational Education Training

Formatted: Heading 2,Reset numbering,Numbered Heading 2,h2 main heading,B Sub/Bold,hd2,h2,Header 2,B Sub/Bold1,B Sub/Bold2,B Sub/Bold11,h2 main heading1,h2 main heading2,B Sub/Bold3,B Sub/Bold12,h2 main heading3,B Sub/Bold4,B Sub/Bold13,h2 main heading4,l2,list 2

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 486 of 1295

VDS	VET Data Streamlining
IVETS	Integrated Vocational Education and Training System
DB	Database
DDS	Data Definition Language
SQL	Structured Query Language
TP	Training Provider
STA	State Training Authority
SMS	Student Management System
DET	Data Entry Tool
RAM	Relationship Authorisation Manager
PWA	Progressive Web App
AD	Active Directory
RBAC	Role Based Access Control
Micro-TP	A training organisation that does not use a Student Management System

2.5 Definitions

Formatted: Heading 2,Reset numbering,Numbered Heading 2,h2 main heading,B Sub/Bold,hd2,h2,Header 2,B Sub/Bold1,B Sub/Bold2,B Sub/Bold11,h2 main heading1,h2 main heading2,B Sub/Bold3,B Sub/Bold12,h2 main heading3,B Sub/Bold4,B Sub/Bold13,h2 main heading4,l2,list 2

Term	Definition
Micro-TP	A training organisation that does not use a Student Management System
STARS Collection Model	The physical representation of the VET-IS data packets
Activity Record	A training activity data packet

3 Design Decisions

The Provider Service will act as the single point of interaction with STARS for all training providers using the 'Direct Path'. This covers two cases:

1. External SMSs
2. Basic SMS-like features for micro-TPs (also defined by this service)

~~'Indirect Path' training providers will also be supported through the Provider Service API, via their STA systems.~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 487 of 1295

Note that FR074 refers to "exemption requests". In this design, these are referred to as "validation outcome review requests".

For a broad overview, see the STARS Solution Design (B). The original source of diagrams specific to the Provider Service are included in the Provider Service Diagrams Visio file (F).

~~3.1~~ STA systems will also use the Provider Service API but will also interact with STARS through the STA ~~Gateway~~ Bulk Submission Service.

~~For a broad overview, see the STARS Solution Design (B). The original source of diagrams specific to the Provider Service are included in the Provider Service Diagrams Visio file (F).~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 488 of 1295

4 Feature Overview

The Provider Service, as mentioned, provides the key interface for external systems, and end-user Training Providers to interact with STARS. It provides the ability to prepare, and submit training activity data, and manage that data through the submission pipeline.

The following diagram is a portion of the overall STARS Architecture diagram. It shows the role of the Provider Service in relation to the other Source Systems & Service on one side, and the Submission Pipeline Services, and Decision Services on the other.

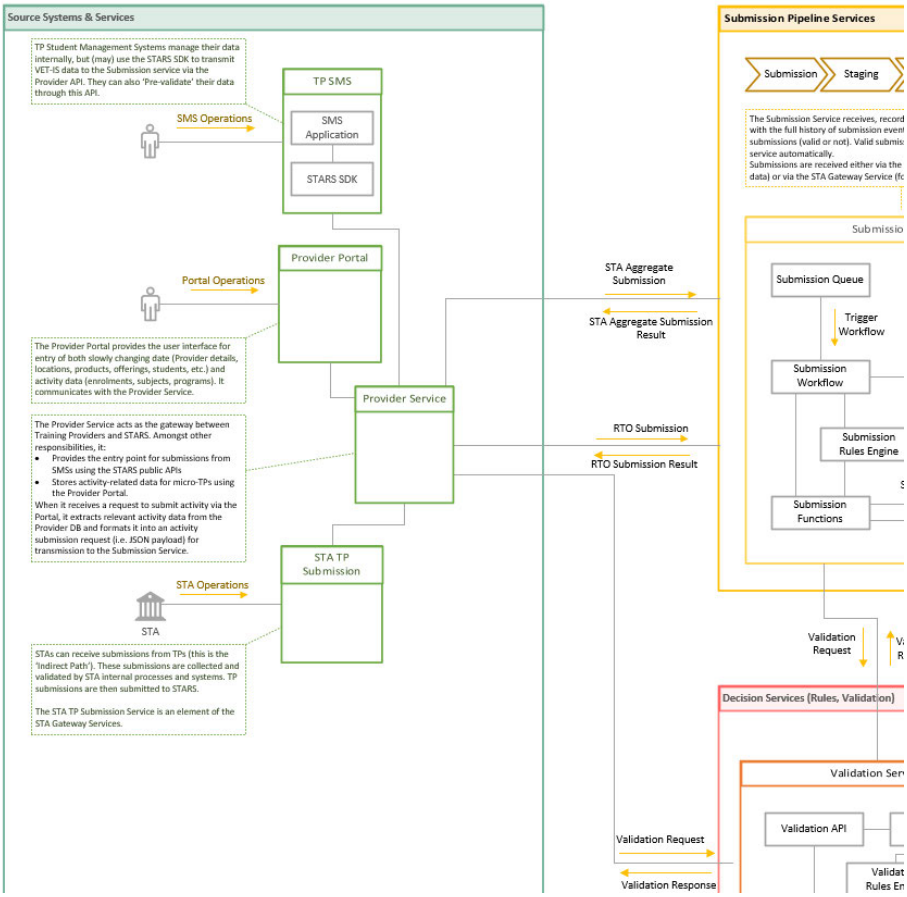


Figure 111: Provider Service Context (F:12)

The components of the Provider Service are as follows, in relation to the STARS user interfaces that depend primarily on the Provider Service. Note that the Provider Event DB is logically part of the Provider Service, but in practice will be implemented as a single Event DB for all Events across STARS.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 489 of 1295

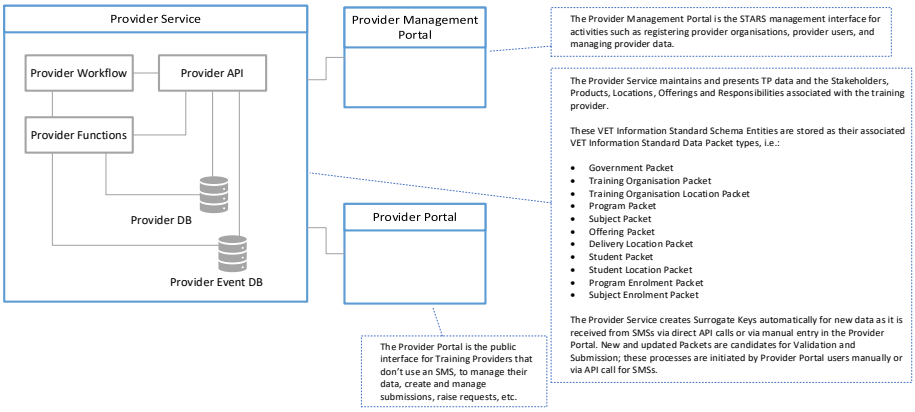


Figure 222: Provider Service Components (F:12)

4.1 Provider Workflow

This component is made up of several workflows, implemented as Logic Apps, that respond to events to be consumed for action by the Provider Service. These workflows facilitate asynchronous communication with other STARS services.

4.2 Provider Functions

The Submission Service code-based logic will be implemented as Azure Function Apps. These functions will support the operation of the service's workflows.

4.3 Provider API

The API will allow other services to flexibly query and provide data in JSON format, including validation and submission data. The API will also be used by the Portal and Management Portal user interfaces to service UI requests.

The API will conform to the STARS Design Principles and the STARS API Design Document. The endpoints described below will be standard CRUD REST endpoints. See the STARS API Design Document for success / failure status codes and expected response headers. See 6 Technical Design Patterns section below for general endpoint descriptions.

Filtering (including free-text search) will be handled using OData, as detailed below in 6 Technical Design Patterns.

The service provides API endpoints for external SMSs to use directly or via the SMS SDK. It also provides endpoints for the Provider Portal, for entering, uploading, and maintaining VET-IS Data Packet entities. Finally, there will be additional endpoints for the Provider Management Portal for handling administrative tasks.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 490 of 1295

4.4 Provider Service Internal Processes

4.5 The Provider Service internal processes can be categorised as follows, with further details described in the description of the API operations that implement these processes in section 5.4 Provider API

Table 111: Provider Service internal processes

Process Name	Description
User and Training Provider Access	Register a Training Organisation, or a User within an existing Training Organisation. This includes presenting and accepting Terms and Conditions for submitting training activity data through STARS. This category also covers operations for logging in and out of the Provider Portal.
Training Activity Data Preparation	This includes uploading training activity data from files, adding or updating through APIs, and validating the data prior to submission.
Training Activity Data Submission	A formal acknowledgement that the prepared data is ready to be reported for National and State funding and reporting purposes.
Manage Submission and Validation Results	View the progress of validations (either before or after submission), drill into validation outcome details, and request a review of validation outcomes.
Manage Training Provider Data	Enter or update reference data specific to this Training Organisation if STARS is the Source of Truth for such data. If the Source of Truth is outside STARS, such data can be viewed only.
Preference Management	User preferences for Notifications can be managed through this service.
User & System Information	This category of processes includes providing notifications and messages and content for user information, as well as providing form details for data entry.

Formatted: Caption, Keep with next

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 491 of 1295

5 Technical Design

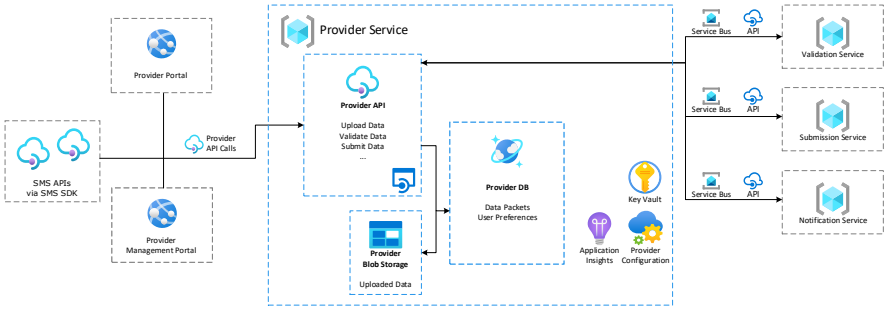


Figure 33: API Design Overview (F:1)

5.1 Provider Service Config

Azure App Configuration will be used across all services to store configuration values. Any secret values will be stored in KeyVault and referenced by App Configuration. Each of the components will require their own set of configuration values and these will be refined closer to development.

5.1.1 Provider DB Config

The Provide DB will be configured for Single-region write account with data distributed across multiple regions in production and hotfix, and Single-region write for other environments.

It will use Autoscale provisioned throughput.

Analytical Storage is not required for the Provide DB.

The following table can be updated as specific configuration values are identified.

Table 22: Provider DB Configuration

Formatted: Caption, Keep with next

Key	Value	Comments
DB:ConnectionString	-	Stored in Key Vault.

5.1.2 Provider API Config

The Provider Service will follow the STARS pattern for App Service configuration, covering appropriate configures for ~~Non-~~Non-PROD and PROD environments).

The following table can be updated as specific configuration values are identified.

Table 33: Provider API Configuration

Formatted: Caption, Keep with next

Key	Value	Comments
	-	Stored in Key Vault.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 492 of 1295

5.1.3 Blob Storage Config

The Blob Storage accounts used for the Provider Service will use LRS for ~~Non-~~Non-PROD environments, and GZRS for PROD environments.

The following table can be updated as specific configuration values are identified.

Table ~~444~~: Provider Blob Storage Configuration

Formatted: Caption, Keep with next

Key	Value	Comments
	-	Stored in Key Vault.

5.1.4 Logic Apps Config

For each logic app, the following will need to be defined, and used as Parameters within the Logic App, or within content of the Logic App bicep templates.

As per general STARS technical design, all Logic Apps will be implemented with the Logic Apps (Standard) resource type in a single-tenant Azure Logic Apps host environment.

The following table can be updated as specific configuration values are identified.

Table ~~555~~: Provider Logic App Configuration

Formatted: Caption, Keep with next

Key	Value	Comments
Name	-	
Resource Group	Specific to the Provider Service	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 493 of 1295

5.2 Provider Event DB

Events will be managed and stored by the Event Service.

5.3 Provider DB

The Provider Database will be implemented in Azure Cosmos DB, storing non-relational data. It will utilise a standardised document-oriented JSON schema to ensure consistency in the management and presentation of provider data. The file schemas stored in this database are described below.

5.3.1 User Configuration

The service needs to track when users have accepted the terms and conditions of using the Provider Service.

Table 666: Provider Service User Configuration Fields

Formatted: Caption, Keep with next

Field	Type
UserID	Key to User.
HasAcceptedTCsDateTime	BooleanDateTime. The point in time at which the Has the user accepted the terms and conditions required before using this service?.

5.3.2 Validation Request

When a provider requests a set of packets for validation, a Validation Request entity is created to track the relevant packets for this validation. See the Validation Service Detailed Design Document (H) for further details.

Table 777: Validation Request Fields

Formatted: Caption, Keep with next

Field	Type
RelevantPackets	
ValidationStatus	
RequestDateTime	DateTime. Set to creation time of the validation request.

5.3.3 Submission

When a provider submits a set of packets for submission, a Submission entity is created to track the relevant packets for this validation.

Table 888: Submission Fields

Formatted: Caption, Keep with next

Field	Type
RelevantPackets	
SubmissionStatus	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 494 of 1295

SubmissionDateTime	DateTime. Set to creation time of the submission.
PreviousSubmission	Submissions can be a 'resubmission', where a previous submission failed validation and is now being updated, or where the original submission passed all validation, but has since been updated. This field records the relationship to a previous submission.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 495 of 1295

5.3.4 Angular Dynamic Forms

The Provider Portal forms will be defined using Angular dynamic forms. This will provide the field types and associated details (e.g., options for dropdown selection controls, field lengths, etc).

The Provider Portal can retrieve these forms and use them to build a Direct Entry form for each type of activity record. See the official Angular "Building dynamic forms" page for more implementation details.[1]

Table 999: Dynamic Form Fields

Formatted: Caption, Keep with next

Field	Type
FormCode	Unique identifier for retrieving a relevant form.
ActivityRecordType	The activity record type that this form is associated with.
Fields	A collection of fields metadata for use in Angular.

5.3.4.1 Fields Metadata Properties

Metadata values will include the following:

Table 101010: Dynamic Form Field Metadata

Formatted: Caption, Keep with next

Metadata	Description
Id	Unique Identifier
ControlType	Specifies if this is e.g., an input field, dropdown, checkbox etc.
Label	Label / heading text for field.
HelpText	Additional text to describe this field in more detail.
ValidationChecks	A collection of basic validation types relevant to this field.
DefaultValue	The initial value for this field.
DropdownOptions	The set of options if this is a dropdown control. A collection of objects.
DropdownValueKey	When a dropdown option is chosen, this is the object property to use as the value.

[1] https://angular.io/guide/dynamic-form

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 496 of 1295

GroupName	Multiple fields may specify a common group name for layout purposes.
Order	Value to specify the order of each field.

5.3.4.2 Field Validation Checks

Note that these validation checks will be for basic informational purposes only and will not block users from saving incomplete data.

Table 111~~11~~: Dynamic Form Field Validation Checks

Formatted: Caption, Keep with next

ValidationCheck	Description
MustBeTrue	Field must be set to true.
Required	Field cannot be empty nor null.

5.3.4.3 Field Control Types

ControlTypes will be one of the following:

Table 121~~2~~: Dynamic Form Control Types

Formatted: Caption, Keep with next

ControlType
Text
Currency
Dropdown
Checkbox

5.3.4.4 Example Form Object

```
{
  "FormCode": "Student",
  "ActivityRecordType": "Student",
  "Fields": [
    {
      "ControlType": "Checkbox",
      "Label": "Accept Terms and Conditions",
      "HelpText": "To use STARS, you must accept the terms and conditions.",
      "Order": 100,
      "ValidationChecks": ["Required", "MustBeTrue"]
    },
    {
      "ControlType": "Dropdown",
      "Label": State / Territory",
      "DropdownOptions": [
        {"Key": "tas", "Name": "Tasmania", "Country": "Aus"},
        {"Key": "wa", "Name": "Western Australia", "Country": "Aus"},
        {"Key": "nt", "Name": "Northern Territory", "Country": "Aus"},
      ],
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 497 of 1295

```
      "DropdownValueKey": "Key",
      "GroupName": "Location Details",
      "ValidationChecks": ["Required"],
      "Order": 50
    }
  ]
}
```

5.3.5 VET-IS Data Packets

The Provider Service helps prepare and manage VET-IS data packets for submission:

- Government Packet
- Training Organisation Packet
- Training Organisation Location Packet
- Program Packet
- Subject Packet
- Offering Packet
- Delivery Location Packet
- Student Packet
- Student Location Packet
- Program Enrolment Packet
- Subject Enrolment Packet

Details of these data packet schemas can be found in the VET-IS Interchange Specification in GovTeams and is also referenced below in section 1.1~~7.1 VET-IS Data Packet Schema~~.

Field Code Changed

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 498 of 1295

5.4 Provider API

The Provider API will be implemented in Azure API Management, with the backend implemented as one or more Azure API Apps. API Apps will support the implementation of OData APIs to provide consistent API interactions between STARS Services and for consumption by external services.

~~The Provider Service API will be presented at the following path:~~

~~<STARS Base URL>/providerService/~~Details of the API are specified in the Provider Service API Design document (I).

~~5.4.1 Provider Service Endpoints~~

~~Provider API Endpoint details are listed here, with details of the implementations of these endpoints found in section 5.5 Provider API Operations below.~~

~~API Endpoint Name~~	~~Base Definition~~	~~path~~
~~Get Terms and Conditions~~	~~Read Endpoint - List~~	~~GET <serviceurl>/Terms~~
~~Register Training Provider~~	~~Create Endpoint~~	~~POST <serviceurl>/ProviderRegistration~~
~~Register User~~	~~Create Endpoint~~	~~POST <serviceurl>/UserRegistration~~
~~Login~~	~~-~~	~~Azure AD B2C Single Sign On~~
~~Logout~~	~~-~~	~~Azure AD B2C Single Log Out~~
~~Upload Data~~	~~Create Endpoint~~	~~POST <serviceurl>/activity/files~~
~~Validate Data~~	~~Create Endpoint~~	~~POST <serviceurl>/validationRequests~~
~~Create Activity Record (General)~~	~~Create Endpoint~~	~~POST <serviceurl>/activity/{activityType}~~
~~Update Activity Record (General)~~	~~Update Endpoint~~	~~PUT <serviceurl>/activity/{activityType}/{id}~~
~~Bulk Upsert Activity Records~~	~~Create Endpoint~~	~~POST <serviceurl>/activity/bulkUpdate~~
~~Delete Activity Record (General)~~	~~Delete Endpoint~~	~~DELETE <serviceurl>/activity/{activityType}/{id}~~
~~Get List of Activity Records (General)~~	~~Read Endpoint - List~~	~~GET <serviceurl>/activity/{activityType}~~
~~Get Activity Record by ID (General)~~	~~Read Endpoint - Detail~~	~~GET <serviceurl>/activity/{activityType}/{id}~~
~~Purge Activity Records~~	~~Delete Endpoint~~	~~DELETE <serviceurl>/activity/ProviderService.Purge~~
~~Get Local Validation Request History~~	~~Read Endpoint - List~~	~~GET <serviceurl>/validationRequestSummaries~~

Formatted Table

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 499 of 1295

API Endpoint Name	Base Definition	path
Submit Data	Create Endpoint	POST <serviceurl>/submissions
Get Validation Results	Read Endpoint - List	GET <serviceurl>/validationRequests
Get Submissions	Read Endpoint - List	GET <serviceurl>/submissions
Get Local Submission History	Read Endpoint - List	GET <serviceurl>/submissionSummaries
Request Validation Outcome Review	Create Endpoint	POST <serviceurl>/validationRequest/[id]/results[id]/reviews
Get Validation Outcome Reviews	Read Endpoint - List	GET <serviceurl>/validationRequest/[id]/results[id]/reviews
Get Training Organisation	Read Endpoint - Detail	GET <serviceurl>/trainingProviders/[id]
Get Subjects	Read Endpoint - List	GET <serviceurl>/trainingProviders/[id]/subjects
Get Programs	Read Endpoint - List	GET <serviceurl>/trainingProviders/[id]/programs
Get Delivery Locations	Read Endpoint - List	GET <serviceurl>/trainingProviders/[id]/DeliveryLocations
Get Notification Preferences	Read Endpoint - Detail	GET <serviceurl>/users/[id]/notificationPreferences
Update Notification Preferences	Update Endpoint	PUT <serviceurl>/users/[id]/notificationPreferences
Get Notifications	Read Endpoint - List	GET <serviceurl>/notifications
Get Business Glossary	Read Endpoint - Detail	GET <serviceurl>/reference/businessGlossary
Get Form by Record Type	Read Endpoint - Detail	GET <serviceurl>/forms/{activityType}

Formatted Table

5.5 Provider API Operations

The Provider Service internal processes are facilitated through API operations defined in the Provider API. The processes for each of these operations are defined below. The implementation details (paths, schemas, security, etc) are documented in the Provider Service API Design document (I).Details of these operations are as follows:

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 500 of 1295

5.5.1 User and Training Provider Access

5.5.1.1 Get Terms and Conditions

For Training Providers and users to be granted access to report their training activity through STARS, certain Terms and Conditions must be read and accepted. The Reference Data Service stores and manages the Terms and Conditions for STARS, so the Provider Service retrieves this data from there.

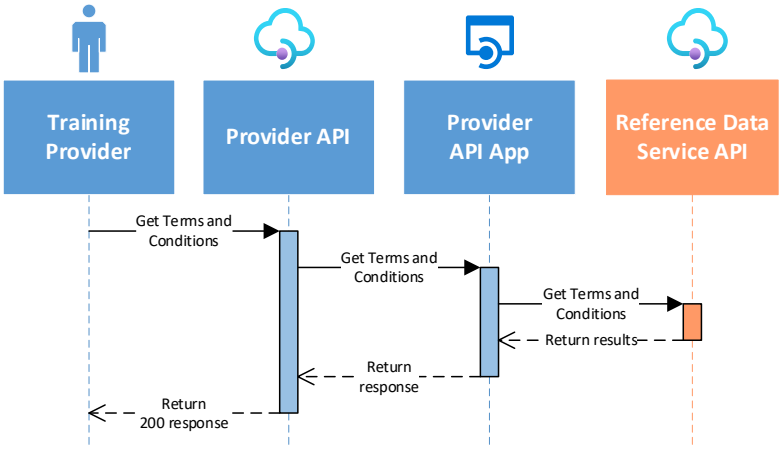


Figure 444: Sequence - Get Terms and Conditions (F:33)

5.5.1.2 Register Training Provider

This operation onboards a Training Provider to STARS prior to them reporting their Training Activity. It initiates a workflow for approval of their onboarding, and preparation of resources required for the new Training Provider to prepare and submit their Training Activity data to STARS. This request must contain agreement that Training Provider terms and conditions have been read and accepted. The registration workflow occurs largely outside of STARS. If there are any issues with the registration, the initiator of the registration request will be contacted to resubmit their registration with updated details. Once the onboarding processes are complete, the Training Organisation will have all details they need to begin submitting their training activity records to STARS. The operation supports RTOs and Non-registered Training Organisations. The details of the broader registration process are not within the scope of this document.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 501 of 1295

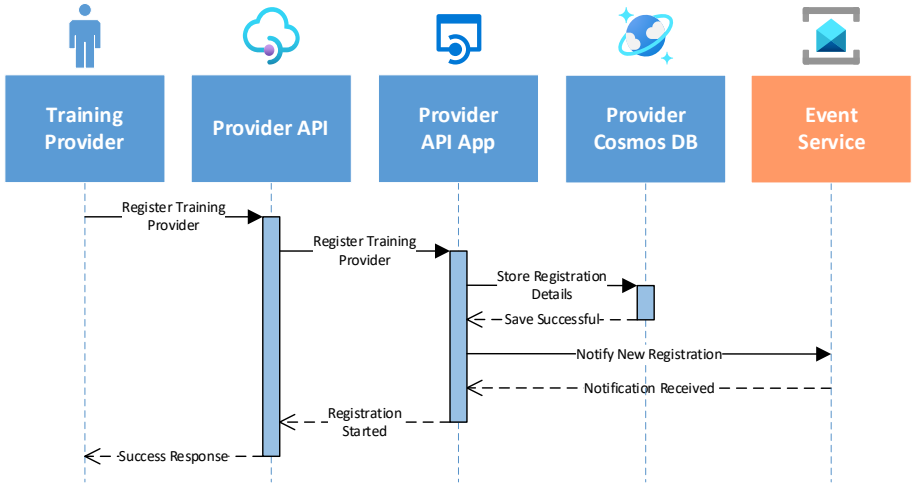


Figure 555: Sequence - Register Training Provider (F:25)

5.5.1.3 Register User

Once the Training Provider has been registered, specific users can be registered to begin preparing and submitting their Training Activity Records. This process is quite similar and involves manual approval and preparation steps outside of STARS. This request must contain agreement that Training Provider User terms and conditions have been read and accepted. The details of the broader process are not within the scope of this document.

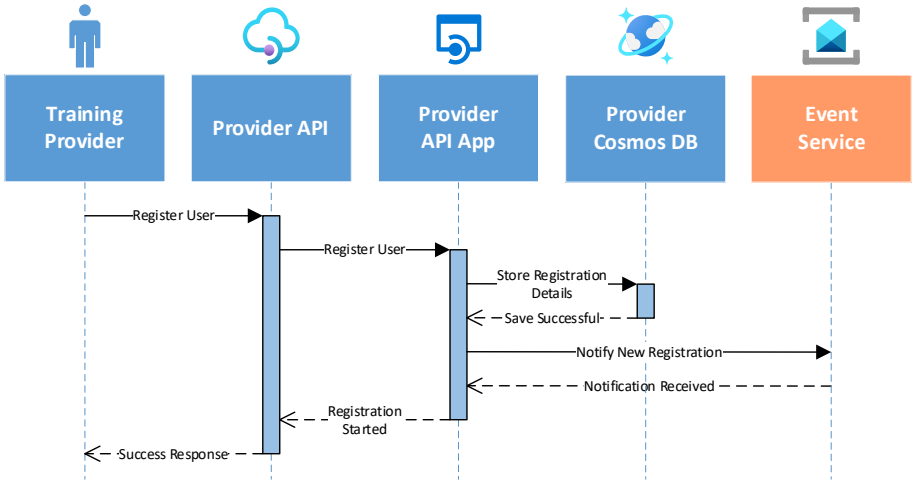


Figure 666: Sequence - Register User (F:26)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 502 of 1295

5.5.1.4 Login

Facilitates Azure AD B2C Single Sign-On

5.5.1.5 Logout

Facilitates Azure AD B2C Sign Out.

5.5.2 Training Activity Data Preparation

It is anticipated that the primary interaction for Training Activity Data Preparation will be through the Provider Service API, using JSON payloads to add/update training activity data in the Provider Database, rather than uploading files.

~~5.5.2~~ We do have requirements to support data entry through file upload, with a potential use case being a Training Organisation that doesn't use an SMS but has a different system of some sort that isn't capable of interacting with the API but is capable of generating files. A specific example of such a Training Organisation has not been identified to date, though, so we currently expect all Training Providers to enter and update training activity data through the Provider Service API, either via an SMS or via the Provider Portal.

Formatted: Normal

5.5.2.1 Upload Data

When files are uploaded, they will be immediately stored as-is. This will allow the raw files to be listed and downloaded as required. This uploaded data will then be converted to the STARS Collection Model before being stored in Cosmos DB where it will be in the same format as if it was directly entered through forms, or added through the API, allowing consistent behaviour for managing the data moving forward.

Blob Storage and Cosmos DB will both have separate containers for each Training Provider. Training providers will have access only to their own container, and other parties (e.g., STAs) will not be able to access this data.

Only one upload process can be running at any given time for a TP. There is a maximum file size configured.

Details of the file upload format are pending meetings with Training Providers that intend to use this feature, so DXC can understand what capabilities they have for generating these files, in order to design a viable file format.

5.5.2.2 List Uploaded Files

Any uploaded file can be viewed in a list in the Provider Portal. This operation provides the list of uploaded files for the Training Organisation to support that functionality.

5.5.2.3 Download File

Any file that has been uploaded, can be downloaded again. This will just be a direct download of exactly what was uploaded originally.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 503 of 1295

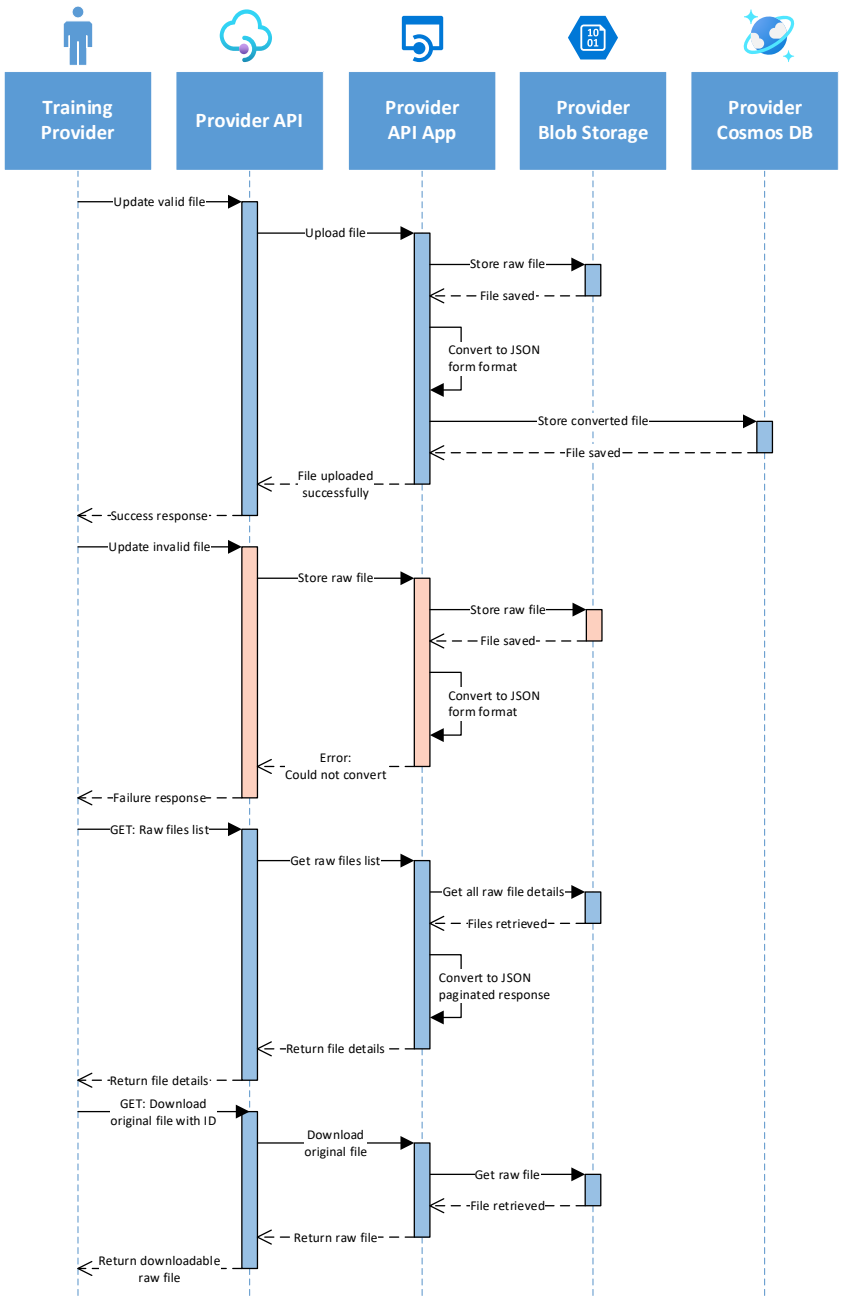


Figure 777: Sequence – Upload, List and Download ~~Data~~ Files (F:4)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 504 of 1295

~~5.5.2.2~~5.5.2.4 Validate Data

The Provider service facilitates data validation before, or after submission. This operation covers 'pre-validation', or validation in preparation for submission. This operation will allow for validation of any size data set, while providing a rapid response to the user, and a clear view of expected outcomes along with progress towards those outcomes. This operation is part of this process and provides a quick response that gives visibility to any subsequent actions that are yet to be completed. Other operations can be used to receive updates on progress towards completion of those subsequent actions.

As with submission, validation requests can contain one or more records. The concept of a record here, can be a single Data Packet with no dependencies, or a set of related Data Packets. For example, if a Subject Enrolment packet is to be validated, it will be submitted for validation with all associated Data Packets that make up the 'Unit Record' for that Enrolment. This detail is not reflected in the operation details, as it relates to the payloads that will be submitted to the operation, rather than the mechanics of the operation itself. Another such detail is that state-specific fields (SSF) will be provided along with national fields, in these training activity records.

Also, it is expected that any validation request will be related to data that is already stored in the Provider DB. SMS vendors that store the source data outside of STARS will need to perform a Save operation prior to this Validate operation. Such data can then be removed after validation is complete.

Here is the sequence of events for initiating a Validation Request:

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 505 of 1295

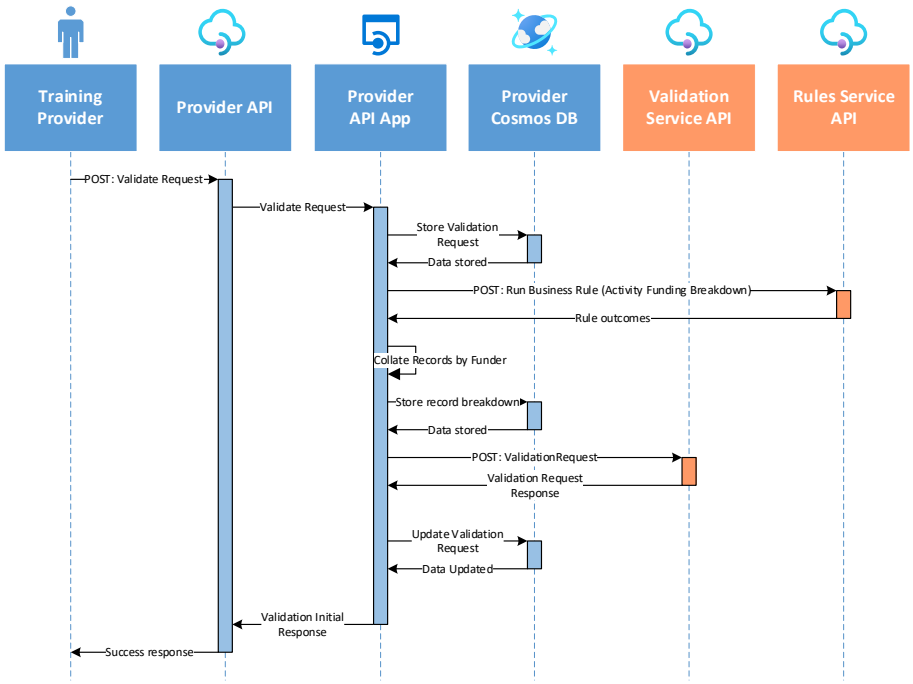


Figure 888: Sequence – Validate Data (F:5)

For larger sets of records, it is likely that some validation steps will be pending a response. In this case, it is expected that the Provider Portal or SMS system will request updated information regarding the Validation Request. This request could also be made at any time, when viewing a previously submitted validation request or the like.

The following diagram shows the process for retrieving details of a specific Validation Request:

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 506 of 1295

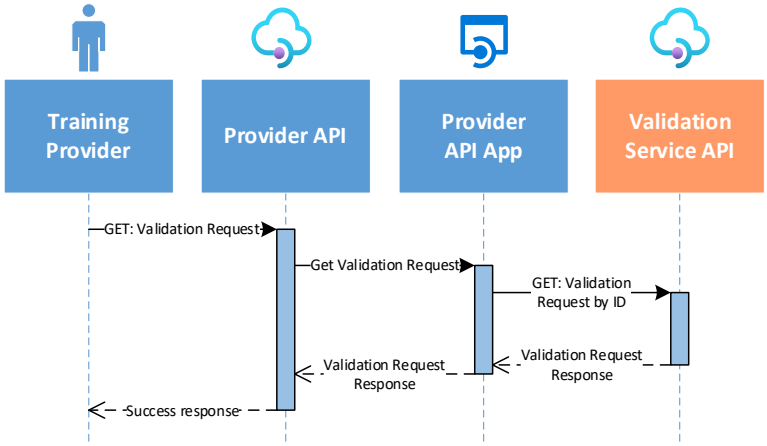


Figure 999: Sequence - Get Validation (F:7)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 507 of 1295

~~5.5.2.3~~5.5.2.5 Create Activity Record (General)

As a 'General' operation, the implementation of this will cover all Training Activity records defined in the Collection Specification.

The Provider Service supports creating and updating activity data. This operation would be invoked by a 'Save' action in the UI for newly entered activity data or could be used by SMS applications to send through data for validation of a single record.

The Provider Service will apply initial duplicate detection to see if this is, in fact, a new Record, before saving to the Provider DB. In this context, the Check Exists action that performs this initial duplication detection is specifically within the context of the Training Organisation that is entering the data and will be based on identifiers and rules per entity. Details about what constitutes a duplicate in this context for each entity (data packet) are yet to be determined.

When the Record is saved, the Provider Service will generate a STARS Identifier for the Entity, as a unique identifier for use in STARS.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 508 of 1295

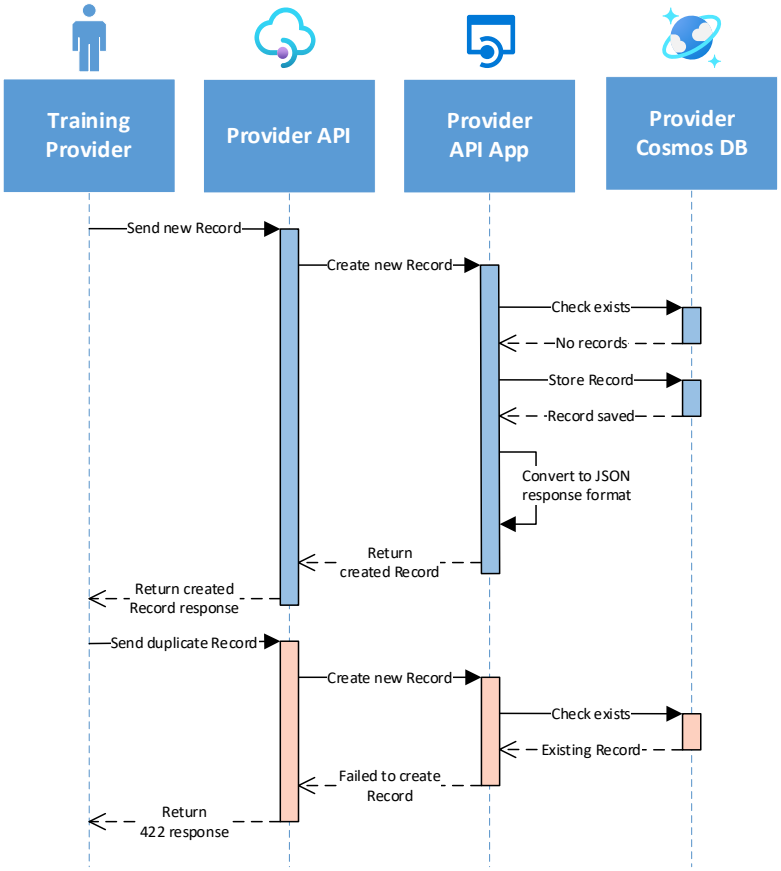


Figure 101~~0~~10: Sequence – Create Activity Record (General) (F:12)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 509 of 1295

~~5.5.2.4~~5.5.2.6 Update Activity Record (General)

As a 'General' operation, the implementation of this will cover all Training Activity records defined in the Collection Specification.

5.5.2.7 Activity records that have already been given STARS identifiers can be modified and updated through this operation. This operation just updates the data locally, even if the record has previously been submitted. The Create Local Submission

Once data has been entered into the Submission DB, a collection of Records can be selected to be submitted for reporting & processing. This involves creating a Submission record in the Provider DB which contains some header information, as well as a list of Record IDs that are being submitted. The response will include the Submission ID.

It's important to note that this operation just creates the Submission within the Provider Service without actually Submitting it to the Submission Service. That is done through the Submit Data operation.

Formatted: Normal

Formatted: Keep with next

Figure 1818: Sequence – Create Local Submission (F:35)

Formatted: Caption, Centered

5.5.2.8 Update Local Submission

It's possible for a user to update the header information for a Submission, or to add Records to, or remove Records from the Submission. These changes are handled through this Update Local Submission operation. Again, these changes just update the local data for the Submission, without actually Submitting it to the Submission Service. That is done through the Submit Data operation.

Formatted: Keep with next

Figure 1919: Sequence - Update Local Submission (F:36)

Formatted: Caption, Centered

Submit Data~~Submit Data~~ operation is to be used to submit such updates. This operation also doesn't validate the data. I can be used to save progress even if some validation errors exist.

This operation can be used to undo a delete action, as well (see 5.5.2.10~~5.5.2.6~~ Delete Activity Record (General)).

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 510 of 1295

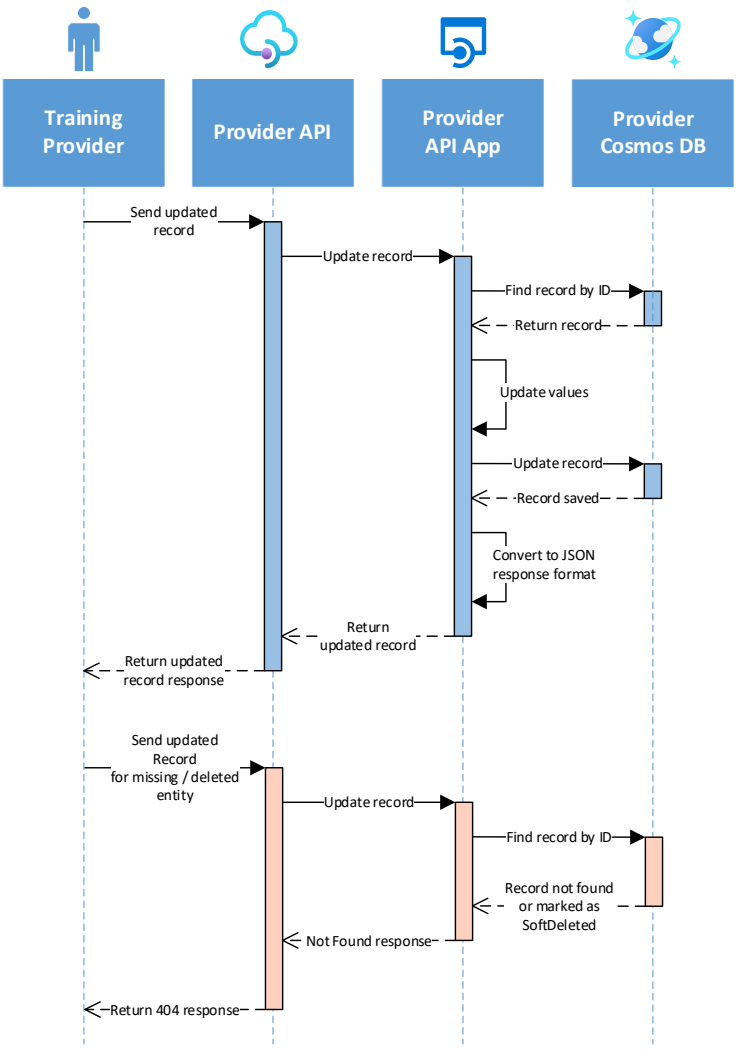


Figure 111~~11~~: Sequence – Update Activity Record (General) (F:13)

~~5.5.2.5~~5.5.2.9 Bulk Upsert Activity Records

This operation takes a collection of Activity Records, and adds or updates them in the Provider DB. It can be used for Bulk Edit operations, or to load data to the Provider Service prior to Validation (e.g., for SMS or STA pre-submission validation).

If any data us new, the Provider Service will generate a STARS Identifier as each new Entity as an internal unique identifier for use within STARS.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 511 of 1295

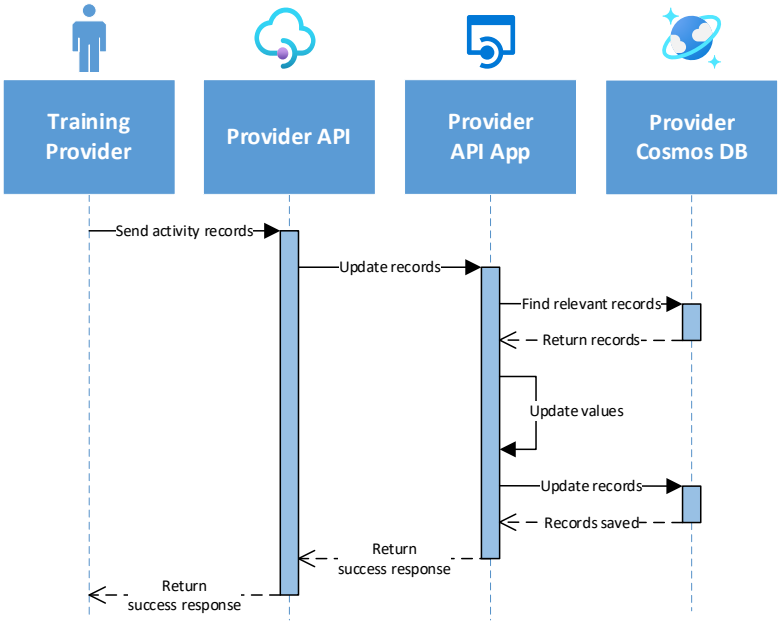


Figure 121~~212~~: Sequence – Bulk Edit Activity Records (General) (F:14)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 512 of 1295

~~5.5.2.6~~5.5.2.10 Delete Activity Record (General)

As a 'General' operation, the implementation of this will cover all Training Activity records defined in the Collection Specification.

Since the Provider Service is used to prepare data prior to Submission (as well as managing this data after it has been submitted), it's possible that data may have been added in error and should be deleted. This operation provides that functionality. The record will be soft-deleted, though, in case the delete is to be undone. This operation will check that the record has not been submitted first, though, to check if it is allowed or not.

Figure 131~~313~~: Sequence – Delete Activity Record (General) (F:15)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 513 of 1295

~~5.5.2.7~~5.5.2.11 Get List of Activity Records (General)

As a 'General' operation, the implementation of this will cover all Training Activity records defined in the Collection Specification.

A list of Training Activity Records that have been prepared for submission (or submitted), can be retrieved through this operation. As with all similar operations, OData filters and selections can be used to manage the results.

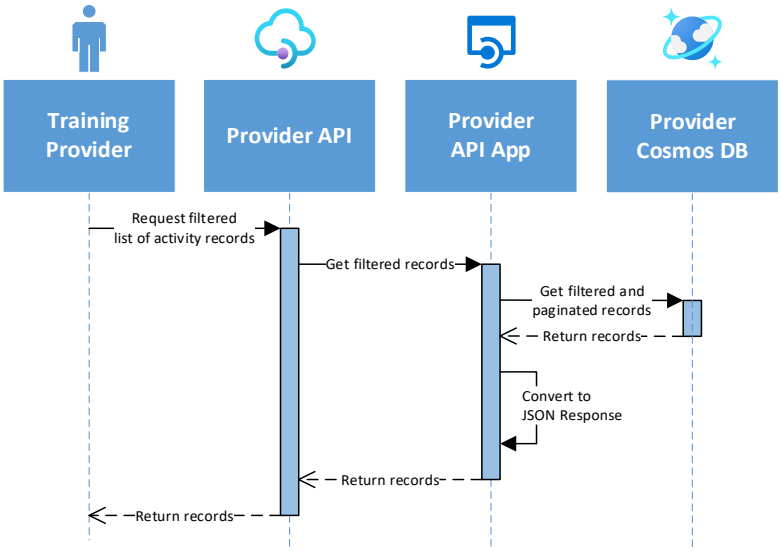


Figure ~~141~~414: Sequence – Get List of Activity Records (General) (F:16)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 514 of 1295

~~5.5.2.8~~5.5.2.12 Get Activity Record by ID (General)

As a 'General' operation, the implementation of this will cover all Training Activity records defined in the Collection Specification.

This operation returns a single Training Activity record by its STARS identifier.

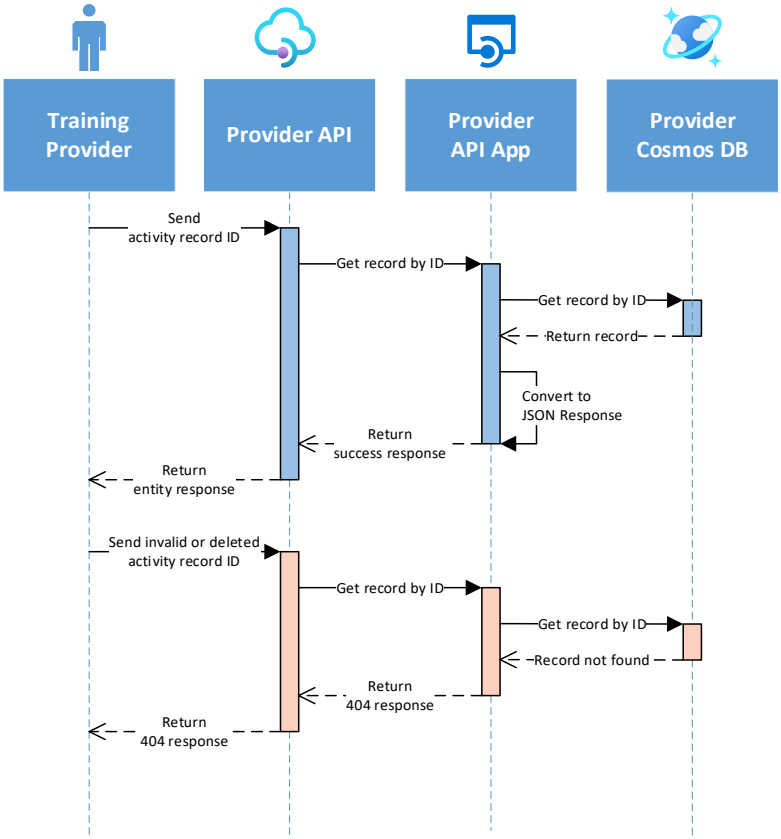


Figure 151~~5~~15: Sequence – Get Activity Record by ID (General) (F:17)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 515 of 1295

~~5.5.2.9~~5.5.2.13 Purge Activity Records

For SMS providers, or STAs, the Bulk Upsert Activity Records and Validate Data operations can be used to validate data prior to submission. This data, however, may not want to be retained in the Provider Database, given the source of truth is outside STARS. This operation allows all, or selected data from a Training Organisation to be removed from the Provider DB, provided it has not yet been submitted. Any activity records that are requested to be purged, but have been submitted, will not be deleted. The response will include IDs of records that we not deleted.

Figure 1616~~1621~~: Sequence - Purge Activity Records (F:24)

~~5.5.2.10~~5.5.2.14 Get Local Validation Request History

This operation provides a summary list of previous Validations requested for the given context (Training Organisation or User). While similar to 5.5.4.1 Get Validation Results, this operation only provides summary information/metadata relating to the Validation Requests, not the actual details of the validation results.

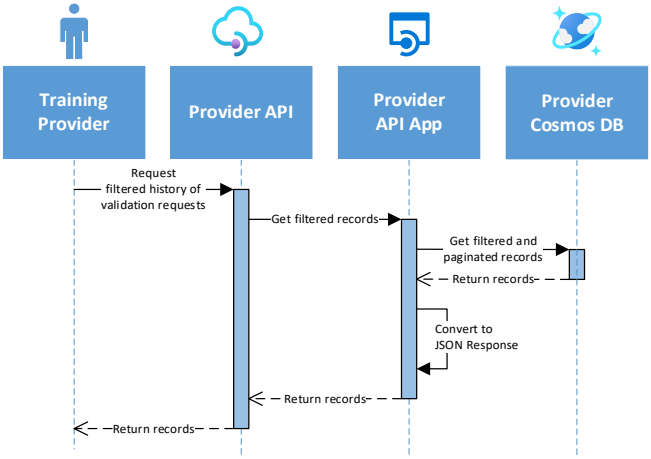


Figure 1717~~1716~~: Sequence – Get Local Validation Request History (F:20)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 516 of 1295

5.5.3 Training Activity Data Submission

Providers can submit data to the Submission Service and receive the validation results. The provider can view a log of submissions and associated statuses, including for in-progress submissions. This process is used both by SMSs via the SMS SDK, and by micro-TPs.

"Minimum datasets" can be submitted. Nil report can be submitted. These are largely validation concerns, more than specific Provider Service details, though. The Provider Service accepts VET-IS data in all its forms.

Individual users of a TP can see submission results for all users within that TP.

5.5.3.1 Create Local Submission

Once data has been entered into the Submission DB, a collection of Records can be selected to be submitted for reporting & processing. This involves creating a Submission record in the Provider DB which contains some header information, as well as a list of Record IDs that are being submitted. The response will include the Submission ID.

It's important to note that this operation just creates the Submission within the Provider Service without actually Submitting it to the Submission Service. That is done through the Submit Data operation.

Formatted: Normal

Formatted: Keep with next

Figure 181~~818~~: Sequence – Create Local Submission (F:35)

Formatted: Caption, Centered

5.5.3.2 Update Local Submission

It's possible for a user to update the header information for a Submission, or to add Records to, or remove Records from the Submission. These changes are handled through this Update Local Submission operation. Again, these changes just update the local data for the Submission, without actually Submitting it to the Submission Service. That is done through the Submit Data operation.

Formatted: Keep with next

Figure 191~~919~~: Sequence - Update Local Submission (F:36)

Formatted: Caption, Centered

~~5.5.3.1~~5.5.3.3 Submit Data

As with validation, the submission process will save the collection of records that are to be submitted. It will then send this off to the Submission service to handle the rest. The initial response will include information about what steps are to be taken for each record to complete processing, along with whatever progress has been completed before this initial response was returned. It is expected that additional calls to get submission details will be made to provide updates on progress through the submission pipeline.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 517 of 1295

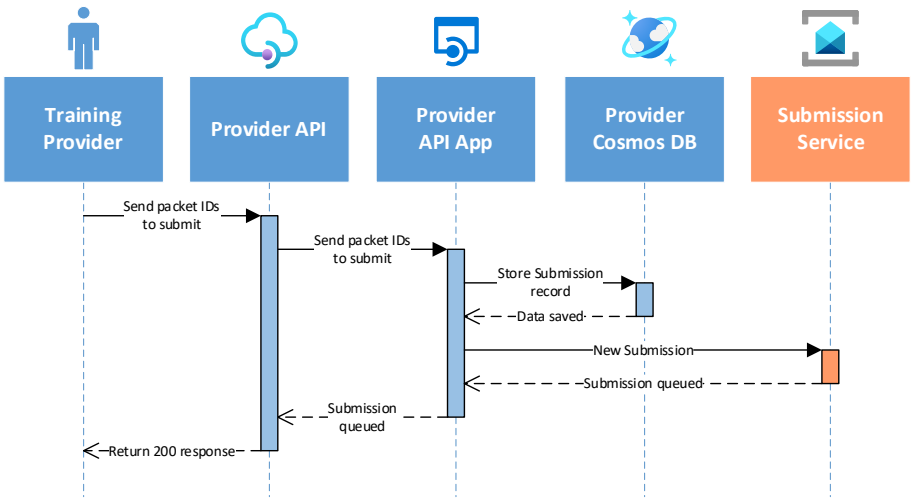


Figure 20202017: Sequence – Submit Data (F:6)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 518 of 1295

5.5.4 Manage Submissions and Validation Results

5.5.4.1 Get Validation Results

This is a simple operation, where the Provider Service facilitates retrieval of Validation information from the Validation Service API. As with all 'List' type operations, pagination query params will be provided in the initial query and passed to the Validation Service API query to ensure we only get the data we intend to display.

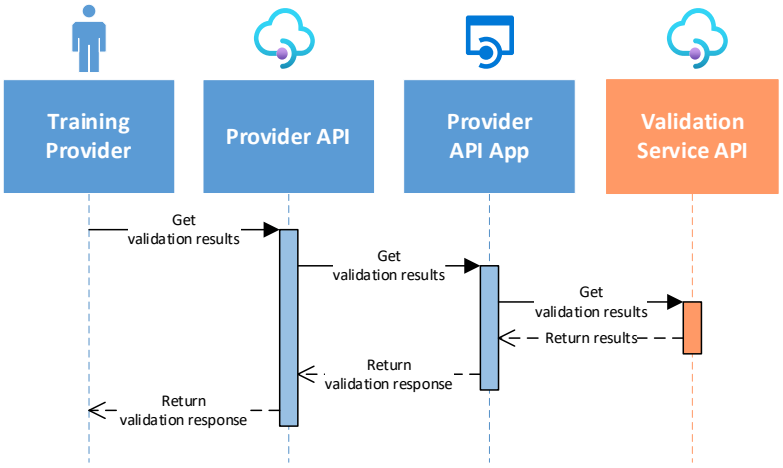


Figure 21212118: Sequence – Get Validation Results (F:8)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 519 of 1295

5.5.4.2 Get Submissions

Another straight-forward operation, this time for retrieving Submissions that have previously been reported. The results will include a summary of Records within the submission, including metrics regarding the progress of these records along the submission pipeline. It will not include a list of the Records themselves, though. Pagination will be supported, as well as other filtering support for submitters, date ranges and statuses.

This operation is used for form pre-population when creating new Records.

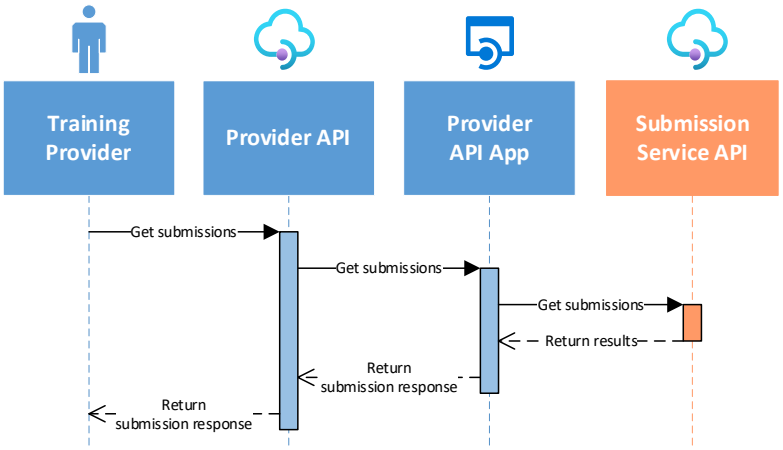


Figure 22222219: Sequence – Get Submissions (F:9)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 520 of 1295

5.5.4.3 Get Local Submission History

This operation provides a summary list of previous Submissions for the given context (Training Organisation or User). While similar to 5.5.4.2 Get Submissions, this operation only provides summary information/metadata relating to the Submissions, not the actual details of the submission results.

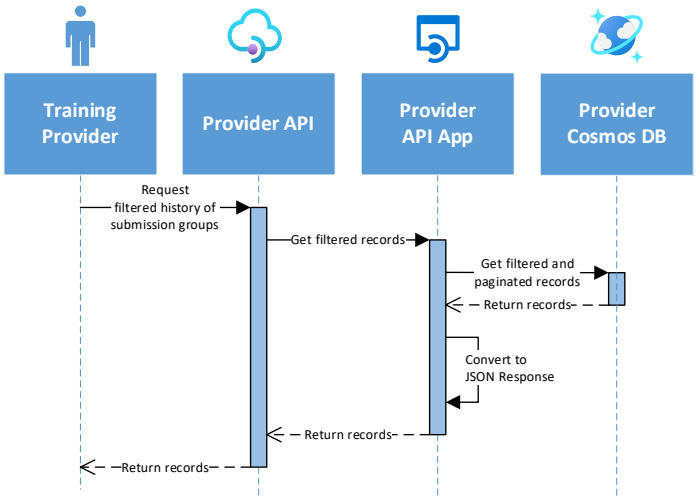


Figure 23232320: Sequence – Get Local Submission History (F:21)

5.5.4.4 Request Validation Outcome Review

If validation errors are received, it's possible that the Provider is confident that the data is correct, and incorrect reference data or problems with validation rules are causing the validation failure. In this case, the Provider can choose to create a Validation Outcome Review. This will be submitted through to STARS and processed through the Rules Service so the appropriate stewards of the rules and reference data can investigate and respond. It is expected that Notifications will be sent to advise of outcomes/progress for the Review, and any such updates to Review outcomes can be retrieved periodically through the Get Validation Outcome Reviews operation.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 521 of 1295

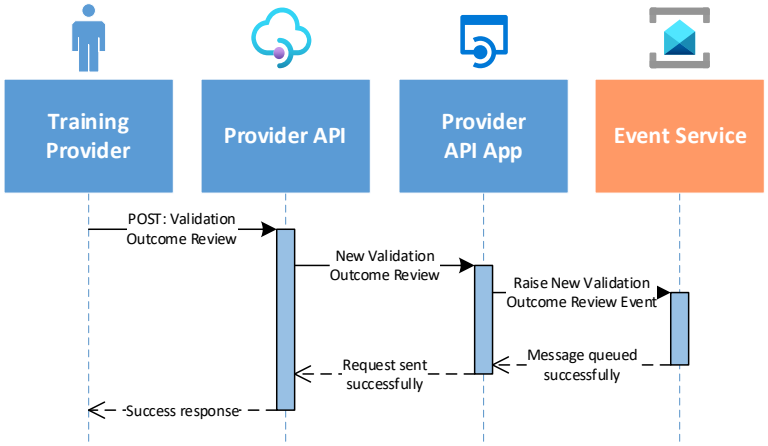


Figure 24242422: Sequence – New Validation Outcome Review (F:18)

5.5.4.5 Get Validation Outcome Reviews

A list of previously submitted Reviews can be requested through this operation. The Rules Service is the owner of the Review details, and the Provider Service will facilitate access to that data. It is expected that the list of reviews will be filtered to specific recipients, but the service will also allow the list of reviews to cover multiple users for a single Training Provider, so that an administrator can have visibility across all Reviews related to their organisation.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 522 of 1295

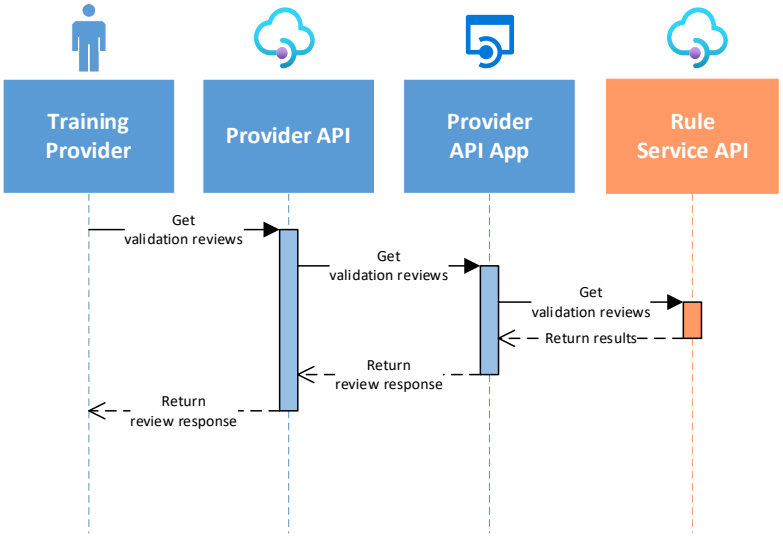


Figure 2525~~2523~~: Sequence – Get Validation Reviews (F:11)

5.5.5 Manage Training Provider Data

Training Provider Data refers to Reference Data related to the Training Provider that can be used to support training activity submissions.

AWAITING ACTION RESPONSE (W12 – Action #05a): It's possible roles of users within a Training Organisation would be managed through the RAM. If not, Training Organisation admins would use STARS to update user roles within their organisation. Assumption – roles will be handled in RAM.

There's an outstanding action (W29 – Action #08) pertaining to what Training Provider Data would be managed within STARS. Until this is resolved it is assumed all Training Provider Data will be managed outside STARS but can be viewed in STARS. Training Providers would have visibility of their reference data, to support their submissions.

5.5.5.1 Get Training Organisation

This operation retrieves details of a single Training Organisation relating to the current scope of access. The data will be retrieved from the Reference Data Service API, which will manage synchronisation of this data with the Source of Truth.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 523 of 1295

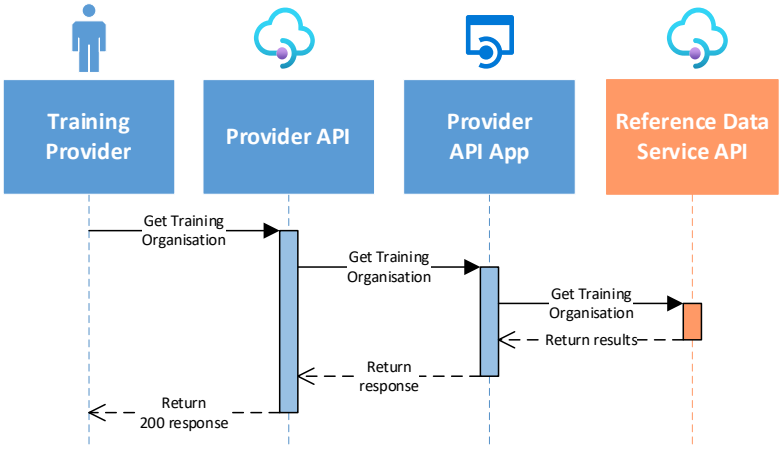


Figure 26262624: Sequence - Get Training Organisation (F:29)

5.5.5.2 Get Subjects

This operation retrieves the Subjects for a single Training Organisation. The data will be retrieved from the Reference Data Service API, which will manage synchronisation of this data with the Source of Truth.

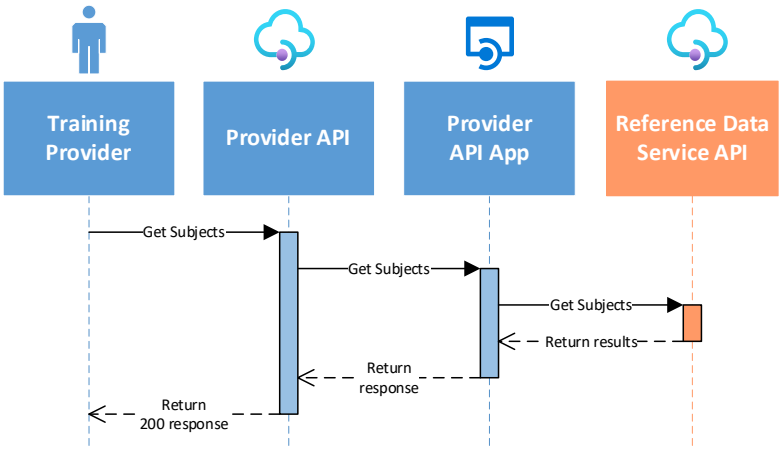


Figure 27272725: Sequence - Get Subjects (F:30)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 524 of 1295

5.5.5.3 Get Programs

This operation retrieves the Programs for a single Training Organisation. The data will be retrieved from the Reference Data Service API, which will manage synchronisation of this data with the Source of Truth.

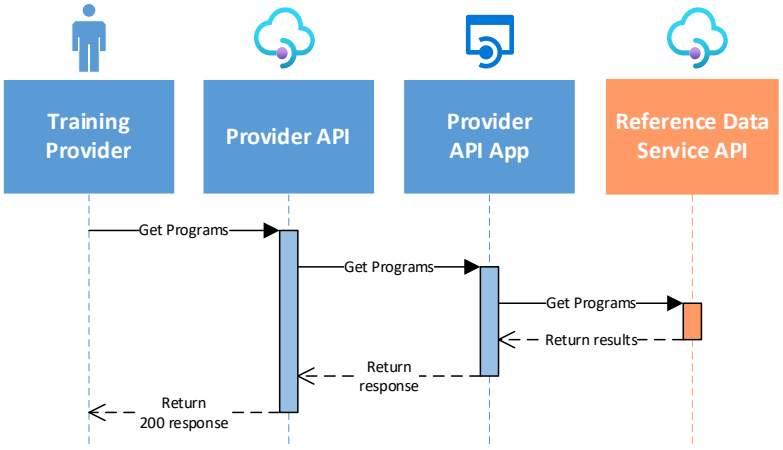


Figure 2828~~2826~~: Sequence - Get Programs (F:31)

5.5.5.4 Get Delivery Locations

This operation retrieves the Delivery Locations for a single Training Organisation. The data will be retrieved from the Reference Data Service API, which will manage synchronisation of this data with the Source of Truth (which may be TGA or an STA).

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 525 of 1295

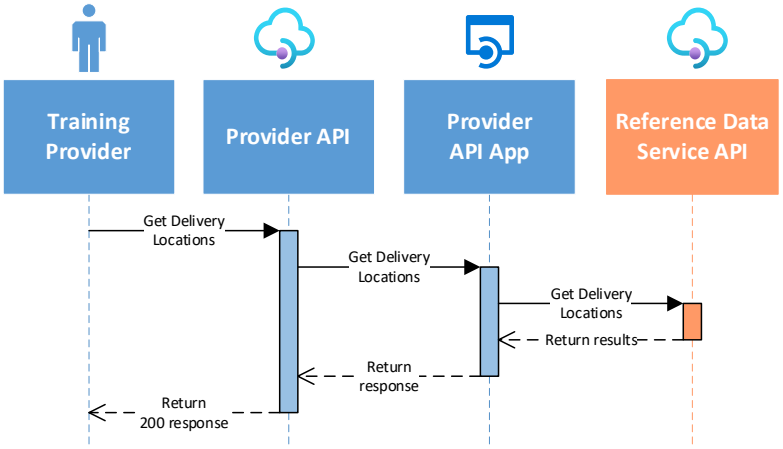


Figure 2929~~2927~~: Sequence - Get Delivery Locations (F:32)

5.5.6 Preference Management

~~5.5.6~~ The following section refers to User preferences, primarily relating to Notifications. The Notification Service Detailed Design Document (G) contains details about the types of preferences that can be updated (e.g., email or text message delivery etc).

Formatted: Normal

5.5.6.1 Get Notification Preferences

User preferences for notifications are stored in the Provider DB. These details can be accessed for display in the Provider Portal or an SMS, or by internal STARS services (like the Notification Service for sending out Notifications). As with all OData endpoints, the request can be filtered. Typical filters expected would be for a single User, or for a collection of Users.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 526 of 1295

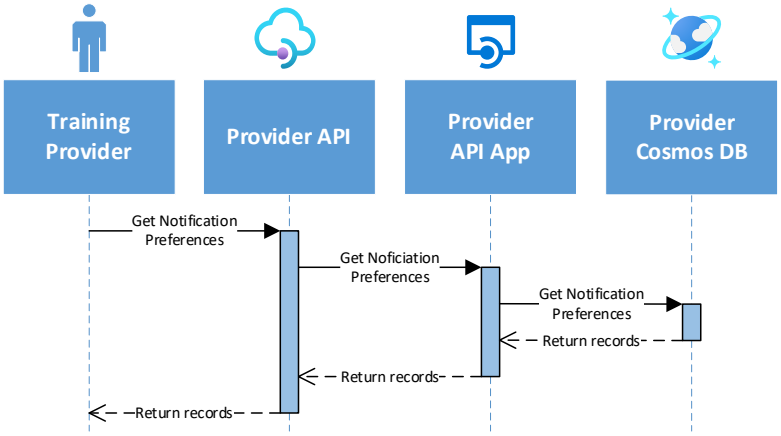


Figure 30303028: Sequence - Get Notification Preferences (F:27)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 527 of 1295

5.5.6.2 Update Notification Preferences

When a User is onboarded, their initial Notification Preferences will be recorded in the Provider DB. These details can then be updated through this operation. The updated values will be validated prior to being stored.

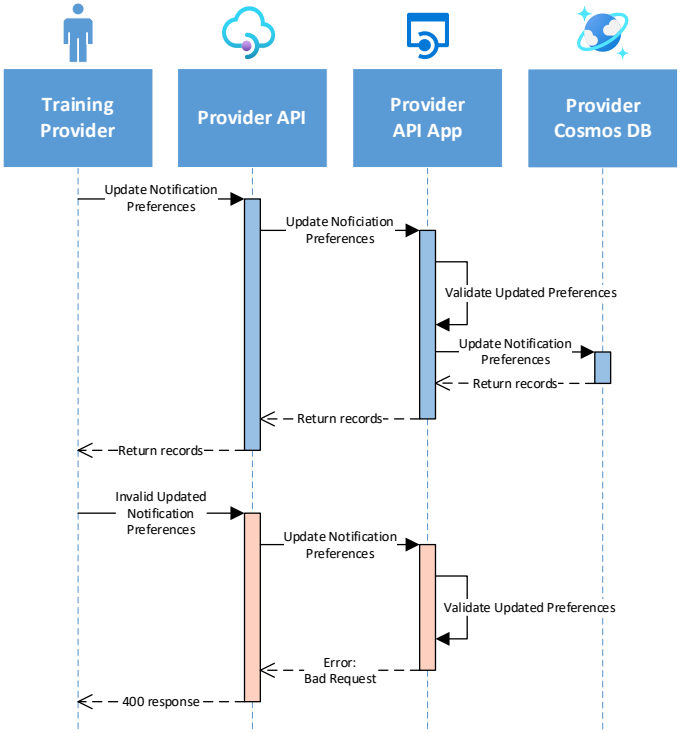


Figure 3131~~3129~~: Sequence - Update Notification Preferences (F:28)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 528 of 1295

5.5.7 User/System Information

5.5.7.1 Get Notifications

This operation will be used to retrieve 'Active' notifications for display in the Provider Portal, or in an SMS. It will also be used for SMS applications to retrieve Notifications that need to be sent out to users of their SMS. OData filters will be used to specify the scope of Notifications to be returned. The response will include details to help the UI decide how the notification is to be displayed (e.g., broadcast style highly visible notifications, news style headlines, or user-specific notifications etc.). The full list of Notification types that can be retrieved can be found in the Notification Service Detailed Design Document (G).

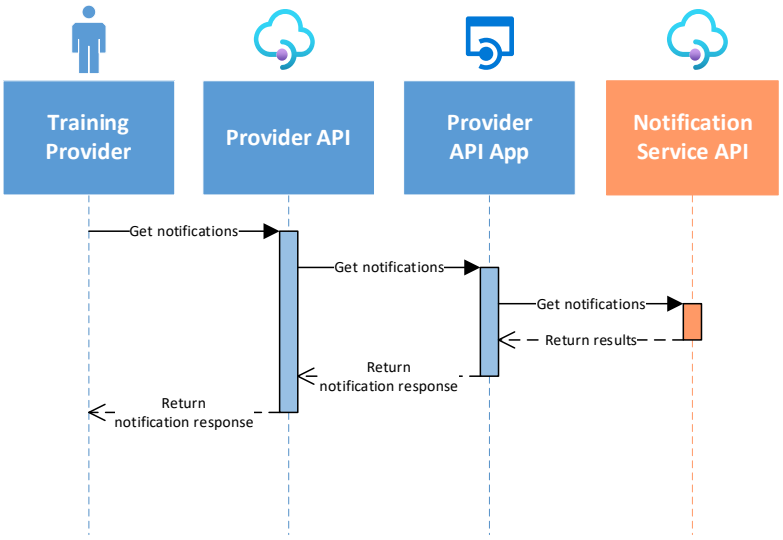


Figure 32323230: Sequence – Get Notifications (F:10)

5.5.7.2 Get Business Glossary

~~5.5.7.2~~ This operation provides the URL to a publicly available Business Glossary. There will be a VET-IS Business Glossary that describes the data elements to be reported to STARS. There will also be a STARS Business Glossary developed as part of the Data Governance Framework, but it is yet to be determined which Glossary will be used here.

Formatted: Normal

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 529 of 1295

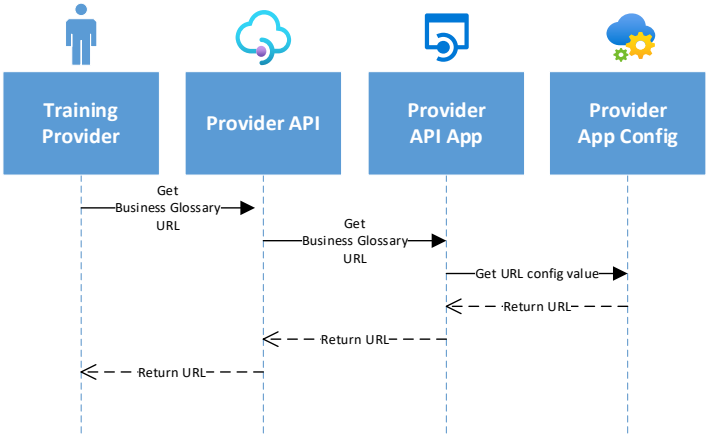


Figure 33333331: Sequence – Get Business Glossary (F:19)

5.5.7.3 Get Form by Record Type

Return the details of a Form associated with a Training Activity Record type. The response will contain all information required to generate a form for direct entry of that type of Training Activity Record. It will include simple validation instructions specific to the published VET-IS, drop-down list options and the like. These validations will not cover all national validation rules, but will provide sanity checking of data types etc. The forms will be specific to 1 or more versions of the VET-IS standard. The standard version can be specified as query parameter or will default to the latest version. These forms can be used to View training activity Records, as well as Create or Update them.

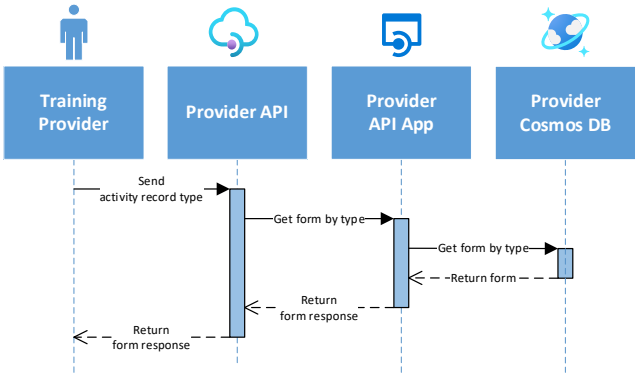


Figure 34343432: Sequence – Get Form by Record Type (F:23)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 530 of 1295

5.6 Provider Service Events

5.6.1 Raised Events

Table 131314: Events raised by the Provider Service

Formatted: Caption, Keep with next

Event	Description
New Submission	Sent to the Event Service to notify that data has been formally Submitted. Expected to be handled by the Submission Service. Properties: • Submission Id (GUID) • Submitter Id (GUID) • Training Organisation Id (GUID) • RecordCount (Positive Integer)
Training Provider Registration	Sent to the Event Service to notify that a new Training Provider is to be Registered to submit training activity data to STARS. Expected to be consumed by the Notification Service to initiate a manual approval process. Properties: TBD
User Registration	Sent to the Event Service to notify that a new Training Provider User is to be Registered for access to the STARS Provider Portal. Expected to be consumed by the Notification Service to initiate a manual approval process. Properties: TBD
Validation Outcome Review Requested	Sent to the Event Service to notify that a review of a particular Validation outcome has been requested. Expected to be handled by the Rules Service. Properties: • Submission Id (GUID) • Submitter Id (GUID) • Training Organisation Id (GUID) • Validation Request Id (GUID) • Validation Result Item Id (GUID)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 531 of 1295

	• Rule Id (GUID) • Justification (Text)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 532 of 1295

5.6.2 Consumed Events

Table 141415: Events consumed by the Provider Service

Formatted: Caption, Keep with next

Event	Description
Validation Complete	Raised by the Validation Service. The Provider Service will listen for these events and will send a "Validation results are available (asynchronous results)" notification to the relevant provider.
Submission Rejected	Raised by Submission Service. The Provider Service will listen for this event and send a "Submission Results available" notification via the Notification Service.
Submission Approved	Raised by Submission Service. The Provider Service will listen for this event and send a "Submission Results available" notification via the Notification Service.

5.6.3 Notifications

Table 151516: Notifications raised by the Provider Service

Formatted: Caption, Keep with next

Notification Type	Description
~~New submission successfully uploaded~~submitted and ~~passes validation~~	~~Sent to STAs when data has been uploaded successfully.~~
Validation results are available (asynchronous results)	Validation Service will raise an event when validation is complete and the results are available. The provider service in turn will send an event to the Notification Service.
Validation results are reviewed and resolved	When training activity data has been updated
~~Submission was rejected (on submission)~~	~~Sent when validation passes but submission fails.~~
~~Submission was approved (on submission)~~	~~Sent when validation passes and submission passes.~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 533 of 1295

5.7 Data Quality Reporting

STAs will require validation metrics/results to be reportable through PowerBI. Reporting metadata may need to be stored as part of the Provider Service. This section will be updated with details as part of the Reporting and Analytics detailed designs.

6 Technical Design Patterns

~~6~~ The design of the Provider Service API follows specific patterns and conforms to the OData specification use for standardising interactions with STARS APIs. Details of the patterns used for the different types of operations in the Provider Service API are documented in the Provider Service API Design document (I).

Formatted: Normal

~~6.1.1 General Endpoint Descriptions~~

~~6.1.1.1 Read Endpoint – List~~

~~A GET endpoint that will return a list of entities with optional filtering. OData will be used for filtering and paginating results. Client-driven paging will be used for pagination.~~

~~6.1.1.1.1 Request Parameters~~

~~The parameters are used to control pagination and to filter the list of results.~~

~~Query parameter~~	~~Comments~~
~~$top~~	~~Limits the number of results returned for this query.~~
~~$skip~~	~~The number of records to "skip over" when retrieving results. Used for pagination. If not included, default to 20.~~
~~$format=json~~	~~How the format of the response data. Should be set to "json".~~
~~$inlinecount=allpages~~	~~Ensures odata.count is included in the response with the total number of relevant results, regardless of pagination. Set to "allpages".~~
~~(filters)~~	~~Varies per endpoint. Values used to filter results to return.~~

~~6.1.1.1.2 Request Body~~

~~None.~~

~~6.1.1.1.3 Response Body~~

~~A JSON response is returned, with describing the number of results and how to retrieve adjacent pagination pages. The page of results is also returned.~~

~~Field~~	~~Comments~~
~~odata.count~~	~~Total number of results (regardless of pagination).~~
~~value~~	~~List of relevant entities. Record structure depends on individual endpoints.~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 534 of 1295

```
{
  "odata.count": 155,
  "value": [
    {
      "recordID": 101,
      "field1": "value"
    }]}
```

6.1.1.2 Read Endpoint – Detail

A GET endpoint for retrieving a specific record.

6.1.1.2.1 Request Parameters

The parameters are used to control pagination and to filter the list of results.

Query parameter	Comments
id	The unique identifier for the desired record.

6.1.1.2.2 Request Body

None.

6.1.1.2.3 Response Body

A JSON object describing the relevant record. The fields will depend on the record relevant to each specific endpoint.

```
{
  "recordID": 101,
  "field1": "value"
}
```

6.1.1.3 Create Endpoint

A POST endpoint for creating a specific record.

6.1.1.3.1 Request Parameters

None.

6.1.1.3.2 Request Body

The request body will be a JSON object that matches part of the relevant record schema, containing the initial object values.

```
{
  "field1": "value",
  "field2": "another value"
}
```

6.1.1.3.3 Response Body

A JSON object describing the newly created record. The fields will depend on the record relevant to each specific endpoint.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 535 of 1295

```
{
  "recordID": 101,
  "field1": "value",
  "field2": "another value"
}
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 536 of 1295

6.1.1.4 Update Endpoint

A POST endpoint for updating a specific record.

6.1.1.4.1 Request Parameters

None.

6.1.1.4.2 Request Body

The request body will be a JSON object that matches part of the relevant record schema, containing the fields that need to be updated. The ID field is also provided to match a single record.

```
{
  "ID": 101,
  "field1": "updated value",
  "field2": "updated value 2"
}
```

6.1.1.4.3 Response Body

202 Accepted response, with a JSON object describing the now-updated record. The fields will depend on the record relevant to each specific endpoint.

```
{
  "recordID": 101,
  "field1": "updated value",
  "field2": "updated value 2"
}
```

6.1.1.5 Delete Endpoint

A DELETE endpoint for deleting a specific record (soft delete). This will ultimately set a "Deleted" flag on the record which will no longer be returned in GET queries.

6.1.1.5.1 Request Parameters

None.

6.1.1.5.2 Request Body

The request body will be a JSON object that matches part of the relevant record schema, containing the record ID for the record to delete.

```
{
  "ID": 101
}
```

6.1.1.5.3 Response Body

A 202 Accepted response, with a JSON object describing the deleted record. The fields will depend on the record relevant to each specific endpoint.

```
{"ID": 101, "field1": "value"...}
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 537 of 1295

7 Data

The core data entities involved with the Provider Service are the Submission Data Packets prepared and submitted by Training Providers, which are based on the VET-IS data standard.

All STARS services will be using the STARS Collection data model, based the VET-IS Collection specification.

The data schema used in the Provider Service API is documented in the Provider Service API Design document (I).

~~7.1 VET-IS Data Packet Schema~~

~~The VET-IS Data Packet Schema forms both a logical information model and the basis for implementing data interchange protocols between STARS and external systems, including Student Management Systems (SMS). The Data Packet Model informs system-to-system data presentation formats and, for example, Application Programming Interfaces (API).~~

~~The model is a de-normalised projection of the VET-IS logical schema into 'Packets' of data elements that represent the common primary conceptual entities found in the VET sector, e.g., Students & Enrolments. Data Packets primary use, within the STARS context, are as Data Transfer Objects (DTO) between external systems (e.g., RTO SMS, STA systems) and STARS.~~

~~In this sense, the Data Packet Schema is a 'projection' of the Logical Schema into commonly recognised topics. For example, a Student data packet is represented as a Stakeholder within the Logical Schema, as is a Training Provider. An Enrolment data packet is represented as a Responsibility.~~

~~Data Packets will also likely be the 'public face' of the VET-IS in terms of describing, reporting and exchanging data with external stakeholders, such as RTOs, SMS providers and STAs.~~

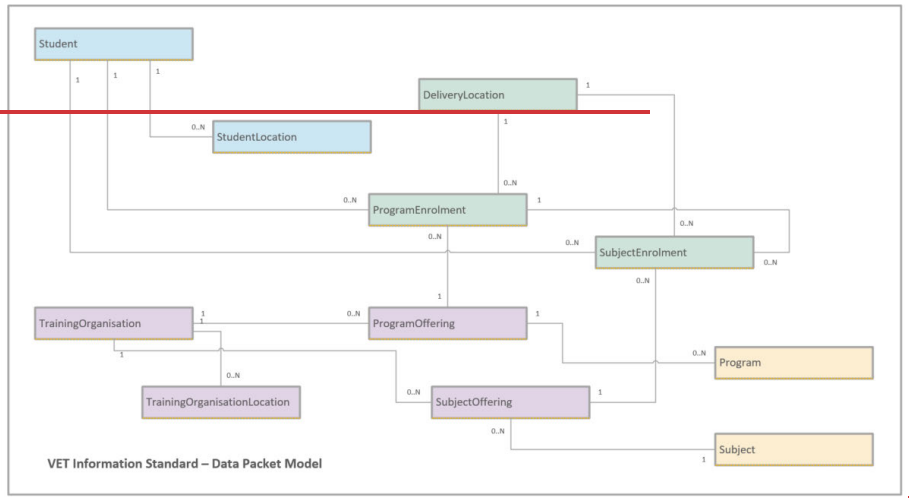
~~The Data Packet types currently defined in VET-IS R1 are:~~

- ~~Government Packet~~
- ~~Training Organisation Packet~~
- ~~Training Organisation Location Packet~~
- ~~Program Packet~~
- ~~Subject Packet~~
- ~~Offering Packet~~
- ~~Delivery Location Packet~~
- ~~Student Packet~~
- ~~Student Location Packet~~
- ~~Program Enrolment Packet~~
- ~~Subject Enrolment Packet~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 538 of 1295

~~VET IS Release 1 is targeted to specifying and shaping the information and data required for the VET Provider Collection and thus there is not currently support for the VET Apprentice and Trainee collection (including Employers).~~

~~The following diagram shows the high-level relationships between the Data Packet types.~~



~~**Figure 33: VET-IS Data Packet Schema**~~

~~By way of terminology, Training Activity Records referred to in this document are a collection of VET-IS Data Packets. Records (also called Packages) only contain packets that relate to each other and contain a complete set of information. E.g., a package containing an Enrolment packet must also contain the details of the student enrolled in a Student packet, along with Delivery Location, Subject, Offering and Training Organisation packets.~~

~~**This model is common across many STARS services, and will be defined in a separate document, to be written, that will form the basis for the OData model used by the Provider Service API, and other STARS APIs that communicate Training Activity data.**~~

Formatted: Caption, Centered

~~7.2~~7.1 Data Flow

The following Data Flow Diagram covers the high-level processes/features of the Provider Services. Note that 'Micro-Training Providers' are specifically training providers that don't use a Student Management System. 'Training Providers' refers to all Training Providers, regardless of whether they use an SMS or not, or through STA systems.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 539 of 1295

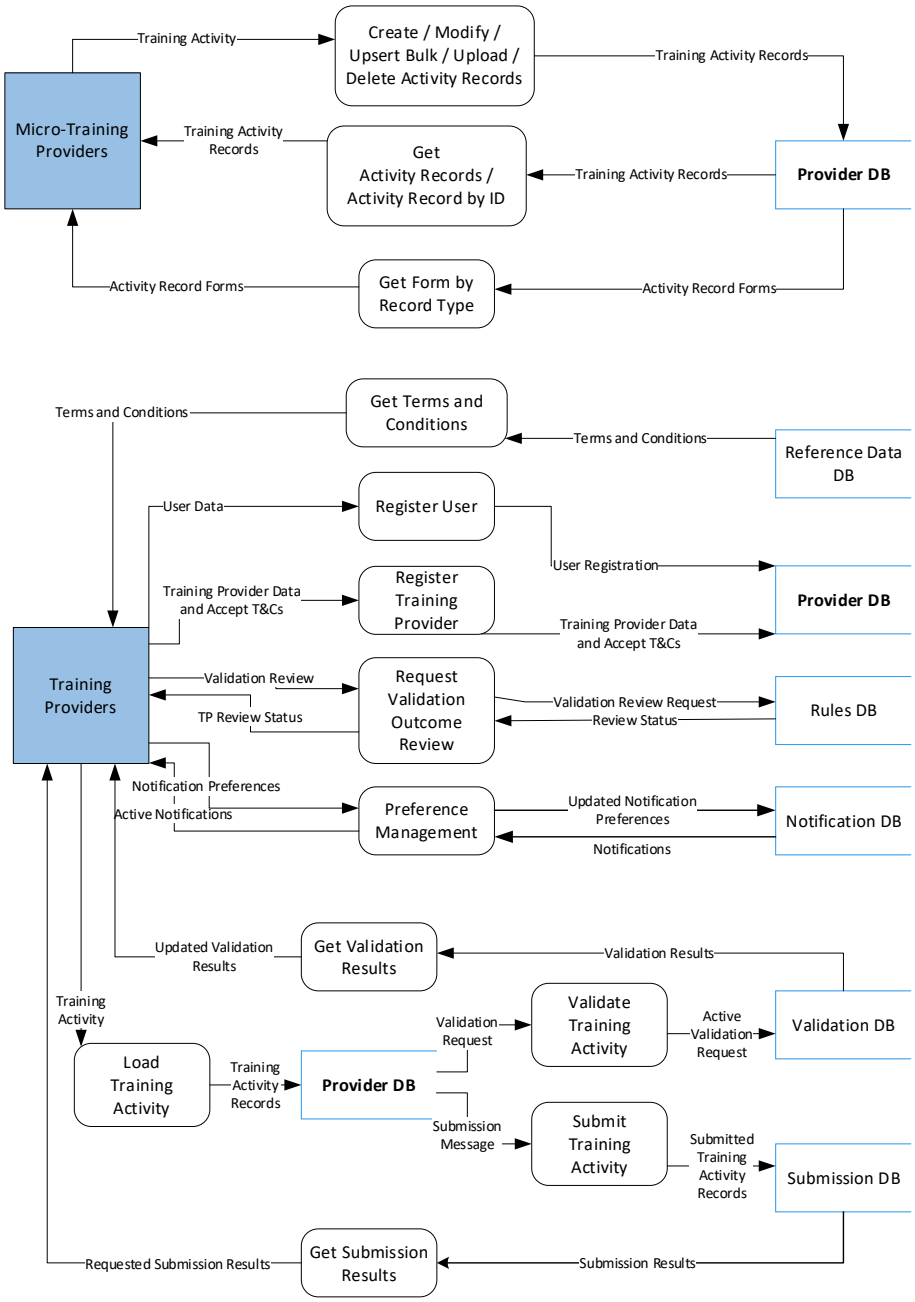


Figure 35353734: Provider Service Data Flow

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 540 of 1295

8 Other Service Dependencies

8.1 Validation Service

The Validation Service API will be used for handling validation requests.

8.2 Submission Service

The Submission Service API will be used for Submitting training activity and retrieving details of Training Activity that has been submitted.

8.3 Notification Service

The Notification Service API will be used when sending notifications.

8.4 Rules Service

The Rules Service API will be used as part of the validation request process, specifically, to run Business Rules to determine the funding relationships for submitted records.

8.5 Reference Data Service

The Reference Data Service API will be used retrieve configurable static content, and VET-IS classification data for pre-population of forms and for Training Provider Data management.

9 Dependent Services

9.1 Provider Portal

The Provider Portal will depend on this service to provide API endpoints for all functionality and static content. This will include retrieving Angular dynamic forms.

9.2 Provider Management Portal

The Provider Management Portal will depend on this service to provide API endpoints for Training Provider management.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 541 of 1295

10 Access Management

The Provider Service will conform to the Data Governance Access and Policy Protocol documents and IDAM detailed design. Access to the Provider Service will be controlled by Azure RBAC, with all calling services identified and granted access through Azure AD Managed Identifies and B2C. For detail about access the Provider Services, please refer to the IDAM detailed design document.

10.1 Control of Data

The Governance Framework sets out the key data governance roles surrounding VET Data Streamlining. The two core central roles for accessing data from STARS are that of the Data Owner and the Data Custodian, summarised below:

10.1.1 Data Owner

Data Owner has authority for establishing rules surrounding the quality, protection, and appropriate access to data. They individually and collectively have a responsibility to protect the personal and sensitive information provided by Data Providers. Data can be jointly or co-owned.

10.1.2 Data Custodian

Data Custodians receive authorised data from Data Providers or from other Data Custodians. Data Custodians are authorised to collect and handle that data to perform their functions. They are responsible for ensuring that the personal information collected and stored is reasonably necessary for them to conduct their functions and activities in relation to the VET sector.

For detail about Data access and control please refer to Data Governance Access and Privacy protocol document.

10.2 Security

STARS will be built to meet existing Australian Government policies, frameworks, standards and design principles, including:

- Australian Government API Design Standards
- Digital Service Standard
- Protective Security Policy Framework (PSPF)
- Information Security Registered Assessors Program (IRAP)
- Australian Government Information Security Manual (ISM)
- the Open Web Application Security Project® (OWASP)
- Hosting Certification Framework.

All data within STARS will be handled in accordance with relevant privacy laws and record keeping requirements, including the Privacy Act, the *Archives Act 1983* (Cth) and any other applicable legislation (e.g., applicable State and Territory privacy legislation or archiving requirements).

For detail about security please refer to the following

- Data Governance Access and Privacy Control document

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 542 of 1295

- IDAM detailed design document

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 543 of 1295

11 Requirements

The following section lists the relevant Functional and Non-functional requirements relevant to this service, with traceability to the individual sections of the document that speak to each requirement. Some of these requirements are "owned" by this service, and others are owned by different services but are still relevant for referencing. "Owned" requirements are highlighted in green.

11.1 Functional Requirements

Table 161620: Function Requirements relevant to the Provider Service

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR002	(Provider Portal) Data submission formats through user interface	See 5.5.2 Training Activity Data PreparationTraining Activity Data Preparation	The VDS system must allow a data to be imported in the following formats: • CSV • Excel • Text • JSON • Other dataset formats as determined such as XML	Given Training Organisation has submittable batch Training Activity When they upload a file (or files) in the agreed data format Then STARS will accept the file and record the data.
FR041	(Provider Portal) Pre-population of data	See 5.5.2 Training Activity Data PreparationTraining Activity Data Preparation and 5.5.4 Manage Submissions and Validation Results	The VDS system must provide a mechanism to enable the pre-population of student information and training activity data from data already held in the master database. • This will require the VDS system to seek and record permission/consent from the student authorising pre-population	Given data previously submitted by a training organisation When that training organisation is entering data Then STARS will pre-populate form data as per agreed designs Given data previously submitted by a training organisation When SMS's request data for pre-population Then STARS will provide the training organisation's data as per agreed designs

Formatted: Caption, Keep with next

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR074	(Provider Portal) Data integrity adherence	See 5.5.4 Manage Submissions and Validation Results~~Manage Submissions and Validation Results~~	The system must be flexible enough to identify incorrect data upfront and also have the functionaility to accept 'exceptional' data. For example: • Numerical value in a name • Name is NULL • Students are twins with same last name, DOB, address and other personal details	Given training activity data is submitted for validation And the data fails validation rules When an exemption is requested by the provider Then STARS will record justification for the exemption And seek approval of the exemption through a defined workflow Given training activity data is submitted for validation And the data fails validation rules When an exemption is approved through the defined walkthrough Then STARS will bypass the specific validation rule(s) included in the exemption

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR079	(Provider Portal) View Student and Enrollment Info	See 5.5.2 Training Activity Data Preparation~~Training Activity Data Preparation~~	As a TO User using data entry functionality, I want to be able to view the student information of a particular student associated with my organisation so that I can view/maintain the student details	Given a Training Organisation user And student data previously provided by the training organisation When a Training Organisation searches a student related to my organisation by fields specified as search criteria in the student search design Then STARS will present a list of potential matches to view Given search results for a student When a result is selected Then STARS will display the details of the student Given a detailed view of a student's details When enrolment details are requested Then STARS will display the current enrolment of the student And STARS will display past enrolments of the student with my training organisation, along with outcomes And the list of enrolments will be filterable

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR080	(Provider Portal) Update of student info	See 5.5.2 Training Activity Data Preparation~~Training Activity Data Preparation~~	As a TO User using data entry functionality, I want to be able to update the student information of a particular student associated with my organisation so that I can update the student details	Given a Training Organisation user And student data previously provided by the training organisation When a Training Organisation searches a student related to my organisation by fields specified as search criteria in the student search design Then STARS will present a list of potential matches to view Given search results for a student When a result is selected Then STARS will display the details of the student Given a detailed view of a student's details When enrolment details are requested Then STARS will display the current enrolment of the student And STARS will display past enrolments of the student with my training organisation, along with outcomes And the list of enrolments will be filterable Given a detailed view of a student record When the student is selected for update Then STARS will display the current enrolment of the student And STARS will display past enrolments of the student with my training organisation, along with outcomes And the list of enrolments will be filterable Given search results for a student When the student is selected for update Then STARS will pre-populate the student data in data entry form(s) And allow submission of the updated student details Given a detailed view of a student record

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
				When the student is selected for update Then STARS will pre-populate the student data in data entry form(s) And allow submission of the updated student details

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR081	(Provider Portal) Update enrollment	See 5.5.2 Training Activity Data Preparation~~Training Activity Data Preparation~~	As a TO User using data entry functionality, I want to be able to update the student enrollment information of a particular student associated with my organisation so that I can update the student details	Given previously submitted training activity When a training organisation searches for students by specific fields agreed in design Then STARS will return matching results from previously entered student data And the results may be filtered by specific fields as per design And a result may be selected to be updated Given previously submitted student data And a student record is being viewed When enrolments for the student are viewed Then the enrolments may be filtered by specific fields as per design And an enrolment may be selected to be updated Given previously submitted enrolment data And an enrolment has been selected to be updated When the updated enrolment details are saved Then STARS will persist the changes And make them available to be loaded at a later time Given a saved Enrolment update When the Enrolment update is loaded Then the saved details will be populated to the Edit Enrolment form(s) And the details may be updated further And the Enrolment changes may be submitted

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR082	(Provider Portal) Bulk Enrollment	See 5.5.2 Training Activity Data Preparation~~Training Activity Data Preparation~~	As a TO User using data entry functionality, I want to be able to batch enroll a list of students to selected program/subjects so that I do not need to enroll them one by one	Given a list of students When selecting to enrol all students in the list Then STARS will prepare data for submission of enrolments for all students And allow entry of enrolment information consistent with regular data entry forms And allow the bulk enrolment to be saved for later submission And details of students will not be stored with the saved bulk enrolment data Given a bulk enrolment When the bulk enrolment is submitted Then details of included students will be retrieved to include in the submission And enrolment packets for all students will be submitted

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR083	(Provider Portal) Bulk Update of Enrollment	See 5.5.2 Training Activity Data Preparation~~Training Activity Data Preparation~~	As a TO User using data entry functionality, I want to be able to perform batch update for students involved in a particular training so that I do not need to update for each student one by one.	Given previously submitted enrolments When viewing a Program Then STARS will show all Students enrolled in the program Given previously submitted enrolments When creating a Bulk Edit of students related to a program Then STARS will allow outcome, start date and end date (and other specific fields identified during design) to be edited for all students enrolled in the program And the bulk edit may be saved for later submission Given previously submitted enrolments When viewing a Subject Then STARS will show all Students enrolled in the subject Given previously submitted enrolments When creating a Bulk Edit of students related to a subject Then STARS will allow outcome, start date and end date (and other specific fields identified during design) to be edited for all students enrolled in the subject And the bulk edit may be saved for later submission Given a bulk student enrolment When the bulk edit is submitted Then STARS will prepare data packets for all students included in the bulk edit And other student details required for the packets, but not editable in a bulk edit will be retrieved and included in the submitted packets And STARS will process the Bulk Edit submission

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR085	FR085 - Capturing whether UAN has been performed and date of assessment.	See 5.5.2 Training Activity Data Preparation~~Training Activity Data Preparation~~	As a TO User providing community training (ACE Program), I want to be able to record whether I have completed the Upfront Assessment of Need (UAN) process and date of UAN so that I can submit this info to STA and get paid.	Given Training Organisation with community training (ACE Programs) When entering the training activity Then STARS will accept all state specific fields included in the accepted version(s) of the VET Information Standard
FR086	FR086 - Capturing Pathway Outcome(whether the training has met the training objective (UAN))	See 5.5.2 Training Activity Data Preparation~~Training Activity Data Preparation~~	As a TO User providing community training (ACE Program), I want to be able to record the pathway outcome in terms of how/whether the training objective has been met.	Given Training Organisation with community training (ACE Programs) When entering the training activity Then STARS will accept all state specific fields included in the accepted version(s) of the VET Information Standard
FR096	(Provider Portal) Management of Training Organisation Delivery Location via UI for non-Registered Training Org without SMS	See 5.5.3 Training Activity Data Submission~~Training Activity Data Submission~~	As a non-Registered Training Organisation using data entry functionality, I want to be able to maintain the training delivery location of my organisation so that I can use them for data submission purpose.	Given a non-Registered Training Organisation When delivery location data is submitted through VET Information Standard packets Then STARS will validate and record the submitted data And progress it through the submission and mastering process
FR097	(Provider Portal) Management of Training Organisation Delivery Location for Registered Training Org without SMS	See 5.5.3 Training Activity Data Submission~~Training Activity Data Submission~~	As a Registered Training Organisation using data entry functionality, I want to be able to maintain the training delivery location of my organisation and keep source of truth in one system so that I do not need to manually maintain/update in multiple system/government agencies (e.g ASQA/TGA).	Given a Registered Training Organisation When delivery location data is submitted through VET Information Standard packets Then STARS will validate and record the submitted data And progress it through the submission and mastering process And STARS will send the delivery location to external systems that require delivery location data as per integration design

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR098	(related) Management of Training Organisation Delivery Location for Training Org using SMS	See 5.5.3 Training Activity Data Submission~~Training Activity Data Submission~~	As a Training Org using SMS, I want to be able to maintain my training delivery location in SMS and the information will be submitted/synced to VDS system so that I do not need to maintain the information manually in multiple systems.	Given a Training Organisation When delivery location data is submitted through VET Information Standard packets via API Then STARS will validate and record the submitted data And progress it through the submission and mastering process And STARS will send the delivery location to external systems that require delivery location data as per integration design
FR111	(Provider Portal) Error and Warning Report for Training Org	See 5.5.4 Manage Submissions and Validation Results~~Manage Submissions and Validation Results~~	As a TO user, I want to be able to view the list of unresolved errors and warnings related to my organisation so that I am aware of the data quality and can follow up from there.	Given submitted training activity data And the data has failed validation rules And the errors have not be resolved When a training organisation user views a list of unresolved errors and warnings Then STARS will show the list of unresolved errors and warnings for their training organisation And show details of the errors and warnings And allow the list to be filtered based on training organisation, submission date/time, error/warning reason, student id, student name or USI And allow the record(s) to be edited if the user has permission to do so
FR133	(Provider Portal) Activity Data Upload Capability during Transition Period	See 5.5.3 Training Activity Data Submission~~Training Activity Data Submission~~	While Activity Data Files are being uploaded or processed via UI, system will not allow relevant training organisation's to re-upload files	Given Training Organisation has submitted training activity data When that submission has not yet completed processing and validation Then STARS will not allow the Training Organisation to upload or resubmit any training activity

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR006	(related) National rules error notification	See 5.5.3 Training Activity Data Submission~~Training Activity Data Submission~~	The VDS system must notify a training organisation of the VDS National/Reference Data validation result (errors/warnings) in near real-time via: • Student Management System • Adaptive Web User Interface • Mobile App [Mobile App out of scope]	Given Training Activity submission When a Training Organisation submits data via the SMS API Then the submission is validated against the relevant internal validation rules And the response via the API will include all errors, warnings and notifications relevant to the data validation result Given Training Activity submission When a Training Organisation submits data via the Web Interface Then the submission is validated against the relevant internal validation rules And the response in the Web Interface will include all errors, warnings and messages relevant to the data validation result
FR008c	(Provider Portal) STA/Funder Validation	See 5.5.3 Training Activity Data Submission~~Training Activity Data Submission~~	The VDS system must validate the data submitted by the training organisation against the STA/Funder validation rules. • training organisations can see which STA(s) is validating their data. The reason for this is transparency.	Given a submission is being validated by an STA When the providing Training Organisation queries/views the submission Then they can identify which STA(s) is performing the data validation activities Given a submission has been validated by an STA When the providing Training Organisation queries/views the submission Then they can identify which STA(s) performed the data validation activities

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
				Given a submission that has failed validation When the provider queries STARS for validation results Then the provider may include criteria to filter the results And STARS will provide the validation results in a structured format that supports filtering and review initiation
FR089	(Provider Portal) Exceptional Data Handling (Training Org request for review)	See 5.5.4 Manage Submissions and Validation Results~~Manage Submissions and Validation Results~~	As a TO/STA user, I want to be able to request for review of the data submission violation so that my data can be accepted in exceptional scenario.	Given a submission that has failed validation And the validation results have been reported to the submission And the validation error(s) are able to be reviewed When the provider requests a review of the one or more errors Then STARS will initiate a process to review the validation rule that failed, along with any reference data that the rule depends on And a justification for the review will be recorded

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR122	(Provider Portal) Validation Results	See 5.5.4 Manage Submissions and Validation Results~~Manage Submissions and Validation Results~~	In continuation to TSNSW FR13 and FR045, as a training organisation, I would like to also have access to the errors & warnings generated from the most recent VET activity data submission stored and accessible when I relogin or if the session times out so that I can have access to the history of my submission status and be able to perform error corrections on the screen for the data and be able to resubmit the specific student activity record . Applicable only until the next subsequent submission Note: FR050 and FR45 looks similar and needs to be reviewed	Given training activity submission and validation error and warning When the Training Organisation user logins into STARS Then the user can access the error and warning messages for actioning and resubmission

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR123	(Provider Portal) Validation Results	See 5.5.4 Manage Submissions and Validation Results~~Manage Submissions and Validation Results~~	The error/warning/info messages must be granular and quote the data/reference item which is under error in reference to notification requirements in FR056b, FRo56c Note: For S&S program specfic VET student activity data the following data elements should not be exposed on the screen: • Payment Stage • Reject Id • Reject Reason • Addendum: o Reject Reason o Course of Action Reject ID will be concatenated at the end of the text message	Given training activity submission and validation error and warning When the Training Organisation user views the errors and warnings Then the following data elements will NOT be displayed in STARS: • Payment Stage • Reject Id • Reject Reason • Addendum: o Reject Reason o Course of Action

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR001a	Data submission via API	See 5.5.3 Training Activity Data Submission~~Training Activity Data Submission~~	A training organisation must be able to submit/resubmit a dataset in real-time to the VDS system via a Student Management System (SMS). SMS and training organisations should be able to setup API unaided based on the API interface design (set out in Annexure B) documentation and the training provided by the seller.	TC Verify authorised user data submission TC (-) Verify unauthorised user data submission fails Given Authorised Access When VET Provider submits training activity Then Provider will be given successful notification of submission TC Verify submission of authorised training activity Given Authorised Access AND a previous submission When VET Provider re-submits an updated submission Then Provider will be given successful notification of re-submission AND the previously submitted data will be updated in VDS (STARS) TC Verify successful re-submission of authorised training activity TC (-ive) Verify un-successful re-submission of un-authorised training activity Given Authorised Access to APIM Developer Portal AND access to API interface design documentation AND completed training activities When an SMS sets up interaction with the APIs Then no additional support is required Given Authorised Access to APIM Developer Portal AND access to API interface design documentation AND completed training activities When a training organisation sets up interaction with the APIs Then no additional support is required

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR001b	(Provider Portal) Data entry via user interface	See 5.5.3 Training Activity Data Submission~~Training Activity Data Submission~~	A training organisation must be able to submit/resubmit a dataset in real-time to the VDS system via an adaptive Web User Interface, including options to upload (per FR2) or to directly enter data.	Given Training Organisation has logged into the Web User Interface When the Training Organisation enters Training Activity Then STARS records the training activity And STARS allows the Training organisation to Submit the data they have entered Given Training Organisation has logged into the Web User Interface And a previous submission of training activity When the Training Organisation resubmits the Training Activity Then STARS records the updated training activity Given Training Organisation has logged into the Web User Interface When they upload data into STARS Then STARS records the training activity And allows the training organisation to Submit the data they have uploaded
FR011	(Provider Portal) Submit data	See 5.5.6 Preference Management~~Preference Management~~	A training organisation must authorise the submission of data into the VDS system. • If the time limit for submission has been reached, the VDS system will notify the Provider if data has completed a validation check, but has not been authorised for submission.	Given Training Organisation has entered or uploaded data into STARS And has not submitted that data When the time limit for a submission has been reached Then STARS will notify the Training Organisation

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR078	(Provider Portal) Search Student	See 5.5.2 Training Activity Data Preparation~~Training Activity Data Preparation~~	As a TO User using data entry functionality, I want to be able to search for a student that is enrolled (or has previously enrolled) with my organisation so that I can view/maintain the student details	Given a Training Organisation user And student data previously provided by the training organisation When a Training Organisation searches a student related to my organisation by fields specified as search criteria in the student search design Then STARS will present a list of potential matches to view Given search results for a student When the student is selected for update Then STARS will pre-populate the student data in data entry form(s) Given search results for a student When a result is selected Then STARS will display the details of the student
FR110	(Provider Portal) Data Submitted for Training Org	See 5.5.3 Training Activity Data Submission~~Training Activity Data Submission~~	As a TO user, I want to be able to view the data submitted by my organisation so that I am aware of what data has been submitted and the validation status.	Given submitted training activity data And the data has failed validation rules And the errors have not be resolved When a training organisation user views a list of unresolved errors and warnings Then STARS will show the list of unresolved errors and warnings for their training organisation And show details of the errors and warnings And allow the list to be filtered based on training organisation, submission date/time, error/warning reason, student id, student name or USI And allow the record(s) to be edited if the user has permission to do so

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR112	(Provider Portal) Training Organisation Delivery Location for training org	See 5.5.2 Training Activity Data Preparation~~Training Activity Data Preparation~~	As a TO user, I want to be able to view the list of training delivery locations related to my organisation so that I can use them for data submission purpose.	Given training delivery location data in STARS When a training organisation user is submitting data Then STARS will provide a way to view a list of training delivery locations for their training organisation And allow the list to be filtered based on active/inactive flag, start date or end date.
FR127	BI Reporting on Student Details	See 5.5.2 Training Activity Data Preparation~~Training Activity Data Preparation~~	The SMS or UI must be able to search for students saved/submitted in activity file submissions by the training organisation that submitted the data only. The information displayed reflects current most recently updated/submitted student details.	Given student details previously submitted by a training organisation When the training orgnisation searches for students Then STARS will return the most recent matching student data that they had previously submitted
FR094	(related) Data of Below the Line Fields	See 5.5.2 Training Activity Data Preparation~~Training Activity Data Preparation~~	As a STA user, I need the data in current below the line fields to be migrated to VDS system (Currently some of the below the line fields do not appear in NCVER data though we report them) and we need to evaluate how those info would be supported during data migration	Given previous training activity When the training activity is provided to STARS in accordance with the VET Information Standard via data entry portal, API or File Upload Then STARS will accept the provided data Given previous training activity When the training activity data is migrated to STARS Then state-specific fields will be included

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR033	(Provider Portal) T&C	See 5.5.1 User and Training Provider Access~~User and Training Provider Access~~	The VDS system must send the current Terms and Conditions of using the VDS system to a user.	Given users authority to access STARS When users access STARS for the first time Then users are prompted to read the Terms and Conditions of use Given users authority to access STARS When Terms and Conditions are updated Then users are prompted to read the Terms and Conditions of use
FR034	(Provider Portal) Accept T&C	See 5.5.1 User and Training Provider Access~~User and Training Provider Access~~	The VDS system must require a user to accept the VDS system's Terms and Conditions prior to submitting data to the VDS system.	Given users authority to access STARS When users are presented with Terms and Conditions Then the user can accept the Terms and Conditions Given users authority to access STARS And accepted Terms and Conditions When users submit data to STARS Then STARS will accept the data Given users authority to access STARS And Terms and Conditions have not been accepted When users submit data to STARS Then STARS will not accept the data And present the Terms and Conditions for acceptance

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR035	(Provider Portal) Acknowledgement of T&C acceptance	See 5.5.1 User and Training Provider Access~~User and Training Provider Access~~	The VDS system must require a user has accepted the VDS system's Terms and Conditions prior to accepting data to the VDS system.	Given users authority to access STARS And accepted Terms and Conditions When users enter or upload data to STARS Then STARS will accept the data Given users authority to access STARS And Terms and Conditions have not been accepted When users enter or upload data to STARS Then STARS will not accept the data And present the Terms and Conditions for acceptance
FR128	(Provider Portal) Near Real - Time error Notification	See 5.5.3 Training Activity Data Submission~~Training Activity Data Submission~~	As a training organisation, I should have near realtime capabilitiy to view Errors and Warnings online especially when sanity checks stated in FR52 has failed; once the sanity checks are compelted and if no errors, system should notify the user with the message - Thank You for submitting your training activity data, it may taken between X to Y minutes for the files to be processed. with an acknolwedgement button.	Given a Training Organisation data When the data is submitted Then STARS will perform all relevant validations And respond to the training organisation with all validations that were able to be run synchronously And record all validation results for later access And notify the provider of any validations that could not be run synchronously And provide status information to the provider regarding progress of the submission
FR132	(Provider Portal) Activity Data Upload Capability during Transition Period	See 5.5.2 Training Activity Data Preparation~~Training Activity Data Preparation~~	System should be able to configure limits of Activity Data upload file sizes to support the peformance needs for real time processing	Given configured maximum file size for uploads to STARS When the system operator modifies the maximum file size for uploads to STARS Then STARS will only accept files within the configured maximum file size

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR136	(related) Notifications	See 5.5.1 User and Training Provider Access~~User and Training Provider Access~~	Training organisation must have the capability (perhaps via GUI interface) to view up-to-date business glossary to consume easily and to understand the meaning of the data elements associated wtih data packets in the new VET information standard	Given a business glossary developed by NCVER When the glossary updated by NCVER Then STARS will update the business glossary And make the updated glossary available to users
FR068	(related) Data Submission - USI dataset	See 5.5.3 Training Activity Data Submission~~Training Activity Data Submission~~	The VDS system must provide regulators the ability to submit minimum datasets to support Student Transcripts for students/former students with a USI number	Given training activity And the training provider is not able to submit the regular training activity data When regulators submit a minumum dataset Then STARS will record the data on behalf of the training provider And only validations for minimum datasets will be triggered
FR125	(related) TSNSW - S&S Program Specific Validations	See 5.5.3 Training Activity Data Submission~~Training Activity Data Submission~~	Once a student training activity data record is successfully validated, VDS system should be able to flag the latest valid activity information received for a student for use by TSNSWs milestone payment reporting.	Given a Training Organisation submits training activity When it is successfully validated Then the latest valid activity data for an individual student can be identified
FR113	(Provider Portal) User Sign In	See 5.5.1 User and Training Provider Access~~User and Training Provider Access~~	As a System User, I want to be able to sign in to the system so that I can perform the actions required.	Given a System User in STARS And role(s) have been assigned to the user When logging in with valid user credentials Then the user is given access to STARS according to their assigned role(s) Given a System User in STARS And roles have been assigned to the user When logging in with invalid user credentials Then user will be denied access to STARS

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR114	(Provider Portal) User Sign Out	See 5.5.1 User and Training Provider Access~~User and Training Provider Access~~	As a System User, I want to be able to sign out from the system so that I do not remain logged on unnecessarily	Given a System User in STARS And the user has signed in When the user signs out Then all access to STARS will be denied until the user signs in again And any active sessions will be invalidated

Formatted: Font: (Default) Calibri, 10 pt, Font color: Black

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

11.2 Non-functional Requirements

~~11.2~~ Table 171~~21~~: Non-functional Requirements related to the Provider Service

NFR No.	Requirement Name	Description	Comment/Traceability
NFR001a	Use Case scalability	The VDS System will be able to scale elastically in response to demand and remain functional and available to users regardless of dynamic changes in volume, processes, and number of users leveraging the VDS System. • The VDS System will be designed to achieve scalability of additional use cases as needed • The VDS System will support both horizontal and vertical scaling	Each of the Azure components used in this service will be configured to scale dynamically. See 5.1 Provider Service Config~~Provider Service Config~~
NFR001b	Concurrent users scalability	• The VDS System will be designed, developed, tested and deployed to accommodate increased concurrent users, volumes and workloads specifically: • Real time scalability to accommodate increases in the number of concurrent users. • Consistent response times (e.g. user interface, data submission, application workflow, and data consumption) with increased volumes and concurrent users.	Each of the Azure components used in this service will be configured to scale dynamically, See 5.1 Provider Service Config~~Provider Service Config~~. In addition, the design of the operations, with asynchronous processing in mind, allows initial responses to be returned quickly, with subsequent requests to retrieve additional details.
NFR001c	Baseline user volumes	The VDS System supports the following baseline volumes: • Over **2,000** concurrent users across training organisations and STAs • **5,000** total users across training organisations and STAs	The Provider Service will ensure App Service configurations for appropriate scaling in Production, for standard and peak usage patterns. This scaling configuration will be adjusted over time as Training Providers on board to STARS.
NFR001d	Baseline transaction volumes	The VDS System supports the following baseline volumes: • Over **xxx** transactions for the submission of data by training organisations per week • Over **xxx** data submissions per second	Please see outstanding action item 1977 (Azure Infrastructure) Submission activity estimates

Formatted: Caption

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

NFR No.	Requirement Name	Description	Comment/Traceability
NFR002a	Minimum Uptime	The VDS System will support continual operations by being available and functioning with minimal downtime (High Availability). • The VDS System will support a minimum proposed uptime for the users, excluding scheduled downtime or outage	Each of the Azure components used in this service will be configured for high availability in Production. See 5.1 Provider Service Config~~Provider Service Config~~
NFR003a	User response time	The VDS System will provide adequate response times for all users. The VDS System will be designed, developed and deployed in such a manner that response times are achieved at all times, including during peak periods of user activity. • The VDS System will provide a response time of 1 second to meet the standard response times expected by a user.	The Provider Service API will use asynchronous communication patterns to provide a response within 1 second, where the overall processing time may be greater than 1 second. Things like uploading large files may be an exception where it's not possible to respond within 1 second.
NFR003b	Monitor response time	The VDS System will have the ability to monitor and measure the response times it provides to its users.	Standard STARS Auditing and Logging practices will allow traceability of response times.
NFR005a	Digital Identity	The VDS System will fully support Digital Identity in accordance with the Trusted Digital Identity Framework (TDIF). •The VDS System will utilise the myGovID machine to machine authentication model.	See 5.5.1 User and Training Provider Access~~User and Training Provider Access~~
NFR006b	Consistent experience	The experience for all users of the VDS System will be respectively consistent and integrated across all supported channels.	The fundamental design of the Provider Service is to provide a single interface for all Provider interactions.
NFR006d	APIs are AVETMIS Standard agnostic	The VDS System's APIs are agnostic to which version of AVETMISS is used	The API Versioning for the Provider Service will be independent of the VET-IS version associated with the data payload.

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

NFR No.	Requirement Name	Description	Comment/Traceability
NFR008c	Future security policies	The VDS System will comply with an architecture that can adapt to future security policies, standards or frameworks that are enacted during the term of any Contract.	Deployment of the Provider Service resources will be defined in bicep templates. These base templates will be shared across services, and updates to configuration related to security policies can be made there and redeployed into each environment.
NFR009a	Risk and Threat model	The VDS System will support security controls and associated security architecture driven by sound risk assessments that reflect the value and sensitivity of the digital assets and data that are being protected, as well as the threats to the VDS System. The VDS System will be designed, delivered and supported using risk and threat model driven design and in accordance with the Australian Government security requirements (including the PSPF and ISM).	Deployment of the Provider Service resources will be defined in bicep templates. These base templates will be shared across services and contain the appropriate configurations to meet the relevant ISM controls for this service.
NFR009d	Strong accountability	• The VDS System will support and enable strong accountability measures for the use of and access to information and resources. • The VDS System will be designed with mechanisms (audit, access logging) which enable Users to be held accountable for their actions.	The Provider Service will comply with the measures in place for STARS Auditing and Logging.
NFR010a	Role based access controls	Access to the VDS System will be restricted to authorised users to safeguard against deliberate, intrusive and/or unauthorised access from internal and external sources. • The VDS System will support role-based security access controls to ensure that only those with appropriate authority can access a particular user interface. • The VDS System will timeout after 30 minutes of inactivity.	Role-based access control will be applied to the API operations described in this document, as well as to the management interfaces for the Azure resources that implement this service.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

NFR No.	Requirement Name	Description	Comment/Traceability
NFR010d	Systems Operator role	• For a Systems Operator role, the VDS System will allow access to easily change access to data, user interfaces and/or functionality.	The User and Training Provider Access operations (see 5.5.1) provide part of the picture for Training Provider access to the system. Other manual processes, Azure AD management, and MyGovID/RAM configuration all work together to support user access management in STARS.
NFR010f	Flexible authentication	• The VDS System will support secure modern and flexible authentication and authorisations methods, including SAML federation with claims-based authorisation, OIDC etc.	Azure AD B2C single sign on will be supported by this service.
NFR010g	Identity Proofing	• The VDS System will implement the required identity proofing and credential strength for each user cohort in accordance with the TDIF identity proofing standards/guidelines. This will be driven by a TDIF assessment of the business risk of high importance transactions, decisions or activity.	The Provider Service supports MyGovID access and associated Azure AD B2C configuration and policies, as well as secure system to system authentication.
NFR010i	Reporting security	The VDS System restricts unauthorised users from accessing, viewing, and changing data and reports not relevant to their role.	All non-public facing operations will require authentication according to the STARS IDAM policies and practices
NFR017d	Daily backups	All server images and system data will be backed up on at least a daily basis.	Provider Service data stores will apply backup and lifecycle policies in line with the STARS Data Governance and Data Management Frameworks
NFR017e	Backup retention period	A full monthly backup must be retained securely for a period of seven (7) years.	Provider Service data stores will apply backup and lifecycle policies in line with the STARS Data Governance and Data Management Frameworks

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

NFR No.	Requirement Name	Description	Comment/Traceability
NFR018a	Data storage	The VET data stored in the VDS System will be retained indefinitely.	Provider Service data stores will apply backup and lifecycle policies in line with the STARS Data Governance and Data Management Frameworks
NFR018b	Access to data retained	The **seven (7)** number of years data is retained in the VDS system can be modifed, in order to reflect industry policy and subject to the data amendment policy.	Provider Service data stores will apply backup and lifecycle policies in line with the STARS Data Governance and Data Management Frameworks
NFR019a	Offline working	The VDS System should work offline (when network connectivity is unavailable) so that work can be carried out and the system will sync with the offline data when network connectivity is available.	The Provider Service API operations relating to Training Activity Data Preparation will support offline interactions by the Provider Portal PWA.
NFR021c	Integrated Omni channel	The VDS System must ensure that users of the service can have an integrated omni channel experience.	The fundamental design of the Provider Services is to ensure consistent interactions with STARS for Training Providers, regardless of whether their initial interaction is through the Provider Portal, an SMS or an STA system.
NFR021j	Useability during intermittent internet connectivity	The VDS System retains data entered and retries the validation process in the event of intermittent internet connectivity during one or across multiple user sessions.	All STARS communication will adhere to a consistent retry policy.
NFR021k	Data Submission Methods	The VDS system supports different forms of event - driven data submission in parallel such as single record entry and batch upload	The data entry and upload operations support any number of records.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

NFR No.	Requirement Name	Description	Comment/Traceability
NFR021I	API failure does not impact user session	The VDS System does not inhibit the user's ability to continue the session or continue entering data, in the event that the validation rules API fails to provide a response.	The Provider Service Validation process, described in section 5.5.2.4~~5.5.2.2~~, shows that the validation process continues regardless of any initial failure. Once the request is made, the user can proceed with other actions, and come back to check validation results at any time.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

12 Appendices

12.1 STARS Solution Design Diagrams

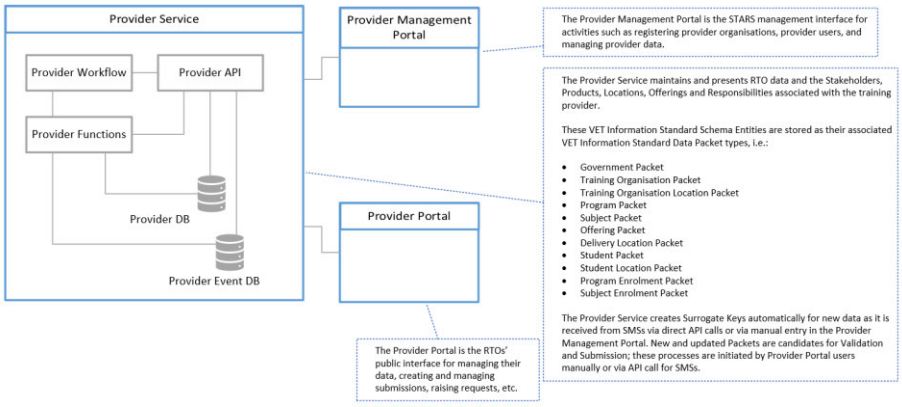


Figure 3636~~3835~~: Appendix: Provider Service (D:12)

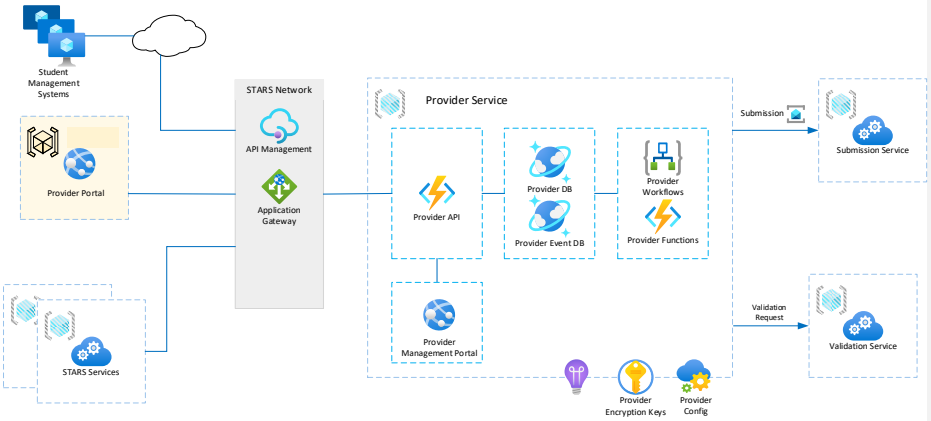


Figure 3737~~3936~~: Appendix: Provider Service (E:4)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 572 of 1295

~~13 Acronyms~~

Formatted: Space After: 8 pt, Line spacing: Multiple 1.08 li

~~Acronym~~	~~Definition~~
~~API~~	~~Application Programming Interface~~
~~CRUD~~	~~Create Read Update Delete~~
~~REST~~	~~Representational State Transfer~~
~~FR~~	~~Functional Requirement~~
~~NFR~~	~~Non-functional Requirement~~
~~STARS~~	~~Student Training & Activity Reporting System~~
~~VET~~	~~Vocational Education Training~~
~~VDS~~	~~VET Data Streamlining~~
~~IVETS~~	~~Integrated Vocational Education and Training System~~
~~DB~~	~~Database~~
~~DDS~~	~~Data Definition Language~~
~~SQL~~	~~Structured Query Language~~
~~TP~~	~~Training Provider~~
~~STA~~	~~State Training Authority~~
~~SMS~~	~~Student Management System~~
~~DET~~	~~Data Entry Tool~~
~~RAM~~	~~Relationship Authorisation Manager~~
~~PWA~~	~~Progressive Web App~~
~~AD~~	~~Active Directory~~
~~RBAC~~	~~Role Based Access Control~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 573 of 1295

Australian Government
Department of Employment and Workplace Relations

Detailed Design Document

Submission Service

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 574 of 1295

1 Document Management

1.1 Document Control

Version	Change Description	Date
0.1	Draft Version	10/02/2023
0.2	Updates to address review feedback	06/03/2023
0.3	Updates to address further review feedback	27/03/2023
1.0		

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This is a living document and will be updated as required. If another component's design scope requires an update to this design document, then the appropriate change will be made and a new version issued.

Formatted: Space Before: 12 pt

1.2 Document Author

Role	Name
Design Lead	s 47F(1)
Lead Solution Architect	s 47F(1)
Senior Technical Consultant	s 47F(1)

1.3 Document Reviewers

Role	Name

1.4 Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 575 of 1295

1.5 Related Documents

Reference	Document
A	STARS Solution Requirements Document – V.1.3.docx
B	STARS Solution Design – V.1.0.docx
C	STARS Solution Design - Attachment 3 - API Design - V.1.0.docx
D	-Submission Service Diagrams – D.0.2.vsdx
E	Detailed Design Document – Notification Service D.0.5.docx
F	Governance Framework VET Student and Training Activity Data.docx
G	VET Data Access and Privacy Protocol Final - 27 February 2023.docx

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 576 of 1295

1.6 All STARS System Components

This list outlines the high-level service dependencies with **Submission Service**. This is a reflection of what is known at the time of Acceptance and may be subject to change throughout the design process.

Green = This Design Document Grey = Other Service Dependencies

STARS Service
Azure Infrastructure and Networking
Identity & Access Management (IAM)
Event Service
Logging & Audit Service
Master Data Service
Reference Data Service
Data Governance Infrastructure
Data Quality Framework
Notification Service
Submission Service
Validation Service
Rules Service
Provider Service
Mastering Service
Downstream Consumer ETL Pipeline Services
Release Service
Downstream Consumer Reporting & Analytics
R & A Provisioning Service
STA Batch Submission Services
STA Gateway Services
USI Transcript Service
STA Bulk Data Consumption Service
SMS SDK Service

Formatted Table

Formatted Table

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 577 of 1295

Contents

1 Document Management 2
1.1 Document Control 2
1.2 Document Author 2
1.3 Document Reviewers 2
1.4 Document Approval 2
1.5 Related Documents 32
1.6 All STARS System Components 4
2 Introduction 168
2.1 Purpose 168
2.2 Intended Audience 179
2.3 Scope and Exclusions 179
2.3.1 In Scope 179
2.3.2 Out of Scope 179
3 Design Decisions 1911
4 Feature Overview 2012
4.1 Submission API 2412
4.2 Submission DB 2413
4.3 Process Submission Logic App 2613
4.4 Submission Encryption Keys 2613
4.5 Submission Config 2613
5 Business Processes 2714
5.1 Submit Activity data 2714
5.2 Review Submission Data 2814
5.3 Cross Jurisdiction Submission 2914
5.3.1 NT Funded and WA Submitting Submission (Indirect Submission) 2914
5.3.2 WA Funded and Reporting to WA (Indirect Submission) 3015
5.3.3 Two Training Activities NT Funded and WA Funded (Direct Submission) 3117
In this submission cross jurisdiction scenario, the student is enrolled in two different training activity in same training provider, one training activity is being funded by WA and the other training activity being funded by NT. The training provider submits the training activities to STARS directly. 3117
6 Submission Workflow 3419
6.1 Additional Context 3820

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 578 of 1295

7 Submission Events ..... 47~~21~~
7.1 Raised Events ..... 47~~21~~
7.2 Consumed Events ..... 50~~22~~
8 API Design ..... 54~~24~~
8.1 Submission API Endpoints ..... 55~~24~~
8.1.1 Get List of Submissions ..... 59~~24~~
8.1.2 Get Submission by ID ..... 59~~24~~
8.1.3 Get List of Activity Records (General) ..... 59~~24~~
8.1.4 Get Activity Record by ID (General) ..... 60~~24~~
9 Technical Design Patterns ..... 61~~25~~
9.1 General Endpoint Descriptions ..... 61~~25~~
9.1.1 Read Endpoint – List ..... 61~~25~~
9.1.2 Read Endpoint – Detail ..... 62~~26~~
9.2 Error Handling ..... 63~~27~~
9.2.1 Error Handling Submission API ..... 63~~27~~
9.2.2 Error Handling in Submission Logic Apps and Azure Function ..... 63~~27~~
10 Data ..... 66~~29~~
10.1 Submission Schema ..... 101~~29~~
10.2 Submission Data Processing ..... 101~~30~~
11 Access Management ..... 102~~31~~
11.1 Control of data ..... 102~~31~~
11.1.1 Data Owner ..... 102~~31~~
11.1.2 Data Custodian ..... 102~~31~~
11.2 Security ..... 104~~31~~
12 Dependency on Other Services ..... 105~~33~~
12.1 Rules Service ..... 105~~33~~
12.2 Validation Service ..... 105~~33~~
12.3 Event Service ..... 105~~33~~
12.4 Notification Service ..... 105~~33~~
13 Requirements ..... 107~~34~~
13.1 Functional ..... 107~~34~~
13.1.1 Fulfilled Requirements ..... 107~~34~~
13.1.2 Informing Requirements ..... 110~~37~~
13.2 Non-Functional ..... 124~~52~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 579 of 1295

14 Appendices 13361
Appendix 1. Acronyms 13361
1 Document Management 2
1.1 Document Control 2
1.2 Document Author 2
1.3 Document Reviewers 2
1.4 Document Approval 2
1.5 Related Documents 2
2 Introduction 6
2.1 Purpose 6
2.2 Intended Audience 7
2.3 Scope and Exclusions 7
2.3.1 In Scope 7
2.3.2 Out of Scope 7
3 Design Decisions 8
4 Feature Overview 9
4.1 Submission API 9
4.2 Submission DB 10
4.3 Process Submission Logic App 10
4.4 Submission Encryption Keys 10
4.5 Submission Config 10
5 Business Processes 11
5.1 Submit Activity data 11
5.2 Review Submission Data 11
5.3 Cross Jurisdiction Submission 11
5.3.1 NT Funded and WA Submitting Submission (Indirect Submission) 11
5.3.2 WA Funded and Reporting to WA (Indirect Submission) 12
5.3.3 Two Training Activities NT Funded and WA Funded (Direct Submission) 14
In this submission cross jurisdiction scenario, the student is enrolled in two different training activity in same training provider, one training activity is being funded by WA and the other training activity being funded by NT. The training provider submits the training activities to STARS directly. 14
6 Submission Workflow 16
7 Submission Events 18
7.1 Raised Events 18

7.2 Consumed Events 19
8 API Design 21
8.1 Submission API Endpoints 21
8.1.1 Get List of Submissions 21
8.1.2 Get Submission by ID 21
8.1.3 Get List of Activity Records (General) 21
8.1.1 Get Activity Record by ID (General) 21
9 Technical Design Patterns 22
9.1 General Endpoint Descriptions 22
9.1.1 Read Endpoint – List 22
9.1.2 Read Endpoint – Detail 23
9.2 Error Handling 24
9.2.1 Error Handling Submission API 24
9.2.2 Error Handling in Submission Logic Apps and Azure Function 24
10 Data 26
10.1 Submission Schema 26
10.2 Submission Data Processing 27
11 Access Management 28
11.1 Control of data 28
11.1.1 Data Owner 28
11.1.2 Data Custodian 28
11.2 Security 28
12 Dependency on Other Services 30
12.1 Rules Service 30
12.2 Validation Service 30
12.3 Event Service 30
12.4 Notification Service 30
13 Requirements 31
13.1 Functional 31
13.2 Non-Functional 40
14 Appendices 49
Appendix 1. Acronyms 49
1 Document Management 2
1.1 Document Control 2

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 581 of 1295

1.2 Document Author 2
1.3 Document Reviewers 2
1.4 Document Approval 2
1.5 Related Documents 2
2 Introduction 7
2.1 Purpose 7
2.2 Intended Audience 8
2.3 Scope and Exclusions 8
2.3.1 In Scope 8
2.3.2 Out of Scope 8
3 Design Decisions 9
4 Feature Overview 10
4.1 Submission API 10
4.2 Submission DB 11
4.3 Process Submission Logic App 11
4.4 Submission Encryption Keys 11
4.5 Submission Config 11
5 Business Processes 12
5.1 Submit Activity data 12
5.2 Review Submission Data 12
5.3 Cross Jurisdiction Submission 12
5.3.1 NT Funded and WA Submitting Submission (Indirect Submission) 12
5.3.2 WA Funded and Reporting to NT (Indirect Submission) 13
5.3.3 Two Training Activities NT Funded and WA Funded (Direct Submission) 14
6 Submission Workflow 16
7 Submission Events 18
7.1 Raised Events 18
7.2 Consumed Events 19
8 API Design 21
8.1 Submission API Endpoints 21
8.1.1 Get List of Submissions 21
8.1.2 Get Submission by ID 21
8.1.3 Get List of Activity Records (General) 21
8.1.1 Get Activity Record by ID (General) 21

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 582 of 1295

9 Technical Design Patterns 22
9.1 General Endpoint Descriptions 22
9.1.1 Read Endpoint – List 22
9.1.2 Read Endpoint – Detail 23
9.2 Error Handling 24
9.2.1 Error Handling Submission API 24
9.2.2 Error Handling in Submission Logic Apps and Azure Function 24
10 Data 26
10.1 Submission Schema 26
10.2 Submission Data Processing 27
11 Access Management 28
11.1 Control of data 28
11.1.1 Data Owner 28
11.1.2 Data Custodian 28
11.2 Security 28
12 Dependency on Other Services 30
12.1 Rules Service 30
12.2 Validation Service 30
12.3 Event Service 30
12.4 Notification Service 30
13 Requirements 31
13.1 Functional 31
13.2 Non-Functional 40
14 Appendices 49
Appendix 1. Acronyms 49
1 Document Management 2
1.1 Document Control 2
1.2 Document Author 2
1.3 Document Reviewers 2
1.4 Document Approval 2
1.5 Related Documents 2
2 Introduction 5
2.1 Purpose 5
2.2 Intended Audience 6

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 583 of 1295

2.3 Scope and Exclusions 6
2.3.1 In Scope 6
2.3.2 Out of Scope 6
3 Design Decisions 7
4 Feature Overview 8
4.1 Submission API 8
4.2 Submission DB 9
4.3 Process Submission Logic App 9
4.4 Submission Encryption Keys 9
4.5 Submission Config 9
5 Business Processes 10
5.1 Submit Activity data 10
5.2 Review submission data 10
6 Submission Workflow 11
7 Submission Events 12
7.1 Raised Events 12
7.2 Consumed Events 13
8 API Design 14
8.1 Submission API Endpoints 14
8.1.1 Get List of Submissions 14
8.1.2 Get Submission by ID 14
8.1.3 Get List of Activity Records (General) 14
8.1.1 Get Activity Record by ID (General) 14
9 Technical Design Patterns 15
10 Data 16
10.1 Submission Schema 16
11 Access Management 17
11.1 Control of data 17
11.1.1 Data Owner 17
11.1.2 Data Custodian 17
11.2 Security 17
12 Dependency on Other Services 19
12.1 Rules Service 19
12.2 Validation Service 19

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 584 of 1295

12.3 Event Service 19
13 Requirements 20
13.1 Functional 20
13.2 Non-Functional 31
14 Appendices 40
Appendix 1. Acronyms 40
1 Document Management 2
1.1 Document Control 2
2 Introduction 6
2.1 Purpose 6
2.2 Intended Audience 7
2.3 Scope and Exclusions 7
2.3.1 In Scope 7
2.3.2 Out of Scope 7
3 Design Decisions 8
4 Feature Overview 9
4.1 Submission Queue 10
4.2 Submission Workflow 10
4.3 Submission Functions 10
4.4 Submission Rules Engine 10
4.5 Submission API 10
4.6 Submission DB 11
4.7 Submission Event DB 11
4.8 Overview of Submission Service 11
4.8.1 Submission API 12
4.8.2 Submission Queue 12
4.8.3 Monitor Submission 12
4.8.4 Process Submission 12
4.8.5 Submission database 12
4.8.6 Submission Event database 12
4.8.7 Submission Encryption Keys 12
4.8.8 Submission Config 13
5 Business Processes 14
5.1 Funded Activity 14

Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Tab stops: Not at 0.78 cm
Formatted: Default Paragraph Font, Check spelling and grammar

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 585 of 1295

5.2 Fee For Service (FFS) 14
5.3 Submission by Training Provider 14
5.4 Submission by STA 15
6 Submission Service Workflow 16
6.1 Funded Training Activity Submission by Training Provider 16
6.2 FFS Training Activity Submission by Training Provider 17
6.3 Training Activity Submission by STA 18
6.4 Funded Training Activity Resubmission by Training Provider 19
6.5 FFS Training Activity Resubmission by Training Provider 20
6.6 Training Activity Resubmission by STA 21
6.7 Get Operations by Submission API 22
7 Submission Events 24
7.1 Submission Event 24
7.2 Resubmission Event 25
7.3 Submission Verification Event 25
7.4 Submission Validation Event 26
8 API Design 27
8.1 Submission API 27
8.2 Submission database 27
8.3 Submission Event database 27
8.4 Submission Encryption Keys 28
8.5 Submission Config 28
8.6 Submission API End Points 28
8.6.1 Save Submission 28
8.6.2 Get Submission 29
8.6.3 Get Submission Status 29
8.6.4 Get Submission Historical Data 29
8.6.5 Get Student 30
8.6.6 Get Student By Id 30
8.6.7 Get Student Enrolments 30
8.6.8 Get Training Provider 31
8.6.9 Get Training Provider By Id 31
9 Technical Design Patterns 32
10 Data 33

Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted ...
Formatted: Tab stops: Not at 0.78 cm
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted: Tab stops: Not at 0.78 cm
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted: Tab stops: Not at 0.78 cm
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...

10.1 VET IS Conceptual Model 33
10.2 VET IS Logical Schema 34
10.3 VET IS Data Packet Schema 34
10.4 Schemas 36
10.4.1 Training Provider 36
10.4.2 Student 38
10.4.3 Program 42
10.4.4 Subject 45
10.4.5 Training Provider Location 46
10.4.6 Student Location 48
10.4.7 Delivery Location 57
10.4.8 Program Enrolment 58
10.4.9 Subject Enrolment 63
11 Access Management 68
11.1 Control of data 68
11.1.1 Data Owner 68
11.1.2 Data Custodian 68
11.2 Security 68
12 Dependency on Other Services 70
12.1 Validation Service 70
12.2 Rules Service 70
12.3 Notification Service 70
12.4 Event Service 70
13 Requirements 71
13.1 Functional 71
13.2 Non-Functional 82
14 Appendices 91
Appendix 1. Acronyms 91

Figure 1: Submission Service Design Overview (D:1) 2211
Figure 2: Cross Jurisdiction Scenario (D:3) 2914
Figure 3: Cross Jurisdiction Scenario (D:4) 3015
Figure 4: Cross Jurisdiction Scenario(D:5) 3216

Table 1: Events raised by the Submission Service 4720

Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Font: Check spelling and grammar
Formatted: Default Paragraph Font, Font: Check spelling and grammar
Formatted: Default Paragraph Font, Font: Check spelling and grammar
Formatted: Default Paragraph Font, Font: Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Font: Check spelling and grammar
Formatted: Default Paragraph Font, Font: Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 587 of 1295

Table 2: Events Consumed by Submission Service 5021
Table 3: Submission API Endpoints 5523
Table 4: List Endpoint Query Parameters 6124
Table 5: List Endpoint Response Fields 6225
Table 6: Read Endpoint Query Parameters 6225
Table 7: Functional Requirements 11033
Table 8: Non-Functional Requirements 12442
Figure 1: API Design Overview (D:1) 8
Figure 2: Training Activity Submission (D:2) 11

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 588 of 1295

2 Introduction

This document covers the design of the STARS Submission Service.

Note: This is a living document that will be updated as designs for various STARS services are developed and evolve. Accordingly, this document will evolve to reflect and document infrastructure and connectivity as the STARS solution build progresses.

The design is informed by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework, which provide a comprehensive set of best practices and guidance for designing, deploying, and operating secure, high-performing, and resilient cloud infrastructure and applications.

The Submission service is one of the core services in Submission pipeline services. It has the role of receiving information from Training Providers or STAs via the provider service, correlating and recording the information for later reporting and audit, validating information and then passing information on to the Staging Service or Release Service. Invalid submissions are referred to the Training Provider or STA via the provider service for correction.

Formatted: Highlight

The Submission Service receives, records and validates all submissions along with the full history of submission events. It is the source of truth for all data submissions, including submission history. Valid submissions are passed into the STARS submission pipeline automatically.

All Ssubmissions are received ~~either~~ via the Provider Service ~~(for RTO sourced data) or via the STA Gateway Service (for STA sourced RTO data.)~~

The Submission Service has two primary roles:

- Manage the workflows and processes involved with receipt, validation and hand-off of submissions to the Staging or Releases s~~S~~ervices.
- Managing and presenting submission data through a consistent schema and API, for use by other STARS solution services.

2.1 Purpose

The purpose of this detailed design document is to provide a technical description of the Submission Service. The document is intended for technical stakeholders, including architects, software engineers, administrators, and operations teams, who are responsible for designing, developing, deploying, and operating the STARS solution in Azure.

The document also serves as a reference for ongoing management and evolution of the STARS solution and to ensure that the STARS solution is deployed and operated in accordance with the best practices and guidance provided by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 589 of 1295

The document is intended to be a living document, updated as the STARS solution evolves and changes over time. It provides a clear and concise understanding of the Azure infrastructure and networking components and their configurations, helping to ensure the STARS solution is deployed, operated, and managed in a consistent and efficient manner.

With respect to expected RTO/SMS engagement activities, it is DXC's understanding that there is no component in this Deliverable that impacts RTO/SMSs. All RTO/SMS impacts are dealt with in the Provider Portal, Provider Service and/or SMS SDK Service as understood and agreed previously.

2.2 Intended Audience

This detailed design document provides technical information and guidance to help technical stakeholders understand the Submission components of the STARS solution and how they integrate with each other. The document is expected to be used by technical stakeholders who have a good understanding of cloud computing concepts and are experienced in the Azure platform.

This document is not intended for non-technical readers, nor for readers who are not experienced in the design and configuration of Azure-based solutions; informational and educational resources regarding Azure are readily available at https://Azure.microsoft.com

The primary audience for this document includes:

- Program technical staff: This includes architects, software engineers, administrators, and operations teams within the program responsible for the STARS solution.
- External technical stakeholders: This includes technical staff from other federal, state, and territory governments agencies, including VDS embedded team members.
- ICT Technical Staff: This includes a variety of technical roles involved during the VDS program and post-go live, including network engineers, systems administrators, database administrators, security engineers, and DevOps engineers.

2.3 Scope and Exclusions

2.3.1 In Scope

The scope of this document is restricted to describing the Submission service and its dependencies on other STARS services and various Azure services. Other services are described in their respective design documents.

This document includes Submission service workflows design, Azure Function design, Azure Service bus, Events specification, API Specification, and Cosmos database schemas.

Business rules, Business process and security matrix are defined in this document to guide the development of Logic App workflows.

2.3.2 Out of Scope

Submission service has dependency on ~~many other services, e.g.,~~ the Validation Service, Rules Service, Events Service and~~,~~ Notification Service. The~~se other services~~ design details and business processes for these services are out of scope for this detailed design document.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 590 of 1295

Similarly, the Submission service is a part of Submission pipeline services, which has other services including Staging service, ~~Release service,~~ Mastering service and Report and Analytics service. These are also out of scope.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 591 of 1295

3 Design Decisions

STARS will utilise Service Oriented Architecture (SOA) principles for design and implementation, coupled with Master Data Management (MDM) principles for population of the Master Database. The solution will be hosted in Microsoft Azure and will conform to the Microsoft Well-Architected Framework[1], utilising recognised Azure Reference Architectures[2].

The architecture and design principles are described in detail in section **Error! Reference source not found.**~~7~~ '**Error! Reference source not found.**~~Solution Architecture & Design Principles~~' of the STARS Solution Design document.

Of note is that the entire STARS solution will be implemented within the Microsoft Azure cloud platform and will heavily utilise Platform as a Service (PaaS) offerings. Accordingly, the Solution Design is oriented around utilising Azure features and resources that support the design and implementation of loosely coupled services. These include, for example, having services implemented in separate Resource Groups, using features such as Azure Functions, Azure SQL, and service-specific Azure App configurations.

Loose coupling will be achieved using ~~Message and~~ Event-Driven approaches for communication between services~~,~~ using ~~features such as~~ Azure Service Bus ~~and Azure Event Grid~~. Services that present HTTPS/REST/JSON APIs will have their URLs controlled through Azure Application Gateway. This will enable a consistent URL scheme that also facilitates loose coupling.

The Submission API will be implemented using OData, in line with the common OData implementation for STARS APIs.

[1] https://docs.microsoft.com/en-us/azure/architecture/framework/

[2] https://docs.microsoft.com/en-us/azure/architecture/

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 592 of 1295

4 Feature Overview

The Submission Service plays a core role within the pipeline where it receives and processes raw information. It also provides submission information to other services both within the Submission pipeline services and elsewhere needed by other STARS services.

The Submission Service has two primary roles:

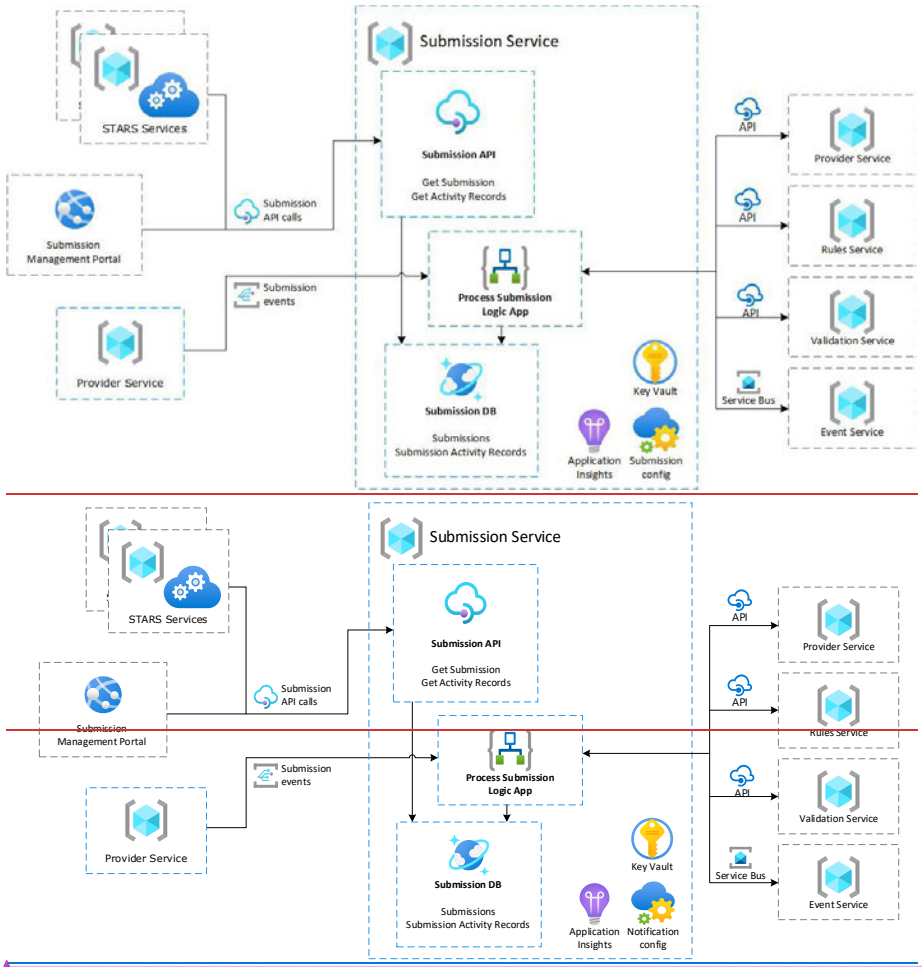
- Manage the workflows and processes involved with receipt, validation and hand-off of submissions to the Staging and Mastering ~~S~~services.
- ~~Managing and p~~Making ~~roviding~~ submission data available through a consistent schema and API, for use by other STARS solution services.

Submissions are raised as events by other STARS solution services (at this stage expected to just be the Provider Service) which the Submission Service subscribes to. The Submission Service receives, ~~records~~ stores and validates all submissions. ~~along with the full history of submission events.~~ It is the source of truth for all submissions (valid or not). Once records within a~~Valid~~ submission are validated, an event is raised so that they can be ~~s are notified~~ progressed to the Staging or Mastering ~~service~~services ~~automatically~~.

~~Submissions are received either via the Provider Service (for RTO sourced data) or via the STA Gateway Service (for STA sourced RTO data.). Submissions are received from these services as messages in the Submission Queue.~~

Submissions across the entire submission pipeline are held and processed in VET-IS JSON schema format. Services that feed-in submissions to the pipeline (via the Submission Service) are responsible for transforming VET data from the original format, into the VET-IS JSON schema.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 593 of 1295



Field Code Changed

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 594 of 1295

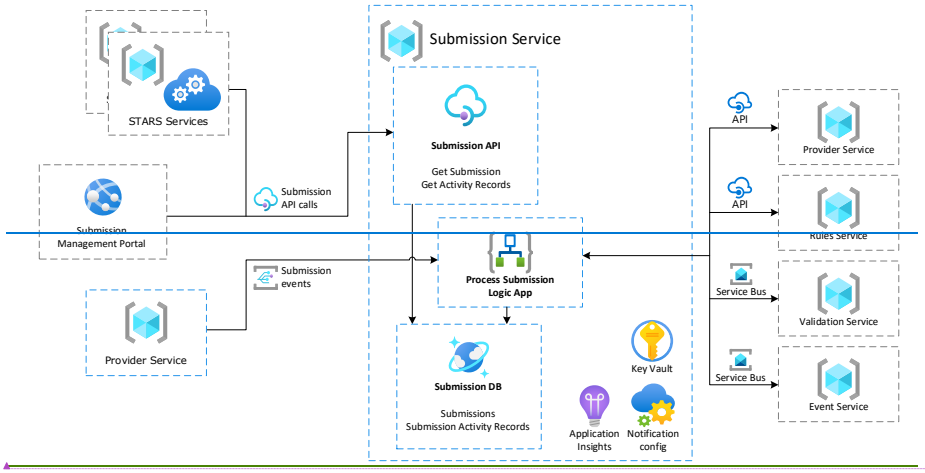
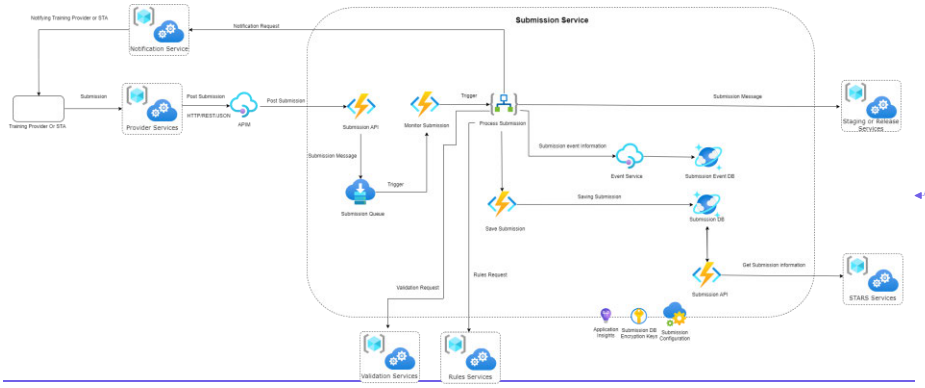


Figure 11~~1~~: ~~API~~Submission Service Design Overview (D:1)



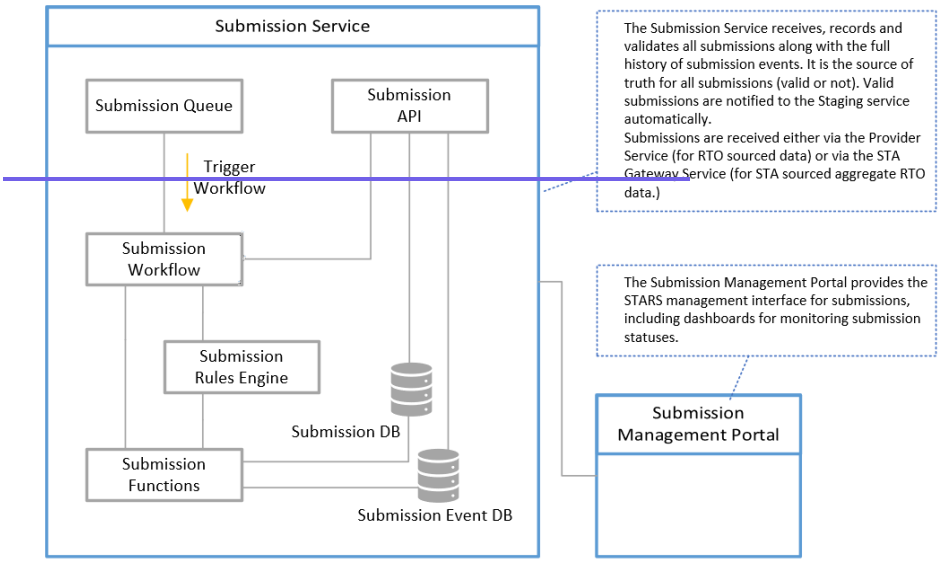
~~Figure 21: Submission Service overview in Azure~~

Field Code Changed

Formatted: Caption

Formatted: Justified

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 595 of 1295



~~**Figure 1: Submission Service**~~

~~Following is the description of different Submission Service components:~~

~~4.1 Submission Queue~~

~~The Submission Service does not receive submissions directly from RTOs or STAs; submissions are placed into the Submission Queue, in VET IS JSON schema format, by the Provider or STA Batch Submission services.~~

~~The Submission Queue will be implemented within Azure Service Bus. It is likely that multiple queues will be utilised to support various STARS submission operational purposes (e.g., to support different VET IS versions, or different STA business processes.) In this regard, the Submission Queue is a 'logical' construct that supports loose coupling between the Submission Service and any upstream services that are collecting submissions for the submission pipeline service.~~

~~4.2 Submission Workflow~~

~~The Submission Service internal processes will be controlled by one or more Azure Logic App workflows.~~

~~The Submission Queue is monitored by the Submission Workflow Logic App(s). As submissions arrive in the queue, the workflow will record the submission in the Submission DB, prior to initiating the relevant validation workflows (see Validation Service for details.).~~

~~Valid records in a submissions are handed off to the Staging Service (funded) or Release Service (FFS) via messages to their respective input queues, and a 'success' message returned to the originating service.~~

~~Invalid records in a submissions are returned to the originating service for attention by either thethe RTO Training Provider or STA(possibly via the STA), along with detailed reasons for validation failure.~~

~~The details are scenario specific, but generally submission results are returned to the upstream service via 'call-back' URLs hosted within the upstream service, with the call-backs uniquely associated with each input submission.~~

~~4.3 Submission Functions~~

~~The Submission Service code-based functionality will be implemented as Azure Function Apps. These functions will support the operation of the service's workflows as well as functionality provided through the Submission Management Portal. For example, the portal will have functions to monitor submission data and results, the actual code to implement this at the database level will be within the Function Apps.~~

~~4.4 Submission Rules Engine~~

~~The Submission Rules Engine, and any associated rules, is intended to assist with the operation of the Submission Workflows. Please note that this does not include submission validation rules – these are handled by the Validation Service. The submission rules engine role will apply any business rules associated with processing of entire submissions (not the content of the submissions).~~

~~4.5~~4.1 Submission API

The API will allow STARS services to flexibly query data in JSON format, including validation and submission data. It will conform to the STARS Design Principles and the STARS API Design Document. Filtering (including free-text search) will be handled using OData.

~~The Submission API will be implemented as an Azure Function App, utilising HTTP endpoints.~~

~~The API will allow other services to flexibly query and extract submission data in JSON format, including information on the submissions themselves and all the submission lifecycle data, including submission status.~~

The API will be refined during development, pursuant to the design and implementation of the Submission DB and associated VET-IS SQL and JSON schemas.

~~Care will be taken to ensure submission~~The Submission data can be presented via the API to workflows and the rules engine(s) in a format to allow rapid processing of rules – please see the discussion in section **Error! Reference source not found.**~~10.4~~ '**Error! Reference source not found.**~~Rules Engine~~' in STARS Solution Design Document~~ V 1.0~~. Technical details of the Submissions API are in Section 8 API Design~~8 API Design 8: API Design~~.

~~The API will conform to the STARS Design Principles and the STARS API Design Document.~~

~~4.6~~4.2 Submission DB

The Submission Database will be implemented in Azure Cosmos DB. It will utilise a standardised document-oriented JSON schema to ensure consistency in the management and presentation of submission data. It will store VET-IS Activity packets alongside submission information including validation status. For more detail see Section 10 Data.~~ 10: Data~~

~~4.7 Submission Event DB~~

Formatted: Normal

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 597 of 1295

All service events relating to the lifecycle and management of submission data will be recorded, in the STARS Event schema, in Cosmos DB.

4.8 Overview of Submission Service

Formatted: Normal

Following diagram provides an overview of Submission Service in Azure.

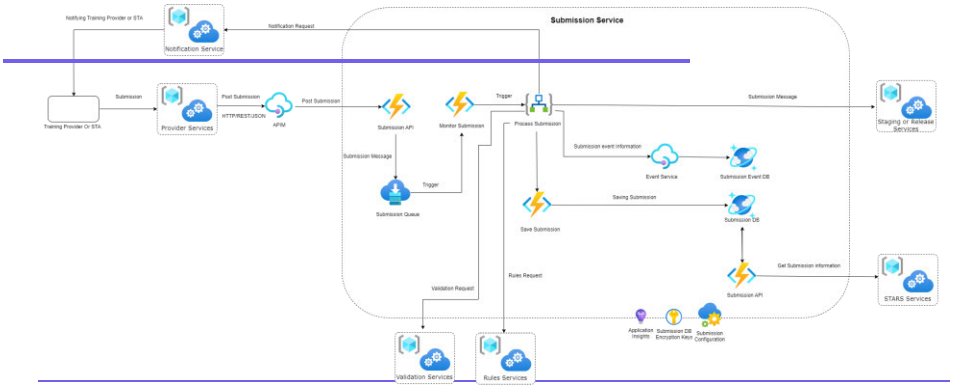


Figure 2: Submission Service overview in Azure

Formatted: Normal

Following are different Azure Services utilized in Submission service:

- Azure Logic App

Formatted: Normal, No bullets or numbering

- Azure Functions
- Azure Service Bus
- Cosmos Database
- Key Vault
- Application Insight
- App Config

Following are the details about different services in Submission service:

4.8.1 Submission API

Formatted: Normal

The Submission API is an HTTP-triggered function app. Submission data management and query operations will be mapped to relevant function app endpoints. The API will be factored around VET-IS data packet entities (e.g., students, enrolments, delivery locations).

4.8.2 Submission Queue

Formatted: Normal

The Submission Queue will be implemented within Azure Service Bus. It is likely that multiple queues will be utilised to support various STARS submission operational purposes (e.g., to support different VET-IS versions, or different STA business processes.) In this regard, the Submission Queue is a 'logical' construct that supports loose coupling between the Submission Service and any upstream services that are initiating submissions for the pipeline.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 598 of 1295

~~4.8.3 Monitor Submission~~

~~Monitor submission is an Azure Function that will be triggered when Submission queue receives a new submission message from Training Providers or STA VIA Submission API. The submission message is passed to the Process Submission logic app for further processing.~~

~~4.9~~4.3 Process Submission Logic App

The ~~P~~Process ~~s~~Submission Workflow will be implemented as an Azure ~~logic app which will be triggered by Monitor Submission function~~Logic App, responsible for monitoring for submission events and directing them through the submission workflow. Details on the submission workflow can be found in Section ~~6~~6: Submission Workflow.Submission Workflow~~Submission Workflow~~

~~and it will pass the submission message to process submission logic app which will send the submission for rules verification, validation and triggering Save Azure Function to save submission data in Cosmos database. This process submission logic app will send the event information request to event service to save it in the submission event Cosmos database. It will send notification request to notification service for notifying Training Provider or STA if the submission is not validated. If validated it will send the submission message to the Staging or release service.~~

~~4.9.1 **Submission database**~~

~~**The Submission Database is an Azure Cosmos DB database that stores submission data in VET-IS Data Packet Schema which will be JSON based.**~~

~~4.9.2 **Submission Event database**~~

~~**The Submission Event Database is an Azure Cosmos DB database instance that stores all submission service, data management and lifecycle events.**~~

~~4.10~~4.4 Submission Encryption Keys

~~-~~The Submission data will be encrypted at rest, encryption keys to be held in Azure key vault.

~~4.11~~4.5 Submission Config

Any Resource Group ~~G~~ specific config, including for function apps, ~~to~~ will be held ~~in a common~~ Azure ~~a~~App ~~c~~Configuration.

Formatted: Normal

Field Code Changed

Formatted: Heading 3,Level 1 - 1,Numbered Heading 3,hd3,h3,h3 sub heading,Heading 3 - old,H3,Head 3,C Sub-Sub/Italic,Head 31,Head 32,C Sub-Sub/Italic1,Head 33,C Sub-Sub/Italic2,Head 311,Head 321,C Sub-Sub/Italic11,h31,3,not in TOC,Heading 3 Char1,Para3,H31,sub-sub, Left

Formatted: Heading 2,Reset numbering,Numbered Heading 2,h2 main heading,B Sub/Bold,hd2,h2,Header 2,B Sub/Bold1,B Sub/Bold2,B Sub/Bold11,h2 main heading1,h2 main heading2,B Sub/Bold3,B Sub/Bold12,h2 main heading3,B Sub/Bold4,B Sub/Bold13,h2 main heading4,l2,list 2

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 599 of 1295

5 Business Processes

There are threewo main business processes supported by the Submission Service: submitting activity data, and reviewing the submitted data and cross jusridiction submission.

The Submission Service receives, records and validates all submissions along with the full history of submission events. It is the source of truth for all submission data. Valid submissions are notified to the Staging service automatically.

Submissions are received either via the Provider Service (for Training Provider sourced data) or via the STA Gateway Service (for STA sourced Training Provider data.) Submissions are received from these services as messages in the Submission Queue.

There are two types of Submission Training activities:

- Funded Activity (FA)
- Fee for Service Activity (FFS)

Training Providers conduct both fee-for-service (FFS) and government funded training (FA).

As part of VET training regulatory requirements, Training Providers must collect and report on training activity throughout the course of a year, including both fee-for-service (FFS) and government funded training. At intervals, ranging from monthly through to annually depending on circumstances, Training Providers report training activity to either or both of NCVER and the relevant STA. Currently this is done via transmission of data files in AVETMISS format.

In the current process FFS training activity data is submitted by Training Providers to NCVER directly. FFS data may be submitted to Training Providers local STA, depending on the STA's requirements, but activity reporting to NCVER is required by national and state training regulatory requirements.

4.12 Funded Activity

Funded training is where student fees are paid for by government(s) on behalf of students to Training Providers. Funded training is conducted under a contract between Training Provider and a STA and/or Federal Government, with the contracts covering many conditions on the training services provided.

4.13 Fee For Service (FFS)

FFS training is not conducted under any contracts between Training Providers, STAs or the Federal Government and is charged direct to students. In either instance, Training Providers must comply to many regulatory requirements covering delivering training services in the VET sector, including reporting.

In case of STARS Training providers will be submitting both types of training activities to Submission service in Submission Pipeline services. STAs will be submitting Training Providers sourced submission into the Submission service in Submission Pipeline services.

4.145.1 Submit Activity datassion by Training Provider

It is a requirement that TPTraining Providerss and STAs cancan submit activity data to the STARS system either via the Provider Portal, or via an external SMS. Regardless of where submissions

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 600 of 1295

~~Following will be the business process flow of Submission if the Submission is submitted by Training Provider:~~

originate, they will come to the Submission Service via the Provider Service. As such, the scope of the Submission Service within the broader submission business process is quite contained. The Submission Service will monitor for submissions, save all submissions (valid or not), trigger and monitor validation, raise events for each package as it finishes validation so that the appropriate service in the submission pipeline can progress the submission, and finally raise an event to inform the Provider Service once validation on all packages in the submission has completed (successful or not). The submission workflow is covered in detail in Section ~~6~~6: Submission Workflow ~~Submission Workflow~~.

- ~~**Review submission data S**~~**Submissions containing complete Enrolment information are received from Training Providers.**
- **The submission request and the submission data are correlated and recorded in the Submission Data Store.**
- **Data quality and business rules are applied via validation request to the Rules service and Validation Service.**
- **If the submission fails validation, it is rejected, with details of the rejected submission recorded in the Submission Data Store, with control passing back to the Training Provider for their attention.**
- **After successful validation, a distinction is made between funded and fee-for-service (FFS) activity, based on defined rules.**

5.2 Review S~~s~~ubmission D~~d~~ata~~FFS data is released to mastering via a message to the Release Service. Funded activity is forwarded on via a message to the Staging Service for further attention by STAs.~~

It is a requirement that Training Providers ~~TPs~~ and STAs can review all data that they have submitted. The Submission Service supports this by providing an API which can be accessed via STARS services such as the STA Gateway service and the Provider Service. The API will provide the mechanism for accessing all activity data that has been submitted, as well as data about the submissions themselves such as when submissions occurred, who made the submission, and what was included in the submission. Full technical details of the Submission Service API can be found in Section 8 ~~API Design8: API Design~~.

Formatted: Heading 2,Reset numbering,Numbered Heading 2,h2 main heading,B Sub/Bold,hd2,h2,Header 2,B Sub/Bold1,B Sub/Bold2,B Sub/Bold11,h2 main heading1,h2 main heading2,B Sub/Bold3,B Sub/Bold12,h2 main heading3,B Sub/Bold4,B Sub/Bold13,h2 main heading4,l2,list 2

Formatted: Heading 2,Reset numbering,Numbered Heading 2,h2 main heading,B Sub/Bold,hd2,h2,Header 2,B Sub/Bold1,B Sub/Bold2,B Sub/Bold11,h2 main heading1,h2 main heading2,B Sub/Bold3,B Sub/Bold12,h2 main heading3,B Sub/Bold4,B Sub/Bold13,h2 main heading4,l2,list 2, No bullets or numbering

Field Code Changed

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 601 of 1295

5.3 Cross Jurisdiction Submission

Following are different examples of cross jurisdiction business process scenarios, NT and WA are being used for these scenarios:

5.3.1 NT Funded and WA Submitting Submission (Indirect Submission)

In this submission cross jurisdiction scenario, the NT resident of the border community training activity is being funded by NT (Northern Territory) and training activity is being carried out in training provider in WA (Western Australia). Training Provider submits the submission to WA STA and WA STA submits the submission to STARS.

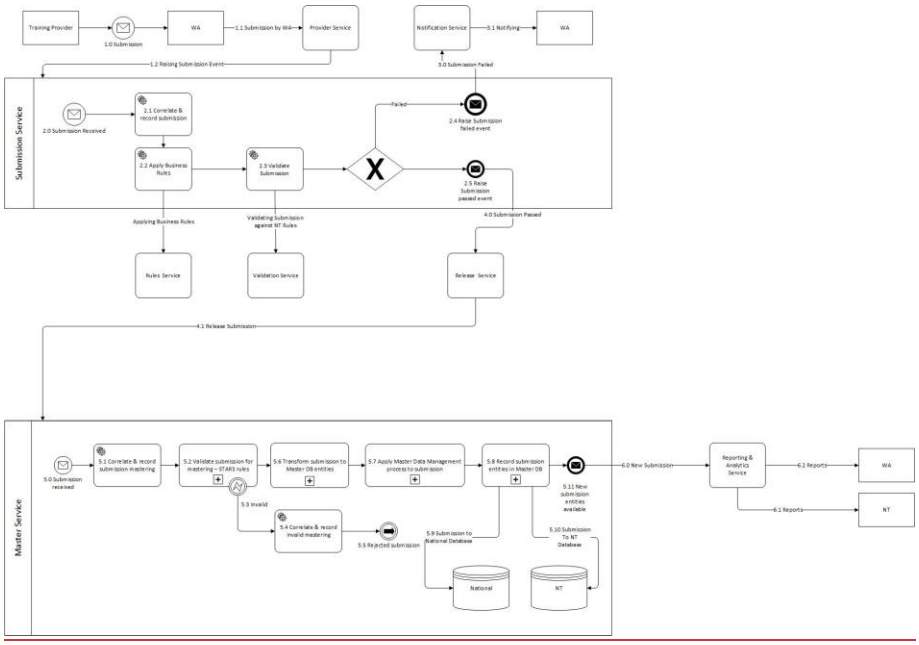


Figure 222: Cross Jurisdiction Scenario (D:3)

The training provider submits the submission data to WA STA. WA STA submits the submission data to Provider Service. The Provider Service raises an event for submission which is consumed by Submission service. Submission service gets the detail of submission from Provider Service and saves the submission in submission service database. The Submission service applies business rules by sending a request to Rules service. By these rules funding arrangement of the training activity is determined, in this scenario this training activity is being funded by NT. The Submission service

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 602 of 1295

validates the submission data against NT (and National) rules by sending the request to Validation Service. If ~~the~~any of the submitted ~~submission data~~records fail~~ed~~ validation, then WA STA is notified via Notification Service. If any of the submitted ~~submission data~~ records pass~~es~~ validation, then the Submission Service raises ~~an event of~~a Submission Passed event, which is consumed by Release Service. The Release ~~s~~Service receives submission data and processes the valid submission data to release it to Master Service to save the submission data after validation to National master database and NT master database. After completing Data Quality checks, t~~T~~his new submission ~~entity~~data is made available to Reporting and Analytics Service which exposes the submission data reports to NT and WA STA.

Formatted: Normal

5.3.2 WA Funded and Reporting to ~~NT~~WA (Indirect Submission)

In this submission cross jurisdiction scenario, the NT (Northern Territory) resident of the border community training activity is being funded by WA (Western Australia) and training activity is being carried out in training provider in WA (Western Australia). Training Provider submits the submission to WA STA and WA STA submits the submission to STARS.

Formatted: Normal

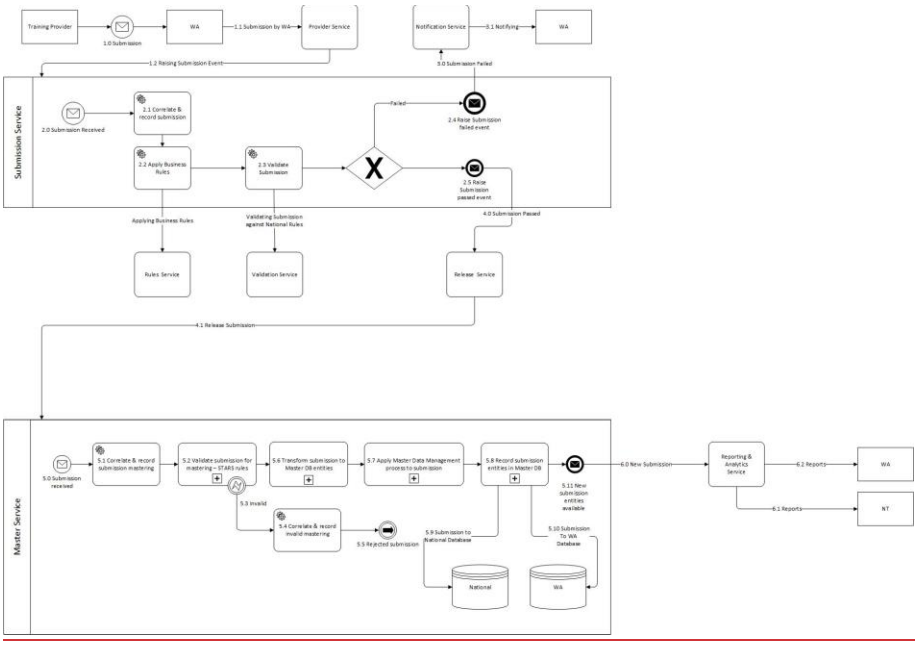


Figure ~~333~~: Cross Jurisdiction Scenario (D:4)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 603 of 1295

The training provider submits the submission data to WA STA. WA STA submits the submission data to Provider Service. The Provider Service raises an event for submission which is consumed by Submission service. Submission service gets the detail of submission from Provider Service and saves the submission in submission service database. The Submission service applies business rules by sending a request to Rules service. By these rules funding arrangement of the training activity is determined, in this scenario this training activity is being funded by WA and it will be validated against the National rules, (as WA rules are run prior to submitting the submission data to submission service)~~.~~ ~~The Submission service validates the submission data against national rules by sending the request to Validation Service.~~ If any the submitted ~~submission data~~records fail~~ed~~ validation, then WA STA is notified via Notification Service. If any of the submitted ~~submission data~~ records pass~~es~~ validation, then the Submission Service raises ~~an event of~~ a Submission Passed event (referencing the valid records), which is consumed by Release Service. ~~–~~The Release service receives submission data and process the submission data to release it to Master Service to save the submission data after validation to National master database and WA master database. This new submission entity is made available to Reporting and Analytics Service which exposes the submission data reports to NT (pending confirmation of data access rules) and WA STA.

5.3.3 Two Training Activities NT Funded and WA Funded (Direct Submission)

In this submission cross jurisdiction scenario, the student is enrolled in two different training activity in same training provider, one training activity is being funded by WA and the other training activity being funded by NT. The training provider submits the training activities to STARS directly.

Formatted: Font: 11 pt

Formatted: Heading 3,Level 1 - 1,Numbered Heading 3,hd3,h3,h3 sub heading,Heading 3 - old,H3,Head 3,C Sub-Sub/Italic,Head 31,Head 32,C Sub-Sub/Italic1,Head 33,C Sub-Sub/Italic2,Head 311,Head 321,C Sub-Sub/Italic11,h31,3,not in TOC,Heading 3 Char1,Para3,H31,sub-sub

Formatted: Font: 11 pt

Formatted: Font: 11 pt

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 604 of 1295

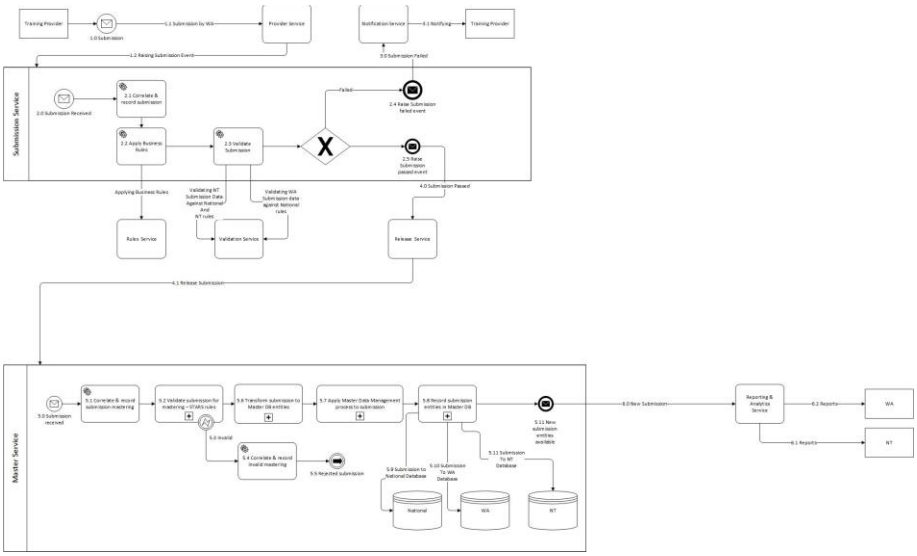


Figure ~~444~~: Cross Jurisdiction Scenario(D:5)

The training provider submits the submission data to the STARS Provider Service and it contains two different training activities, one of the training activity funded by NT and other by WA. The Provider Service raises an event for submission which is consumed by Submission service. Submission service gets the detail of submission from Provider Service and saves the submission in submission service database. The Submission service applies business rules by sending a request to Rules service. By these rules funding arrangement of the training activity is determined~~.~~. ~~I~~In this scenario one of the training activity records is being funded by NT and other ~~training activity~~ by WA. The Submission service validates the submission data of training activity funded by NT against National and NT rules by sending the request to Validation Service and the submission service validates the submission data of training activity record funded by WA against the ~~WA rules~~National rules by sending the request to Validation service. NOTE: this record will not be validated against WA state rules, as STARS does not have this capability for indirect states). If either of the submitted ~~submission~~ records data failed validation, then training provider is notified via Provider Service and Notification Service. If ~~the~~either ~~submitted submission data~~ record passes validation, then the Submission Service raises ~~an event of~~ a Submission Passed event for each funding arrangement. ~~, which is consumed by Release Service.~~ The Release service ~~receives~~subscribes to Submission Passed events for NT funded activity. This record can then be Released and Mastered, being recorded in the NT master database, and released for National Reporting in the Reporting and Analytics database. The Mastering service subscribes to Submission Passed events for WA funded activity This record can then be recorded in the WA master database, and released for National Reporting (for NCVER and all Jurisdictions) in the Reporting and Analytics database. ~~submission data and process the submission data to release it to Master Service to save the submission data after validation to National master database, NT training activity submission data in NT master database and WA training activity submission data in WA~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 605 of 1295

master database. These new submission entities are made available to Reporting and Analytics Service which exposes the submission data reports to NT and WA STA.

4.15 Submission by STA

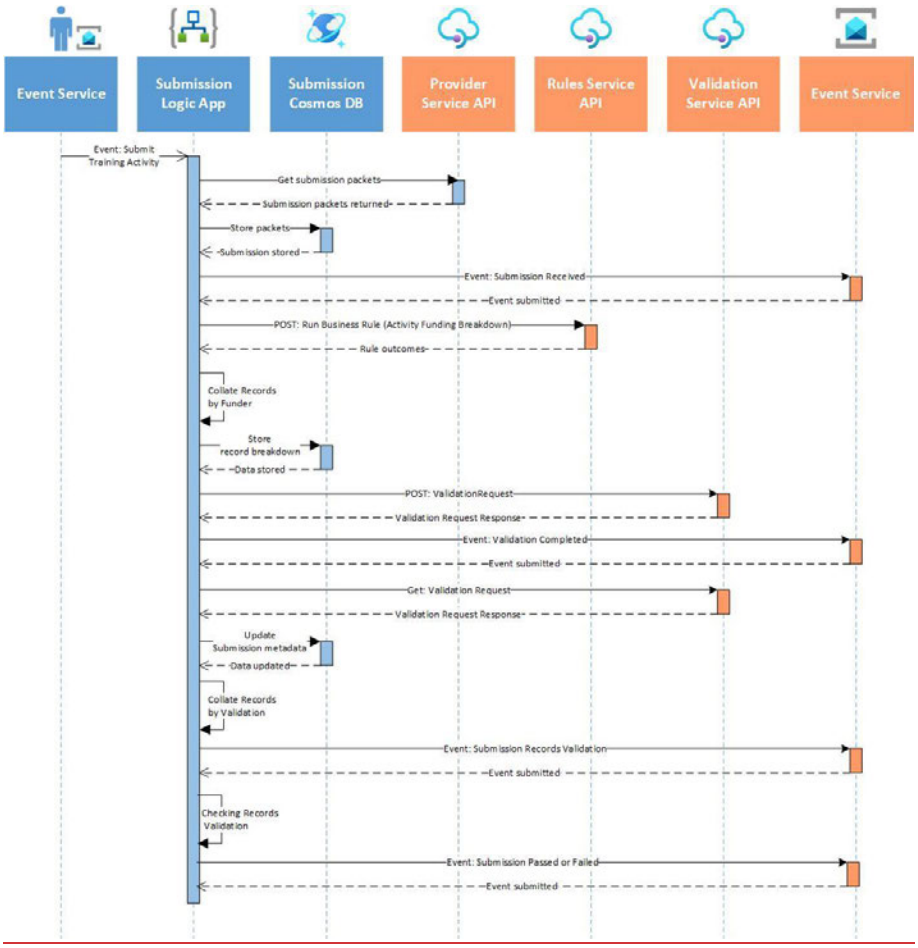
Following will be the business process flow of Submission if the Submission is submitted by STA:

- Submissions containing complete Enrolment information are received from STA.
- The submission request and the submission data are correlated and recorded in the Submission Data Store.
- Data quality and business rules are applied via validation request to the Rules and Validation Service.
- If the submission fails validation, it is rejected, with details of the rejected submission recorded in the Submission Data Store, with control passing back to the STA for their attention.
- After successful validation, a distinction is made between funded and fee-for-service (FFS) activity, based on defined rules.
- FA (Funded Activity) and FFS (Fee For Service) data is released to mastering via a message to the Release Service.

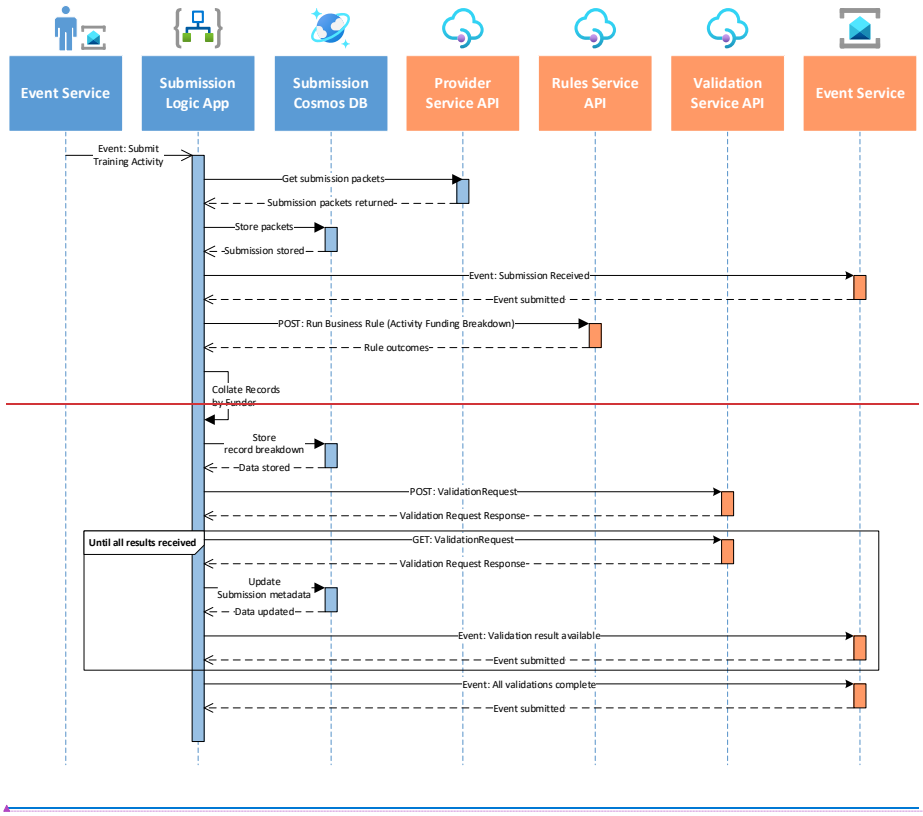
Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 606 of 1295

6 Submission ~~Service~~ Workflow

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 607 of 1295



Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 608 of 1295



Field Code Changed

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 609 of 1295

Field Code Changed

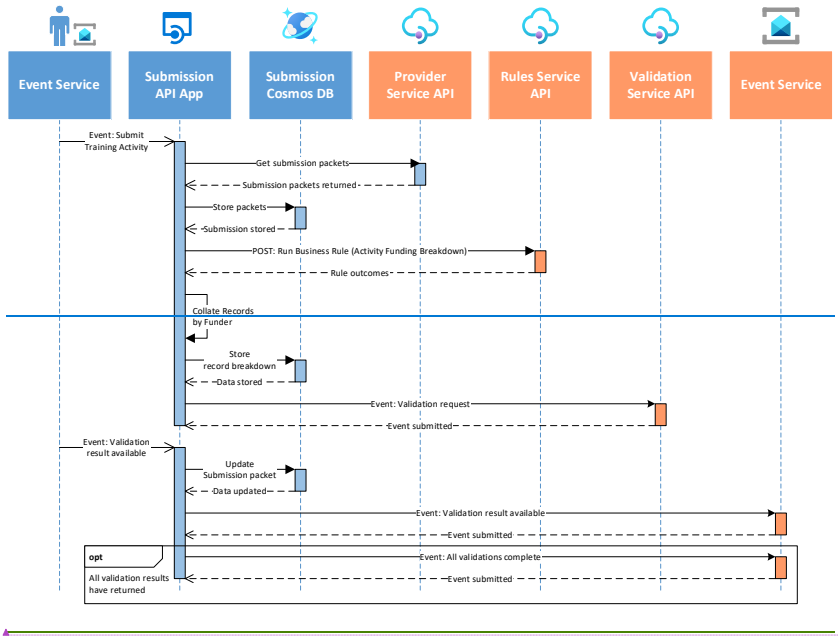


Figure ~~2~~5: Training Activity Submission (D:2)

The ~~s~~Submissions can be made by STARS services by raising a submission event containing a reference to the submission details. ~~It is expected that all~~The submission events will be raised by the Provider Service. The Submission Service will have a Logic ~~a~~App that has a subscription to submission events and orchestrates the processing of submissions. When a submission event ~~comes in~~arrives, the ~~l~~Logic ~~a~~App will request the full submission details from the provider service API and save them in the Submission Service Cosmos DB. ~~There is still more requirements gathering needed to determine how update and delete submissions will be stored.~~ After the details are saved, an event will be raised to confirm that the submission has been received. The process and details around storing data in the Submission Cosmos DB (e.g., updates vs inserts etc) will be updated once Test Data and associated scenarios are made available.

To determine funding arrangements, a request will be made to the Rules Service API. The specific details for what data is needed to be supplied to the rules service and the shape of the data that is returned will be clarified as part of the ~~requirements gathering for the~~ detailed design of the Rules Service. The result of this, will be a clear depiction of which Rulesets need to be run on which records (based on the Funder(s) of each record).~~After any additional processing to determine validation rulesets required for each record within the submission, t~~ These details will be stored in the ~~s~~Submission ~~cosmos~~ DB ~~along with validation statuses of 'pending'.~~

A request will then be made to the ~~v~~Validation ~~s~~Service to validate all packages against their required validation rulesets ~~and the response will be stored in the submission cosmos DB~~. This is an asynchronous process, so each record will be considered 'Pending Validation' at this point, as the validation request has been initiated but no results have been returned yet. The Validation Service

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 610 of 1295

will raise an event of validation completed after validating all records in the submission. ~~-~~The Submission service~~Immediately~~ will send~~,~~ a request ~~will be made~~ to the Validation API to retrieve the status of the Validation Request (including all results). The metadata for the packets will be updated to track progress against any rulesets that have returned results.

~~The Submission Service will continue to poll for new results, until all results have arrived. Note: upon receipt of expected data volumes for STARS, and performance testing in line with these estimates, it's expected that the polling nature of this process will be verified, with potential to move to event-driven processing of validation results.~~

~~If the response from the vValidation sService indicates that all validation is completed or any package, a 'Record Validated' event will be raised so that the sSubmission can be progressed through the pipeline by the appropriate service (e.g., Staging or Release)~~ If the response from the Validation Service indicates that all validation is completed or any package, a "Submission Records Passed Validation" event will be raised so that the Submission can be progressed through the pipeline by the appropriate service (e.g., Release). If all records in the submission have completed validation, a "~~'~~Submission Passed~~Successful~~"~~'~~ or "~~'~~Submission Failed~~Failed~~"~~'~~ event will be raised based on whether all rules passed for all records, or there was at least one error (this event is intended for the Provider Service).

6.1 Additional Context

The funding relationships determined during the process will be 'attached' to all STARS internal communications regarding those funding activity records, and used to guide processing activities along the Submission Pipeline.

For example, Fee for Service training records will only require validation against the National Rules, won't require data enrichment, and won't require explicit Release checking. This will be enabled by adding funding relationships for groups of records within a Submission to Event properties.

Then subscribers to those events can filter specific relationships as required. In this case, the Release Service would have multiple subscriptions for Validation Passed events, each filtering to different Jurisdictions where enrichment and/or explicit release is required. It wouldn't subscribe to any Validation Passed events for Fee for Service records, but the Mastering Service would subscribe to Validation Passed events for Fee For Service records as they can be written to the NCVER Master DB instance once they are validated. ~~Since~~ ~~The~~ Vvalidation ~~will~~can be asynchronous, ~~there will~~ may be a ~~pending response response of pending for some rules. Therefore, the validation service will be polled on a to-be-determined interval to get up to date validation status which will be stored in the submission cosmos DB as it changes. Once all necessary validation is complete for a record, a 'Record Validated' event will be raised so that the submission can be progressed through the pipeline by the appropriate service (eg. Staging or Release). Once all records in the submission have completed validation, a 'Submission Approved or 'Submission Rejected' event will be raised based on whether all rules passed for all records, or there was at least one error (this event is intended for the Provider Service).~~

Formatted: Heading 2,Reset numbering,Numbered Heading 2,h2 main heading,B Sub/Bold,hd2,h2,Header 2,B Sub/Bold1,B Sub/Bold2,B Sub/Bold11,h2 main heading1,h2 main heading2,B Sub/Bold3,B Sub/Bold12,h2 main heading3,B Sub/Bold4,B Sub/Bold13,h2 main heading4,l2,list 2

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 611 of 1295

Since Vvalidation willcan be asynchronous, there willmay be pendinga response of pending for some rules. Therefore, there will be a second Logic App which subscribes to validation events and updates the submission cosmos DB when submission for a record completes. Once all necessary validation is complete for a record, a 'Record Validated' event will be raised so that the submission can be progressed through the pipeline by the appropriate service (eg. Staging or Release). Once all records in the submission have completed validation, a 'Submission Approved' or 'Submission Rejected' event will be raised based on whether all rules passed for all records, or there was at least one error (this event is intended for the Provider Service).

5 Nathan to add details here

Formatted: Highlight

Formatted: Normal

Submission service have the following different workflows:

- Funded Training Activity Submission by Training Provider
- FFS Training Activity Submission by Training Provider
- Training Activity Submission by STAs
- Funded Training Activity Resubmission by Training Provider
- FFS Training Activity Resubmission by Training Provider
- Training Activity Resubmission by STAs
- Get Operations by Submission API

Formatted: No bullets or numbering

Note: Please refer to the Visio file Submission Service Diagrams.vsdx for a full-size representation.

5.1 Funded Training Activity Submission by Training Provider

Formatted: No bullets or numbering

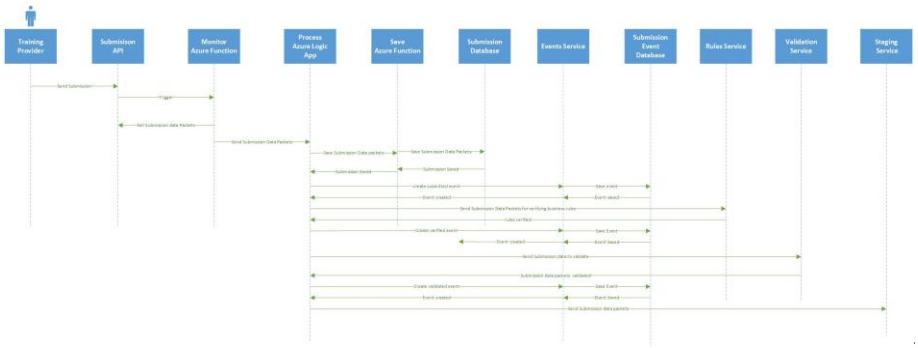


Figure 3: Funded Training Activity Submission by Training Provider Validated (I:1)

The Submission service will receive the submission message from Training providers via the Submission API or via Submission Azure Service Bus queues which will be monitored by Submission Monitor Azure Function. This function will be triggered when Submission Azure Service Bus has received a new message from Training Provider.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 612 of 1295

Submission Monitor Azure Function will trigger Submission Process Azure logic app which will trigger Save Azure Function to save the data packets in Submission database and it will trigger Event service to save the event information in Submission Event database.

Submission process logic app will send a request to Validation service to validate the submission data packets – the Validation Service delegates validation rules processing to the Rules Service.

If submission data packets are validated from Validation service, Submission Process logic app will send the request to Event service to update the submission event information in Submission event database. It will also send the Submission data packets to Staging service Azure Service Bus queue.

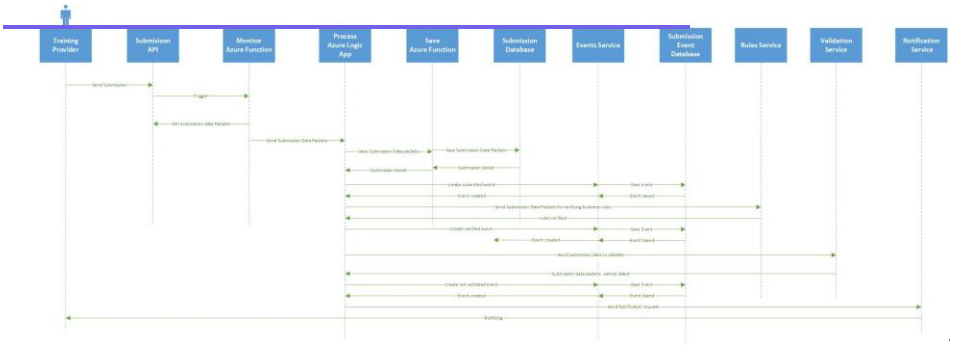


Figure 4: Funded Training Activity Submission by Training Provider Not Validated (I:2)

In case submission data packets send by Submission process logic to the Validation service are not valid, Process Submission logic app will send the request to Notification service to notify the Training Provider. It will trigger Save Azure Function to update the status in Submission Cosmos DB database. Submission Process logic app will send the request to Event Service to update the submission event information in Submission event database.

5.2 FFS Training Activity Submission by Training Provider

Formatted: No bullets or numbering

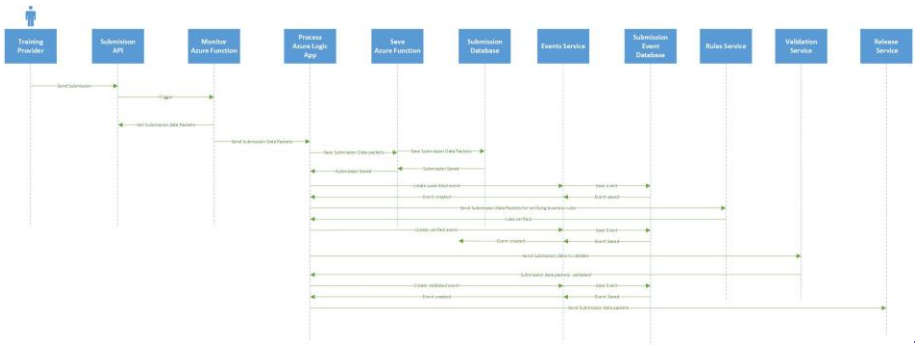


Figure 5: FFS Training Activity Submission by Training Provider Validated (I:3)

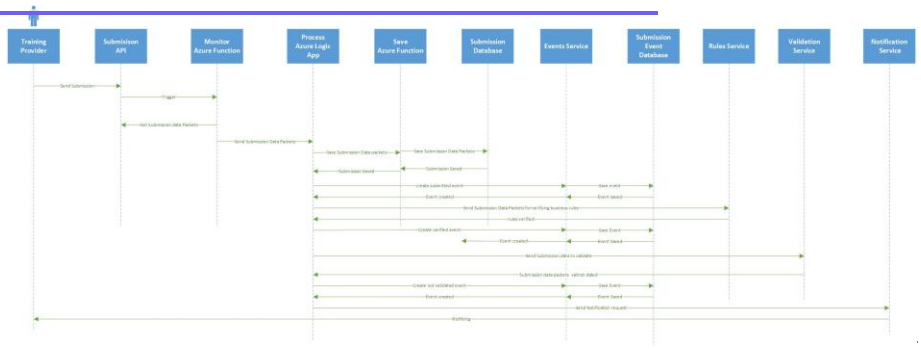
Submission service will receive the submission message from Training providers VIA Submission API in Submission Azure Service Bus queues which will be monitored by Submission Monitor Azure

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 613 of 1295

~~Function and it will get triggered when Submission Azure Service Bus had received a new message from Training Provider.~~

~~The Submission Monitor Azure Function will trigger Submission Process Azure logic app which will trigger Save Azure Function to save the data packets in Submission database and it will trigger Event service to save the event information in Submission Event database. Submission process logic app will send request to Rules service to verify the business rules once verified it will send the request to Validation service to validate the submission data packets.~~

~~If submission data packets are validated from Validation service, Submission Process logic app will send the request to Event service to update the submission event information in Submission Event database. And it will send the Submission data packets to Release service Azure Service Bus.~~

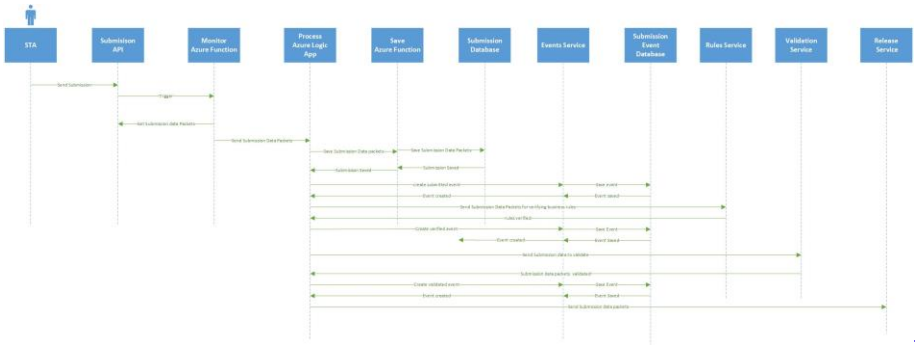


~~**Figure 6: FFS Training Activity Submission by Training Provider Not Validated (I:4)**~~

~~In case submission data packets send by Submission process logic to Rules services or Validation service is not validated, Process Submission logic app will send the request to Notification service to notify the Training Provider. It will trigger Save Azure Function to update the status in Submission Cosmos DB database. Submission Process logic app will send the request to Event Service to update the submission event information in Submission Event database.~~

~~5.3 Training Activity Submission by STA~~

Formatted: No bullets or numbering



~~**Figure 7: Training Activity Submission by STA Validated (I:5)**~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 614 of 1295

Submission service will receive the submission message from STA VIA Submission API in Submission Azure Service Bus queues which will be monitored by Submission Monitor Azure Function and it will be triggered when Submission Azure Service Bus has received a new message from an STA.

Submission Monitor Azure Function will trigger Submission Process Azure logic app which will trigger Process Azure Function to save the data packets in Submission database and it will trigger Event service to save the submission event information in Submission Event database. Submission process logic app will send request to Rules service to verify the business rules once verified it will send the request to Validation service to validate the submission data packets. If submission data packets are validated from Validation service, Submission Process logic app will send the request to Event service to update the resubmission event information in Submission event database. It will send the Submission data packets to Release service Azure Service Bus queue(s).

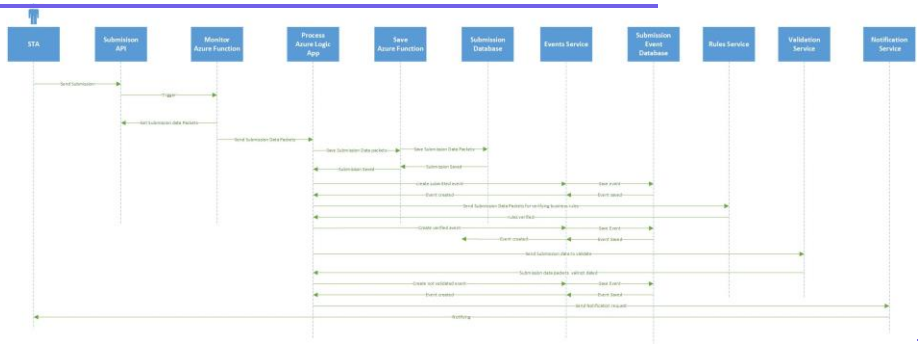


Figure 8: Training Activity Submission by STA Not Validated (I:6)

In case submission data packets send by Submission process logic to Rules or Validation service is not validated, Process Submission logic app will send the request to Notification service to notify the STA. It will trigger Save Azure Function to update the status in Submission Cosmos DB database. Submission Process logic app will send the request to Event Service to update the submission event information in Submission event database.

5.4 Funded Training Activity Resubmission by Training Provider

Formatted: No bullets or numbering

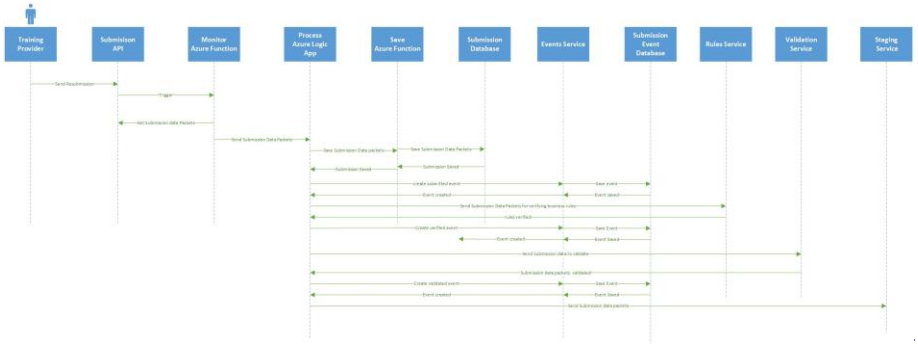


Figure 9: Funded Training Activity Resubmission by Training Provider Validated (I:7)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 615 of 1295

Submission service will receive the resubmitted submission message from Training providers via the Submission API or Submission Azure Service Bus queues which will be monitored by Submission Monitor Azure Function, which will be triggered when Submission Azure Service Bus has received a new message from Training Provider.

Submission Monitor Azure Function will trigger Submission Process Azure logic app, which will trigger Save Azure Function to save the data packets in Submission database. It will trigger Event service to save the event information in Submission Event database.

Submission process logic app will send request to Rules service to verify the business rules once verified it will send the request to Validation service to validate the submission data packets. If submission data packets are validated from Validation service, Submission Process logic app will send the request to Event service to update the submission event information in Submission event database, then will send the Submission data packets to Staging service Azure Service Bus.

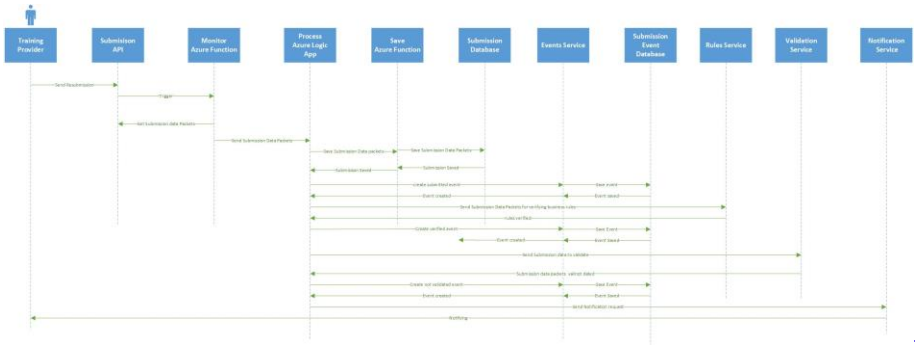
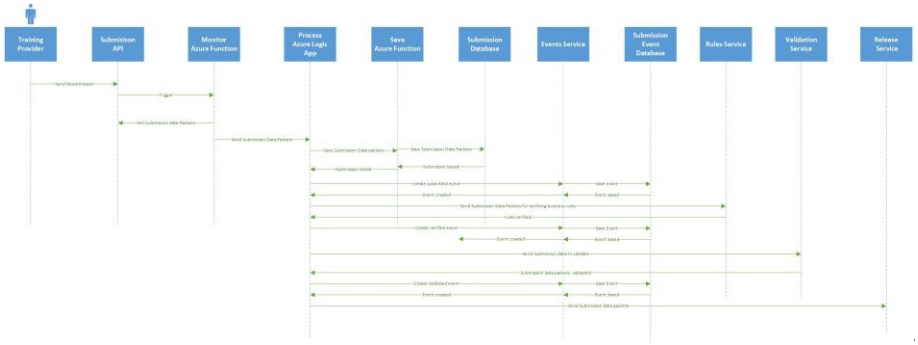


Figure 10: Funded Training Activity Resubmission by Training Provider Not Validated (I:8)

In case resubmitted submission data packets send by Submission process logic to Validation service is not validated, Process Submission logic app will send the request to Notification service to notify the Training Provider. It will trigger Save Azure Function to update the status in Submission database. Submission Process logic app will send the request to Event Service to update the submission event information in Submission event database.

5.5 FFS Training Activity Resubmission by Training Provider

Formatted: No bullets or numbering



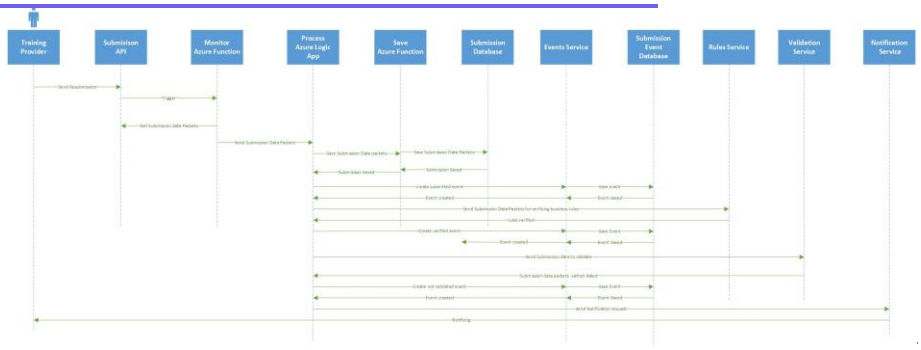
Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 616 of 1295

~~**Figure 11: Funded Training Activity Resubmission by Training Provider Validated (I:9)**~~

~~Submission service will receive the resubmitted submission message from Training providers via Submission API or Submission Azure Service Bus queues monitored by Submission Monitor Azure Function, which will get triggered when Submission Azure Service Bus had received a new message.~~

~~Submission Monitor Azure Function will trigger Submission Process Azure logic app which will trigger Save Azure Function to save the data packets in Submission database, and trigger Event service to save the event information in Submission Event database.~~

~~Submission process logic app will send request to Validation service to validate the submission data packets. If submission data packets are validated from Validation service, Submission Process logic app will send the request to Event service to update the submission event information in Submission Event database. It will send the Submission data packets to Release service Azure Service Bus.~~

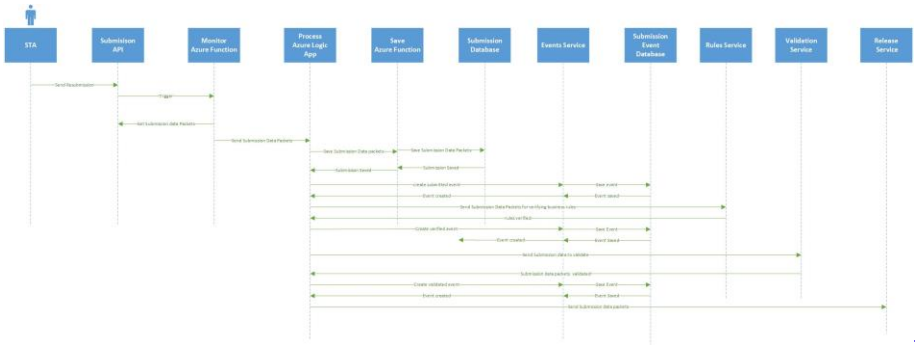


~~**Figure 12: Funded Training Activity Resubmission by Training Provider Not Validated (I:10)**~~

~~In case submission data packets send by Submission process logic to Rules services or Validation service is not validated, Process Submission logic app will send the request to Notification service to notify the Training Provider. And it will trigger Save Azure Function to update the status in Submission Cosmos DB database. Submission Process logic app will send the request to Event Service to update the submission event information in Submission Event database.~~

~~5.6 Training Activity Resubmission by STA~~

Formatted: No bullets or numbering



~~**Figure 13: Training Activity Resubmission by STA Validated (I:11)**~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 617 of 1295

Submission service will receive the resubmitted submission message from STA via Submission API in Submission Azure Service Bus queues which will be monitored by Submission Monitor Azure Function, triggered when Submission Azure Service Bus has received a new message from STA.

Submission Monitor Azure Function will trigger Submission Process Azure logic app which will trigger Process Azure Function to save the data packets in Submission database and it will trigger Event service to save the submission event information in Submission Event database.

Submission process logic app will send request to Rules service to verify the business rules once verified it will send the request to Validation service to validate the submission data packets. If submission data packets are validated from Validation service, Submission Process logic app will send the request to Event service to update the resubmission event information in Submission event database. It will send the Submission data packets to Release service Azure Service Bus.

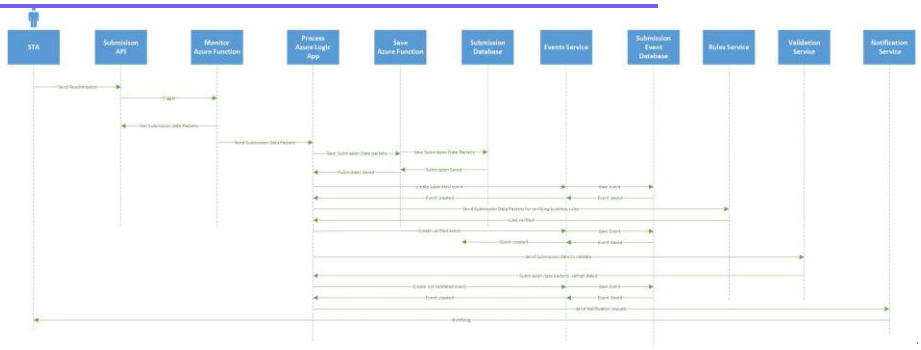


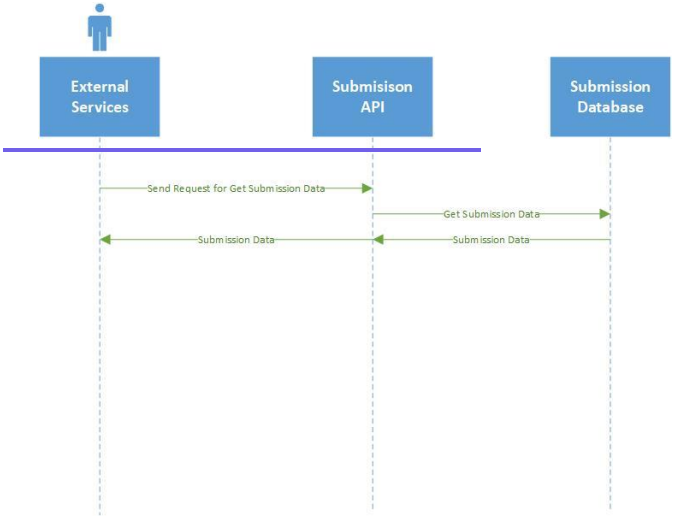
Figure 14: Training Activity Resubmission by STA Not Validated (I:12)

In case submission data packets send by Submission process logic to Rules or Validation service is not validated, Process Submission logic app will send the request to Notification service to notify the STA. It will trigger Save Azure Function to update the status in Submission Cosmos DB database. Submission Process logic app will send the request to Event Service to update the submission event information in Submission event database.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 618 of 1295

~~5.7 Get Operations by Submission API~~

Formatted: No bullets or numbering



~~Figure 15: Get Operations by Submission API (I:13)~~

~~All external services if required submission data will get the submission data from Submission API. Submission API will be exposing API end points on HTTP and it will be based on Azure Function.~~

~~When an external service requires any submission data, these external services will send the request to Submission API end points on HTTP, which will go to the Submission API Azure Function to Cosmos database to get the submission information and send the response back to external services in JSON.~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 619 of 1295

7 Submission Events

The ~~s~~Submissions are triggered in the Submission Service by raising a "Submission" event in the Event Service. The Submission~~Submission service will be raising events on different stages of Submission Service. Submission service depends upon Event service to save event information in Submission Event database. Events raised will contain the information regarding the events which will be passed to Event service to save in Submission Event database.~~ Service will also raise various events in the Event Service at various stages of submission. The Event Service is responsible for saving the events (for more information on the Event Service, see the Event Service detailed design document).

7.1 Raised Events

Table 111: Events raised by the Submission Service

Event ~~Code~~Type	Description
Submission Received	This event is raised after successfully retrieving a submission but before the submission is saved or any processing is done on it. It is expected that the Provider Service will subscribe to these events Properties: • Submission Id (GUID) • Record Count (Integer)
Submission Records Passed Validation	This event is raised by Submission service after the records within a submission has passed all validation. It is expected Release and Master services will subscribe to these events. Properties: • Submission Id (Guid) • Record Id (Guid) • Is Funded By {Jurisdiction} (Boolean) ○ Each Jurisdiction (including Commonwealth) listed • Submitter Id (Guid) • Provider Id (Guid)

Formatted: Outline numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0 cm + Indent at: 0 cm

Formatted: Caption, Keep with next

Formatted Table

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 620 of 1295

Submission Records Failed Validation	This event is raised by Submission service after the records within a submission has failed validation. It is expected Provider and Notification services will subscribe to these events. Properties: • Submission Id (Guid) • Record Id (Guid) • Submitter Id (Guid) • Provider Id (Guid)
~~Record Validated~~	~~This event is raised after a record within a submission has passed all validation. It is expected that the Staging, Release and Mastering services will subscribe to these events~~ ~~Properties TBC:~~ ~~• Submission Id (Guid)~~ ~~• Record Id (Guid)~~ ~~• Is Funded By {Jurisdiction} (Boolean)~~ ~~• Each Jurisdiction (including Commonwealth) listed~~ ~~• Submitter Id (Guid)~~ ~~• Provider Id (Guid)~~
~~SubmissionValidationComplete~~Submission Failed~~Rejected~~	This event is raised after all records within a submission have completed validation if any of the of the validations failed. It is expected that the Provider Service will subscribe to these events Properties: • Submission Id (GUID) • Error Count (Integer) • Warning Count (Integer) • Success Count (Integer)

Formatted

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 621 of 1295

Submission Passed~~Approved~~	This event is raised after all records within a submission have completed validation if all the validations passed. It is expected that the Provider Service will subscribe to these events Properties: • Submission Id (GUID) • Record Count (Integer)

Formatted Table

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 622 of 1295

7.2 Consumed Events

Table 222: Events Consumed by Submission Service

Event Code	Description
New Submission	Expected to be raised by the Provider Service. The Submission Processing logic app will subscribe to submission events to trigger the processing of a submission Properties: • Submission Id (GUID) • Submission URI (URI) • Submitter Id (GUID) • Training Organisation Id (GUID) • Record Count (Positive Integer)
Validation Completed	This event is raised by Validation service after the submission records have been validated. The Submission service will subscribe to this event. Properties: • Submission Id (Guid) • Submitter Id (Guid) • Provider Id (Guid)

~~Following will be different attributes of the Events entity:~~

- ~~EventId~~
- ~~EventType~~
- ~~EventStatus~~
- ~~Identifier~~

~~Following is an example of Event entity JSON structure~~

```
{
        "EventId": "6F9619FF-8B86-D011-B42D-00C04FC964FF",
```

Formatted: Caption, Keep with next

Formatted Table

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 623 of 1295

~~"EventType ":"Submission",~~

~~"EventStatus":"Submitted",~~

~~"Identifier":"6F9619FF-8B86-D011-B42D-00C04FC964EE"~~

~~}~~

~~There will be two types of Submission Event Types:~~

- ~~Submission~~
- ~~Resubmission~~

~~Following will be different types of Submission Events Status:~~

- ~~Submitted~~
- ~~Verified~~
- ~~Not Verified~~
- ~~Validated~~
- ~~Not Validated~~

~~Following are different Submission Events raised in Submission service:~~

~~Submission Event~~

~~5.8 A submission is triggered by raising a submission event. The submission processing logic app will subscribe to submission events~~

Formatted: Normal

~~Submission Submitted Event~~

~~Trigger:Training Provider or STA send submission data packets.~~

~~Inputs: Submission data packets~~

~~Outputs:~~

~~Submission gets submitted successfully and event status is set to submitted~~

~~Error message if submission is not submitted~~

~~Description:~~

~~When Training provider or STA send submission data to submission service, Submission service will save the submission data packets in Submission database.~~

~~If the submission is saved in Submission database, request is sent to Event service to save the event information in Submission event database and set the event status to "Submitted".~~

~~If Submission is not saved Training Provide or STA will be notified with the error message.~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 624 of 1295

~~5.9 Resubmission Event~~

~~Resubmission Submitted Event~~

~~Trigger: Training Provider or STA send submission data packets.~~

~~Inputs: Submission data packets~~

~~Outputs:~~

~~Resubmission gets submitted successfully and event is recorded to submitted~~

~~Error message if submission is not submitted~~

~~Description:~~

~~When Training provider or STA send resubmission data to submission service, Submission service will save the resubmission data packets in Submission database.~~

~~If the resubmission is saved in Submission database, request is sent to Event service to save the event information in Submission event database and set the event status to "Submitted".~~

~~If resubmission is not saved Training Provide or STA will be notified with the error message.~~

~~5.10 Submission Verification Event~~

~~Submission Verification Event~~

~~Trigger: When submission data packets are saved in Submission database and its Status in Submission Event database is set to " Submitted".~~

~~Inputs: Submission data packets~~

~~Outputs:~~

~~Submission gets verifies by Rules service successfully and event status is set to verified if submission is not verified by Rules service it will set the submission status to Not Verified and will notify to Training Provider or STA.~~

~~Description:~~

~~When submission is sent by Training provider or STA becomes in submitted status, Submission service will send the submission data packets to Rules service for verification.~~

~~If the submission has been verified by Rules service, request is sent to Event service to update the event information in Submission event database and set the event status to "Verified".~~

~~If submission is not verified by Rules service, request will be sent to Event service to update the event status to "Not Verified". Training Provide or STA will be notified.~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 625 of 1295

~~5.11 Submission Validation Event~~

~~Submission Validation Event~~

~~Trigger: When submission gets verified from Rules service and its Status in Submission Event database is set to" Verified".~~

~~Inputs: Submission data packets~~

~~Outputs:~~

~~Submission gets validated by Validation service successfully and event status is set to validated and Submission message are passed to Staging or Release service. If submission is not validated by Validation service it will set the submission status to Not Validated and will notify to Training Provider or STA.~~

~~Description:~~

~~When submission send by Training provider or STA becomes in verified status,~~

~~Submission service will send the submission data packets to Validation service for validation.~~

~~If the submission has been validated by Validation service, request is sent to Event service to update the event information in Submission event database and set the event status to "Validated" and send the submission message to Staging or Release service.~~

~~If submission is not validated by Validation service, request will be sent to Event service to update the event status to "Not Validated". Training Provide or STA will be notified.~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 626 of 1295

~~6~~8 API Design

The Submission API will be implemented in Azure API Management, with the backend implemented as one or more Azure API Apps. API Apps will support the implementation of OData APIs to provide consistent API interactions between STARS Services and for consumption by external services.

The Submission Service API will be presented at the following path:

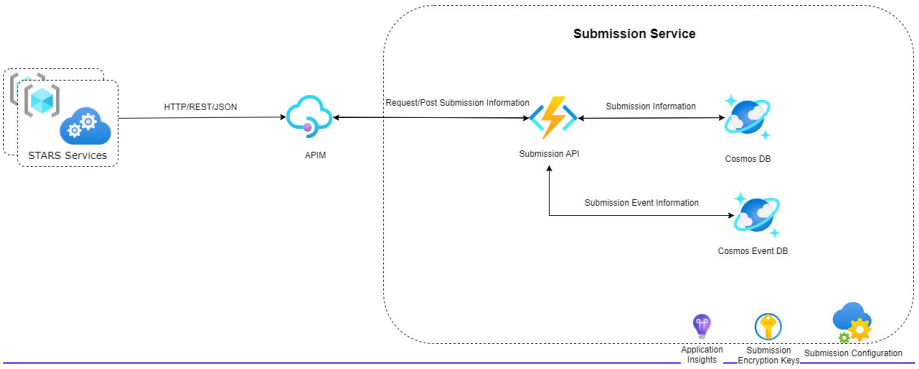
<STARS Base URL>/submissionService/

~~Submission API will be REST API. REST (Representational State Transfer) is a software architectural style that defines a set of constraints and properties for creating web services. RESTful APIs use HTTP requests to POST (create), PUT (update), GET (read), and DELETE (delete) data. REST is a lightweight alternative to other web service technologies like SOAP and is often used for building APIs for web-based applications.~~

~~Submission API will be using to HTTP requests to GET, POST, PUT, PATCH and DELETE data. The service will follow standard REST standards for requests. Submission API will receive and sent the data in JSON format conforming to JSON Schema standards.~~

~~The API will conform to the STARS Design Principles and the STARS API Design Document. The endpoints described below will be standard CRUD REST endpoints. See the STARS API Design Document for success / failure status codes and expected response headers. See Technical Design Patterns section below for general endpoint descriptions.~~

~~Filtering (including free-text search) will be handled using OData.~~



~~**Figure 16: Submission API Design Overview**~~

~~6.1 Submission API~~

~~The Submission API is an HTTP-triggered function app. Submission data management and query operations will be mapped to relevant function app endpoints. The API will be factored around VET-IS data packet entities (e.g., students, enrolments, delivery locations).~~

~~6.2 Submission database~~

~~The Submission Database is an Azure Cosmos DB database that stores submission data in VET-IS Data Packet Schema.~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 627 of 1295

6.3 Submission Event database

The Submission Event Database is an Azure Cosmos DB database instance that stores all submission service, data management and lifecycle events.

6.4 Submission Encryption Keys

Submission data will be encrypted at rest, encryption keys to be held in Azure key vault.

6.5 Submission Config

Any RG specific config, including for function apps, to be held in a common Azure app config.

6.68.1 Submission API Endpd Points

Following are different API End-points supported by the Submission API. Some of these operations are listed as 'General', meaning there will be an instance of that endpoint for each Data Packet in the VET-IS Collection Specification (represented as {activityType} in the path structure below).

Table 333: Submission API Endpoints

Formatted: Caption, Keep with next

Formatted Table

API Endpoint Name	ActionBase Definition	Comments pPath	
	SaveSubmission	POST	Save Submission API end point will get submission and resubmission data packets, which will be saved in Submission database.
Get Submission by ID	GETRead Endpoint - Detail	GET <serviceurl>/submission/[id]	This API end point will get the submission details against the submission identifier.
Get Submissions	Read Endpoint – List	GET <serviceurl>/submission	
	GetSubmissionStatus	GET	This API end point will get the submission status from Submission event database.
	GetSubmissionHistoricalData	GET	This API end point will get the list of submission historical data.
Get List of Activity Records (General)GetStudent	GETRead Endpoint - List	GET <serivceurl>/activity/{activityType}/[id]	This API end point will get the list of students with matching criteria.
Get Activity Record by ID (General)GetStudentById	GETRead Endpoint -Detail	GET <serviceurl>/activity/{activityType}	This API end point will get the student details against the student identifier.
GetStudentEnrolments	GET	This API end point will get the list of student enrolments.	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 628 of 1295

GetTrainingProvider	GET	This API end point will get the list of training provider.
GetTrainingProvider	GET	This API end point will return the training provider details against the training provider identifier.

6.6.1 Save Submission

API Operation	SaveSubmission
Operation Type	POST
Input Parameter	Submission data packets
Data Format	Request and Response will be in JSON Format
Description	Training Provider or STA will submit the submission data packets VIA Provider Service or STA Gateway Service to SaveSubmission end point to save the submission data in Submission database. This operation will be working as an upsert it will be inserting the new data and updating the existing data in the Submission database
Response	SaveSubmission will send back the Submission Identifier and status header 200. In case of error it will send back the error status header with the information message.

6.6.2 Get Submission

API Operation	GetSubmission
Operation Type	GET
Input Parameter	Submission Identifier
Data Format	Request and Response be in JSON Format
Description	This API end point will return the submission from Submission database. It will get submission identifier in input parameter and will get the status of submission from Submission event database.
Response	It will return the Submission details in JSON format and header status will be 200. In case if submission details are not found it will return header status 404 and in case of error it will return information message along with error header status.

6.6.3 Get Submission Status

API Operation	GetSubmissionStatus

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 629 of 1295

Operation Type	GET
Input Parameter	Submission Identifier
Data Format	Request and Response will be in JSON Format
Description	This API end point will return the status of submission from Submission Event database. It will get submission identifier in input parameter and will get the status of submission from Submission event database.
Response	It will return the Submission Status in JSON format and header status will be 200. In case if submission is not found it will return header status 404 and in case of error it will return information message along with error header status.

6.6.4 Get Submission Historical Data

API Operation	GetSubmissionHistoricalData
Operation Type	GET
Input Parameter	Submission Identifier
Data Format	Request and Response will be in JSON Format
Description	This API end point will return the list of historical submissions against the submission Identifier from Submission database and Submission Events database for the status. Submission historical data will be returned in JSON format.
Response	It will return the list of historical submission data in JSON format and header status will be 200. In case if submission historical data is not found it will return header status 404 and in case of error it will return information message along with error header status.

6.6.5 Get Student

API Operation	GetStudent
Operation Type	GET
Input Parameter	First Name, Last Name, Date of Birth, Training Provider
Data Format	Request and Response will be in JSON Format
Description	This API end point will return the student list with the matching criteria details against the student first name, last name, date of birth and training provider from Submission database and it will be returned in JSON format.
Response	It will return the student list in JSON format and header status will be 200. In case if student is not found it will return header status 404 and

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 630 of 1295

	Student not found message. In case of error, it will return information message along with error header status.

6.6.6 Get Student By Id

API Operation	GetStudentById
Operation Type	GET
Input Parameter	Student Identifier
Data Format	Request and Response will be in JSON Format
Description	This API end point will return the student information against the student Identifier from Submission database and it will be returned in JSON format.
Response	It will return the student information in JSON format and header status will be 200. In case if student is not found it will return header status 404 and Student not found message. In case of error, it will return information message along with error header status.

6.6.7 Get Student Enrolments

API Operation	GetStudentEnrolments
Operation Type	GET
Input Parameter	Student Identifier
Data Format	Request and Response will be in JSON Format
Description	This API end point will return the student enrolments detail list against the student identifier from Submission database and it will be returned in JSON format.
Response	It will return the student enrolments detail list in JSON format and header status will be 200. In case if student enrolment is not found it will return header status 404 and Student enrolment not found message. In case of error it will return information message along with error header status.

6.6.8 Get Training Provider

API Operation	GetTrainingProvider
Operation Type	GET
Input Parameter	Training Provider Name, State
Data Format	Request and Response will be in JSON Format

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 631 of 1295

~~Description~~	~~This API end point will return the training provider list again the training provider name or State Submission database and it will be returned in JSON format.~~
~~Response~~	~~It will return the training provider list in JSON format and header status will be 200. In case if student enrolment is not found it will return header status 404 and Student enrolment not found message. In case of error it will return information message along with error header status.~~

~~6.6.9 Get Training Provider By Id~~

~~API Operation~~	~~GetTrainingProviderById~~
~~Operation Type~~	~~GET~~
~~Input Parameter~~	~~Training Provider Identifier~~
~~Data Format~~	~~Request and Response will be in JSON Format~~
~~Description~~	~~This API end point will return the training provider information against the training provider Identifier from Submission database and it will be returned in JSON format.~~
~~Response~~	~~It will return the training provider information in JSON format and header status will be 200. In case if student enrolment is not found it will return header status 404 and Student enrolment not found message. In case of error it will return information message along with error header status.~~

8.1.1 Get List of Submissions

A list of Submissions can be retrieved through this operation. OData filters and selections can be used to manage the results.

8.1.2 Get Submission by ID

This operation retrieves a single submission using its submission id. OData selection can be used to expand which data is returned. For example, this endpoint could be used to fetch the current validation status of a submission.

8.1.3 Get List of Activity Records (General)

As a 'General' operation, the implementation of this will cover all Training Activity records defined in the Collection Specification.

A list of Training Activity Records that have been submitted can be retrieved through this operation. OData filters and selections can be used to manage the results.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 632 of 1295

8.1.4 Get Activity Record by ID (General)

As a 'General' operation, the implementation of this will cover all Training Activity records defined in the Collection Specification.

This operation returns a single Training Activity record by its STARS identifier.

Please refer to the access management section in the document for details about the access of the data from STARS database by the Submission Service API.

Formatted: Outline numbered + Level: 3 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0 cm + Indent at: 0 cm

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 633 of 1295

~~7~~9 Technical Design Patterns

Submission service will follow SOA (Service Oriented Architecture) design pattern like other STARS services. Loose coupling will be achieved using message- and event-driven architectures, primarily using Azure Service Bus. Message and event payloads conforming to several standardised information JSON-based schemas.

Many services present Application Programming Interfaces (API), typically as HTTP/REST/JSON based endpoints. The general API design~~s~~ principles and details are presented in a separate document, please refer to API Design D.0.1.docx[3]. API-based services (both internal and external) will be managed through Azure API Management. ~~For external-facing services STARS will utilise Azure Application Gateway as an intermediary between the APIs and their clients.~~The APIs will conform to the OData standard, to allow consistent behaviour across all STARS APIs. This approach is model-based, so allows for the definition of the STARS Collection Model (the physical implementation of the VET-IS Data Packet Collection Specification) to provide the implementation for the API endpoints, filters and relationships.

9.1 General Endpoint Descriptions

9.1.1 Read Endpoint – List

A GET endpoint that will return a list of entities with optional filtering. OData will be used for filtering and paginating results. Client-driven paging will be used for pagination.

9.1.1.1 Request Parameters

The parameters are used to control pagination and to filter the list of results.

Table ~~44~~4: List Endpoint Query Parameters

Formatted: Caption, Keep with next

Query parameter	Comments
$top	Limits the number of results returned for this query.
$skip	The number of records to "skip over" when retrieving results. Used for pagination. If not included, default to 20.
$format=json	How the format of the response data. Should be set to "json".
$inlinecount=allpages	Ensures odata.count is included in the response with the total number of relevant results, regardless of pagination. Set to "allpages".
(filters)	Varies per endpoint. Values used to filter results to return.

[3] Please note the API design document will evolve as the solution is implemented. At the time of authoring, the design document lays out the API principles and standards to be followed during implementation.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 634 of 1295

9.1.1.2 Request Body

None.

9.1.1.3 Response Body

A JSON response is returned, with describing the number of results and how to retrieve adjacent pagination pages. The page of results is also returned.

Table 555: List Endpoint Response Fields

Formatted: Caption, Keep with next

Field	Comments
odata.count	Total number of results (regardless of pagination).
Value	List of relevant entities. Record structure depends on individual endpoints.

```
{
  "odata.count": 155,
  "value": [
    {
      "recordID": 101,
      "field1": "value"
    }]}
```

9.1.2 Read Endpoint – Detail

A GET endpoint for retrieving a specific record.

9.1.2.1 Request Parameters

The parameters are used to control pagination and to filter the list of results.

Table 666: Read Endpoint Query Parameters

Formatted: Caption, Keep with next

Query parameter	Comments
~~i~~id	The unique identifier for the desired record.

9.1.2.2 Request Body

None.

9.1.2.3 Response Body

A JSON object describing the relevant record. The fields will depend on the record relevant to each specific endpoint.

```
{
  "recordID": 101,
  "field1": "value"
}
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 635 of 1295

8

9.2 Error Handling

Error handling is an important aspect of building robust and reliable applications.

Following are some of the best practices of error handling that will be used in Submission Service:

- **Try/Catch Blocks:** Critical code sections will be wrapped in try/catch blocks to handle exceptions gracefully. In case of an error, the catch block will handle the error, and the error will be logged.
- **Log Errors:** Logging is an essential part of error handling. Submission service will be using Event, Logging and Audit service for logging the errors and exception.
- **Retry Policies:** In case of transient errors (such as network timeouts) in Submission Service Logic Apps and Azure functions, retry policies will be used to retry the failed operation.
- **Monitor App:** Azure Application Insights will be used to monitor app's health and performance.
- **Custom Error Responses:** In case of an error, Submission Service API will return a custom error response to the client. This can include HTTP error codes, such as 404 or 500, and a custom error message that explains what went wrong.

Formatted: Font: Bold

Formatted: List Paragraph,NFP GP Bulleted List,TOC style,lp1,Bullet OSM,Proposal Bullet List,List Paragraph1,Recommendation,List Paragraph11,List Paragraph111,L,F5 List Paragraph,Dot pt,CV text,Medium Grid 1 - Accent 21,Numbered Paragraph,Bullets,NAST Quote,lp11,n, Bulleted + Level: 1 + Aligned at: 0.63 cm + Indent at: 1.27 cm

Formatted: Font: Bold

Formatted: Font: Bold

Formatted: Font: Bold

Formatted: Font: Bold

9.2.1 Error Handling Submission API

The Submission API will be using try catch block to handle the unexpected errors and exception gracefully. Exceptions caught in catch block will be logged using Logging Service, details about logging service can be found in detail design document of Event, Logging and Audit service. In case of an error Submission Service API will return a custom error response to the client. This will include HTTP error codes, such as 404 or 500, detail about API design and HTTP status codes returned from API can be found in API Design - V.1.0.

9.2.2 Error Handling in Submission Logic Apps and Azure Function

The Submission Logic Apps and Azure functions will be following all the best practices mentioned above for handling the error. Try Catch block will be used in the code of Azure functions to handle the unexpected behaviour. Exception will be logged using Logging service, details about logging service can be found in detail design document of Event, Logging and Audit service. Azure Application Insights will be used to monitor the logic apps and azure functions.

Formatted: Font: Not Bold

Formatted: Caption

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 636 of 1295

The Logic Apps and Azure functions have a useful feature of retry policies to handle the transient errors. Retry policies can help you improve the reliability and availability of your application by automatically retrying failed actions or functions. Instead of returning an error to the user or terminating the application, the application will retry the failed operation according to the retry policy.

In Azure Functions, retry policies can be configured using the host.json file. And in logic apps retry policies are configure using the "Control" action.

Following are the retry policies options available in Logic Apps and Azure Functions:

Retry Count: Defines how many times a failed operation should be retried before giving up.

Retry Interval: Defines the interval between retry attempts.

Retry Strategy: Defines the retry strategy to use, such as linear or exponential backoff.

Following is an example of exponential backoff retry strategy in host.json file of azure function:

```
{
  "version": "2.0",
  "extensions": {
    "queues": {
      "retry": {
        "strategy": "exponentialBackoff",
        "maxRetryCount": 3,
        "delayInterval": "00:00:05"
      }
    }
  }
}
```

Following is an example of a retry policy in Logic Apps:

```
"RetryPolicy": {
    "type": "Exponential",
    "maximumRetryCount": 3,
```

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Normal

Formatted: Font: Bold

Formatted: Font: 11 pt

Formatted: Font: Bold

Formatted: Font: 11 pt

Formatted: Font: Bold

Formatted: Font: 11 pt

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 637 of 1295

```
  "minimumInterval": "PT5S"
}
```

Please refer to Ssection 4 Feature Overview 4: Feature Overview4.8 Overview of Submission Service for detail design overview of the Submission service.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 638 of 1295

~~9~~10 Data

Submission data packets submitted by training providers or STA will be based on VET-IS data standard.

All The STARS services will be utilising the data based on new VET Information Standard (VET-IS), which replaces the AVETMISS 8 information standard with a modern fit-for-purpose and backwards-compatible. The Data Packet types currently defined in VET-IS R1 are:

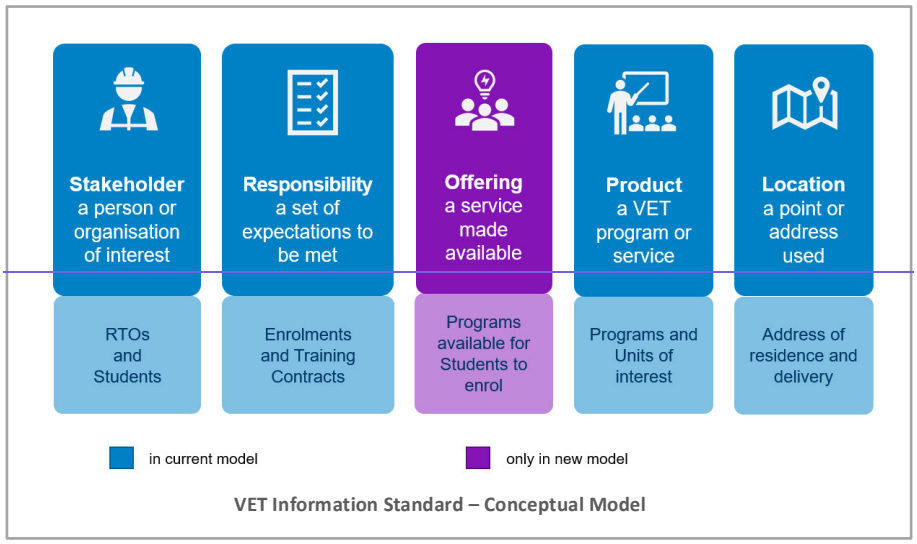
- Government Packet
- Training Organisation Packet
- Training Organisation Location Packet
- Program Packet
- Subject Packet
- Offering Packet
- Delivery Location Packet
- Student Packet
- Student Location Packet
- Program Enrolment Packet
- Subject Enrolment Packet

Details of these data packet schemas can be found in the VET-IS Interchange Specification in GovTeams

~~9.1 VET-IS Conceptual Model~~

~~The VET-IS aims to achieve a higher level of flexibility and extensibility that AVETMISS. It does this by abstracting people, organisations, enrolments, subjects, etc.~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 639 of 1295



~~Figure 17: VET IS Conceptual Model~~

~~Stakeholder: representing people, organisations. NB: employers will be covered in future releases.~~

~~Location: representing addresses, virtual spaces.~~

~~Product: representing programs, subjects.~~

~~Offering: representing a product offered by a training organisation.~~

~~Responsibility: representing many things, including enrolments, training contracts, financial arrangements. Responsibilities are between Stakeholders and address obligations between them. For example, an enrolment is a responsibility between a training organisation and student.~~

~~Event (representing a business event – currently not in R1)~~

~~The logical model (below) and physical model (described in the Solution Design) give effect and materialise these concepts within the STARS Master Database.~~

~~9.2 VET IS Logical Schema~~

~~The intention of the VET IS Logical Model is to flexibly represent the many types of conceptual entities in the VET sector, without having to model them in (very many) separate tables.~~

~~The VET IS Logical Schema Release 1 is a normalised relational schema that models the following primary logic entities and their relationships:~~

~~Stakeholder (representing people, organisations. NB: employers will be covered in future releases)~~

~~Location (representing addresses, virtual spaces)~~

~~Product (representing programs, subjects)~~

~~Offering (representing a product offered by a training organisation)~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 640 of 1295

Responsibility (representing many things, including enrolments, training contracts, financial arrangements)

Event (representing a business event, but not lifecycle events – currently not in R1)

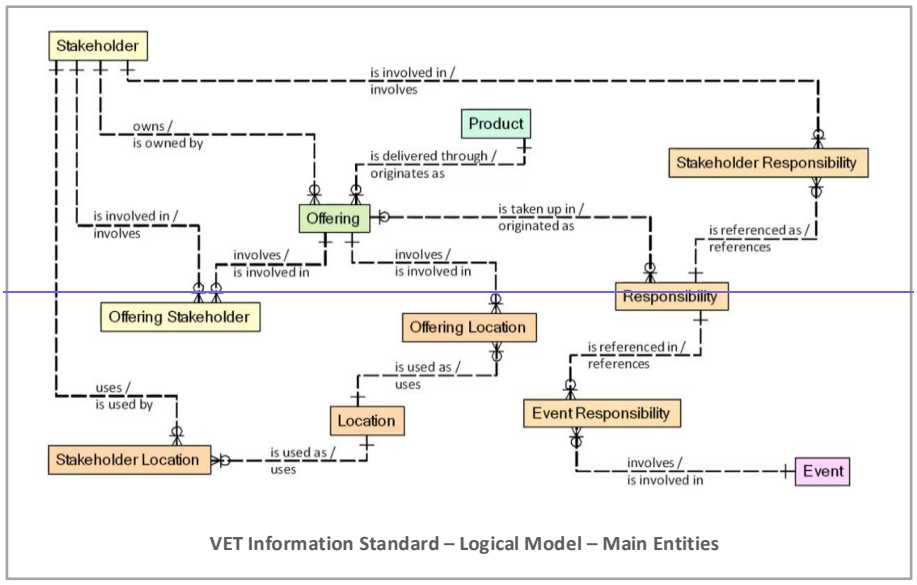


Figure 18: VET-IS Logical Model

Figure 18: VET-IS Logical Model presents a high-level view of the main entities in the logical model. Detailed logical schemas are presented in Appendix A: 'VET-IS Logical Model Entity Relationship Diagrams'.

9.3 VET-IS Data Packet Schema

The VET-IS Data Packet Schema forms both a logical information model and the basis for implementing data interchange protocols between STARS and external systems, including Student Management Systems (SMS). The Data Packet Model informs system-to-system data presentation formats and, for example, Application Programming Interfaces (API).

The model is a de-normalised projection of the VET-IS logical schema into 'Packets' of data elements that represent the common primary conceptual entities found in the VET sector, e.g., Students & Enrolments. Data Packets primary use, within the STARS context, are as Data Transfer Objects (DTO) between external systems (e.g., RTO SMS, STA systems) and STARS.

In this sense, the Data Packet Schema is a 'projection' of the Logical Schema into commonly recognised topics. For example, a Student data packet is represented as a Stakeholder within the Logical Schema, as is a Training Provider. An Enrolment data packet is represented as a Responsibility.

Data Packets will also likely be the 'public face' of the VET-IS in terms of describing, reporting and exchanging data with external stakeholders, such as RTOs, SMS providers and STAs.

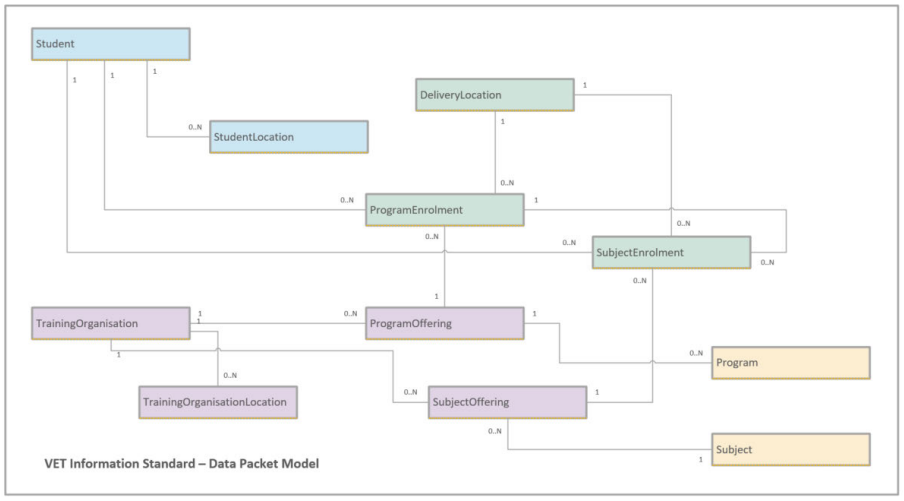
Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 641 of 1295

~~The Data Packet types currently defined in VET IS R1 are:~~

- ~~Government Packet~~
- ~~Training Organisation Packet~~
- ~~Training Organisation Location Packet~~
- ~~Program Packet~~
- ~~Subject Packet~~
- ~~Offering Packet~~
- ~~Delivery Location Packet~~
- ~~Student Packet~~
- ~~Student Location Packet~~
- ~~Program Enrolment Packet~~
- ~~Subject Enrolment Packet~~

~~VET-IS Release 1 is targeted to specifying and shaping the information and data required for the VET Provider Collection and thus there is not currently support for the VET Apprentice and Trainee collection (including Employers).~~

~~The following diagram shows the high-level relationships between the Data Packet types.~~



~~**Figure 19: VET IS Data Packet Schema**~~

~~By way of terminology, VET-IS Data Packages are a collection of VET-IS Data Packets. Packages only contain packets that relate to each other and contain a complete set of information. E.g., a package containing an Enrolment packet must also contain the details of the student enrolled in a Student packet, along with Delivery Location, Subject, Offering and Training Organisation packets.~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 642 of 1295

9.4 Schemas

Following are the data schemas of the entities those are passed by Training provider or STA to Submission service as data packets.

9.4.1 Training Provider

Note: Training Provide entity is mapped to Stake Holder – Training Org sheet in draft collection specification document.

Attributes	Mapping Collection Specification Model Name	Data Type	Comments
Identifier	Training organisation identifier	String	
Name	Training organisation name	String	
OrganisationType	Training organisation type code	String	Following will be valid Organisation types: School - government School - Catholic School - independent Technical and further education institute or similar public institutions University - government University - non-government Catholic University - non-government independent Enterprise - government Enterprise - non-government Community based adult education provider Private education/training business or centre:

Formatted: Normal

Formatted: Normal, Line spacing: single, No bullets or numbering

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 643 of 1295

			privately operated registered training organisation Professional association Industry association Equipment and/or product manufacturer or supplier Other - not elsewhere classified
ABN	ABN	String	

Example Training Provider entity JSON structure:

```
{
"Identifier": "6F9619FF-8B86-D011-B42D-00C04FC964FF",
" Name ": " TnH",
" OrganisationType ": " School - government",
" ABN ": " 32432432523"
}
```

Formatted: Normal, Line spacing: single, No bullets or numbering

9.4.2 Student

Formatted: Normal

Formatted: Normal, Line spacing: single, No bullets or numbering

Attributes	Mapping Collection Specification Model Name	Data Type	Comments
UniqueStudentIdentifier	Unique student identifier/USI	String	
StudentIdentifierNumber	Student identifier number	String	
ClientIdentifier	Client Identifier - apprenticeships	String	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 644 of 1295

SurveyContactStatus	Survey contact status	String	available not available
ReasonSurveyContactStatus	Reason for survey contact status	String	
DeceasedIndicator	Deceased indicator	String	
IndigenousStatusCode	Indigenous status code	String	
Title	Person title code	String	
FirstName	First given name	String	
OtherName	Other given names	String	
FamilyName	Family Name	String	
BirthDate	Birth date	String	DDMMYYYY format MMYYYY format YYYY format blank (for when not provided)
Gender	Gender type code	String	Following will be the valid options: Female Male Non-binary Prefer not to answer

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 645 of 1295

HighestSchoolCompleted	Highest school level completed code	String	Following will be the valid options: Did not go to school Year 8 or below Year 9 or equivalent Completed Year 10 Completed Year 11 Completed Year 12 Not specified
LabourStatusCode	Labour force status code	String	Following will be valid options: Full-time employee Part-time employee Self-employed - not employing others Self-employed - employing others Employed - unpaid worker in a family business Unemployed - seeking full-time work Unemployed - seeking part-time work Not employed - not seeking employment Not specified
SecondarySchoolCode	Secondary school type code	String	Following will be valid options: Non-VET in Schools student at school Not specified (Non-VET in Schools students only) School - Government

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 646 of 1295

			School - Catholic School - Independent Technical and Further Education Institute Community-based adult education provider Privately operated registered training organisation Home school arrangement
LanguageSpokenHome	Language spoken at home	String	It defines main language spoken at home
BirthCountry	Country of birth	String	
DisabilityType	Disability type code	List of String	Following will be the valid options: Hearing/deaf Physical Intellectual Learning Mental illness Acquired brain impairment Vision Medical condition Autism spectrum disorder Deafblind (dual sensory) Neurological (including epilepsy) Speech Other Prefer not to specify

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 647 of 1295

EducationalAchievementType	Educational achievement type code	List of String	Following will be the valid options: Bachelor degree or higher degree level (defined for AVETMISS use only) Advanced diploma or associate degree level Diploma level Certificate IV Certificate III Certificate II Certificate I Miscellaneous education Not specified

Example of Training Provider entity JSON structure

{

“ UniqueStudentIdentifier ": "6F9619FF-8B86-D011-B42D-00C04FC964FF",

" StudentIdentifierNumber ": " 7F9619FF-8B86-D011-B42D-00C04FC964FF",

" ClientIdentifier": " 7F9619FF-8B86-D011-B42D-00C04FC964FF",

" SurveyContactStatus": "Available",

"ReasonSurveyContactStatus": "",

" IndigenousStatusCode": " No, neither Aboriginal nor Torres Strait Islander ",

" Title": " Mr",

" FirstName": " Michael",

" OtherName": " ",

" FamilyName": " Jordan",

" BirthDate": " 22/10/2005",

" DeceasedIndicator": " No",

" Gender": " Male",

" HighestSchoolCompleted": " Completed Year 10",

" LabourStatusCode ": " Full-time employee",

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 648 of 1295

" SecondarySchoolCode": " School - Government",
" LanguageSpokenHome ": "English ",
" BirthCountry": "Australia ",
" DisabilityType": [" "],
" EducationalAchievementType": [" Diploma level"]
}

9.4.3 Program

Formatted: Normal

Attributes	Mapping Collection Specification Model Name	Data Type	Comments
Identifier	Program identifier	String	
Name	Program name	String	
ClassCode	Program class code	String	Following are valid classifications: Recognised Program Not Recognised Program
CategoryCode	Program category code	String	Following are valid classifications: State Recognised Program Not Nationally Recognised Program
TypeCode	Program type code	String	Other program Higher level qualification - self-accrediting authority Jurisdictionally recognised program
StreamReferenceNumber	Stream Reference Number	String	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 649 of 1295

StreamName	Stream Name	String	
EaducationCode	ASCED narrow field of education code	String	
DetailedEducationCode	ASCED detailed level of education code	String	Following are valid Classifications : Graduate diploma Graduate certificate Bachelor degree (honours) Bachelor degree (pass) Advanced diploma Year 12 Year 11 Year 10 Other non-award courses Statement of attainment not identifiable by level Bridging and enabling courses not identifiable by level Education not elsewhere classified Associate degree Diploma
VocationalEducationCode	Vocational education purpose code	String	Following are valid classifications: Yes No
ANZSCOCode	ANZSCO code	String	
NominalHours	Nominal hours quantity	Number	Must be a number between 0 and 6000. WARNING* It is expected the value is between 0 and 3000.

Formatted: Normal, Line spacing: single, No bullets or numbering

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 650 of 1295

Formatted: Indent: Left: 0 cm

Example of Program entity JSON structure

Formatted: Normal, Line spacing: single, No bullets or numbering

```
{
 " Identifier ": "6F9619FF-8B86-D011-B42D-00C04FC964FF",
 " Name": "Diploma",
 " ClassCode": " Recognised Program",
 " CategoryCode": " State Recognised Program",
 " TypeCode ": " Higher level qualification",
 " StreamReferenceNumber": " ",
 " StreamName": "",
 " DetailedEducationCode": " Diploma",
 " VocationalEducationCode": " No",
 " ANZSCOCode ": "",
 " NominalHours ": 4000,

}
```

9.4.4 Subject

Formatted: Normal

Attributes	Mapping Collection Specification Model Name	Data Type	Comments
Identifier	Subject identifier	String	
Name	Subject Name	String	
ClassCode	Subject Class Code	String	Following are valid classifications: Not Recognised Subject
CategoryCode	Subject Category Code	String	Following are valid classifications: Not Nationally Recognised Subject

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 651 of 1295

TypeCode	Subject Type Code	String	Following are valid classifications: Module
SubjectEducationFieldCode	Subject field of education code	String	
VocationalEducationCode	Vocational education purpose code	String	
NominalHours	Nominal hours quantity	Number	Must be a number between 0 and 1500. WARNING* Nominal hours is expected to be less than 400.
ScheduledHours	Scheduled hours quantity - state element	String	

Example of Subject entity JSON structure

Formatted: Normal, Line spacing: single, No bullets or numbering

```
{
" Identifier ": "6F9619FF-8B86-D011-B42D-00C04FC964FF",
" Name": "Subject",
" ClassCode": " Not Recognised Subject",
" CategoryCode": " Not Nationally Recognised Subject",
" TypeCode ": " Module",
" SubjectEducationFieldCode": "",
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 652 of 1295

~~" VocationalEducationCode": "",~~

~~" NominalHours ": 500,~~

~~" ScheduledHours ": " "~~

~~}~~

Formatted: Normal, Line spacing: single, No bullets or numbering

~~9.4.5 Training Provider Location~~

Formatted: Normal

Formatted: Normal, Line spacing: single, No bullets or numbering

~~Note: Training Provider Location entity is mapped to Stake Holder – Training Org Location sheet in draft collection specification document.~~

~~Attributes~~	~~Mapping Collection Specification Model Name~~	~~Data Type~~	~~Comments~~
~~Identifier~~	~~Training provider identifier~~	~~String~~	
~~PropertyName~~	~~Building property name~~	~~String~~	
~~Suburb~~	~~Suburb name~~	~~String~~	~~Suburbs found on Australia Post~~
~~State~~	~~State code~~	~~String~~	~~New South Wales~~ ~~Victoria~~ ~~Queensland~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 653 of 1295

			South Australia Western Australia Tasmania Northern Territory Australian Capital Territory
Postcode	Postcode	String	Postcodes found on Australia Post
Address1	Address line 1 description	String	
Address2	Address line 2 description	String	
EmailId	email address	String	
Telephone	telephone number	String	

Formatted: Normal, Line spacing: single, No bullets or numbering

Example of Training Provider Location entity JSON structure

Formatted: Normal, Line spacing: single, No bullets or numbering

```
{
"Identifier": "6F9619FF-8B86-D011-B42D-00C04FC964FF",
" PropertyName ": "George Place",
" Suburb": "Paramatta",
" State": "New South Wales",
" Postcode": "2500",
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 654 of 1295

~~" Address1": "10 Queen Street",~~

~~" Address2": "",~~

~~" EmailId ": " ",~~

~~" Telephone ": " "~~

~~}~~

Formatted: Normal, Line spacing: single, No bullets or numbering

~~9.4.6 Student Location~~

Formatted: Normal

Formatted: Normal, Line spacing: single, No bullets or numbering

~~Attributes~~	~~Mapping Collection Specification Model Name~~	~~Data Type~~	~~Comments~~
~~Identifier~~	~~Student Identifier Number~~	~~String~~	
~~CountryCode~~	~~Country code~~	~~String~~	
~~PrimaryEmailId~~	~~Email address (primary)~~	~~String~~	
~~SecondaryEmailId~~	~~Email address (secondary)~~	~~String~~	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 655 of 1295

WorkTelephoneNumber	Work telephone number	String	
HomeTelephoneNumber	Home telephone number	String	
PostalDeliveryType	Postal delivery type	String	PO Box Community Mail Bag GPO Box Mail Service Roadside delivery Roadside mail service Community mail agent Community postal agent Locked Bag Roadside Mail Box/Bag Private Bag Care of Post Office
PostalDeliveryReferenceNumber	Postal delivery reference number	String	
PostalSuburb	Postal delivery suburb name	String	Suburbs found on Australia Post
PostalState	Postal State code	String	New South Wales Victoria Queensland South Australia Western Australia Tasmania Northern Territory

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 656 of 1295

			Australian Capital Territory
PostalPostcode	PostalPostcode	String	Postcodes found on Australia Post
PostalBuildingName	Postal Building property name	String	
PostalBuildingType	Postal Building complex sub-unit type	String	Apartment Factory Flat Marine berth Office Room Shed Shop Site Stall Suite Unit Villa Warehouse
PostalBuildingNumber	Postal Building complex sub-unit number	String	
PostalFloorNumber	Postal Floor level number	String	
PostalStreetNumber	Postal Street property number	String	
PostalLotNumber	Postal Lot section number	String	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 657 of 1295

PostalStreet	Postal Street name	String	
PostalStreetType	Postal Street type code	String	Alley Arcade Avenue Boulevard Close Court Crescent Drive Esplanade Grove Highway Lane Parade Place Road Square Street Terrace
PermanentSuburb	Permanent delivery suburb name	String	Suburbs found on Australia Post
PermanentState	Permanent State code	String	New South Wales Victoria Queensland South Australia Western Australia Tasmania Northern Territory Australian Capital Territory

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 658 of 1295

PermanentPostcode	PermanentPostcode	String	Postcodes found on Australia Post
PermanentBuildingName	Permanent Building property name	String	
PermanentBuildingType	Permanent Building complex sub-unit type	String	Apartment Factory Flat Marine berth Office Room Shed Shop Site Stall Suite Unit Villa Warehouse
PermanentBuildingNumber	Permanent Building complex sub-unit number	String	
PermanentFloorNumber	Permanent Floor level number	String	
PermanentStreetNumber	Permanent Street property number	String	
PermanentLotNumber	Permanent Lot section number	String	
PermanentStreet	Permanent Street name	String	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 659 of 1295

PermanentStreetType	Permanent Street type code	String	Alley Arcade Avenue Boulevard Close Court Crescent Drive Esplanade Grove Highway Lane Parade Place Road Square Street Terrace
TermSuburb	Term delivery suburb name	String	Suburbs found on Australia Post
TermState	Term State code	String	New South Wales Victoria Queensland South Australia Western Australia Tasmania Northern Territory Australian Capital Territory
TermPostcode	Term Postcode	String	Postcodes found on Australia Post

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 660 of 1295

TermBuildingName	Term Building property name	String	
TermBuildingType	Term Building complex sub-unit type	String	Apartment Factory Flat Marine berth Office Room Shed Shop Site Stall Suite Unit Villa Warehouse
TermBuildingNumber	Term Building complex sub-unit number	String	
TermFloorNumber	Term Floor level number	String	
TermStreetNumber	Term Street property number	String	
TermLotNumber	Term Lot section number	String	
TermStreet	Term Street name	String	
TermStreetType	Term Street type code	String	Alley Arcade Avenue Boulevard Close Court

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 661 of 1295

			~~Crescent~~ ~~Drive~~ ~~Esplanade~~ ~~Grove~~ ~~Highway~~ ~~Lane~~ ~~Parade~~ ~~Place~~ ~~Road~~ ~~Square~~ ~~Street~~ ~~Terrace~~

Formatted: Normal, Line spacing: single, No bullets or numbering

~~Example of Student Location entity JSON structure~~

~~{~~

~~" Identifier": "6F9619FF-8B86-D011-B42D-00C04FC964FF",~~

~~" CountryCode": "",~~

~~" PrimaryEmailId ": " ",~~

~~" SecondaryEmailId ": " ",~~

~~" WorkTelephoneNumber ": " ",~~

~~" HomeTelephoneNumber ": " ",~~

~~" PostalDeliveryType": "PO Box ",~~

~~" PostalDeliveryReferenceNumber ": " ",~~

~~" PostalSuburb": "Paramatta",~~

~~" PostalState": "New South Wales",~~

~~" PostalPostcode": "2500",~~

~~" PostalBuildingName": "",~~

~~" PostalBuildingType": " Apartment",~~

~~" PostalBuildingNumber": "",~~

~~" PostalFloor ": "2",~~

" PostalStreetNumber ": "27",

" PostalLotNumber ": "27",

" PostalStreet": "Queen Street",

" PostalStreetType": "Street",

" PermanentSuburb": "Paramatta",

" PermanentState": "New South Wales",

" PermanentPostcode": "2500",

" PermanentBuildingName": "",

" PermanentBuildingType": " Apartment",

" PermanentBuildingNumber": "",

" PermanentFloor ": "2",

" PermanentStreetNumber ": "27",

" PermanentLotNumber ": "27",

" PermanentStreet": "Queen Street",

" TermStreetType": "Street",

" TermSuburb": "Paramatta",

" TermState": "New South Wales",

" TermPostcode": "2500",

" TermBuildingName": "",

" TermBuildingType": " Apartment",

" TermBuildingNumber": "",

" TermFloor ": "2",

" TermStreetNumber ": "27",

" TermLotNumber ": "27",

" TermStreet": "Queen Street",

" TermStreetType": "Street"

}

9.4.7 Delivery Location

Formatted: Normal

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 663 of 1295

Attributes	Mapping Collection Specification Model Name	Data Type	Comments
TrainingProviderIdentifier	Training provider identifier	String	
DeliveryLocationCode	Delivery location code	String	
DeliveryLocationName	Delivery location name	String	
Suburb	Suburb name	String	
State	State code	String	New South Wales Victoria Queensland South Australia Western Australia Tasmania Northern Territory Australian Capital Territory
Postcode	Postcode	String	Postcodes found on Australia Post
CountryCode	Country code	String	
VenueTypeCode	Venue type code	String	On site facility Off site location Workplace

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 664 of 1295

			Other (undefined) Not applicable

Example of Delivery Location entity JSON structure

```
{
" TrainingProviderIdentifier ": "6F9619FF-8B86-D011-B42D-00C04FC964FF",
" DeliveryLocationCode ": "",
" DeliveryLocationName": "",
" Suburb": "Paramatta",
" State": "New South Wales",
" Postcode": "2500",
" CountryCode": "",
" VenueTypeCode ": " Workplace "
}
```

9.4.8 Program Enrolment

Formatted: Normal

Attributes	Mapping Collection Specification Model Name	Data Type	Comments
Identifier	Program enrolment identifier	String	
ScheduledStartDate	Scheduled start date	String	
ScheduledCompletionDate	Scheduled completion date	String	
TraineeTrainingContractIdentifier	Trainee Training contract identifier	String	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 665 of 1295

~~ApprenticeTrainingContractIdentifier~~	~~Apprentice Training contract identifier~~	~~String~~	
~~TrainingClientIdentifier~~	~~Training Client identifier - apprenticeship~~	~~String~~	
~~ApprenticeClientIdentifier~~	~~Apprentice Client identifier - apprenticeship~~	~~String~~	
~~PurchasingContractIdentifier~~	~~Purchasing contract identifier (state specific)~~	~~String~~	
~~PurchasingContractScheduleIdentifier~~	~~Purchasing contract schedule identifier~~	~~String~~	
~~VoucherNumber~~	~~Voucher Number~~	~~String~~	
~~IntakeNumber~~	~~Intake Number~~	~~String~~	
~~StudyReasonCode~~	~~Study reason code~~	~~String~~	
~~SupersededProgramEnrolmentIdentifier~~	~~Superseded program enrolment identifier~~	~~String~~	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 666 of 1295

TransferProgramEnrolmentIdentifier	Transfer program enrolment identifier	String	
ClientIdentifier	Client identifier	String	
TrainingProviderIdentifier	Training provider identifier	String	
ProviderAssessmentIdentifier	Provider of assessment identifier	String	
EmployedIdentifier	Employed by Identifier	String	
InternationalStudentTypeCode	International Student Type Code	String	
ProgramIdentifier	Program identifier	String	
LifecycleStateTypeCode	Program enrolment lifecycle state type code	String	
LifecycleStateReasonTypeCode	Program enrolment lifecycle state reason type code	String	
LifecycleStateEffectiveDate	Program enrolment lifecycle state effective date	String	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 667 of 1295

LifecycleStateExpiryDate	Program enrolment lifecycle state expiry date	String	
FinancialDetailCategoryCode	Financial detail category code	String	
FinancialDetailTypeCode	Financial detail type code	String	
PaymentContributionType	Payment contribution type	String	Revenue earned from another training organisation
IssuedDate	Issued date	String	
ParchmentNumber	Parchment number	String	

Formatted: Normal, Line spacing: single, No bullets or numbering

Example of ProgramEnrolment entity JSON structure

```
{
 " Identifier ": "6F9619FF-8B86-D011-B42D-00C04FC964FF",
 " ScheduledStartDate": "21102022",
 " ScheduledCompletionDate ": "21102023",
 " TraineeTrainingContractIdentifier": "",
 " ApprenticeTrainingContractIdentifier": " ",
 " TrainingClientIdentifier": " ",
 " ApprenticeClientIdentifier": "",
 " PurchasingContractIdentifier": "",
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 668 of 1295

" PurchasingContractScheduleIdentifier": "",

" VoucherNumber ": "",

" IntakeNumber": "",

" StudyReasonCode": " Superseded by",

" SupersededProgramEnrolmentIdentifier": "",

" TransferProgramEnrolmentIdentifier": "",

" ClientIdentifier": "",

" TrainingProviderIdentifier": "",

" ProviderAssessmentIdentifier ": "",

" EmployedIdentifier": "",

" InternationalStudentTypeCode": "International onshore",

" LifecycleStateTypeCode": "Registered",

" LifecycleStateReasonTypeCode": " Student successfully completed all learning activities ",

" LifecycleStateEffectiveDate": "09/03/2023",

" LifecycleStateExpiryDate": "09/07/2023",

" FinancialDetailCategoryCode ": " Commonwealth specific funding program",

" FinancialDetailTypeCode": " Subsidy - (ACT) Job Trainer ",

" PaymentContributionTypeCode": "",

" IssueDate": "20102022",

" ParchmentNumber ": ""

}

9.4.9 Subject Enrolment

Formatted: Normal

Formatted: Normal, Line spacing: single, No bullets or numbering

Attributes	Mapping Collection Specification Model Name	Data Type	Comments
Identifier	Subject enrolment identifier	String	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 669 of 1295

ScheduledStartDate	Scheduled start date	String	
ScheduledCompletionDate	Scheduled completion date	String	
StudyReasonCode	Study reason code	String	
SourceProgramEnrolmentIdentifier	Source Program enrolment identifier	String	
SupersededSubjectEnrolmentIdentifier	Superseded Subject enrolment identifier	String	
TransferSubjectEnrolmentIdentifier	Transfer Subject enrolment identifier	String	
InternationalStudentTypeCode	International Student Type Code	String	
ClientIdentifier	Client identifier	String	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 670 of 1295

~~ProviderAssessmentIdentifier~~	~~Provider of assessment identifier~~	~~String~~	
~~SubjectIdentifier~~	~~Subject Identifier~~	~~String~~	
~~LifecycleStateTypeCode~~	~~Subject enrolment lifecycle state code~~	~~String~~	
~~LifecycleStateReasonCode~~	~~Subject enrolment lifecycle state reason code~~	~~String~~	
~~LifecycleStateEffectiveDate~~	~~Subject enrolment lifecycle state effective date~~	~~String~~	
~~LifecycleStateExpiryDate~~	~~Subject enrolment lifecycle state expiry date~~	~~String~~	
~~LearningChannelTypeCode~~	~~Learning Channel Type Code~~	~~String~~	
~~PredominantLearningChannelIndicator~~	~~Predominant Learning Channel Indicator~~	~~String~~	

EnrolmentLearningDeliverySkey	Enrolment Learning Delivery Skey	String	
DeliveryTypeCode	Delivery type code	String	
PredominantLearningDeliveryIndicator	Predominant Learning Delivery Indicator	String	
PredominantVenueIndicator	Predominant venue indicator	String	
TrainingProviderDeliveryLocationIdentifier	Training organisation delivery location identifier	String	
VETSchoolIndicator	VET in School Indicator	String	Yes
SubjectAssessmentTypeCode	Subject Assessment Type Code	String	
FinancialDetailCategoryCode	Financial detail category code	String	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 672 of 1295

FinancialDetailTypeCode	Financial detail type code	String	
PaymentContributionType	Payment contribution Type	String	

Example of SubjectEnrolment entity JSON structure

{

" Identifier": "6F9619FF-8B86-D011-B42D-00C04FC964FF",

" ScheduledStartDate": "21102022",

" ScheduledCompletionDate ": "21102023",

" StudyReasonCode": "Get a job",

" SourceProgramEnrolmentIdentifier": "",

" SupersededSubjectEnrolmentIdentifier": "",

" TransferSubjectEnrolmentIdentifier": "",

" InternationalStudentTypeCode": "International onshore",

" ClientIdentifier": "",

" ProviderAssessmentIdentifier": "",

" SubjectIdentifier": "",

" LifecycleStateTypeCode": "Registered",

" LifecycleStateReasonTypeCode": " Student successfully completed all learning activities ",

" LifecycleStateEffectiveDate": "09/03/2023",

" LifecycleStateExpiryDate": "09/07/2023",

" LearningChannelTypeCode": "",

" PredominantLearningChannelIndicator": "",

" DeliveryTypeCode": "Directed",

" PredominantLearningDeliveryIndicator": "",

" PredominantVenueIndicator": "",

" TrainingProviderDeliveryLocationIdentifier": "",

" VETSchoolIndicator ": "",

" SubjectAssessmentTypeCode": "Standard",

" FinancialDetailCategoryCode": " Commonwealth and state general purpose ",

" FinancialDetailTypeCode": " Subsidy - (ACT) Job Trainer ",

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 673 of 1295

```
  " PaymentContributionTypeCode": ""

}
```

10.1 Submission Schema

Submitted VET-IS packets will be accompanied by submission related meta-data. The exact shape of a submission object is still to be finalised and will depend on the design of the Rules Service and Validation Service, but will be similar toin line with the following example:

```
Submission_id:
Previous_submission_id:
Submission_Datetime:
Submitter_id:
Records: [{
   Record_id:
   RelevantPackets:  [ packet_id, …]
   Validation: [{
   Ruleset_id:
   Validation_status [{
      Rule_id:
      Status:
      Message:
      }, …]
   }, …]
}, …]
```

10.2 Submission Data Processing

This section is awaiting test data and associated scenarios to understand how the Training Activity Data in a Submission is to be stored (added, updated, deleted etc).

At a high level, when the Submission Service begins processing a Submission event, it will extract the current details of the submitted records from the Provider DB, and write a snapshot of the data submitted to the Submission DB. This snapshot will be the official record of data that was submitted. It will be what the Rules Service uses for validation, and what the Master Data~~ing~~ Service uses for writing data to the Master DB.

The Collection Specification Data Packets will inform the minimum set of data elements the Submission Service can expect to be available in the Provider DB.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 674 of 1295

~~10~~11 Access Management

Submission service will conform to Data Governance Access and Privacy Protocol document and IDAM detail design. Access to the Submission Service will be controlled by Azure RBAC, with all calling services identified and granted access through Azure AD Managed Identities. For detail about accessing submission service please refer to IDAM detail design document.

~~10.1~~11.1 Control of data

The Governance Framework sets out the key data governance roles surrounding VET Data Streamlining. The two core central roles for accessing data from STARS are that of the Data Owner and the Data Custodian, summarised below:

~~10.1.1~~11.1.1 Data Owner

Data Owner has authority for establishing rules surrounding the quality, protection and appropriate access to data. They individually and collectively have a responsibility to protect the personal and sensitive information provided by Data Providers. Data can be jointly or co-owned.

~~10.1.2~~11.1.2 Data Custodian

Data Custodians receive authorised data from Data Providers or from other Data Custodians. Data Custodians are authorised to collect and handle that data to perform their functions. They are responsible for ensuring that the personal information collected and stored is reasonably necessary for them to conduct their functions and activities in relation to the VET sector.

VET data stored in STARS will be controlled based on the data's characteristics, namely:

- Whether the training was government funded of delivered on a fee for service basis
- Whether the data is part of the National Minimum Data Set (NMDS) or is a jurisdiction specific data item
- Whether the data is in the STARS staging database or the STARS master database.

11.1.3 Fee for service Data

Fee for service data contained in STARS (reflecting the NMDS) will be owned by VET regulators, regardless of funding arrangement or provider type (this includes data relating to fee for service activity delivered by TAFEs and other government training bodies). Jurisdictions will retain joint custodianship with NCVER of any fee for service data provided through the STA, until that data is processed to the STARS master database. Once in the master database, NCVER will be the sole data custodian.

Formatted: Font: 11 pt

Formatted: Font: 11 pt, Not Highlight

Formatted: Not Highlight

Formatted: Font: 11 pt, Not Highlight

Formatted: Font: 11 pt, Not Highlight

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 675 of 1295

11.1.4 Government Funded Data

Government funded training data in the STARS staging database will be controlled by the responsible jurisdiction for quality assurance and to support payment approvals to training providers.

Government funding bodies will have responsibility for the quality of data provided by training providers and will control the release of any data from STARS staging to the STARS master database, following quality assurance against program/contract reference data held by the funding body.

11.1.5 National VET Dataset

The national VET data set (which excludes jurisdiction specific data items) will be available from the STARS master database, under the control of the National VET Data Custodian. The NVETR Act, the SI Act and the SI regulation collectively set out a range of authorised users and their authorised uses of VET data. NCVER has a key role in collecting VET data on behalf of governments and disclosing the data to authorised users in order for them to fulfil their legislated functions and purposes. Such purposes include regulation, administrating the VET sector, consumer information, market and policy analysis, and generating public reports and student transcripts.

11.2 Data Access Levels

The levels of data access for each jurisdiction and VET regulators will be granted by NCVER to data stored in STARS. As National VET Data Custodian, NCVER will have access to all data in STARS.

STAs may be granted access to unpublished identified student and training activity data in the STARS master database for training undertaken nationally (NMDS only), including for training delivered on a fee for service basis. The Commonwealth may also be granted access to data from the master database for purposes related to the administration of the VET sector and/or the collection and preparation of statistics relating to VET.

Unique Student Identifiers (USIs) will be accessible for all VET students as authorised under the Student Identifiers Act.

The Training Provider will provide single acknowledgement while submitting the data to STARS, they are making an agreement that the submission data is fine to be release for National Reporting and state funding purposes. Any submission data that passes national validation, will be made available for STA validation and payment processing as part of the process in case of Govt Funded training.

For detail about Data access and control please refer to the following documents:~~Data governance Access and Privacy protocol document.~~

- Data governance Access and Privacy protocol document

Formatted: Font: 11 pt, Not Highlight
Formatted: Font: 11 pt
Formatted: Font: 11 pt, Not Highlight
Formatted: Normal, Line spacing: single, No bullets or numbering
Formatted: Not Highlight
Formatted: Font color: Text 1
Formatted: Normal
Formatted: Font: 11 pt
Formatted: Normal
Formatted: Font: 11 pt
Formatted: Space After: 0 pt, Keep with next

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 676 of 1295

- Governance Framework for Vocational Education and Training Student and Training Activity Data
- VET Data Access and Privacy Protocol Final - 27 February 2023

~~10.2~~11.3 Security

STARS will be built to meet existing Australian Government policies, frameworks, standards and design principles, including:

- Australian Government API Design Standards
- Digital Service Standard
- Protective Security Policy Framework (PSPF)
- Information Security Registered Assessors Program (IRAP)
- Australian Government Information Security Manual (ISM)
- the Open Web Application Security Project® (OWASP)
- Hosting Certification Framework.

All data within STARS will be handled in accordance with relevant privacy laws and record keeping requirements, including the Privacy Act, the *Archives Act 1983* (Cth) and any other applicable legislation (e.g. applicable State and Territory privacy legislation or archiving requirements).

Submission service will conform to the Security Architecture defined for STARS in Data Governance Access and Privacy Control document and IDAM detail design document. Access to the Submission Service will be controlled by Azure RBAC, with all calling services identified and granted access through Azure AD Managed Identities.

For detail about security please refer to the following

- Data Governance Access and Privacy Control document
- IDAM detail design document
-

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 677 of 1295

12 Dependency on Other Services

~~11~~

~~Following are STARS services upon which Submission service has dependencies.~~

12.1 Rules Service

The Rules Service will be used to run business rules to determine the funding relationships for submitted records prior to validation. For details about the Rules Service please refer to Rules Service detailed design document.

~~11.1~~12.2 Validation Service

The ~~Submission service depends upon~~ Validation service ~~for~~ will be used to ~~validation~~ validate ~~of Submission~~ submitted data packets~~ submitted by Training Provider or STA.~~ For details about the Validation service please refer to Validation service detailed design document.

~~11.2 Rules Service~~

~~Submission service depends upon Rules service which is used to validate business rules of Submission data packets submitted by Training Provider or STA which will be using Drools rules engine. For details about Rules service please refer to Rules service detail design document.~~

~~11.3 Notification Service~~

~~Submission service will depend upon Notification service for sending notifications when the submission gets rejected or got approved and have been moved to next service which could be Staging or Release. For details about Notification service please refer to Notification service detail design document.~~

~~11.4~~12.3 Event Service

~~Submission service will depend upon Event service for saving event information in Submission event database. For details about Event service please refer to Event service detail design document.~~

Submissions enter the Submission Service via a subscription to a Topic within the Event Service. The Submission Service will also raise events at various stages in the submission workflow as outlined in Section ~~7.~~7 Submission Events. The Event Service will be responsible for storing the history of all events raised. For details about the Event Service, please refer to the Event Service detailed design document.

12.4 Notification Service

It is expected that the Submission Service will need to raise notifications, such as when Submission Failed or Submission Succeeded events are raised. Further investigation is required to confirm the precise details of these events, but they will be raised as API calls to the Notification Service API, in line with the Notification Service Detailed Design Document (E).

Formatted: Normal

Formatted: No bullets or numbering

Field Code Changed

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 678 of 1295

12.5 Provider Service

The Training Providers or STAs will submit the submission using Provider service. The Provide service will raise an event which will be consumed by the Submission service and the submission service will get the details of the submission from the Provider Service API. Please refer to the Provider Service detail design document for more detail.

Formatted: Normal

12.6 Logging And Audit Service

The Logging and Audit Service will be used for logging and audit purpose by Submission service. Please refer to the Logging and Audit detail design document for more detail.

12.7 Master Data Service

The Master Data Service is one of an important service in Submission pipeline service, and it stores the submission data which is mastered by the Release service. The Submission service will get the dependency of the submission from the Master Data Service. Please refer to the Master Data Service detail design document for more details.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 679 of 1295

~~12~~13 Requirements

13.1 Functional

The following section lists the relevant Functional and Non-functional requirements relevant to the submission ~~is~~ service, with traceability to the individual sections of the document that speak to each requirement. ~~Some of these requirements are "owned" by this service, and others are owned by different services but are still relevant for referencing. "Owned" requirements are highlighted in green.~~

Formatted: Normal

13.1.1 Fulfilled Requirements

Following is the list of all functional requirements that will be fulfilled by the Submission Service.

Table ~~77~~: Functional Requirements - Fulfilled Requirements

Formatted: Caption

FR No	Requirement Name	Traceability	Description	Acceptance Criteria
FR010	FR010 - Reject data	See 6 Submission Workflow	The VDS system must accept records that have passed validation checks, and reject only records that have failed validation checks, during submission. • The VDS system will notify validation errors/warnings/messages in near real time at the time of submission, where possible	Given Training Organisation activity data that violates internal or external validation rules When the data is submitted Then STARS rejects invalid records And valid data is accepted into STARS Given Training Organisation activity data that violates internal validation rules

Formatted: Font: Bold

Formatted Table

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description	Acceptance Criteria
				When the data is submitted Then STARS rejects invalid records And the submission response contains all errors/warnings/messages from internal validation Given Training Organisation activity data that violates external validation rules When the data is submitted Then STARS rejects invalid records And notifies the submitter of all errors/warnings/messages from external validation
FR138	FR138 - Validation	See 5.3 Cross Jurisdiction Scenarios	If a training organisation is contracted with multiple STAs, VDS system should be able to isolate funded activity data that is specific to my STA and should apply my STAs specific state validation rules only on the activity data that is	Given a Training Organisation that operates across multiple jurisdictions And the Training Organisation has submitted training activity data across multiple jurisdictions in a single submission When STARS validates the training activity data

Formatted: Font: Bold

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description	Acceptance Criteria
			required and relevant for funding and NCVER reporting for my STA.	Then STARS will apply appropriate validation rules for each record according to its jurisdiction Given a Training Organisation that operates across multiple jurisdictions And the Training Organisation has submitted training activity data across multiple jurisdictions in a single submission When STARS processes the training activity data Then STARS will apply appropriate processing logic according to each record's jurisdiction

13.1.2 Informing Requirements

~~12.1~~ Following is the list of Informing Requirements which are not "owned" by the Submission service but are still relevant for referencing.

~~The following section lists the relevant Functional and Non-functional requirements relevant to this service, with traceability to the individual sections of the document that speak to each requirement. Some of these requirements are "owned" by this service, and others are owned by different services but are still relevant for referencing. "Owned" requirements are highlighted in green.~~

Formatted: Normal

Table ~~88~~87: Functional Requirements – Informing Requirements

Formatted: Caption, Keep with next

FR No	Requirement Name	Traceability	Fulfilled In	Description	Acceptance Criteria
FR001a	FR001a - Data submission via API	See 5.1 Submit Activity Data	Provider Service	A training organisation must be able to submit/resubmit a dataset in real-time to the VDS system via a Student Management System (SMS). SMS and training organisations should be able to setup API unaided based on the API interface design (set out in Annexure B) documentation and the training provided by the seller.	TC Verify authorised user data submission TC (-) Verify unauthorised user data submission fails Given Authorised Access When VET Provider submits training activity Then Provider will be given successful notification of submission TC Verify submission of authorised training activity Given Authorised Access and a previous submission When VET Provider re-submits an updated submission Then Provider will be given

Formatted Table

Formatted: Font: Bold

FR No	Requirement Name	Traceability	Fulfilled In	Description	Acceptance Criteria
					successful notification of re-submission and the previously submitted data will be updated in VDS (STARS) TC Verify successful re-submission of authorised training activity TC (-ive) Verify un-successful re-submission of un-authorised training activity Given Authorised Access to APIM Developer Portal and access to API interface design documentation and completed training activities When an SMS sets up interaction with the APIs Then no additional support is required Given Authorised Access to APIM Developer Portal and access to API interface design documentation and completed training activities When a training organisation sets up interaction with the APIs

Formatted Table

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Fulfilled In	Description	Acceptance Criteria
					t~~T~~hen no additional support is required
FR003	FR003 - Re-submitting data	See 5.1 Submit Activity Data and 5.2 Review Submission Data	Provider Service	A VET Provider must be able to submit/re-submit (including correcting historical data) at any point in time all data specified in the VET Information Standard. A VET Provider must be able to re-submit data that has previously been validated at any time. The system must be able to distinguish which record is latest.	Given errors are identified in previously Entered or Uploaded VET Information Started data When the VET Provider re-submits updates to that data Then STARS will update and record required changes Given errors are identified in previously Staged VET Information Started data When the VET Provider re-submits updates to that data Then STARS will update and record required changes Given errors are identified in previously Mastered VET Information Started data When the VET Provider re-submits updates to that data Then STARS will update and record required changes Given VET Information Started data that has been re-submitted

Formatted Table

Formatted: Font: Bold

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Fulfilled In	Description	Acceptance Criteria
					When viewing the submitted data Then the latest record(s) may be distinguished
~~FR010~~	~~FR010 – Reject data~~			~~The VDS system must accept records that have passed validation checks and reject only records that have failed validation checks during submission.~~ ~~• The VDS system will notify validation errors/warnings/messages in near real time at the time of submission, where possible~~	~~Given Training Organisation activity data that violates internal or external validation rules~~ ~~When the data is submitted~~ ~~Then STARS rejects invalid records~~ ~~And valid data is accepted into STARS~~ ~~Given Training Organisation activity data that violates internal validation rules~~ ~~When the data is submitted~~ ~~Then STARS rejects invalid records~~ ~~And the submission response contains all~~

Formatted Table

FR No	Requirement Name	Traceability	Fulfilled In	Description	Acceptance Criteria
					~~errors/warnings/messages from internal validation~~ ~~Given Training Organisation activity data that violates external validation rules~~ ~~When the data is submitted~~ ~~Then STARS rejects invalid records~~ ~~And notifies the submitter of all errors/warnings/messages from external validation~~
~~**FR011**~~	~~FR011 - Submit data~~			~~A training organisation must authorise the submission of data into the VDS system.~~ ~~• If the time limit for submission has been reached, the VDS system will notify the Provider if data has completed a validation check but has not~~	~~Given Training Organisation has entered or uploaded data into STARS and has not submitted that data.~~ ~~When the time limit for a submission has been reached~~ ~~Then STARS will notify the Training Organisation.~~

Formatted Table

Formatted: Font: Bold

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Fulfilled In	Description	Acceptance Criteria
				been authorised for submission.	
FR042a	FR042a – Activity data			The VDS system must be able to collect training activity data. For example, this includes funded and unfunded training activity data, subject only and program only activity, estimated/forecast and actual training outcome data, and related fields e.g. unique ID which identifies the training contract in STA systems.	Given training activity data When the data is entered/uploaded according to the VET Information Standard Then the data will be validated and stored
FR042b	FR042b – Non-accredited courses and training organisations			The VDS system must be able to collect non-accredited training activity data. • This will require additional reference data from STAs to support validations. For example, courses not recognised by TGA, and data from non-registered training organisations.	Given submittable training activity for non-accredited training activity When Training Organisation submit data to STARS Then the data will be validated and stored Given reference data from STAs related to non-accredited training providers When the reference data is provided to STARS Then STARS will store the reference

Formatted Table

Formatted: Font: Bold

Formatted: Font: Bold

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Fulfilled In	Description	Acceptance Criteria
					~~data~~ ~~And use this data to support validation of non-accredited training activity data~~
FR048	FR048 - State and program data	See 6 Submission Workflow	Rules Service	The VDS system gathers data that distinguishes the relevant STA, program, funding type and funding source (e.g. Commonwealth, state-specific, Fee For Service local, Fee For Service international) that submitted data relates to.	Given submitted and validated training activity data When training activity data is stored in STARS Then the stored training activity data has attributes that distinguish the relevant STA, program, funding type and funding source in accordance with the VET Information Standard
FR049	FR049 - Provisional data submission	See 5.2 Review Submission Data	Provider Service	A training organisation must be able to provisionally validate data, (i.e. view the outcomes of validation checks) without submitting the data for downstream processing.	Given training activity data that has been uploaded or entered to STARS When the provider validates the data prior to submission Then STARS will validate and return the validation results but not submit the data for further processing
FR050	FR050 - Log of historical data and validation results		Management Portal – Submission Service	The VDS system must allow the relevant data custodians to: • View a log of validation results from previous submissions, including result	Given submitted training data When the custodian of that data views associated validation results Then STARS will provide the validation results for each entry And validation results from previous

Formatted Table

Formatted: Not Highlight

Formatted: Not Highlight

Formatted: Font: Bold

Formatted: Font: Bold

Formatted: Font: Bold

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Fulfilled In	Description	Acceptance Criteria
				for each entry • View data the training organisation has previously submitted to the VDS system	submissions Given submitted training data When the custodian of that data views the submitted data Then STARS will provide the submitted data And provide a view of previous submissions
~~**FR068**~~	~~FR068 – Data Submission - USI dataset~~			~~The VDS system must provide regulators the ability to submit minimum datasets to support Student Transcripts for students/former students with a USI number~~	~~Given training activity~~ ~~And the training provider is not able to submit the regular training activity data~~ ~~When regulators submit a minimum dataset~~ ~~Then STARS will record the data on behalf of the training provider~~ ~~And only validations for minimum datasets will be triggered~~
~~**FR076**~~	~~FR076 – Other Collection agencies are able to provide bulk event data to VDS system using any version of the~~			~~The VDS system must allow another collection agency to upload data that does not exist in VDS system and the collection agency will receive validation responses on behalf of training organisation. This~~	~~Given access for data entry or upload~~ ~~And access for data submission~~ ~~And a recorded agreement between the submitter and a training organisation~~ ~~When data is submitted on behalf of the training organisation~~ ~~Then STARS will record the submitter~~

Formatted Table

Formatted: Font: Bold

Formatted: Font: Bold

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Fulfilled In	Description	Acceptance Criteria
	~~VET Information standard that is still accepted nationally.~~			~~can be a partnering or non-partnering jurisdiction.~~	~~of the data~~ ~~And process the submission as if it was submitted by the training organisation~~
~~FR077~~	~~FR077 - Other Collection agencies are able to send near real time event data to VDS system using any version of the VET Information standard that is still accepted nationally.~~			~~The VDS system will allow another collection agency to upload data that does not exist in VDS system and the collection agency will receive validation responses on behalf of training organisation. This can be a partnering or non-partnering jurisdiction.~~	~~Given access for data entry or upload~~ ~~And access for data submission~~ ~~And a recorded agreement between the submitter and a training organisation~~ ~~When data is submitted on behalf of the training organisation~~ ~~Then STARS will record the submitter of the data~~ ~~And process the submission as if it was submitted by the training organisation~~
FR084	FR084 - Handling of Data Resubmission		Management Portal – Provider Service	As a TO User (using data entry functionality or SMS system), I want the previous submission records of the same data to be treated/marked as historical so that those records do not need to be reprocessed again	Given a previous submission of training activity data And the submission contained validation errors And the training organisation has updated the training activity data When the submission is re-submitted via the API Then any unaltered packets that passed validation previously are not

Formatted Table

Formatted: Font: Bold

Formatted: Not Highlight

Formatted: Font: Bold

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Fulfilled In	Description	Acceptance Criteria
					re-validated And any updated packets are validated Given a previous submission of training activity data And the submission contained validation errors And the training organisation has updated the training activity data When the submission is re-submitted via Data Entry UI Then any unaltered packets that passed validation previously are not re-validated And any updated packets are validated Given a resubmission of training activity data that has been prepared but not submitted When the resubmission is saved Then STARS records the resubmission data Given a saved resubmission When the resubmission is loaded Then additional updates can be

Formatted Table

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Fulfilled In	Description	Acceptance Criteria
					made And the data can be submitted
FR108	FR108 - Data Submitted for STA		Management Portal - Submission Service	As a STA user, I want to be able to view the data submitted by a particular training organisation that is delivering training in my state regardless of the method of data submission so that I am aware of what data has been submitted and the validation status.	Given training activity data submitted by a training organisation When an STA user searches for a training organisation Then STARS will provide search results matching the given criteria And allow the user to view the data submitted by the training organisation Given training activity data submitted by a training organisation When an STA user views the data submitted by the training organisation Then STARS will show the date and time of each submission And the status of each record And allow the submission records to be filtered by submission date/time, status, student name, student id or USI
FR130	FR130 - Validate via 2 Standards		Reference Data Service	System should be able to store all the information (Data elements) in the VET training	Given multiple active versions of the VET Information Standard When training data is submitted in

Formatted Table

Formatted: Font: Bold

Formatted: Font: Bold

FR No	Requirement Name	Traceability	Fulfilled In	Description	Acceptance Criteria
				activity data irrespective of the standard followed during data submissions so that the data has the same meaning and that the data consumer (for ex. STA / NCVER) can consume data in the VET information standard they are still operating in. Note* system should be able to ~~accomodate~~accommodate multiple sets of validations at any given point. The system will record the standard data was validated in, so future changes can be understood.	one of the active versions of the standard Then STARS will allow consumption of the data in any active version of the standard Given multiple active or inactive versions of the VET Information Standard When training data has previously been submitted in a version of the standard that is no longer active Then STARS will allow consumption of the data in any active version of the standard
~~**FR133**~~	~~FR133 - Activity Data Upload Capability during Transition Period~~			~~While Activity Data Files are being uploaded or processed via UI, system will not allow relevant training organisation's to re-upload files~~	~~Given Training Organisation has submitted training activity data When that submission has not yet completed processing and validation Then STARS will not allow the Training Organisation to upload or resubmit any training activity~~
FR140	FR140 - Validations	See 8 Submission API Design	STA Gateway Service	The VET student activity data records once submitted by the	Given Training Organisation has submitted training activity data to STARS When the training activity

Formatted Table

Formatted: Font: Bold

Formatted: Font: Bold

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Fulfilled In	Description	Acceptance Criteria
				contracted training organisations should be accessible by the STA and NCVER with appropriate validation check status set against each activity record. STAs will have access to data it funds across entire data lifecycle.	data has completed synchronous validation Then STAs will be able to access the submitted data for their jurisdiction And NCVER will be able to access the submitted data And status information for each record will be available And will continue to have access through the data lifecycle of each record.
~~FR128~~	~~FR128 – Validation~~	~~If a training organisation is contracted with multiple STAs, VDS system should be able to isolate funded activity data that is specific to my STA and should apply my STAs specific state validation rules only on the activity data that is required and relevant for funding and NCVER~~	~~Given a Training Organisation that operates across multiple jurisdictions And the Training Organisation has submitted training activity data across multiple jurisdictions in a single submission When STARS validates the training activity data Then STARS will apply appropriate validation rules for each record~~		

Formatted Table

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Fulfilled In	Description	Acceptance Criteria
		reporting for eny STA.	according to its jurisdiction Given a Training Organisation that operates across multiple jurisdictions And the Training Organisation has submitted training activity data across multiple jurisdictions in a single submission When STARS processes the training activity data Then STARS will apply appropriate processing logic according to each record's jurisdiction		

Formatted Table

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

~~12.2~~13.2 Non-Functional

'Non-functional requirements' and 'technical requirements' are synonymous; in this document the more standard BABOK term 'non-functional' is used. These Non-Functional Requirements have been loaded into DevOps and linked to the Business and Functional Requirements. Requirement's traceability is leveraged through the use of Azure DevOps and ~~Smart~~ Modern Requirements. These requirements are not directly resolved by the Submission Service but will inform details of the Submission Service implementation.

Table ~~9998~~: Non-Functional Requirements

Formatted: Caption, Keep with next

FR No	VDS NFR Category	Requirement Name	Description
NFR001a	Scalability	NFR001a - Use Case scalability	The VDS System will be able to scale elastically in response to demand and remain functional and available to users regardless of dynamic changes in volume, processes, and number of users leveraging the VDS System. • The VDS System will be designed to achieve scalability of additional use cases as needed • The VDS System will support both horizontal and vertical scaling
NFR001b	Scalability	NFR001b - Concurrent users scalability	• The VDS System will be designed, developed, tested and deployed to accommodate increased concurrent users, volumes and workloads specifically: • Real time scalability to accommodate increases in the number of concurrent users. • Consistent response times (e.g. user interface, data submission, application workflow, and data consumption) with increased volumes and concurrent users.
NFR001c	Scalability	NFR001c - Baseline user volumes	The VDS System supports the following baseline volumes: • Over 2,000 concurrent users across training organisations and STAs • 5,000 total users across training organisations and STAs

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

NFR001d	Scalability	NFR001d - Baseline transaction volumes	The VDS System supports the following baseline volumes: • Over xxx transactions for the submission of data by training organisations per week • Over xxx data submissions per second
NFR002a	Availability	NFR002a - Minimum Uptime	The VDS System will support continual operations by being available and functioning with minimal downtime (High Availability). • The VDS System will support a minimum proposed uptime for the users, excluding scheduled downtime or outage
NFR002b	Availability	NFR002b - Downtime notification	The VDS System will have a notification process to notify all system users of any scheduled or unscheduled downtime or outage, including to address critical issues (e.g. denial of service attack). For scheduled downtime, users will be notified three (3) business days prior to the downtime. For unscheduled downtime, users will be notified as soon as the outage is identified.
NFR002c	Availability	NFR002c - Scheduled downtime	The VDS System will have an agreed scheduled downtime or outage (if applicable) within business hours and/or outside business hours
NFR002d	Availability	NFR002d - Production environment uptime	The VDS System, including all user interfaces, in the Production environment is available 99.86% of the scheduled hours of availability, measured over a month. The system is scheduled to be available for all hours, except those in approved change windows (i.e. planned outages). This equates to 60 minutes of unavailability in a 30 day month with no planned outages.
NFR002e	Availability	NFR002e - Non-Production environments uptime	The VDS System, including all user interfaces, in environments other than Production (per NFR11), is available 99.79% of the scheduled hours of availability, measured over a month. The system is scheduled to be available for all hours, except those in approved change windows (i.e. planned outages). This equates to 90 minutes of unavailability in a 30 day month with no planned outages.
NFR002f	Availability	NFR002f - Outage window for major releases	The VDS System will use a single planned outage window of five (5) hours allowed in a three (3) month period for major releases. The planned outage will allow for a full system outage.
NFR002g	Availability	NFR002g - Outage window for minor releases	The VDS System will use a single planned outage window of one (1) hour allowed in a one (1) month period for minor releases. The planned outage will allow for a full system outage.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

NFR002h	Availability	NFR002h - Outage window for emergency changes	The VDS System will use an outage window for emergency changes, targeting installation within a 30 minute window. The outage will be scheduled between 2:00 am and 4:00 am AEST on any day. The planned outage will allow for a full system outage.
NFR002i	Availability	NFR002i - Overnight extraction	The VDS system will be available for data extraction overnight
NFR003a	Performance	NFR003a - User response time	The VDS System will provide adequate response times for all users. The VDS System will be designed, developed and deployed in such a manner that response times are achieved at all times, including during peak periods of user activity. • The VDS System will provide a response time of 1 second to meet the standard response times expected by a user.
NFR003b	Performance	NFR003b - Monitor response time	The VDS System will have the ability to monitor and measure the response times it provides to its users.
NFR003c	Performance	NFR003c - No performance impact on state-managed systems	The VDS System integrates with state-managed systems in such a way that the traffic does not negatively impact the performance or availability of these systems.
NFR003d	Performance	NFR003d - Currency of reference data	The VDS System provides a mechanism to enable currency of reference data from state-managed systems into VDS.
NFR004a	Hosting	NFR004a - WoG Hosting Strategy	The VDS System will be hosted in accordance with the Whole of Government Hosting Strategy. • The VDS System will meet whole of government sovereignty requirements as specified in the Whole of Government Hosting Strategy.
NFR004b	Hosting	NFR004b - Minimise Supply chain risks	The VDS System will minimise supply chain risks in regards to the hosting arrangements, in accordance with the Whole of Government Hosting Strategy.
NFR004c	Hosting	NFR004c - Change Hosting	The VDS System will be capable of supporting a change in hosting provider(s), for any reason, as directed by the Australian Government.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

NFR005a	Authentication	NFR005a - Digital Identity	The VDS System will fully support Digital Identity in accordance with the Trusted Digital Identity Framework (TDIF). • The VDS System will utilise the myGovID machine to machine authentication model.
NFR005b	Authentication	NFR005b - Partner identity federation	The VDS System enables users in each Partner organisation to authenticate using identity federation via single sign on (SSO). This includes: • AUSKey • Add specific requirement for each STA
NFR006a	Compatibility	NFR006a - Operating systems & devices	The VDS System will support a seamless user experience and remain consistent with the Digital Service Platforms Strategy (available at https://www.dta.gov.au/our-projects/digital-service-platforms-strategy). • The VDS System will function for users across a range of operating systems and devices (including laptops, personal computers, and mobile devices such as mobile phones and tablets). This will include the most current minor version, the previous two minor versions and any major version from the past two years.
NFR006b	Compatibility	NFR006b - Consistent experience	The experience for all users of the VDS System will be respectively consistent and integrated across all supported channels.
NFR006c	Compatibility	NFR006c - Access key components	• Users will be able to access key components of the VDS System where there is limited internet connectivity or mobile telecommunications coverage. For example, providing the ability to access the VDS System BI Capability through a basic view even if more sophisticated services are not viable.
NFR006d	Compatibility	NFR006d - APIs are AVETMIS Standard agnostic	The VDS System's APIs are agnostic to which version of AVETMISS is used
NFR008a	Security Compliance	NFR008a - ISM Compliance	The VDS System will meet or comply with the Protective Security Policy Framework (PSPF) (https://www.protectivesecurity.gov.au/). • The VDS System will meet or comply with the Information Security Manual (ISM) (https://www.cyber.gov.au/acsc/view-all-content/ism), including having cyber security controls in place to protect the security and privacy of information processed, stored and transmitted. The VDS System will be

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

			authorised against the Information Security Manual at the appropriate classification level for the information that they are collecting, storing and/or processing.
NFR008b	Security Compliance	NFR008b - ACSC Essential 8 compliance	• The VDS System will meet or comply with the highest maturity level attainable against the ACSC Essential 8 (https://www.cyber.gov.au/acsc/view-allcontent/publications/essential-eight-maturity-model).
NFR008c	Security Compliance	NFR008c - Future security policies	The VDS System will comply with an architecture that can adapt to future security policies, standards or frameworks that are enacted during the term of any Contract.
NFR008d	Security Compliance	NFR008d - Mitigate Cyber Security Incidents	The VDS System will meet the ACSC Strategies to Mitigate Cyber Security Incidents described at https://www.cyber.gov.au/acsc/view-allcontent/publications/strategies-mitigate-cyber-security-incidents. Any instances of non-compliance should be documented and provided to the Department.
NFR008e	Security Compliance	NFR008e - Partner security policy compliance	The VDS System will be compliant with Partner security guidance and policies including: * Add specific requirements for each custodian here
NFR009a	Security Controls and Architecture	NFR009a - Risk and Threat model	The VDS System will support security controls and associated security architecture driven by sound risk assessments that reflect the value and sensitivity of the digital assets and data that are being protected, as well as the threats to the VDS System. The VDS System will be designed, delivered and supported using risk and threat model driven design and in accordance with the Australian Government security requirements (including the PSPF and ISM).
NFR009b	Security Controls and Architecture	NFR009b - Deliberate attack	• The VDS System will be designed to defend against deliberate malicious actions. • The VDS System will be designed so that it can continue to function as intended in a trustworthy, dependable and reliable manner in the event of deliberate attack. The VDS System will be designed to be hardened to have resilience in the face of deliberate attack including able to resist and defend against high level motived nation state threat actors and trusted insiders.
NFR009c	Security Controls and Architecture	NFR009c - Control failure	• The VDS System will be designed so that failure or inadequacy of preventative controls can be detected and response controls are in place to detect and contain the impact of failure. • The VDS System will be designed to support defence-in-depth protection and layers of diverse security controls so that any one control failure will not result in a complete loss of security.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

NFR009d	Security Controls and Architecture	NFR009d - Strong accountability	• The VDS System will support and enable strong accountability measures for the use of and access to information and resources. • The VDS System will be designed with mechanisms (audit, access logging) which enable Users to be held accountable for their actions.
NFR009e	Security Controls and Architecture	NFR009e - Deny access by default	The VDS System will support secure defaults and in the event of failure, deny access by default.
NFR010a	Access Security	NFR010a - Role based access controls	Access to the VDS System will be restricted to authorised users to safeguard against deliberate, intrusive and/or unauthorised access from internal and external sources. • The VDS System will support role-based security access controls to ensure that only those with appropriate authority can access a particular user interface. • The VDS System will timeout after 30 minutes of inactivity.
NFR010b	Access Security	NFR010b - Data attributes access controls	The VDS System will support data attribute access control. This shall include controls to ensure there is no unauthorised access to the Business Rules contained within the VDS System or the configuration of the VDS System.
NFR010c	Access Security	NFR010c - Record of access controls	The VDS System will retain a historical record of the access controls that were applied to each role/individual at any given point in time and a record of all changes to access controls for a particular role/individual.
NFR010d	Access Security	NFR010d - Systems Operator role	For a Systems Operator role, the VDS System will allow access to easily change access to data, user interfaces and/or functionality.
NFR010e	Access Security	NFR010e - IAM systems integration	The VDS System's role and attribute-based security access controls are to integrate with the existing Identity and Access Management (IAM) systems where needed.
NFR010f	Access Security	NFR010f - Flexible authentication	The VDS System will support secure modern and flexible authentication and authorisations methods, including SAML federation with claims-based authorisation, OIDC etc.
NFR010g	Access Security	NFR010g - Identity Proofing	• The VDS System will implement the required identity proofing and credential strength for each user cohort in accordance with the TDIF identity proofing standards/guidelines. This will be driven by a TDIF assessment of the business risk of high importance transactions, decisions or activity.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

NFR010h	Access Security	NFR010h - Reporting security	• VDS must enable secure access and protection of reporting, compliant with Govt security, privacy and cyber policy
NFR010i	Access Security	NFR010i - Reporting security	The VDS System restricts unauthorised users from accessing, viewing, and changing data and reports not relevant to their role.
NFR010j	Access Security	NFR010j - Ability to report on access	The VDS System enables the systems operator to generate reports on users' access patterns to monitor inappropriate access.
NFR011a	Infrastructure	NFR011a - VDS environments	The following environments shall be developed and deployed for VDS which provides a easy to use, debugging and error reporting tools. • Development • Test • Systems Integration Testing • User Acceptance Test • Staging • Production
NFR013a	Audit	NFR013a - Generate Audit information	The VDS System will be able to log activities and provide audit information as needed. • The VDS System will be able to generate all audit information supporting all historical logging for a specified user function.
NFR013b	Audit	NFR013b - Generate business rules	The VDS System will be able to generate a version of the Business Rules.
NFR013c	Audit	NFR013c - Generate changes to business rules	The VDS System will be able to generate a record of all changes to Business Rules, including the specific changes that were made, the user that undertook the change, and the date and time the change was made.
NFR013d	Audit	NFR013d - User access audit log	The VDS System will be able to generate an audit log describing when a user accessed functionality with the integrity of such audit logs maintained to prevent any deletion or modification.
NFR013e	Audit	NFR013e - Information access audit log	The VDS System will be able to generate an audit log describing what information was accessed by who, when and what action was undertaken.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

NFR013f	Audit	NFR013f - Systematic sampling	The VDS System will allow for • Systematic sampling • Program monitoring • Segment level compliance reports for programs
NFR014a	Reliability	NFR014a - Rectify issues	The VDS System will consistently perform its functions without failure. • The VDS System will monitor its performance and health and shall have the ability to rectify issues with the operation of itself with little or no human intervention.
NFR014b	Reliability	NFR014b - Event Management Plan	The VDS System shall have comprehensive event management plan which addresses the entire event lifecycle (including event occurrence, event notification, event detection, event logging, event filtering and correlation), and event response.
NFR015a	Business Continuity	NFR015a - Secondary site operation	The VDS System will support business continuity, such that if there is a disaster at the primary location, the VDS System can continue operations from a secondary site for business continuity purposes.
NFR016a	Disaster Recovery	NFR016a - Recovery Mechanisms	The VDS System will support disaster recovery for all hosting services and systems. Three major recovery mechanisms must include Geo-Replication, Geo-Recovery, and Geo-Restore, to warrant that fail over of primary services and/or databases to a secondary counterpart in the event of a datacentre or regions downtime: • During a failover process, primary services and databases must fail over to secondary services and databases in different regions. • Geo-Restore must be enabled to recover a database from automatically maintained geo-redundant backups. • Repatriation and Geo-Replication must be used to repatriate updated databases to the primary database in their original regions.
NFR016b	Disaster Recovery	NFR016b - Defined tier levels for disaster recovery	The VDS System will support off-site recoverability based on a defined tier level for each of the VDS System's core functions, e.g. to support transaction integrity and data loss within the target RPO (per NFR17).
NFR016c	Disaster Recovery	NFR016c - Maximum allowable outage (MAO)	The maximum allowable outage (MAO) in the event of a disaster for all system components is five (5) hours. Within this the execution of the recovery for the IT systems should take no longer than two (2) hours to execute, which allows three (3) hours for the declaration of a disaster

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

NFR017a	Data Retention and Recovery	NFR017a - Offline Backups	The VDS System will support data recovery and retention by storing secure offline backups in a remote location in accordance with the data classification 'UNCLASSIFIED DLM: SENSITIVE-PERSONAL'.
NFR017b	Data Retention and Recovery	NFR017b - Recovery time objective (training organisation)	In the event of a system crash, the Recovery Time Objective will be within 4 hrs from the time of the system crash.
NFR017c	Data Retention and Recovery	NFR017c - Recovery point objective (RPO)	In the event of a system crash, the Recovery Point Objective will be the most recent 5-minute checkpoint before the system crash.
NFR017d	Data Retention and Recovery	NFR017d - Daily backups	All server images and system data will be backed up on at least a daily basis.
NFR017e	Data Retention and Recovery	NFR017e - Backup retention period	A full monthly backup must be retained securely for a period of seven (7) years.
NFR017f	Data Retention and Recovery	NFR017f - Audit log retention	The retention of audit logs must comply with the National Archives Australia (NAA) Administrative Functions Disposal Authority of a minimum of seven (7) years.
NFR018a	Records Management	NFR018a - Data storage	The VET data stored in the VDS System will be retained indefinitely.
NFR018b	Records Management	NFR018b - Access to data retained	The seven (7) number of years data is retained in the VDS system can be modified, in order to reflect industry policy and subject to the data amendment policy.
NFR019a	Offline Store and Forward	NFR019a - Offline working	The VDS System should work offline (when network connectivity is unavailable) so that work can be carried out and the system will sync with the offline data when network connectivity is available.
NFR020a	Maintainability	NFR020a - System enhancements	The VDS System can be supported, changed, enhanced, and restructured over time

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

~~13~~14 Appendices

Appendix 1. Acronyms

Term/Acronym	Expanded Form
ACSC	Australian Cyber Security Centre
AD	Active Directory
ADF	Azure Data Factory
ADLS	Azure Data Lake (Gen2) Storage
AI	Artificial Intelligence
API	Application Programming Interface
APIM	Application Programming Interface Manager
ASQA	Australian Skills Quality Authority
AVETMISS	Australian Vocational Education and Training Management Information Statistical Standard
BAU	Business as usual
BI	Business Intelligence
CARMS	Classification and Reference Management System
CMS	Contract Master Schedule
DESE	Department of Education, Skills and Employment
DevOps	Development Operations
DEWR	Department of Employment and Workplace Relations
DIU	Data Integration Units
ETL	Extract, Transform and Load
FFS	Fee For Service
FA	Funded Activity
FR	Functional Requirement
Iaas	Infrastructure as a Service
IAM	Identity and Access Management
ICT	Information and Communication Technology
IRAP	Information Security Registered Assessors Program
ISM	Information Security Manual
ITIL	Information Technology Infrastructure Library

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 706 of 1295

Term/Acronym	Expanded Form
IVETS	Integrated Vocational Education and Training System (NSW Dept of Education)
JSON	JavaScript Object Notation
JWT	JSON Web Token
LAW	Log Analytics Workspace
ML	Machine Learning
MDM	Master Data Management
NCVER	The National Centre for Vocational Education Research
NCI	National Careers Institute
NESM	New Employment Services Model
NFR	Non-Functional Requirements
NSA	National Skills Agreement
NSC	National Skills Commission
NVA	Network Virtual Appliance
NVETR	National Vocational Education and Training Regulator
NFR	Non - Functional Requirement
OIDC	OpenID Connect
PaaS	Platform as a Service
PIA	Privacy Impact Assessment
PoC	Proof of Concept
PMP	Project Management Plan
PR	Production Environment
PRISM	Provider Registration and International Student Registration System
PSPF	Protective Security Policy Framework
RAAIDD	Risk Actions, Assumptions, Issues, Decisions, and Dependencies
RBAC	Role Based Access Control
REST	Representational State Transfer
RG	Resource Group
RTO	Registered Training Organisation
SaaS	Software as a Service
SAML	Security Assertion Markup Language

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 707 of 1295

Term/Acronym	Expanded Form
SDK	Software Development Kit
SDLC	Software Development Life Cycle
SEE	Skills for Education and Employment
SHIR	Share-Hosted Integration Runtime
SKU	Stock Keeping Unit
SMS	Student Management System
SSO	Single Sign On
SSON	Skills Senior Officials' Network
STA	State Training Authority
STARS	Student Training & Activity Reporting System
TGA	Training.gov.au (National VET directory website)
TP	Training Provider
TLS	Transport Layer Security
TRA	Trades Recognition Australia
TVA	Total VET Activity
UAT	User Acceptance Testing
USI	Unique Student Identifier
VDP	Unique Student Identifier
VDS	VET Data Streamlining
VET	Vocational Education and Training
VIS	VET Information Standard
VM	Virtual Machine
VNet	Virtual Network
VSL	VET Student Loans
VVCT	VSL, VET Compliance and TRA
WCAG	Web Content Accessibility Guidelines

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 708 of 1295

Australian Government
Department of Employment and Workplace Relations

Detailed Design Document

R&A Provisioning Service

1 Document Management

1.1 Document Control

Version	Change Description	Date
0.1	Draft Version	06 Mar 2023
0.2	Updates from design review	29 Mar 2023
1.0		

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This is a living document and will be updated as required.

Document Author

Role	Name
Data Architect/Tech Lead	s 47F(1)
Data Engineer	s 47F(1)

Document Reviewers

Role	Name

Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

Related Documents

Reference	Document
A	STARS Solution Requirements Document – V.1.3.docx
B	STARS Solution Design – V.1.0.docx
C	VET Data Access and Privacy Protocol.docx
D	R&A Provisioning Service Diagrams - V0.2~~1~~.vsdx
E	Fact-Dimension tables.xlsx
G	Detailed Design Document - Master Data Service D.0.2.docx
H	R&A DW ERD V0.1.vsdx
I	Detailed Design Document – Data Quality Framework D.0.1.docx

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 710 of 1295

Contents

1 Document Management 2
1.1 Document Control 2
Document Author 2
Document Reviewers 2
Document Approval 2
Related Documents 2
2 Introduction 76
2.1 Purpose 76
2.2 Intended Audience 76
3 Scope and Exclusions 87
3.1 In Scope 87
3.2 Out of Scope 87
4 Assumptions 98
5 Design Decisions 98
6 Feature Overview 119
7 R&A Provisioning Solution Components 1310
7.1 Data Source 1411
7.2 Target Database 1513
7.3 Data Refresh from Master database to R&A database 1913
7.3.1 Data Load – ADF Pipeline 2015
7.4 Database Availability 2317
7.5 ADF Logging and Exception Handling 2418
7.5.1 Logs 2418
7.5.2 Alerting and Notification 2418
7.5.3 Exception Handling 2418
7.5.4 Custom Logging 2418
7.5.5 Metadata Extraction and Cataloguing 2519
7.6 ADF Pipeline Monitoring 2519
8 Access Management 2620
8.1 Control of data 2720
8.1.1 Data Owner 2720
8.1.2 Data Custodian 2720
8.2 Security 2721

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 711 of 1295

9 Dependency on other STARS Services 2821

9.1 Master Data Service 2821

9.2 Notification Service 2821

9.3 Data Quality Framework 2822

9.4 IDAM 2822

10 Functional Requirements 2923

11 Non-Functional Requirements 2923

12 Appendix 3125

12.1 Supporting documents 3125

13 Acronyms 3226

~~1 Document Management ..... 2~~

~~1.1 Document Control ..... 2~~

~~Document Author ..... 2~~

~~Document Reviewers ..... 2~~

~~Document Approval ..... 2~~

~~Related Documents ..... 2~~

~~2 Introduction ..... 6~~

~~2.1 Purpose ..... 6~~

~~2.2 Intended Audience ..... 6~~

~~3 Scope and Exclusions ..... 7~~

~~3.1 In Scope ..... 7~~

~~3.2 Out of Scope ..... 7~~

~~4 Assumptions ..... 7~~

~~5 Design Decisions ..... 8~~

~~6 Issues ..... 8~~

~~7 Dependencies ..... 9~~

~~8 Feature Overview ..... 10~~

~~9 R&A Provisioning Solution Components ..... 11~~

~~9.1 Data Source ..... 12~~

~~9.2 Target Database ..... 12~~

~~9.3 R&A Star Schema Model ..... 13~~

~~9.3.1 Logical Star Schema Model ..... 13~~

~~9.3.2 Physical Star Schema Model ..... 17~~

~~9.3.3 R&A Staging tables ..... 18~~

Formatted: Default Paragraph Font, Check spelling and grammar

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 712 of 1295

9.3.4 Dimension and Fact tables 18
9.4 Data Refresh from Master database to R&A database 19
9.4.1 Data Load – ADF Pipeline 20
9.5 R&A Database Auditing and Version History 22
9.6 R&A Database Tables Nullability 22
9.7 ADF Logging 23
9.7.1 Logs 23
9.7.2 Alerting and Notification 23
9.7.3 Exception Handling 23
9.7.4 Custom Logging 23
9.7.5 Metadata Extraction and Cataloguing 24
9.8 ADF Pipeline Monitoring 24
10 Access Management 25
10.1 Control of data 25
10.1.1 Data Owner 25
10.1.2 Data Custodian 25
10.2 Security 25
11 Dependency on other STARS Services 26
11.1 Master Data Service 26
11.2 Notification Service 26
12 Functional Requirements 27
13 Non Functional Requirements 27
14 Appendix 28
14.1 Supporting documents 28
15 Acronyms 29

Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Font: Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Font: Check spelling and grammar
Formatted: Default Paragraph Font, Font: Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted: Default Paragraph Font, Check spelling and grammar
Formatted ...
Formatted ...
Formatted ...

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 713 of 1295

Figure 1: R&A Feature Overview (D1) 129
Figure 2: STARS Azure Data Estate (D2) 1310
Figure 7: Data Quality Aggregated Threshold Check (D7) 1914
Figure 8: ADF Delta Load (D8) 2015
Figure 9: Azure Data Factory Monitoring (Pipeline and activity runs logs) (D9) 2619
Figure 10: Azure Data Factory Monitoring (Results of a debug run) (D9) 2620
Figure 11: Azure Data Factory Monitoring (Rerun pipeline) (D9) 2620
~~Figure 1: R&A Feature Overview (D1) ..... 10~~
~~Figure 2: STARS Azure Data Estate (D2) ..... 11~~
~~Figure 3: Fact Subject Enrolment (D3) ..... 13~~
~~Figure 4: Fact Program Enrolment (D4) ..... 14~~
~~Figure 5: Fact Subsidy Funding Arrangement (D5) ..... 15~~
~~Figure 6: Fact Work Placement (D6) ..... 16~~
~~Figure 7: Data Quality Aggregated Threshold Check (D7) ..... 19~~
~~Figure 8: ADF Delta Load (D8) ..... 20~~
~~Figure 9: Azure Data Factory Monitoring (D9) ..... 25~~

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 714 of 1295

2 Introduction

This document covers the design of the STARS Reporting & Analytics (R&A) Provisioning Service.

Note: This is a living document that will be updated as designs for various other STARS services are developed and evolve. Accordingly, this document will evolve to reflect and document R&A provisioning components as the STARS solution build progresses.

The design is informed by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework, which provide a comprehensive set of best practices and guidance for designing, deploying, and operating secure, high-performing, and resilient cloud infrastructure and applications.

R&A Provisioning service detail design will cover the tools & technology that needs to be provisioned for enabling reporting and analytics capability in STARS. This design document will include~~s~~

- the automated and continuous population of R&A database using Azure Data Factory for the specific purpose of supporting current and as-at reporting requirements.
- Data load mechanism using ADF pipelines
- ADF pipeline monitoring tool to monitor all pipeline runs

Please note that the reporting requirements will be captured in separate detail design "Downstream Consumer Reporting and Analytics".

2.1 Purpose

The purpose of this detailed design document is to provide a technical description of the R&A provisioning Service. ~~Reporting and Analytics database will be designed and implemented to offload national data reporting and visualisation activities.~~ The document is intended for technical stakeholders, including architects, software engineers, administrators, and operations teams, who are responsible for designing, developing, deploying, and operating the STARS solution in Azure.

The document also serves as a reference for ongoing management and evolution of the STARS solution and to ensure that the STARS solution is deployed and operated in accordance with the best practices and guidance provided by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework.

The document is intended to be a living document, updated as the STARS solution evolves and changes over time. It provides a clear and concise understanding of the Azure infrastructure and networking components and their configurations, helping to ensure the STARS solution is deployed, operated, and managed in a consistent and efficient manner.

2.2 Intended Audience

This detailed design document provides technical information and guidance to help technical stakeholders understand the R&A provisioning Service components of the STARS solution and how

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 715 of 1295

they integrate with each other. The document is expected to be used by technical stakeholders who have a good understanding of cloud computing concepts and are experienced in the Azure platform.

This document is not intended for non-technical readers, nor for readers who are not experienced in the design and configuration of Azure-based solutions; informational and educational resources regarding Azure are readily available at https://Azure.microsoft.com

The primary audience for this document includes:

- Program technical staff: This includes architects, software engineers, administrators, and operations teams within the program responsible for the STARS solution.
- External technical stakeholders: This includes technical staff from other federal, state, and territory governments agencies, including VDS embedded team members.
- ICT Technical Staff: This includes a variety of technical roles involved during the VDS program and post-go live, including network engineers, systems administrators, database administrators, security engineers, and DevOps engineers.

3 Scope and Exclusions

3.1 In Scope

The following are in-scope of this detail design solution:

- Data Ingestion from master database into R&A database using Azure Data Factory
- ADF pipeline monitoring and logging
- ~~Creation of star schema driven from the VET-IS Logical Schema~~
- ~~Creation of dimensions and facts tables in R&A database~~
- ~~Data transformation of master data into dimension and facts table in R&A database~~
- ~~Star schema model to enable as-at reporting requirements~~

3.2 Out of Scope

The following are out of scope for this document:

- Creation of star schema driven from the VET-IS Logical Schema (will be covered in Downstream Consumer Reporting and Analytics Detail Design)
- Data enrichment or derived columns in R&A database (will be covered in Downstream Consumer Reporting and Analytics Detail Design)
- Creation of dimensions and facts tables in R&A database (will be covered in Downstream Consumer Reporting and Analytics Detail Design)
- Data transformation of master data into dimension and facts table in R&A database (will be covered in Downstream Consumer Reporting and Analytics Detail Design)

Formatted: Bulleted + Level: 1 + Aligned at: 0.63 cm + Indent at: 1.27 cm

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 716 of 1295

- Creation of stored procedures to apply above data transformation (will be covered in Downstream Consumer Reporting and Analytics Detail Design)
- Star schema model to enable as-at reporting requirements (will be covered in Downstream Consumer Reporting and Analytics Detail Design)
- Data Quality aggregated ~~threshold~~ data checks that are applied on the master database, this is covered in Data Quality Framework detail design
- Historical data requirement and aggregated data output check result to be stored in Master database or R&A database to be covered in Data Quality Framework detail design
- Reporting requirements on R&A database, this will be covered in Downstream Consumer Reporting and Analytics workshop
- Reference Data requirements for reporting will be covered in Downstream Consumer Reporting and Analytics

4 Assumptions

Id #	Assumption
A001	Assume Aggregated threshold checks have already applied to the source transactional database, these checks are covered in Data Quality Framework detail design.
A002	Assume Azure Data Factory will be used as a tool for data ingestion and orchestration.
~~A003~~	~~Assume no data quality checks are required when populating data in R&A database.~~
A00~~3~~4	Assume source mater data to be ingested that have changed ~~is~~are not to be merged with the target reporting and analytics data to maintain version history of the source records.

5 Design Decisions

The STARS solution will utilise Service Oriented Architecture (SOA) principles for design and implementation, coupled with Master Data Management (MDM) principles for population of the Master Database. The solution will be hosted in Microsoft Azure and will conform to the Microsoft Well-Architected Framework[1], utilising recognised Azure Reference Architectures[2].

The architecture and design principles are described in detail in STARS Solution Design document.

Of note is that the entire STARS solution will be implemented within the Microsoft Azure cloud platform and will heavily utilise Platform as a Service (PaaS) offerings. Accordingly, the Solution Design

[1] https://docs.microsoft.com/en-us/azure/architecture/framework/
[2] https://docs.microsoft.com/en-us/azure/architecture/

is oriented around utilising Azure features and resources that support the design and implementation of loosely coupled services. These include, for example, having services implemented in separate Resource Groups, using features such as Azure Functions, Azure SQL, and service-specific Azure App configurations.

Loose coupling will be achieved using Message- and Event-Driven approaches for communication between services, using features such as Azure Service Bus and Azure Event Grid. Services that present HTTPS/REST/JSON APIs will have their URLs controlled through Azure Application Gateway. This will enable a consistent URL scheme that also facilitates loose coupling.

Table 11~~1~~: Design Decisions/Action Items

Decision #	Design Decision	Status
DD01	Data Load into R&A database schedule/trigger frequency depends on the aggregated threshold data quality checks and reporting requirements. Action item 3381	Open
DD02	Data enrichment or derived columns in R&A database depends on the reporting requirements from NCVER as part of their WP02. Action item 3995	Open

~~6 Issues~~

~~After analysing the Master database physical ERD, we have come up with the Star schema model based on all the tables in Master database (requirement is to get all data from Master database that is gold ready). We have identified few issues that needs to be discussed and addressed in this Star Schema.~~

1. ~~Many to many relationships in relationship tables between the master database tables is causing structural issues in the Dimension and fact tables. To be confirmed if we need these relationship tables denormalized into star schema model.~~
2. ~~Size will be huge because of duplication of data as we try to demoralize the entire database, instead creating of some specific reporting requirements based dim and fact tables.~~

~~7 Dependencies~~

~~Following are the dependencies for R&A Provisioning service:~~

- ~~Reporting & Analytics Database Star Schema model will be based on the VIS model in the master database. Any changes to the VIS model will impact the R&A Database structure and will need to be updated as well.~~
- ~~Populating Reporting & Analytics database depends on the aggregated threshold checks performed on the Master database. If the Master database records passes the quality~~

~~checks, then only the data gets loaded into R&A database. These checks are covered in Data Quality Framework detail design.~~

~~8~~6 Feature Overview

R&A Provisioning service in STARS will include the automated and continuous population of the Reporting and Analytics database for the specific purpose of supporting current and as-at reporting requirements. This involves the conversion of normalised master database structures into dimensional models which are optimised for analytics workloads.

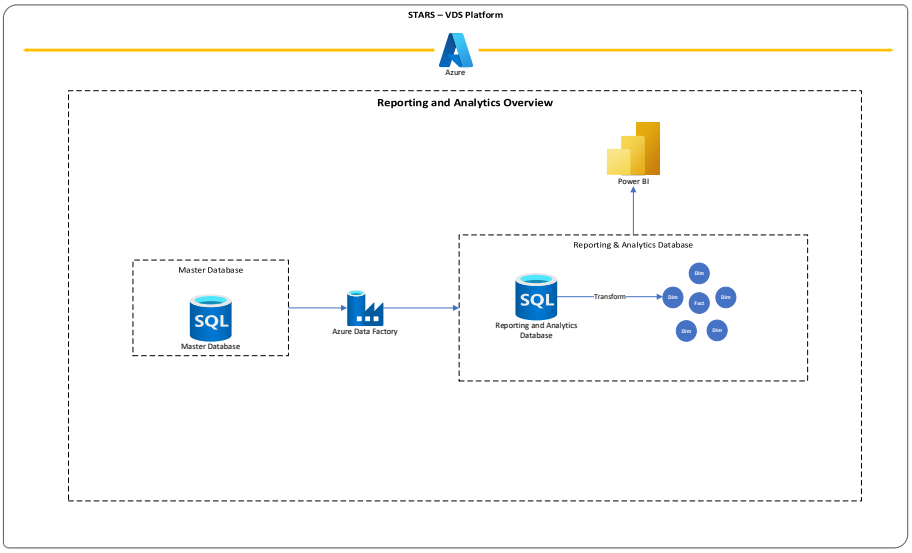


Figure 111: R&A Feature Overview (D1)

For handling of as-at reporting requirements in the R&A database, this service in conjunction with Downstream Consumer Reporting and Analytics (future design) will implement:

- Data Ingestion, load, and monitoring process
- The star schema driven from the VET-IS Logical Schema
- The creation of date and time dimensions
- The creation of slowly changing dimensions (type-2 SCD's) that provide the point in time data selection capabilities
- The fact tables will maintain version history of the mastered records. Capturing versions of the fact table data will enable STARS to analyse the facts as they looked at a chosen point in time. The fact table changes will simply be treated like SCD type 2 changes that are commonly used with dimension processing. Data in fact tables will be stored at a fine-grained level i.e., individual transactions from master database which will have a high level of granularity.
- Transformation of master data into dimension/fact star schemas to optimise reporting workloads

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 720 of 1295

97 R&A Provisioning Solution Components

The below picture shows overall high-level STARS Azure data~~target~~ estate ~~data~~ architecture, this document will cover the R&A Provisioning solution components (numbered 1 & 2).

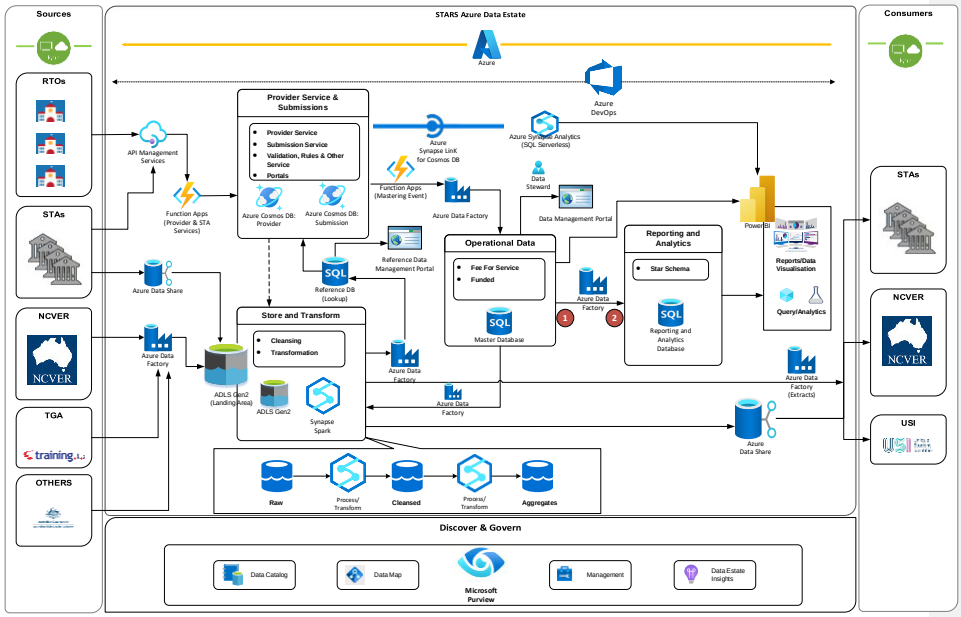


Figure 222: STARS Azure Data Estate (D2)

Table 222: R&A Provisioning Steps

Step No	Description	Technologies
1	Azure Data Factory will load the gold ready and release data from source master database into staging tables in R&A database	Azure Data Factory, Azure SQL Database
2	Azure Data Factory will transform master data from staging tables to dimension and fact tables in R&A database	Azure Data Factory, Azure SQL Database

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 721 of 1295

~~9.1~~7.1 Data Source

STARS Master database will be implemented in Azure SQL Database. Server and database details for each environment as below.

Please note there will be multiple master databases as below, the R&A ADF process will load data from all the databases into R&A Staging tables and after data load completion transformation to dim and fact tables will take place. The staging table's structure will be defined in Downstream Consumer Reporting and Analytics (future design).

Table ~~33~~4: STARS Master Database Environment

Formatted: Normal

Environment	Details*
Dev	Server Name: vds-stars-aue-sqls-master-dev Database Name: vds-stars-sqldb-master-sa-dev Database Name: vds-stars-sqldb-master-nsw-dev Database Name: vds-stars-sqldb-master-nt-dev Database Name: vds-stars-sqldb-master-qld-dev Database Name: vds-stars-sqldb-master-act-dev Database Name: vds-stars-sqldb-master-tas-dev Database Name: vds-stars-sqldb-master-wa-dev Database Name: vds-stars-sqldb-master-vic-dev Database Name: vds-stars-sqldb-feeforservice-dev
Test	Server Name: vds-stars-aue-sqls-master-test Database Name: vds-stars-sqldb-master-sa-test Database Name: vds-stars-sqldb-master-nsw-test Database Name: vds-stars-sqldb-master-nt-test Database Name: vds-stars-sqldb-master-qld-test Database Name: vds-stars-sqldb-master-act-test Database Name: vds-stars-sqldb-master-tas-test Database Name: vds-stars-sqldb-master-wa-test Database Name: vds-stars-sqldb-master-vic-test Database Name: vds-stars-sqldb-feeforservice-test
SIT	Server Name: vds-stars-aue-sqls-master-sit Database Name: vds-stars-sqldb-master-sa-sit Database Name: vds-stars-sqldb-master-nsw-sit

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 722 of 1295

	Database Name: vds-stars-sqldb-master-nt-sit Database Name: vds-stars-sqldb-master-qld-sit Database Name: vds-stars-sqldb-master-act-sit Database Name: vds-stars-sqldb-master-tas-sit Database Name: vds-stars-sqldb-master-wa-sit Database Name: vds-stars-sqldb-master-vic-sit Database Name: vds-stars-sqldb-feeforservice-sit
UAT	Server Name: vds-stars-aue-sqls-master-uat Database Name: vds-stars-sqldb-master-sa-uat Database Name: vds-stars-sqldb-master-nsw-uat Database Name: vds-stars-sqldb-master-nt-uat Database Name: vds-stars-sqldb-master-qld-uat Database Name: vds-stars-sqldb-master-act-uat Database Name: vds-stars-sqldb-master-tas-uat Database Name: vds-stars-sqldb-master-wa-uat Database Name: vds-stars-sqldb-master-vic-uat Database Name: vds-stars-sqldb-feeforservice-uat
Production	Server Name: vds-stars-aue-sqls-master-prd Database Name: vds-stars-sqldb-master-sa-prd Database Name: vds-stars-sqldb-master-nsw-prd Database Name: vds-stars-sqldb-master-nt-prd Database Name: vds-stars-sqldb-master-qld-prd Database Name: vds-stars-sqldb-master-act-prd Database Name: vds-stars-sqldb-master-tas-prd Database Name: vds-stars-sqldb-master-wa-prd Database Name: vds-stars-sqldb-master-vic-prd Database Name: vds-stars-sqldb-feeforservice-prd

**Subject to change during build*

~~9.2~~7.2 Target Database

STARS R&A database will be implemented in Azure SQL Database. Server and database details for each environment as below.

Table ~~44~~5: STARS R&A Database Environment

Environment	Details*
Dev	Server Name: vds-stars-aue-sqls-rna-dev

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 723 of 1295

	Database Name: vds-stars-sqldb-rna-dev
Test	Server Name: vds-stars-aue-sqls-rna-test Database Name: vds-stars-sqldb-rna-test
SIT	Server Name: vds-stars-aue-sqls-rna-sit Database Name: vds-stars-sqldb-rna-sit
UAT	Server Name: vds-stars-aue-sqls-rna-uat Database Name: vds-stars-sqldb-rna-uat
Production	Server Name: vds-stars-aue-sqls-rna-prd Database Name: vds-stars-sqldb-rna-prd

**Subject to change during build*

~~9.3 R&A Star Schema Model~~

~~Reporting & Analytics Database Star Schema model will be based on the VIS model in the master database.~~

~~9.3.1 Logical Star Schema Model~~

~~9.3.1.1 Fact Subject Enrolment~~

~~Figure 3: Fact Subject Enrolment (D3)~~

~~9.3.1.2 Fact Program Enrolment~~

~~Figure 4: Fact Program Enrolment (D4)~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 724 of 1295

~~9.3.1.3 Fact Subsidy Funding Arrangement~~

~~Figure 5: Fact Subsidy Funding Arrangement (D5)~~

~~9.3.1.4 Fact Work Placement~~

~~Figure 6: Fact Work Placement (D6)~~

9.3.2 Physical Star Schema Model

9.3.3 R&A Staging tables

Staging tables will be created under <staging> schema in R&A database. Data from master database will be loaded into these staging tables before they are loaded into Star schema dimension and fact tables.

All staging tables will be of similar structure to the master tables, the staging tables will hold all the source columns including primary key for transformation purposes in R&A database. The source records that have changed (based on 'LastModifiedDateTime' and 'IsGoldReady') will be extracted from Master database into the staging tables in R&A database.

Based on the changed records and the primary keys, the data will be inserted/updated in the R&A database dim and fact tables.

9.3.4 Dimension and Fact tables

Below are the dimension tables that will be provisioned in R&A database with their Slowly Changing Dimension types. All the dimension tables (except Date dimension) will be SCD type 2 to keep history of changes to the dimension data that will enable users to analyse the data as it existed at different point in time.

Table 7: R&A Dimension Tables

Subject Area	Dimension Table	SCD Type
Stakeholder	DimGovernment	SCD Type 2
Stakeholder	DimOrganisation	SCD Type 2
Stakeholder	DimPerson	SCD Type 2
Location	DimPlace	SCD Type 2
Location	DimTelecommunicationAddress	SCD Type 2
Location	DimPostOfficeAddress	SCD Type 2
Location	DimStructureAddress	SCD Type 2
Location	DimSemiStructureAddress	SCD Type 2
Location	DimElectronicAddress	SCD Type 2
Product	DimSubject	SCD Type 2

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 726 of 1295

~~Product~~	~~DimProgram~~	~~SCD Type 2~~
~~Responsibility~~	~~DimEnrolment~~	~~SCD Type 2~~
~~Responsibility~~	~~DimResponsibility~~	~~SCD Type 2~~
~~Responsibility~~	~~DimResponsibilityFinancialDetail~~	~~SCD Type 2~~
	~~DimDate~~	~~NA (Static data)~~

~~Below are the fact tables that will be provisioned in R&A database, version history will be maintained in the fact tables as well to enable point in time reporting.~~

~~**Table 8: R&A Fact Tables**~~

~~Subject Area~~	~~Dimension Table~~
~~Responsibility~~	~~FactSubjectEnrolment~~
~~Responsibility~~	~~FactProgramEnrolment~~
~~Responsibility~~	~~FactSubsidyFundingArrangement~~
~~Responsibility~~	~~FactWorkPlacement~~

~~9.4~~7.3 Data Refresh from Master database to R&A database

Data Load into R&A database depends on the data quality checks that are performed on the Master database. The records that are quality checked and gold ready data are only copied into the R&A database. The following data flow shows the data quality aggregated threshold checks process implementation in STARS.

~~2~~ ?

Figure 3~~3~~7: Data Quality Aggregated ~~Threshold~~ Data Check (D7)

This document will cover steps 6 & 7 in detail shown in the above diagram. Steps 1 to 5 are covered in Data Quality Framework detail design.

Table 55~~10~~: R&A Provisioning Steps

Step No	Description	Technologies
1 & 2	Azure data factory will run the Azure spark notebook that will extract data from Reference, Reporting & Analytics and Master database required for Aggregated ~~threshold~~ data check	Azure Synapse Spark, Azure SQL Database, Azure Data Factory

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 727 of 1295

Step No	Description	Technologies
3	Azure spark will process the aggregated ~~threshold~~ data checks on the master data and publish the results back into the master database for further processing by setting the flag 'IsGoldready' to true for the records that have passed aggregated data checks.	Azure Synapse Spark, Azure SQL Database, Azure Data Factory
4	After successful completion of steps 1 to 3, Azure Data factory will trigger Power BI dataset refresh via Web activity task	Azure Data factory
5	Power BI imports the required data from Master database into its dataset for producing data quality reports to review data trends	Azure SQL Database, Power BI
6	After successful completion of steps 1 to 3, Azure Data factory triggers ADF pipeline to load gold ready data from Master to R&A database for further reporting	Azure SQL Database, Azure Data Factory
7	ADF pipeline extracts the gold ready and release data from the Master database and transforms the data into dimension and fact tables in R&A database (dimension and fact tables structure will be covered in Downstream Consumer Reporting and Analytics Service detail design.)	Azure SQL Database, Azure Data Factory

~~9.4.1~~7.3.1 Data Load – ADF Pipeline

~~9.4.1.1~~7.3.1.1 Incremental Loads

Azure Data Factory will be extracting data (delta load) from Master database to R&A database as a daily refresh or manually on demand (open Action item 3381). Incremental delta loading will be done using Watermark table explained further in this section.

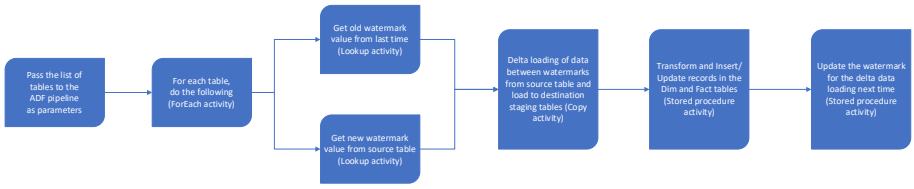


Figure ~~4~~48: ADF Delta Load (D8)

~~9.4.1.2~~7.3.1.2 Steps to create this solution:

1. Select the watermark column: 'Last Modified Time' and 'Is Gold Ready' will be selected in the source tables, which will be used to identify the new or updated gold ready records for every run.
2. Store the Watermark value: This value will be stored in the Azure SQL R&A database.
3. Create ADF pipelines for data load from Master database.
    a. Create a ForEach activity that will iterate through a list of source table names that is passed as a parameter to the pipeline. For each source table, it will invoke

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 728 of 1295

the following activities to perform delta loading for that table.

b. Create two lookup activities. Use the first Lookup activity to retrieve the last watermark value. Use the second Lookup activity to retrieve the new watermark value. These watermark values will be passed to the Copy activity.
c. Create a Copy activity that will copy rows from the source data store with the value of the watermark column greater than the old watermark value and less than the new watermark value. Then, it will copy the delta data from the source data store to Azure Data Lake as a new file.
d. Create a StoredProcedure activity that will Transform and Insert/Update records in the Dim and Fact tables. Based on the source changed records and the primary keys, the data will be inserted/updated in the R&A database dim and fact tables.
e. Create a StoredProcedure activity that will update the watermark value for the pipeline that runs next time.

~~9.4.1.3~~7.3.1.3 Watermark table structure:

Table ~~66~~11: Table name ~~-~~– etl.Watermark

Field Name	Data type
WatermarkKey	Integer, IDENTITY (1, 1) NOT NULL
TableName	NVARCHAR(255)
WatermarkValue	DATETIME
LastStartTime	DATETIME
LastEndtime	DATETIME
LastProcessStatus	NVARCHAR(25)

Sample SQL script:

```
CREATE TABLE etl.watermark
(
    WatermarkKey integer IDENTITY (1, 1) NOT NULL,
    TableName nvarchar(255),
    WatermarkValue datetime,
    LastStartTime datetime,
    LastEndTime datetime,
    LastProcessStatus nvarchar(255),
);
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 729 of 1295

7.3.1.4 Other ADF Data Load Functionality

- **Data Load from a specific period:** If for any reason data needs to be loaded from a specific period for a table or a set of tables, the system admin will update the Watermark table column *LastEndTime* appropriately for the required tables and then manually trigger the ADF pipeline load. To not change data already processed by the previous run of the data load, the following data transformation approach will be implemented in ADF or Stored procedure.
  - Lookup incoming new records against existing records in the Dimension/Fact table
  - If they are new, then create a new surrogate key and insert the row into the Dimension/Fact table
  - If the record already exists and the attributes have changed (trigger an update to the table if there is a change in the source record), then update the existing row
  - Add a new row in the Dimension/Fact table with the new data
- **Retry capability:** Azure Data Factory pipelines will be triggered at a periodic time based on the frequency requirement (tumbling window trigger). The retry option will be set to 3 in the ADF pipeline to automatically retry when the pipeline runs fail due to server/throttling limit issues etc.
- **Rerun a failed execution for a data load:** In case a pipeline fails after three retries, system administrator will be notified for remediation actions. The system admin will be able to rerun the pipeline from the point of failure from ADF monitoring dashboard.

Formatted: Indent: Left: 1.27 cm, No bullets or numbering
Formatted: Font: Bold
Formatted: Font: Italic
Formatted: Font: Bold
Formatted: List Paragraph,NFP GP Bulleted List,TOC style,lp1,Bullet OSM,Proposal Bullet List,List Paragraph1,Recommendation,List Paragraph11,List Paragraph111,L,F5 List Paragraph,Dot pt,CV text,Medium Grid 1 - Accent 21,Numbered Paragraph,Bullets,NAST Quote,lp11,n, Bulleted + Level: 2 + Aligned at: 1.9 cm + Indent at: 2.54 cm
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: List Paragraph,NFP GP Bulleted List,TOC style,lp1,Bullet OSM,Proposal Bullet List,List Paragraph1,Recommendation,List Paragraph11,List Paragraph111,L,F5 List Paragraph,Dot pt,CV text,Medium Grid 1 - Accent 21,Numbered Paragraph,Bullets,NAST Quote,lp11,n, Bulleted + Level: 1 + Aligned at: 0.63 cm + Indent at: 1.27 cm

~~9.5 R&A Database Auditing and Version History~~

~~All R&A Star Schema model tables will include the below attributes for auditing purposes.~~

- ~~Effective Date From~~
- ~~Effective Date To~~
- ~~Release Number~~
- ~~Version Number~~
- ~~ETL Batch Id~~
- ~~Created Date Time~~
- ~~Modified Date Time~~
- ~~Is Current~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 730 of 1295

~~Table 12: R&A Database Auditing~~

~~Audit Attribute~~	~~Data Type and Size~~	~~Nullability~~
~~EffectiveDateFrom~~	~~DATETIME2~~	~~NOT NULL~~
~~EffectiveDateTo~~	~~DATETIME2~~	~~NOT NULL~~
~~ETLBatchId~~	~~UNIQUEIDENTIFIER~~	~~NULL~~
~~ReleaseNumber~~	~~NVARCHAR(15)~~	~~NULL~~
~~VersionNumber~~	~~NVARCHAR(15)~~	~~NULL~~
~~CreatedDateTime~~	~~DATETIME2~~	~~NOT NULL~~
~~ModifiedDateTime~~	~~DATETIME2~~	~~NOT NULL~~
~~IsCurrent~~	~~TINYINT~~	~~NOT NULL (Default 1)~~ ~~1- Current~~ ~~0- Not Current~~

~~9.6 R&A Database Tables Nullability~~

~~All columns within the STARS R&A database will be defined as Nullable except for columns which belong to the primary key and audit columns (Table 12 above) of a table, to be consistent with the Master database.~~

7.4 Database Availability

R&A database will be implemented on Azure SQL database. Azure SQL Database offers a 99.995% availability SLA for zone redundant databases in its business-critical tier. This is the highest SLA in the industry among all relational database services. It is also backed by up to a 100% monthly cost credit for when the SLA is not maintained. Secondly, it offers a business continuity SLA for databases in the business-critical tier that are geo-replicated between two different Azure regions. That SLA comes with very strong guarantees of a five second recovery point objective (RPO) and a 30 second recovery time objective (RTO), including a 100% monthly cost credit when the SLA is not maintained. Azure SQL Database is the only relational database service in the industry offering a business continuity SLA.

Platform	Availability		Business continuity			
	Uptime	Max Credit	RTO	Max Credit	RPO	Max Credit

Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri)
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri)
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri)
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri)
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri)
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri)
Formatted Table
Formatted: Font: (Default) +Body (Calibri)
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri), 11 pt

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 731 of 1295

Azure SQL Database	99.995%	100%	30 seconds	100%	5 seconds	100%

~~9.7~~

~~9.8~~7.5 ADF Logging and Exception Handling

~~9.8.1~~7.5.1 Logs

Azure Data Factory (ADF) stores data of pipeline run ~~keeps the pipeline run data~~ for only 45 days by default with very limited information. Azure Monitor (AM) will be integrated with ADF to keep the logs data for ~~a longer time~~indefinite time and route diagnostic logs for further analysis. AM will provide higher visibility on the performance and health metrics of the workloads which will; be used for monitoring and trouble shooting.

Examples: Count of – pipeline failures, succeeded activities, failed trigger runs, IR failures, failed activities due to insufficient or unavailable memory for IR, etc.,

~~9.8.2~~7.5.2 Alerting and Notification

Alerts and notifications will be generated for STARS and Stakeholders by calling Notification Service API (UserNotification) from ADF pipeline to add a new notification to the Notification Service. The notification will be sent via email (depending on the user~~'~~s preferences) and can be retrieved via the Notification API.

For more details refer Notification Services detail design document.

~~9.8.3~~7.5.3 Exception Handling

ADF Pipelines fail for variety of reasons. While this is expected, it is important that the system safely recover from errors, they will be logged appropriately and that the system administrators will be notified immediately for remediation actions.

The activities in ADF have 'activity dependency statuses' such as success, failure, completion and skipped (condition based skipped) built-in which will be used for exception handling scenarios.

The system administrator will investigate the error reported in the ADF monitoring dashboards and take appropriate actions like rerun the pipeline from the point of failure or trigger the execution of ADF pipeline.

~~9.8.4~~7.5.4 Custom Logging

STARS R&A database will maintain a custom pipeline log table for storing the ADF pipeline activity details as records in the log table for each run.

Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri), 11 pt
Formatted: Font: (Default) +Body (Calibri), 11 pt

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 732 of 1295

Sample SQL script to create log table in Azure SQL R&A database:

```
CREATE TABLE [etl].[PipelineLog](
   PipelineLogKey bigint IDENTITY (1, 1) NOT NULL,
   DataFactoryName nvarchar(500) NULL,
   PipelineName nvarchar(500) NULL,
   RunId nvarchar(500) NULL,
   Source nvarchar(500) NULL,
   Destination nvarchar(500) NULL,
   TriggerType nvarchar(500) NULL,
   TriggerId nvarchar(500) NULL,
   TriggerName nvarchar(500) NULL,
   TriggerTime nvarchar(500) NULL,
   RowsCopied nvarchar(500) NULL,
   RowsRead int NULL,
   NumberOfParallelCopies int NULL,
   CopyDurationInSeconds nvarchar(500) NULL,
   EffectiveIntegrationRuntime nvarchar(500) NULL,
   SourceType nvarchar(500) NULL,
   SinkType nvarchar(500) NULL,
   ExecutionStatus nvarchar(500) NULL,
   CopyActivityStartTime datetime NULL,
   CopyActivityEndTime datetime NULL,
   CopyActivityQueuingDurationInSeconds nvarchar(500) NULL,
   CopyActivityTimeToFirstByteInSeconds nvarchar(500) NULL,
   CopyActivityTransferDurationInSeconds nvarchar(500) NULL,
) ;
```

~~9.8.5~~7.5.5 Metadata Extraction and Cataloguing

Microsoft Purview will be connected to the ADF for capturing the metadata lineage, and to discover and explore data assets across the STARS data platform. This will be covered in a separate document Purview – technical Detailed Design.

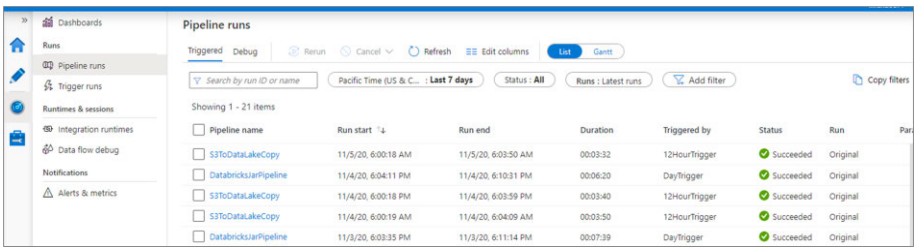
~~9.9~~7.6 ADF Pipeline Monitoring

Azure Data Factory (ADF) provides a monitoring portal for pipeline runs to track the status and performance of their data pipelines.

The monitoring portal displays information about pipeline runs, including the status of each activity within the pipeline, the duration of the run, and any error messages that may have occurred during execution. Users can view the status of individual runs or aggregate data across multiple runs to gain insight into the overall performance of their pipelines.

Please note that the below screenshots are just examples of ADF monitoring dashboard.

Formatted: Font: Bold



Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 733 of 1295

Figure 559: Azure Data Factory Monitoring (Pipeline and activity runs logs) (D9)

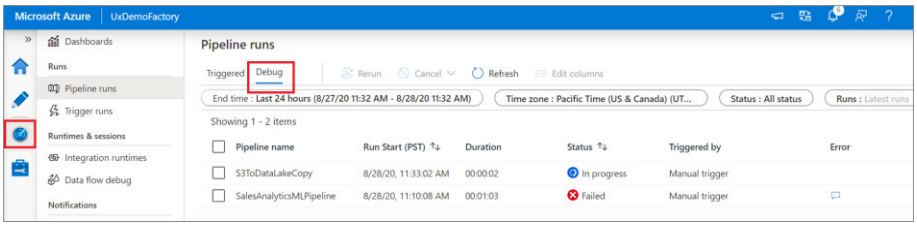


Figure 6610: Azure Data Factory Monitoring (Results of a debug run) (D9)

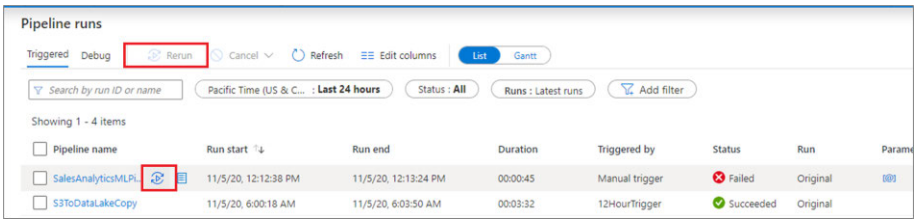


Figure 77119: Azure Data Factory Monitoring (Rerun pipeline) (D9)

Formatted: Normal, Left

In addition to monitoring pipeline runs, the monitoring portal also provides metrics and alerts to help users proactively identify and address issues with their pipelines. Users can set up custom alerts based on specific metrics, such as the number of failed pipelines run or the duration of a pipeline run exceeding a certain threshold. Users can also rerun a failed pipeline from the start.

The monitoring portal also includes a visual interface for monitoring pipeline dependencies, which allows users to identify and resolve issues with pipeline dependencies that may be impacting pipeline performance.

A separate R&A Provisioning Management Portal is therefore not required in STARS as this capability already exists in Azure Portal as above.

108 Access Management

The R&A Provisioning Service will conform to the Data Governance Access and Privacy Protocol document and the IDAM detail design. Access to the R&A Provisioning service will be controlled by Azure RBAC, with all calling services identified and granted access through Azure AD Managed Identities. For detail about accessing the R&A Provisioning Service please refer to IDAM detail design document.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 734 of 1295

~~10.1~~8.1 Control of data

The Governance Framework sets out the key data governance roles surrounding VET Data Streamlining. The two core central roles for accessing data from STARS are that of the Data Owner and the Data Custodian, summarised below:

~~10.1.1~~8.1.1 Data Owner

The Data Owner has authority for establishing rules surrounding the quality, protection, and appropriate access to data. They individually and collectively have a responsibility to protect the personal and sensitive information provided by Data Providers. Data can be jointly or co-owned.

~~10.1.2~~8.1.2 Data Custodian

The Data Custodians receive authorised data from Data Providers or from other Data Custodians. Data Custodians are authorised to collect and handle that data to perform their functions. They are responsible for ensuring that the personal information collected and stored is reasonably necessary for them to conduct their functions and activities in relation to the VET sector.

For detail about the data access and control please refer to the Data governance Access and Privacy protocol document.

~~10.2~~8.2 Security

The STARS solution will be built to meet an existing Australian Government policies, frameworks, standards, and design principles, including:

- Australian Government API Design Standards
- Digital Service Standard
- Protective Security Policy Framework (PSPF)
- Information Security Registered Assessors Program (IRAP)
- Australian Government Information Security Manual (ISM)
- the Open Web Application Security Project® (OWASP)
- Hosting Certification Framework.

All data within STARS will be handled in accordance with relevant privacy laws and record keeping requirements, including the Privacy Act, the *Archives Act 1983* (Cth) and any other applicable legislation (e.g., applicable State and Territory privacy legislation or archiving requirements).

The R&A Provisioning Service will conform to the Security Architecture defined for STARS in Data Governance Access and Privacy Control document and IDAM detail design document. Access to the R&A Provisioning Service will be controlled by Azure RBAC, with all calling services identified and granted access through Azure AD Managed Identities.

For detail about security please refer to the following

- Data Governance Access and Privacy Control document
- IDAM detail design document

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 735 of 1295

~~11~~9 Dependency on other STARS Services

Following are STARS services upon which R&A Provisioning Service has dependencies.

~~11.1~~9.1 Master Data Service

The R&A Provisioning Service will depend upon the Master service for populating gold ready master data into R&A database. Please refer to Master Data service detail design document for details about the Master Data Service.

~~11.2~~9.2 Notification Service

The R&A Provisioning Service will depend upon the Notification service for raising notifications. Please refer to Notification service detail design document for details about the Notification Service.

9.3 Data Quality Framework

Populating Reporting & Analytics database depends on the aggregated threshold checks performed on the Master database. If the Master database records passes the quality checks, then only the data gets loaded into R&A database. These checks are covered in Data Quality Framework detail design.

Formatted: Normal, No bullets or numbering

9.4 IDAM

Access to the R&A Provisioning Service will be controlled by Azure RBAC, with all calling services identified and granted access through Azure AD Managed Identities.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 736 of 1295

~~12~~10 Functional Requirements

Table 77~~13~~: Related Functional Requirements

FR Id	Category	Requirement Name	Description	Traceability
FR023	Data Consumption and Reporting	BI and Analytics	The VDS system must provide a BI and Analytics capability that can be utilised for reporting on data from the Master Database. This includes the ability to disaggregate the data. This is applicable for consumers who want to drill down to unit level records when they get aggregated data. Access to unit record data must be in accordance with privacy and confidentiality requirements.	Section 7~~Section 7 & 8~~

~~13~~11 Non-Functional Requirements

Table 88~~14~~: Related Non-Functional Requirements

NFR Id	Category	NFR Category	Requirement Name	Requirement Description	Traceability
NFR002a	VDS System and Data Governance	Availability	Minimum Uptime	The VDS System will support continual operations by being available and functioning with minimal downtime (High Availability). • The VDS System will support a minimum proposed uptime for the users, excluding scheduled downtime or outage.	Section 7.4

Formatted: Font: (Default) +Body (Calibri), Check spelling and grammar

Formatted: Font: (Default) +Body (Calibri), 11 pt, English (Australia), Check spelling and grammar

Formatted: Font: (Default) +Body (Calibri), Check spelling and grammar

Formatted: Font: (Default) +Body (Calibri), 11 pt, English (Australia), Check spelling and grammar

Formatted: Font: (Default) +Body (Calibri), Check spelling and grammar

Formatted: Space After: 5 pt, Line spacing: Multiple 1.15 li

Formatted: Font: (Default) +Body (Calibri), Check spelling and grammar

Formatted: Font: (Default) +Body (Calibri), 11 pt, English (Australia), Check spelling and grammar

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

~~14~~12 Appendix

~~14.1~~12.1 Supporting documents

Table ~~99~~15: Supporting Documents

Reference	Document
Azure Data Factory Monitoring portal	https://learn.microsoft.com/en-us/azure/data-factory/monitor-visually

~~15~~13 Acronyms

Table 101~~016~~: Acronyms

Acronym	Definition
AAD	Azure Active Directory
ADF	Azure Data Factory
ADLS	Azure Data Lake Storage
AM	Azure Monitor
ETL	Extract, Transform & Load
DEWR	Department of Employment and workplace Relations
FR	Functional Requirement
IDAM	Identity and Access Management
IR	Integration Runtime
IRAP	Information Security Registered Assessors Program
ISM	Australian Government Information Security Manual
MDM	Master Data Management
NCVER	The National Centre for Vocational Education Research
NCI	National Careers Institute
NESM	New Employment Services Model
NFR	Non-functional Requirement
OWASP	Open Web Application Security Project
PSPF	Protective Security Policy Framework
RPO	Recovery Point Objective
RTO	Recovery Time Objective
STARS	Student Training & Activity Reporting System
VET	Vocational Education Training
VDS	VET Data Streamlining
IVETS	Integrated Vocational Education and Training System

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 740 of 1295

DB	Database
DDL	Data Definition Language
R&A	Reporting and Analytics
RBAC	Role Based Access Control
RTO	Registered Training Organisation
SCD	Slowly Changing Dimension
SQL	Structured Query Language
SHIR	Self-Hosted Integration Runtime
SIT	System Integration testing
STA	State Training Authority
SMS	Student Management System
STARS	Student Training & Activity Reporting System
TGA	Training.gov.au (National VET directory website)
TP	Training Provider
UAT	User Acceptance Testing
USI	Unique Student Identifier
VDP	Unique Student Identifier
VDS	VET Data Streamlining
VET	Vocational Education and Training
VIS	VET Information Standard
VM	Virtual Machine
VNet	Virtual Network

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 741 of 1295

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 742 of 1295

Detailed Design Document

Release Service

1 Document Management

1.1 Document Control

Version	Change Description	Date
D.0.1	Initial Draft	03/04/2023

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This is a living document and will be updated as required. If another component's design scope requires an update to this design document, then the appropriate change will be made and a new version issued.

1.2 Document Author

Role	Name
Design Lead	s 47F(1)

1.3 Document Reviewers

Role	Name

1.4 Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

1.5 Related Documents

Reference	Document
A	STARS Solution Requirements V1.3.docx
B	STARS Solution Design - V.1.0.docx
C	STARS Solution Design - Attachment 1 - Collaboration Diagrams - V.1.0.vsdx
D	STARS Solution Design - Attachment 3 - API Design - V.1.0
E	Release Service reference diagrams D.0.1.vsdx
F	ReleaseAPI D.0.1.pdf
G	ReleaseAPI D.0.1.json
H	RelavantRequirementsForRelease D.0.1.xlsx

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 744 of 1295

2 Contents

1 Document Management 2
1.1 Document Control 2
1.2 Document Author 2
1.3 Document Reviewers 2
1.4 Document Approval 2
1.5 Related Documents 2
2 Contents 3
3 Introduction 5
3.1 Purpose 5
3.2 Intended Audience 6
4 Scope and Exclusions 7
4.1 In Scope 7
4.2 Out of Scope 7
5 Design Decisions 8
6 Requirements 9
7 Feature Overview 10
7.1 Data enrichment 11
7.1.1 Data enrichment rule 11
7.1.2 Data enrichment process 12
7.1.3 Use cases 13
7.2 Data release 16
8 Data Flow 17
8.1 Data enrichment flow 18
8.2 Data release flow 19
9 Technical Architecture 20
9.1 Release API 21
9.2 Release Workflow 21
9.3 Release Rules Engine 21
9.4 Release DB 21
9.5 Release Config 21
10 API Design 22
10.1 Data Enrichment Record Endpoint 22
10.2 STA Release Endpoint 22

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 745 of 1295

11 Release Service Workflows 23
11.1 Data enrichment check workflow 24
11.2 Data enrichment request workflow 24
11.3 Data release check workflow 25
11.4 Data release request workflow 25
12 Release Event Service Messages 26
13 Data 27
14 Security and Access Management 28
15 Acronyms 29

Figure 1: Release Service overview (E1) 10
Figure 2: Data enrichment business rule trigger (E2) 11
Figure 3: Data enrichment process (E2) 12
Figure 4: Data enrichment use case (E6) 13
Figure 5: STA Data release (E2) 16
Figure 6: STA Data release API call 16
Figure 7: Flow of data in the submission pipeline (E4) 17
Figure 8: Release Service Architecture (E3) 20
Figure 9: Release workflow (E5) 23

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 746 of 1295

3 Introduction

The purpose of this document is to capture the detailed design requirements for the STARS release service.

The release service facilitates:

- Data enrichment
  - Data enrichment can happen at any time to a record after it has passed submission validation
  - There can be business rules in place to hold a record in the submission pipeline after submission validations have passed for data enrichment
  - Validations happen on enriched data to make sure the record is valid after enrichment
- Data release
  - Certain Records are halted from entering the Reporting and Analytics database based on business rules entered and managed by STAs
  - These records are “released” to the Reporting and Analytics database when explicit “release” is received from the STA
- Store
  - The release service stores information pertaining to data release and data enrichment in the release database
- API
  - The release service API has endpoints to facilitate data release and data enrichment
  - It also allows for querying of data enrichment and data release information stored in the release database

Note: This is a living document that will be updated as designs for various other STARS services are developed and evolve. Accordingly, this document will evolve to reflect and document notification components as the STARS solution build progresses.

The design is informed by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework, which provide a comprehensive set of best practices and guidance for designing, deploying, and operating secure, high-performing, and resilient cloud infrastructure and applications.

3.1 Purpose

The purpose of this detailed design document is to provide a technical description of the Azure release service components of the STARS solution. The document is intended for technical

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 747 of 1295

stakeholders, including architects, software engineers, administrators, and operations teams, who are responsible for designing, developing, deploying, and operating the STARS solution in Azure.

The document also serves as a reference for ongoing management and evolution of the STARS solution and to ensure that the STARS solution is deployed and operated in accordance with the best practices and guidance provided by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework.

The document is intended to be a living document, updated as the STARS solution evolves and changes over time. It provides a clear and concise understanding of the Azure release service components and their configurations, helping to ensure the STARS solution is deployed, operated, and managed in a consistent and efficient manner.

3.2 Intended Audience

This detailed design document provides technical information and guidance to help technical stakeholders understand the Azure release service components of the STARS solution and how they integrate with each other. The document is expected to be used by technical stakeholders who have a good understanding of cloud computing concepts and are experienced in the Azure platform.

This document is not intended for non-technical readers, nor for readers who are not experienced in the design and configuration of Azure-based solutions; informational and educational resources regarding Azure are readily available at https://azure.microsoft.com

The primary audience for this document includes:

- Program technical staff: This includes architects, software engineers, administrators, and operations teams within the program responsible for the STARS solution.
- External technical stakeholders: This includes technical staff from other federal, state, and territory governments agencies, including VDS embedded team members.
- ICT Technical Staff: This includes a variety of technical roles involved during the VDS program and post-go live, including network engineers, systems administrators, database administrators, security engineers, and DevOps engineers.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 748 of 1295

4 Scope and Exclusions

4.1 In Scope

- Release service API specification
- Release service workflow including data enrichment workflow and
- Interaction of release service with STARS and external services

4.2 Out of Scope

- Mastering of data to STA Master Databases and Reporting and Analytics database
- Specification of any service that release service is dependent on or any service that is reliant on the release service. Only interaction mechanisms such as data transfer objects will be defined. Other services will be defined in respective detailed design documents.
- Definition of external applications/services that call the STARS system to release data.
- User interface elements based on results produced by the release service. These will have their respective detailed design documents.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 749 of 1295

5 Design Decisions

The release service is part of the submission pipeline and acts as gate that allows the STAs to review and enrich data before it is passed onto Reporting and Analytics database.

As this service may not be required for every STA, the functionality of this service only activates if the STA chooses to create business rules to stop records for data enrichment or release to Reporting and Analytics database.

It should also be noted that the whole record is validated after enrichment. An alternative was considered where specific rules are developed to validate only the fields that have changed as part of the enrichment process. While it is feasible that only the enriched content should require validation, the benefits of maintaining a single ruleset and the consistency this provides have been decided to be of sufficient value to outweigh the negligible overhead this may impose on the Rules service.

STAs will be provided access to this service through the STA Gateway service.

Submission data will be written to each Jurisdiction's Master Database prior to release. NCVER do not have the ability to access data in this database for reporting purposes, and the release decision is to make the data available for reporting. As such unreleased data will be available in extracts from each Jurisdiction's Master Database prior to being released.

6 Requirements

The list of requirements relevant to the release Service detailed design can be found in related document H Requirements relevant to release Service.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 751 of 1295

7 Feature Overview

The release service is part of the submission pipeline and resides between Submission and Mastering. Its purpose is to facilitate STA processes required on valid submission data, prior to release to the Reporting and Analytics database.

Note: Data enrichment can happen at any time after the record has passed submission validation. But if the STA chooses to it can implement business rules to halt records in the submission service pending data enrichment.

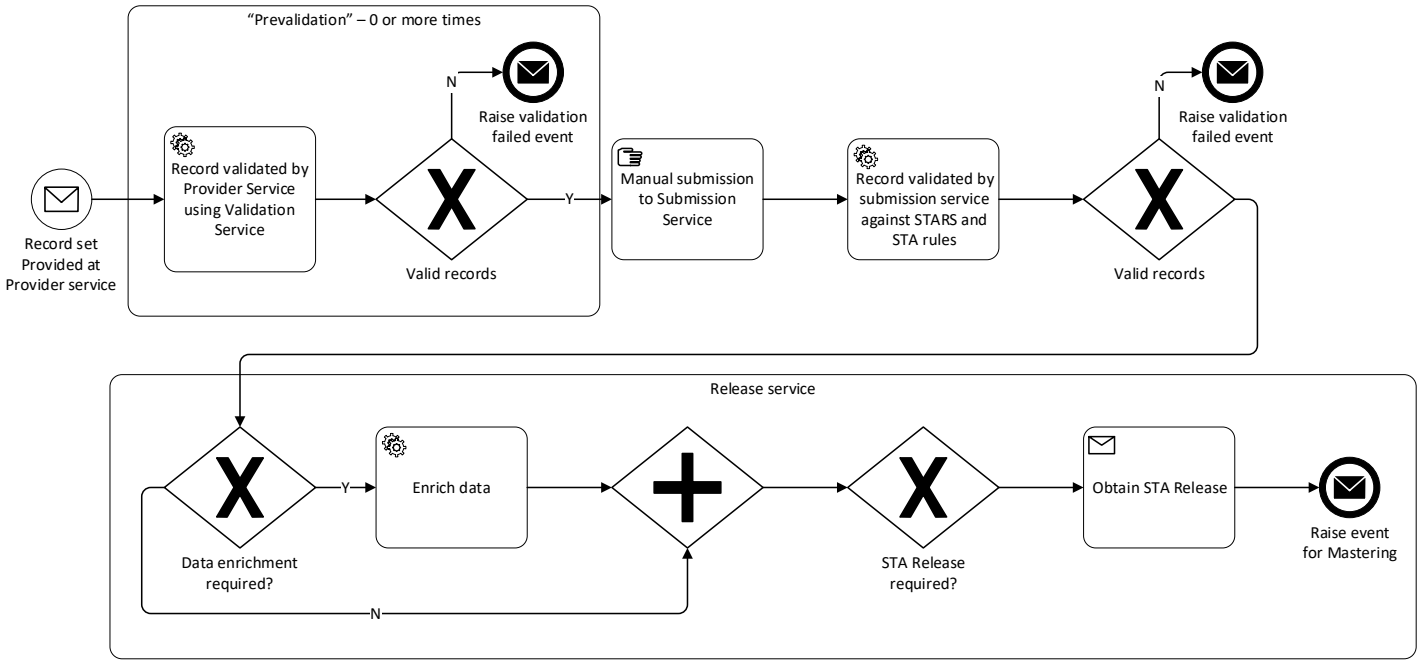


Figure 1: Release Service overview (E1)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

7.1 Data enrichment

The release service facilitates data enrichment by STA if needed. Data enrichment can happen at any time after the record has passed submission service validations. If an STA chooses to, it can implement business rules that flag records that require data enrichment and stop their progress in the submission pipeline.

7.1.1 Data enrichment rule

Records validated by the submission service are checked to determine if the records is from an STA with data enrichment rules if they are not then they are not processed any further for data enrichment. If it is from and STA with business rules for data enrichment are executed on the records in the event using the rules engine to determine if data enrichment is required. These rules are created and maintained by the STA using the rules management portal. If the STA does not want to stop any records for enrichment, they do not need to create any rules as the default behaviour is to let all records pass through.

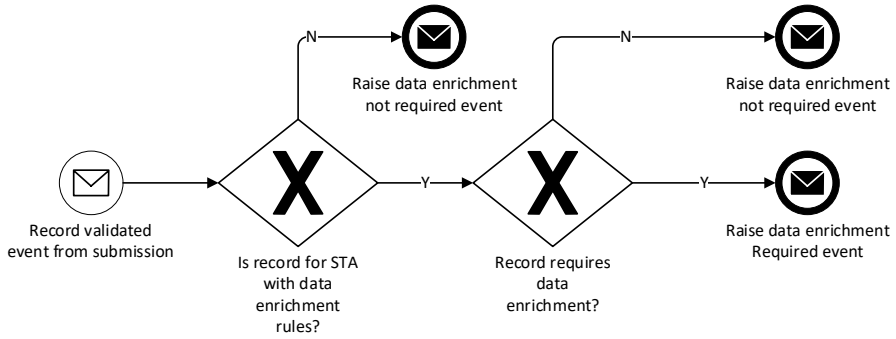


Figure 2: Data enrichment business rule trigger (E2)

If the business rules indicate that data enrichment is required for record(s) then an event is raised, this flags the records and stops their progress through the submission pipeline into the STA master databases and notifies the STA that a record is awaiting data enrichment using the notification service.

Note: While a data enrichment rule can be implemented, it is not necessary for data enrichment to occur. Data can be enriched at any time after submission service validations have passed for a record.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 753 of 1295

7.1.2 Data enrichment process

The Enrichment process happens through the release service API. The details of the data enrichment are received as changes that need to be made (JSON patch) to a particular record.

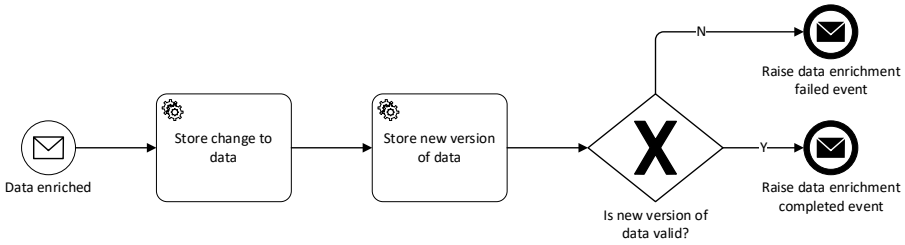


Figure 3: Data enrichment process (E2)

The change details are first stored in the release database. Then the record is retrieved from the submission database and the changes are applied to it to create a modified record which is then stored in the release database.

Once the modified record exists in the release database it is validated using the Validation service against both STARS and STA rules. Only on successful validation the data enrichment completed event is raised. If validation fails however, the enrichment failed event is raised and the STA is notified of the failure.

The STA may also choose to notify the release service that the record does not require enrichment. This is done through the release service API in a similar manner to data enrichment.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 754 of 1295

7.1.3 Use cases

Shown below is a general use case diagram for data enrichment functionality.

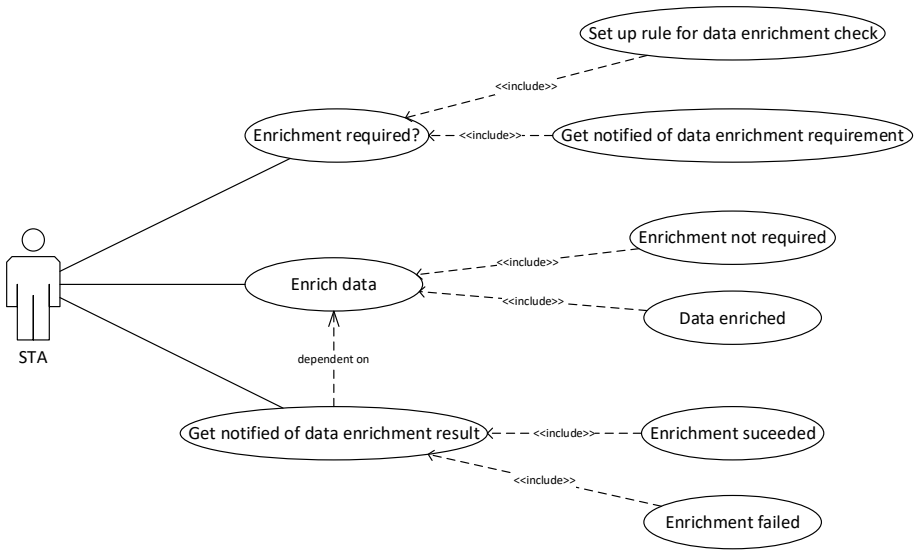


Figure 4: Data enrichment use case (E6)

STA ENRICHES RECORD WITH FUNDING INFORMATION

OVERVIEW	An RTO submits data without funding information as they do not have access to funding information, the STA gets notified and enriches record with funding information.
ACTORS	RTO, STA
PREREQUISITES	1. STA has set up rule to be notified of data enrichment requirements when an RTO submits data without funding information. 2. RTO is submitting data for the STA in question
TRIGGER	Records submitted by RTO pass submission service validation
FLOW 1	1. The STA is notified that there are records that are awaiting data enrichment due to rule set up by the STA to check for records without funding information 2. STA adds the funding information to the record 3. STA notified of successful enrichment

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 755 of 1295

FLOW 2	1. The STA is notified that there are records that are awaiting data enrichment due to rule set up by the STA to check for records without funding information 2. STA adds the funding information to the record 3. STA notified of failed enrichment due to record failing validation due to incorrect funding information for enrolment 4. STA corrects funding information 5. STA is notified of successful enrichment

STA NOTIFIES NO DATA ENRICHMENT REQUIRED AND CHANGES RULE

OVERVIEW	An RTO submits data and that gets incorrectly flagged for data enrichment by an outdated Business rule
ACTORS	RTO, STA
PREREQUISITES	1. STA has set up rule to be notified of data enrichment under certain circumstances that are no longer relevant 2. RTO is submitting data for the STA in question
TRIGGER	Records submitted by RTO pass submission service validation
FLOW 1	1. The STA is notified that there are records that are awaiting data enrichment due to outdated rule 2. STA updates record to say no data enrichment required 3. STA updates business rule so it does not flag records incorrectly anymore

STA IDENTIFIES RECORDS REQUIRING ENRICHMENT FROM MASTER DATA ANDENRICHES DATA

OVERVIEW	An RTO submits data which flows through to master data for STA where the STA identifies that it requires enriching and enriches it
ACTORS	RTO, STA
PREREQUISITES	1. RTO is submitting data for the STA in question

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 756 of 1295

	2. Data passes all validation and makes it through to STA master data
TRIGGER	STA views data in STA master data
FLOW 1	1. The STA views its master data and identifies need for data enrichment 2. STA enriches data 3. STA is notified of successful enrichment

7.2 Data release

The release service facilitates data release by STA if needed. The default behaviour in STARS is to let all records validated by the submission service through to the Reporting and Analytics database. STA Data release business rules need to be created by the STA to stop records from proceeding onto the Reporting and Analytics database. These rules are created and maintained by the STA using the rules management portal.

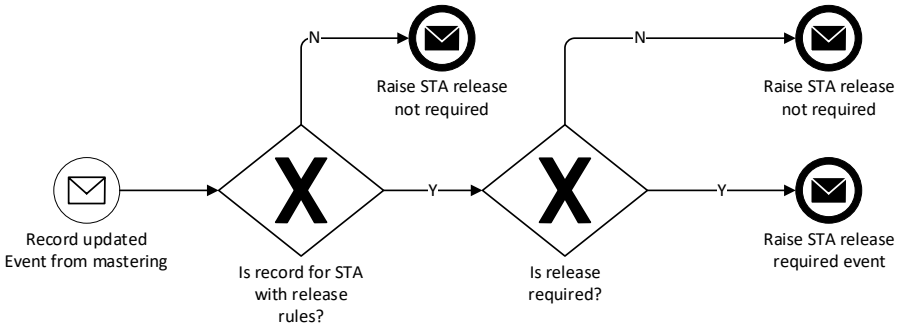


Figure 5: STA Data release (E2)

The release service listens for record updated events from the mastering service. It checks the records in the event first to check if it is from an STA with business rules for release. If it is not, then the record is immediately marked as released. If the rules are found, the release service workflow runs these rules in the Rules engine in the Rules service to determine if the any of the records need to be flagged for STA release. If it finds such records, it flags them, and then they are stopped from being moved to the Reporting and Analytics database and the relevant STA is notified using the notification service.

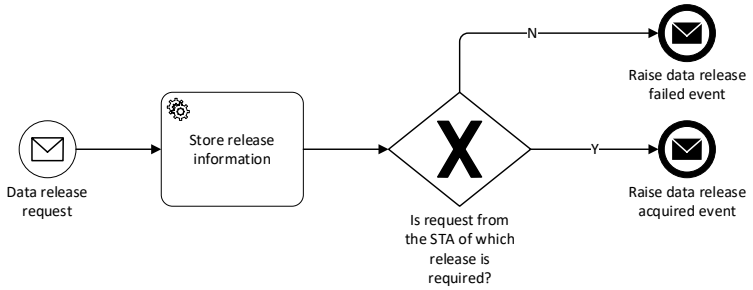


Figure 6: STA Data release API call

Field Code Changed

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 758 of 1295

8 Data Flow

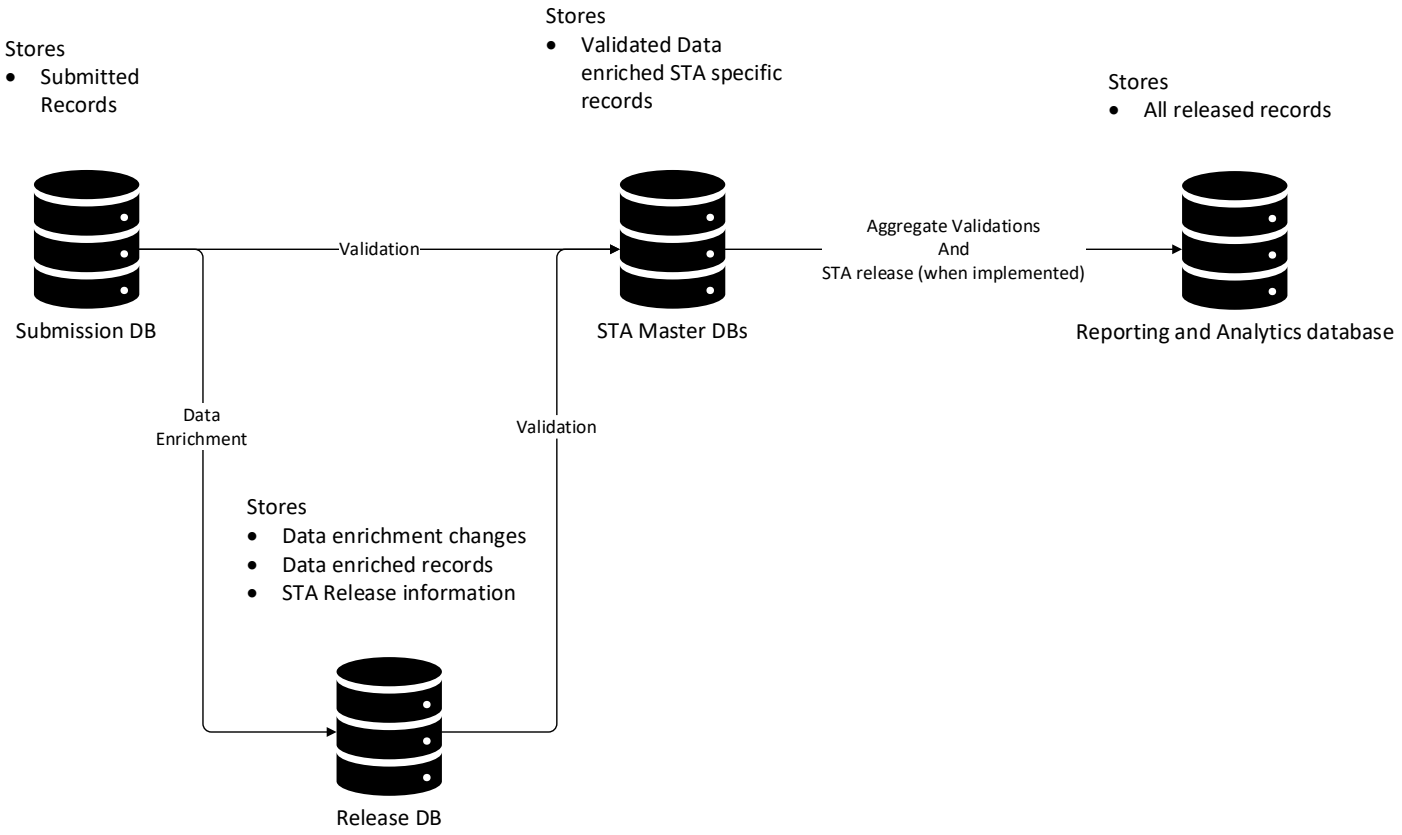


Figure 7: Flow of data in the submission pipeline (E4)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

8.1 Data enrichment flow

The submission DB is a document database that holds all records submitted via the provider service. The submission service facilitates the validation of records in the submission database using the validation database.

The release DB is also a document database that holds data enrichment information, data enriched records and STA release information

On successful validation by the submission service the Record is checked by the release service to determine if it requires data enrichment by the STA.

If the record does require enrichment

- it notifies the STA and acquires the Data Enrichment information from the STA via the release service API (in the form of a JSON patch) and stores this information in the release database.
- it acquires the original record from the Submission DB and applies the patch to create a data enriched version of the record and again stores that in the release database.
- This data enriched record in the release database is used to perform validation on the enriched version of the record using the validation service.
- On successful validation the data enrichment is accepted, and the record is moved on to the STA specific master database.

Note: The STA may also choose to inform the release service that even though the record was flagged for enrichment it was not actually required which will direct the record down the not needing enrichment path.

If the record does not require enrichment, it is moved into the STA specific master database directly.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

8.2 Data release flow

When a record(s) is created or updated in a STA specific master DB

- It is checked by the release service to determine if the record requires STA release
- If it does require STA release the record(s) is flagged and from progressing further in the submission pipeline and the relevant STA is notified.
- The STA can then release the record(s) by sending a STA release request to the Release Service API.
- If any corrections are required to the record(s) they need to be made by the RTO through the provider service and propagated through the submission pipeline
- Once modified correctly by the RTO the STA can approve the release.

Note: For the record(s) to move to the Reporting and Analytics database it need to have STA release (or not require one) and it also needs to pass aggregate data quality checks.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

9 Technical Architecture

The technical architecture of the release service is depicted in the diagram below. Note the interaction with the Rules Service, Validation Service, Submission service, Mastering service and Gateway service. App insights for the STARS solution is also used for audit and logging.

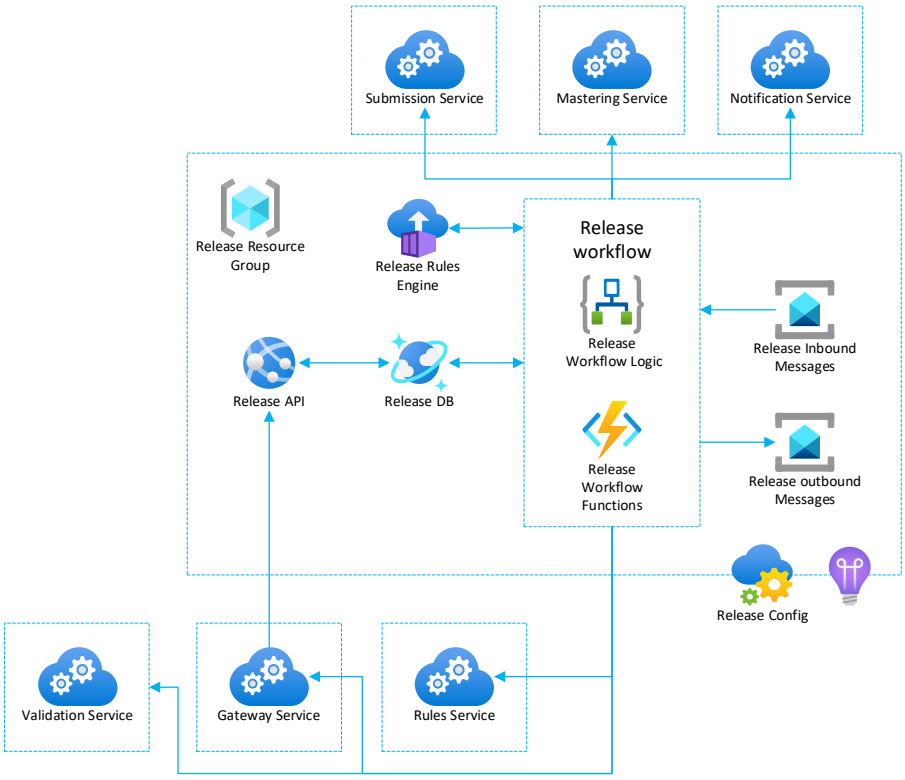


Figure 8: Release Service Architecture (E3)

All incoming requests to be handled by the release service comes in through the API. External requests to the API are mediated through the gateway Service.

The release workflow makes calls to the rules service to execute business rules in relation to data enrichment and release. It also uses the validation service to validate records after data enrichment.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 762 of 1295

9.1 Release API

The release API will be implemented as an ASP.Net core web API. It will be responsible for

- Receiving and responding to requests for data enrichment and raising relevant events
- Receiving and responding to requests for data release and raising relevant events
- Responding to queries relating to data enrichment requests and STA release requests

9.2 Release Workflow

The release workflow component will be implemented using an Azure Logic app with use of Azure Functions for responding to pipeline events relating to dataflow and evaluating records using the rules engine.

9.3 Release Rules Engine

The release rules engine will be used by the release workflow logic app to evaluate business rules related to the data enrichment and data release business rules. It will use the Drools rule engine and be hosted in an azure container.

9.4 Release DB

The validation DB will store all incoming data enrichment and data release requests. It will also store data enriched records.

9.5 Release Config

The Release Config (Key Vault) will store values for configuring the service. Note that common config values that every service will require will be defined in a separate, common location.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 763 of 1295

10 API Design

A Detailed API specification and Open API definition can be found in related documents F and G. JSON object definitions are contained in related document F

Here we will highlight the primary endpoints and their functions. Please note that the API will be implemented using OData. Please refer to reference document D for details on STARS API implementation.

10.1 Data Enrichment Record Endpoint

This endpoint is responsible for receiving data enrichment requests and allowing the querying of DataEnrichmentRecord objects.

- It receives validation requests through the enrich OData post action with a DataEnrichmentRequest JSON object.
- It allows for querying using OData and returns DataEnrichmentRecord JSON objects

10.2 STA Release Endpoint

This endpoint is responsible for receiving STA release requests and allowing the querying of STAReleaseRecord objects

- It receives STA release requests through the release OData post action with a STAReleaseRequest JSON object.
- It allows for querying using Odata and returns STAReleaseRecord JSON objects for all STA release requests received

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 764 of 1295

11 Release Service Workflows

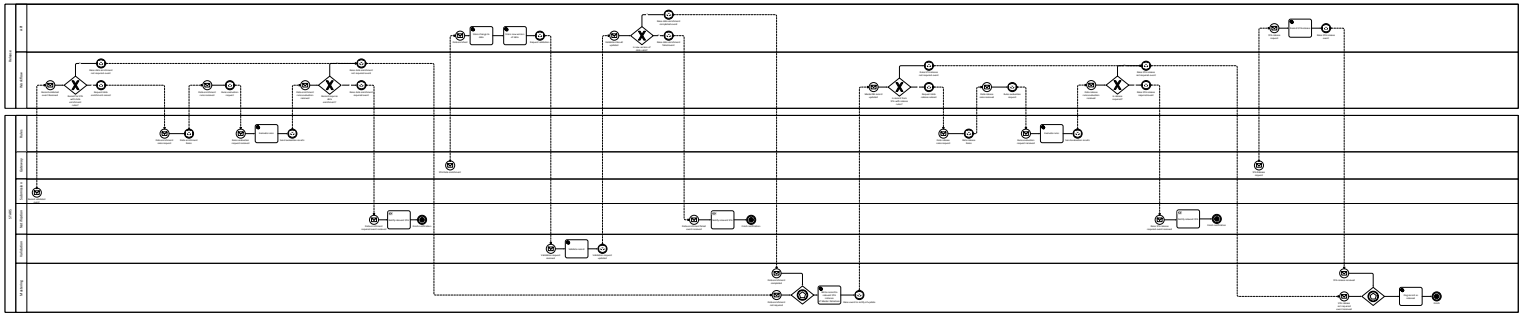


Figure 9: Release workflow (E5)

The fully integrated workflow can be found in related document E which is part of the detailed design pack. Please refer to it for a visual representation of the following workflows.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

11.1 Data enrichment check workflow

- The release service workflow subscribes to Validation Successful events from the submission service indicating a record(s) have been successfully validated in the submission database and filters these events so only STAs requiring explicit release are considered.
- The release Service workflow invokes the Rules Service to decide if data enrichment is required for any of the validated records.
- If the record(s) need enrichment the release service workflow:
    - Raises a data enrichment required event
    - Sends a request to the notification service to notify the relevant STA of the requirement for data enrichment
    - Stops the record from progressing further down the submission pipeline
    - Waits for a response from the STA
- If it does not need enrichment the release service workflow:
    - Raises a data enrichment not required event (this prompts the mastering service to move it into the relevant STA master db)

11.2 Data enrichment request workflow

- A STA sends a DataEnrichmentRequest JSON object to the release service API DataEnrichmentRecord/Enrich endpoint
    - The API Stores the DataEnrichmentRequest as a DataEnrichmentRecord JSON object in the database
    - The API retrieves the relevant record from the submission database and attempts to apply the enrichment patch
    - If successful
        - It stores the modified record in the database
        - It updates the DataEnrichmentRecord
        - Requests the validation service to validate the record
        - Updates the DataEnrichmentRecord according to validation result
            - If valid
                - Raises a data enrichment request completed event
            - If not valid
                - Raises a data enrichment request failed event
        - Notifies the STA of the outcome using the notification service
    - If unsuccessful
        - Updates the DataEnrichmentRecord

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 766 of 1295

- Notifies the calling STA in the response

11.3 Data release check workflow

- The release service workflow subscribes to Data Mastered events indicating a record(s) have been created or updated in the STA master database, and filters these events so only STAs requiring explicit release are considered.
- The Release Service workflow invokes the Rules Service to decide if Explicit Release is required for any of the Mastered records.
- For records that do need explicit release, the release service workflow:
  - Raises a data release required event
  - Sends a request to the notification service to notify the relevant STA of the requirement for data release
  - Waits for a response from the STA
- For records that do not require explicit release, the Release service workflow:
  - Raises an STA release not required event

11.4 Data release request workflow

- The STA sends a STAReleaseRequest JSON object to the STAReleaseRecord/Release endpoint
- The API stores the STAReleaseRequest as a STAReleaseRecord object
- The API checks to see of the requesting authority has the authority to release the record
- If it does
  - It updates the STAReleaseRecord to have a status of released
  - Raises a STA release received event
  - Notifies the calling STA in the response
- If it does not
  - It updates the STAReleaseRecord to have a status of Rejected
  - Notifies the calling STA in the response

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 767 of 1295

12 Release Event Service Messages

Below is a list of release service messages to the event service

Name	Purpose
Data enrichment required	Data enrichment required for record(s)
Data enrichment not required	Data enrichment not required for record(s)
Data enrichment failed	Data enrichment failed for record(s)
Data enrichment completed	Data enrichment completed for record(s)
STA release required	STA release required for record(s)
STA release not required	STA release not required for record(s)
STA release failed event	STA release failed for record(s)
STA release acquired	Valid STA release has been obtained for record(s)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 768 of 1295

13 Data

Please refer to document F for all data schemas and examples.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 769 of 1295

14 Security and Access Management

The release service is an internal service. Access to the release service from systems external to the STARS solution will be mediated through the Gateway Service.

The service will conform to the Security Architecture. Access to the Release Service will be controlled by Azure RBAC, with all calling services identified and granted access through Managed Identities.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 770 of 1295

15 Acronyms

Acronym	Definition
API	Application Programming Interface
CRUD	Create Read Update Delete
REST	Representational State Transfer
FR	Functional Requirement
NFR	Non-functional Requirement
STARS	Student Training & Activity Reporting System
VET	Vocational Education Training
VDS	VET Data Streamlining
IVETS	Integrated Vocational Education and Training System
DB	Database
DDS	Data Definition Language
SQL	Structured Query Language
TP	Training Provider
STA	State Training Authority
SMS	Student Management

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 771 of 1295

Australian Government
Department of Employment and Workplace Relations

Detailed Design Document

Downstream Consumer ETL Pipeline Service

1 Document Management

1.1 Document Control

Version	Change Description	Date
0.1	Draft Version	2023-03-14
0.2	Client Feedback	2023-04-13
1.0		

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This is a living document and will be updated as required.

1.2 Document Author

Role	Name
DXC Data Architect	s 47F(1)

1.3 Document Reviewers

Role	Name
xxx	xxxxx

1.4 Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

1.5 Related Documents

Reference	Document
A	STARS Solution Requirements Document – D.0.3.docx
B	STARS Solution Design – D.0.3.docx
C	DRAFT VET Info Std Mapping to NAT Files - Reverse Engineering R1_V1.0.xlsx
D	DRAFT VET Information Standard Logical Interchange Specification V2 R1_V1.0.xlsx
E	NATFilesFormat - and SA NAT File Extract Requirement.xlsx
F	Detailed Design Document – Rules Service.docx

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 773 of 1295

Reference	Document
G	Detailed Design Document - Master Data Service.docx
H	Detail Design Document – Submission Service.docx

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 774 of 1295

Contents

1 Document Management 2
1.1 Document Control 2
1.2 Document Author 2
1.3 Document Reviewers 2
1.4 Document Approval 2
1.5 Related Documents 2
Contents 4
2 Introduction 8
2.1 Purpose 8
2.2 Intended Audience 8
3 Scope and Exclusions 9
3.1 In Scope 9
3.2 Out of Scope 9
3.2.1 Report Style Filtering or Aggregation 9
3.2.2 Row level data access restrictions 9
3.2.3 Data Validation Rule extracts 9
3.2.4 Submission Meta Data 9
3.2.5 Data extraction via API end points 9
3.2.6 Azure Data Share database sharing 9
4 Design Decisions 1110
4.1 Assumptions 1110
4.1.1 Data Filtering 1110
4.2 Issues 1110
4.2.1 Overloaded Fields 1110
4.2.2 SSF “below the lines fields” 1110
4.2.3 Data Transmission Header Files 1110
4.2.4 State Specific Requirements 1110
4.2.5 Unmastered Data 1210
4.3 Scheduling 1211
4.4 Data Retention of archived file extracts 1211
4.5 Extract Reporting Scope 1211
4.5.1 Mastered Data 1211
4.5.2 Unmastered Data 1311

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 775 of 1295

4.6 Associated Reference Data ..... 13~~12~~
5 Requirements ..... 14~~13~~
5.1 Functional ..... 14~~13~~
5.2 Non-Functional ..... 17~~16~~
6 Feature Overview ..... 20~~19~~
6.1 Mastered Data ..... 20~~19~~
6.1.1 VET Information Standard (VET IS) ..... 20~~19~~
6.1.2 AVETMISS 8 ..... 20~~19~~
6.2 Unmastered Data ..... 20~~19~~
6.3 Data Transformation Outputs ..... 20~~19~~
6.3.1 VET IS to AVETMISS 8 ..... 20~~19~~
6.3.2 VET IS ..... 21~~19~~
6.3.3 VET IS to STARS Collection Specification (Data Packet) ..... 21~~20~~
6.3.4 State Specific Data Transformation ..... 21~~20~~
6.4 File Format ..... 21~~20~~
6.4.1 CSV ..... 21~~20~~
6.4.2 Fixed Width Text ..... 21~~20~~
6.4.3 Json ..... 21~~20~~
6.5 Load Method ..... 21~~20~~
6.5.1 Full Load ..... 21~~20~~
6.5.2 Incremental ..... 21~~20~~
6.5.3 Changed TO Full ..... 21~~20~~
6.6 Data Transfer ..... 22~~21~~
6.6.1 Site-to-Site Virtual Private Network (VPN) ..... 22~~21~~
6.6.2 Azure Data Share ..... 22~~21~~
6.6.3 Data Header Files ..... 22~~21~~
7 Technical Design Patterns ..... 24~~23~~
7.1 Data Transformation ..... 24~~23~~
7.1.1 Data Transformation Staging Schemas (LZ, ST, TZ, LD, EX) ..... 24~~23~~
7.1.2 Landing (LZ) ..... 25~~24~~
7.1.3 Staging (ST) ..... 25~~24~~
7.1.4 Transform (TZ) ..... 25~~24~~
7.1.5 Load (LD) ..... 25~~24~~
7.1.6 Exception (EX) ..... 26~~25~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 776 of 1295

7.1.7 Data Transformation Development Toolset ..... 26~~25~~
8 Data Extract Solution Components ..... 27~~26~~
8.1 Data Target and Environment ..... 28~~27~~
8.2 Extract Data Integration ..... 29~~28~~
8.2.1 Extract Data Refresh Interval ..... 31~~30~~
8.2.2 Full/ Incremental ..... 31~~30~~
8.2.3 Data Volume ..... 31~~30~~
8.3 ADF Pipeline – Data Refresh Process ..... 31~~30~~
8.4 Data Transformation Staging Schemas ..... 33~~32~~
8.5 ADF Data Extraction – Data Transformation Schema to Azure Data Lake ..... 33~~32~~
8.5.1 ADF Pipeline – Generate Files ..... 33~~32~~
8.5.2 Raw Data Storage ..... 34~~33~~
8.5.3 Hierarchical Namespace enabled ADLS Gen2 ..... 34~~33~~
8.5.4 Folder Structure and File Name Convention ..... 34~~33~~
8.5.5 File Format for Storage ..... 36~~35~~
8.6 ADF Data Transfer – STARS Azure Data Lake to Client Stie-to-Site VPN ..... 36~~35~~
8.6.1 ADF Pipeline – data load ..... 36~~35~~
8.7 ADF Logging ..... 37~~35~~
8.7.1 Logs ..... 37~~35~~
8.7.2 Alerting and Notification ..... 37~~36~~
8.7.3 Exception Handling ..... 37~~36~~
8.7.4 Custom Logging ..... 37~~36~~
8.7.5 Metadata Extraction and Cataloguing ..... 38~~37~~
9 Data ..... 39~~38~~
9.1 Schemas ..... 39~~38~~
9.1.1 AVETMISS 8 ..... 39~~38~~
9.1.2 VET IS ..... 39~~38~~
9.1.3 VET IS to STARS Collection Specification (Data Packet) ..... 39~~38~~
9.1.4 State Specific Data Transformation ..... 39~~38~~
10 Access Management ..... 40~~39~~
10.1 Control of data ..... 40~~39~~
10.1.1 Data Owner ..... 40~~39~~
10.1.2 Data Custodian ..... 40~~39~~
10.2 Security ..... 40~~39~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 777 of 1295

11 Dependency on Other Services ..... 42~~41~~

11.1 Event Service ..... 42~~41~~

12 Acronyms ..... 43~~42~~

Figure 1: STARS Data Transformation Architecture ..... 24~~23~~
Figure 2: STARS Azure Data Estate ..... 27~~26~~
Figure 3: Data Extract Integration Technical Design ..... 30~~29~~
Figure 4: ADF Pipeline steps to refresh data ..... 32~~31~~
Figure 5: ADF Pipeline steps to load into Azure Data Lake ..... 34~~33~~
Figure 6: Data Lake Folder Structure ..... 35~~34~~
Figure 7: ADF Pipeline Data Transfer ..... 36~~35~~

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 778 of 1295

2 Introduction

The purpose of this document is to capture the detailed design requirements for the STARS Downstream Consumer ETL Pipeline Service solution.

The STARS Downstream Consumer ETL Pipeline Service will provide the ability to extract bulk data from the STARS Master and Submission databases.

Note: This is a living document that will be updated as designs for various other STARS services are developed and evolve. Accordingly, this document will evolve to reflect and document Downstream Consumer ETL Pipeline Service components as the STARS solution build progresses.

2.1 Purpose

The purpose of this detailed design document is to provide a technical description of the bulk data extract component of the STARS solution. The document is intended for technical stakeholders, including architects, software engineers, administrators, and operations teams, who are responsible for designing, developing, deploying, and operating the STARS solution in Azure.

The document is intended to be a living document, updated as the STARS solution evolves and changes over time.

2.2 Intended Audience

This detailed design document provides technical information and guidance to help technical stakeholders understand the bulk data extract component of the STARS solution. The document is expected to be used by technical stakeholders who have a good understanding of database modelling concepts, are experienced in relational and document-oriented database platforms and data transfer mechanisms.

This document is not intended for non-technical readers, nor for readers who are not experienced in the design and configuration of data extract and data transfer solutions.

The primary audience for this document includes:

- Program technical staff: This includes architects, software engineers, administrators, and operations teams within the program responsible for the STARS solution.
- External technical stakeholders: This includes technical staff from other federal, state, and territory governments agencies, including VDS embedded team members.
- ICT Technical Staff: This includes a variety of technical roles involved during the VDS program and post-go live, including network engineers, systems administrators, database administrators, security engineers, and DevOps engineers.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 779 of 1295

3 Scope and Exclusions

3.1 In Scope

The following are in-scope of this detail design solution:

Scheduled bulk file-based data extracts sourcing data from the STARS Master and Submission databases.

3.2 Out of Scope

The following are out of scope:

3.2.1 Report Style Filtering or Aggregation

This service does not support the filtering or aggregation of data sets other than as per what is specified in the Extract Reporting Scope section of this document. Reporting requirements will be documented as part of the 'Downstream Consumer Reporting & Analytics Service Detailed Design' or as part of the service they are associated with.

3.2.2 Row level data access restrictions

Access to data extracts will be restricted at the jurisdictional/organisational level only. Any more detailed data access restrictions to extracted data will need to be applied client side by the client.

3.2.3 Data Validation Rule extracts

The extraction of data validation rules is out of scope of this service. Please refer to the 'Detailed Design Document – Rules Service' document.

3.2.4 Submission Meta Data

Data extracts of unmastered submissions will be made available in their original submission format. Associated meta data related to the submission (i.e., validation status) will not be available as part of this extract. Reporting requirements related to meta data about the submission should be included as part of the relevant service (i.e., STARS Validation Service etc…).

3.2.5 Data extraction via API end points

Data extraction via API end points will not be available as part of this bulk data transfer service. Please see 'Submission API end points' as described in the 'Detail Design Document – Submission Service.docx' document.

3.2.6 Azure Data Share database sharing

At the time of writing this document Azure Data Share does not support the sharing of Azure Cosmos databases or incremental snapshot sharing of Azure SQL Databases. As such Azure Data Share capabilities for Azure SQL Databases will not be enabled unless there is a specific requirement to refresh an entire target database with a full snapshot.

3.2.7 Notification of delete

Notification of deletes will not be provided as part of bulk data extracts.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 780 of 1295

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 781 of 1295

4 Design Decisions

4.1 Assumptions

4.1.1 Data Filtering

Any custom data filtering will be performed on the client side against the full year to date data collection period data set unless otherwise specified in this document.

4.1.2 Validation status of extracted data

Mastered data will be validated. Unmastered data could be a mixture of validated and unvalidated submissions it is assumed that a mixture of both validated and unvalidated unmastered data will be supplied as part of bulk data extracts.

4.2 Issues

4.2.1 Overloaded Fields

An 'over loaded field' is a term used to describe an attribute that has used in a manner contrary to its original intentions and as such deviates from the National standard.

South Australia has indicated a need to cater for such 'over loaded fields' as defined in the provided mapping 'NATFilesFormat - and SA NAT File Extract Requirement.xlsx'.

4.2.2 SSF "below the lines fields"

The DRAFT VET Information Standard Logical Interchange Specification V2 R1_V1.0.xlsx does not include all the "below the line" fields that are documented in the AVETMISS_VET_8_0_Collection_Specifications.docx. As all the AVETMISS 8 attributes are required to generate an AVETMISS 8 data extract as such the Concordance Mapping Design Verification 24 Nov 22 - DEWR feedback R1_V1.0.xlxs and DRAFT VET Info Std Mapping to NAT Files - Reverse Engineering R1_V1.0.xlsx documentation will be used to perform the associated data mapping. ~~more analysis and data mapping activities are required.~~

In addition to these AVETMISS 8 attributes there are potentially also other State Specific fields (SSF) that require capture and require more investigation. Capture of this data may form part of a state specific data integrations.

4.2.3 Data Transmission Header Files

Where Azure Data Share's inbuilt data transfer reconciliation functionality cannot be utilised header files including meta data about the data transfer will be made available for reconciliation purposes.

These header files will not be included as part of the .zip compressed archives that contain business data so as to be backwards compatible with legacy ingestion processes.

4.2.4 State Specific Requirements

South Australia has provided detailed jurisdictional specific requirements as of the time of writing these are documented in the 'NATFilesFormat - and SA NAT File Extract Requirement.xlsx' document.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 782 of 1295

4.2.5 Unmastered Data

Submission data stored in the Submission database in its raw form may not display a coherent picture of the submission history of an entity without some additional data manipulation that is open to interpretation on the client side. The 'Submission API end points' as described in the 'Detail Design Document – Submission Service.docx' document perform this additional data manipulation.

For this reason, requirements for data extracts should be assessed in the context of these 'Submission API end points and other reporting requirements defined as part of the 'Downstream Consumer Reporting & Analytics Service Detailed Design'.

4.3 Scheduling

Extracts will be scheduled nightly after close of business at the discretion of a STARS administrator to avoid internal system maintenance conflicts. Extracts will be made available for target side client consumption as at 7am each business day (i.e., Monday to Friday).

Any scheduling requirements that are less frequent than daily can be submitted by the client. However, it is worth noting that the same can be achieved by processing the daily delta files that occurred during the period.

4.4 Data Retention of archived file extracts

A retention period of 90 days has been suggested but, this would seem excessive for disaster recovery purposes. Further investigation is required to determine if this retention period can be lowered.

4.5 Extract Reporting Scope

The following will be available as distinct file extract sets as per the defined extract schedule:

4.5.1 Mastered Data

4.5.1.1 Full Cumulative Year to Date extracts

Cumulative Year to Date data from the current collection period will be available for all submitted data. This would form the initial extract of information to a client and will be generated as per the data extract schedule.

4.5.1.2 Incremental Delta extracts

New data that has been created or existing data that has been modified since the last scheduled STARS extract will be available for the current collection period (i.e., only delta records) as per the data extract schedule.

4.5.1.3 ~~Incremental~~ RTO based extracts

Cumulative Year to Date data for the current collection period will be available for data submitted by a Registered Training Organisations (RTO) where the RTO has generated new data or modified existing data belonging to the current collection period since the last extract (i.e., not just delta

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 783 of 1295

records but, all records belonging to that RTO). These extracts will be generated as per the data extract schedule.

4.5.2 Unmastered Data

Unmastered data will only be available as a Json document in its original submitted 'Data Packet' form. This format is detailed in the 'DRAFT VET Information Standard Logical Interchange Specification V2 R1_V1.0.xlsx' document.

At the time of writing this document it has not yet been specified whether State Specific Fields (SSF) in addition to those documented in the 'DRAFT VET Information Standard Logical Interchange Specification V2 R1_V1.0.xlsx' document will enrich the original Json submission document and be available for extract.

These extracts will be generated as per the data extract schedule.

4.6 Associated Reference Data

Only files that are extracted as per VET Information Standard (VET IS) data model will have associated reference data exported as a distinct supporting file set. If the reference data sets are excessively large, then they will be filtered to include only values that form part of the associated VET IS data extract as per that current time slice. Reference data sets will be exported as per the VET IS data model.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 784 of 1295

5 Requirements

5.1 Functional

FR No	Requirement Name	Description	Traceability
BR011	BR011 - VET Information Standard supports quality and integrity for multiple uses	All data elements are defined in the VET Information Standard. • During the transition period national reporting will continue to be based on AVETMISS 8. This means that VDS will need to be compatible to the new standard which is under development and validate against VET Information Standard used by stakeholders who have a hybrid situation with a subset of their training organisations on VDS. • This includes the ability to extract (via API) AVETMISS 8 compliant NAT files for the full collection year per training organisation, to support backwards compatibility with downstream STA/NCVER systems. • VDS will need to convert data from the new Standard back into NAT Files for submission through existing validation tools (AVS) to NCVER. • The VDS system will be compatible with the current and previous version of AVETMISS.	Delta extracts as described in the Extract Reporting Scope section of this document will be provided routinely as described in the Scheduling section of this document. The scope of this service is to provide data transfer methods as described in the Extract Data Integration section of this document. At the time of writing the scope of this service is to provide AVETMISS data extracts in the current version of the standard (i.e., AVETMISS 8).
BR030	BR030 - Reporting based on jurisdiction	A STA/Funder will be able access, extract, and report on all training activity data in the VDS System relevant to its	This service does not support the filtering or aggregation of data sets other than as per what is

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Description	Traceability
		jurisdiction, irrespective of whether the training was government funded or not, subject to privacy requirements. This includes multiple options such as training delivered in the jurisdiction, students based in the jurisdiction, students who live and/or study in areas surrounding the jurisdiction, and training organisations based in the jurisdiction, including coverage of remote learning.	specified in the Extract Reporting Scope section of this document. Reporting requirements should be documented as part of the 'Downstream Consumer Reporting & Analytics Service Detailed Design'.
BR063	BR063 - Access data that failed validation	The VDS system enables the STA/Funder to access and extract submitted data that did not pass National and State Specific validations across all staging areas to enable State Specific QA Checks on Providers and Training Activities and for Legal considerations. • This data can be used to establish trends in errors and providers • This data can be used by support teams to provide help to the providers.	Data extracts of unmastered submissions will be made available in their original submission format. Associated meta data related to the submission (i.e., validation status) will not be available as part of this extract. Reporting requirements related to meta data about the submission should be included as part of the relevant service (i.e., STARS Validation Service etc…).
FR013b	FR013b - Access and use of data in Staging DB	A STA/Funder will be able to access, extract and report on data, including related validation outcomes, from the Staging Database through APIs, e.g. REST or Open API, ETL, and Database replication. This includes the ability to extract only the data that has changed since the previous validation.	Data extraction via API end points will be available as described in the 'Detail Design Document – Submission Service.docx' document. Incremental database replication is not currently supported by Azure Data Share. Delta extracts as described in the Extract Reporting Scope section of this document will be provided

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Description	Traceability
			routinely as described in the Scheduling section of this document.
FR020	FR020 - Export formats	The VDS system must allow a dataset to be exported from staging and master databases in the following formats: • CSV • Excel • Text • JSON • Other dataset formats as determined (potentially PDF)	The scope of this service is to provide extract files in the following formats: • CSV • Text • JSON As described in the File Format section of this document.
FR021a	FR021a - Data Outputs	The VDS system must facilitate data consumption by authorised users via: • BI tools • APIs • ETL	The scope of this service is to provide data transfer methods as described in the Extract Data Integration section of this document.
FR031	FR031 - View and extract validation rules	The VDS system must allow the extraction of the National, Reference or STA/Funder Data Validation Rules. A STA/Funder will be able to view, distinguish, and extract the configured national and state-specific validation rules.	The extraction of data validation rules is out of scope of this service. Please refer to the 'Detailed Design Document – Rules Service' document.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Description	Traceability
FR032c	FR032c - Method of access	Users can access data they are authorised to access database reporting capability for staging and mastered data via: • BI tools • standard reports or extracts as per FR020	Access to data extracts will be restricted at the jurisdictional/organisational level only. Any more detailed data access restrictions to extracted data will need to be applied by the client.

5.2 Non-Functional

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Description	Traceability
NFR002i	NFR002i - Overnight extraction	The VDS system will be available for data extraction overnight	As described as part of the Scheduling section of this document.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Description	Traceability

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

6 Feature Overview

6.1 Mastered Data

Extracts based on mastered data will be sourced from the STARS Master relational database as per the current point in time snapshot (i.e., historical changes to data not provided). The data set available will represent the current year to date collection period with delta versions containing "what has changed since the last extract" also available.

6.1.1 VET Information Standard (VET IS)

The current year to date collection period snapshot will be staged in the STARS Data Transformation Schema (Staging - ST) (see Data Transformation) with delta updates pushed to it from the STARS Master database as per a regular schedule. The data model of this data set will be identical to the physical STARS Master Database. Any outstanding delta updates from the Master database will need to be applied prior to data being extracted from STARS Data Transformation Schema (Load - LD).

The Staging schema (ST) will also serve as a source reconciliation point for any data transformations that are required to support other target formats (i.e., VET IS to AVETMISS 8).

This staged relational data model will require little to no data transformation to enable extraction into an equivalent file-based format.

6.1.2 AVETMISS 8

Data transformation will be required to support the extraction of data in the AVETMISS 8 format. ~~This staged relational data model will require little to no data transformation to enable extraction into an equivalent file-based format.~~

6.2 Unmastered Data

Submission data is stored within the Submission database as Json documents. These documents are stored as per the 'Data Packet' data model as defined in the DRAFT VET Information Standard Logical Interchange Specification V2 R1_V1.0.xlsx document.

The submission data stored within the Submission database may be validated, unvalidated, approved, unapproved, mastered or unmastered as a history of all submissions and resubmissions that have been passed from the Provider Service will be available.

6.3 Data Transformation Outputs

6.3.1 VET IS to AVETMISS 8

Data will be transformed from VET IS to AVETMISS 8 data model as per the data transformation mapping defined within the Concordance Mapping Design Verification 24 Nov 22 - DEWR feedback R1_V1.0.xlsx and DRAFT VET Info Std Mapping to NAT Files - Reverse Engineering R1_V1.0 documents.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 791 of 1295

6.3.2 VET IS

Data will be extracted without transformation from the VET IS physical data model with a sperate file created for each physical database table.

6.3.3 VET IS to STARS Collection Specification (Data Packet)

VET IS data will be transformed back into the data model as per the data transformation mapping defined within the DRAFT VET Information Standard Logical Interchange Specification V2 R1_V1.0.xlsx document.

6.3.4 State Specific Data Transformation

State specific data transformation has been specified as per the following:

6.3.4.1 South Australia

VET IS to AVETMISS 8 transformation will be performed as per the provided data mapping documented as 'NATFilesFormat - and SA NAT File Extract Requirement.xlsx' with a Load Method of 'Changed TO Full' as described in the Load Method section of this document.

6.4 File Format

6.4.1 CSV

The comma separated value format will be used for data extracts based on the VET IS and Data Packet data models.

6.4.2 Fixed Width Text

Each field in this format has fixed number of characters always beginning and ending at the same column number within a single line.

This is the backwards compatible format for AVETMISS 8 data extracts (i.e., NAT files).

6.4.3 Json

Each Json document extracted from the Submission database will be generated in its raw Json format.

6.5 Load Method

6.5.1 Full Load

Files generated using this method will contain the entire year to date data for the collection period.

6.5.2 Incremental

Files generated using this method will contain only records that have changed since the last equivalent extract file was generated.

6.5.3 Changed TO Full

Files generated using this method will contain the entire year to date data for the collection period belonging to Training Organisations (TO) whose data sets have change changed since the last set of

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 792 of 1295

extract files was generated. For instance, if a TO has had a single record change since the last data extract was performed then the entire collection period is extracted for that TO.

6.6 Data Transfer

6.6.1 Site-to-Site Virtual Private Network (VPN)

A Site-to-Site Virtual Private Network (VPN) will be created for clients who are not Azure Data Share capable. This method will be used to enable file-based data transfer.

6.6.2 Azure Data Share

Azure data share will be enabled for file-based data sharing.

Azure Data Share enables organizations to securely share data with multiple customers and partners. Data providers are always in control of the data that they've shared, and Azure Data Share makes it simple to manage and monitor what data was shared, when and by whom.

Using Data Share, a data provider will share data and manage their shares all in one place. They can stay in control of how their data is handled by specifying terms of use for their data share. The data consumer must accept these terms before being able to receive the data. Data providers can specify the frequency at which their data consumers receive updates. Access to new updates can be revoked at any time by the data provider.

Both the data provider and data consumer must have an Azure subscription to share and receive data.

6.6.3 Data Header Files

For non-Azure Data Share based data transfers associated header files will be available containing metadata relating to the file transfer and a summary of each file's contents.

6.6.3.1 File Set

Each set of data extract files as per a scheduled output will form a compressed file archive (i.e., .zip). A header file associated with the files set will list the following information to assist with reconciliation of the file set as a whole and will be transferred separately to the compressed archive:

- Number of Files in the Files Set
- List of File Names within the File Set
- A File Number to identify each file within file set
- Data Transfer Initiation Datetime
- Data Transfer Completion Datetime
- Data Transfer Completion Status
- Archive Size (i.e., KB, MB, GB etc...)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 793 of 1295

6.6.3.2 File

A header file for each file within a file set will list the following information to assist with reconciliation of the file itself:

- File Number within the File Set
- File Name
- File size (i.e., KB, MB, GB etc...)
- Number of rows of data
- Data schema
- File Format

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 794 of 1295

7 Technical Design Patterns

7.1 Data Transformation

The below diagram illustrates the data transformation process relating to VET IS mastered data that is stored within the STARS Master database:

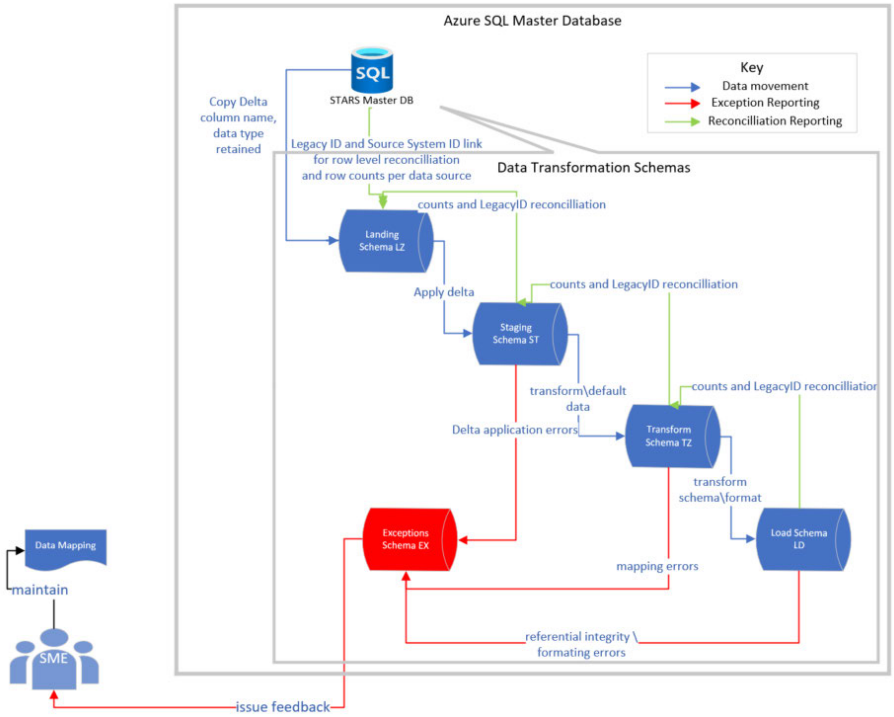


Figure 11~~1~~: STARS Data Transformation Architecture

7.1.1 Data Transformation Staging Schemas (LZ, ST, TZ, LD, EX)

The data transformation staging schemas are where all data transformation activities will take place using primarily SQL database views with Azure Data Factory as the orchestration component (i.e., scheduling jobs, error monitoring etc…). Azure SQL Stored Procedures will materialize data sets within each schema.

The data transformation ETL pipeline will be decoupled from the Master database by maintaining a copy of successfully landed source data in the LZ to enable isolated rerun of the data transformation process.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 795 of 1295

Database schemas will be used to separate data transformation staging zones within the Master database. Each staging schema will materialise processed data sets particular to the step in the data transformation process being executed to promote traceability and minimise the need for re-execution of the entire process in the event of failure.

Post transformation data sets will be available for file-based extraction into the STARS Data Lake from the Load (LD) schema.

7.1.2 Landing (LZ)

All delta refreshes from the STARS Master database will land in the data transformation Landing Schema (LZ) conforming to the VET IS physical data model. Previously landed data will be truncated and overwritten.

A delta load of mastered data will be pulled from STARS Master database temporal tables at regular scheduled intervals. This data set will represent records that have been updated in the Master database since this process was last executed and will be marked with an execution runtime timestamp ('ExtractedAt') to denote its currency.

7.1.3 Staging (ST)

Refreshed delta data from the LZ will be merged with the existing data set that was used to populate the previous data extract. Newly merged records (i.e., ST records that have been inserted or updated as part of the current process) will carry the latest 'ExtractedAt' timestamp pulled through from the LZ data set.

The entire data set now stored in the Staging schema represents the year-to-date collection period (i.e., 'Full' Load).

By using the most recent 'ExtractedAt' timestamp recorded in each table the latest 'Incremental' loads (i.e., delta) can be identified.

7.1.4 Transform (TZ)

Data transformation rules are applied here possibly resulting in modifications to data structure. New temporary staging data entities may be created as a common lookup for subsequent transformations. New data attributes\values that do not exist at source but are required by the target will be created here. Resulting data sets can be system agnostic.

At this point in the extract process the Transformation schema will pull the latest 'Incremental' data set from the Staging schema and perform data transformation activities as described in the Data Transformation Outputs section of this document.

Where a Load Method of 'Changed TO Full' is required (see Load Method section) this is where all associated Training Organisation (TO) year to date collection period data that hasn't changed since the last data extract will be appended with the TOs records that have changed.

7.1.5 Load (LD)

Target system specific transformations and formatting occur here.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 796 of 1295

This is the final stage of the data transformation process where the 'Incremental' data set will be pulled from the transformation schema in its transformed state. Any changes to data schema made in the Load (LD) schema are done so to simplify the subsequent 'Step 4 - Generate extract files into Azure Data Lake' (see Data Extract Solution Components) step performed by an Azure Data Factory Pipeline (ADF).

7.1.6 Exception (EX)

All business data transformation related errors that need to be fed back to SMEs (Subject Matter Expert) for analysis and rectification are recorded here in a standardised format.

7.1.7 Data Transformation Development Toolset

7.1.7.1 Azure Data Factory

ETL (Extract Transform and Load) tool to land data from external data sources.

7.1.7.2 SSMS (SQL SERVER MANAGEMENT STUDIO) – (preferred option)

SQL Server management client for SQL development and non-development tasks (e.g., backup, security).

7.1.7.3 Azure Data Studio

SQL development environment for ETL pipeline data transformations, includes integrated source code control with Git. Azure Data Studio provides external database connectivity support for numerous database technologies, including DB2.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 797 of 1295

8 Data Extract Solution Components

The below picture shows overall high-level STARS target state data architecture, this section will cover the STARS Data Extract Services (numbered 1, 2, 3, 4, 5 for Mastered Data and 1.a, 2.a for Unmastered Data).

NOTE: Multiple Data Lake icons have been used in the diagram below for illustrative purposes only. There will be only one physical STARS Data Lake.

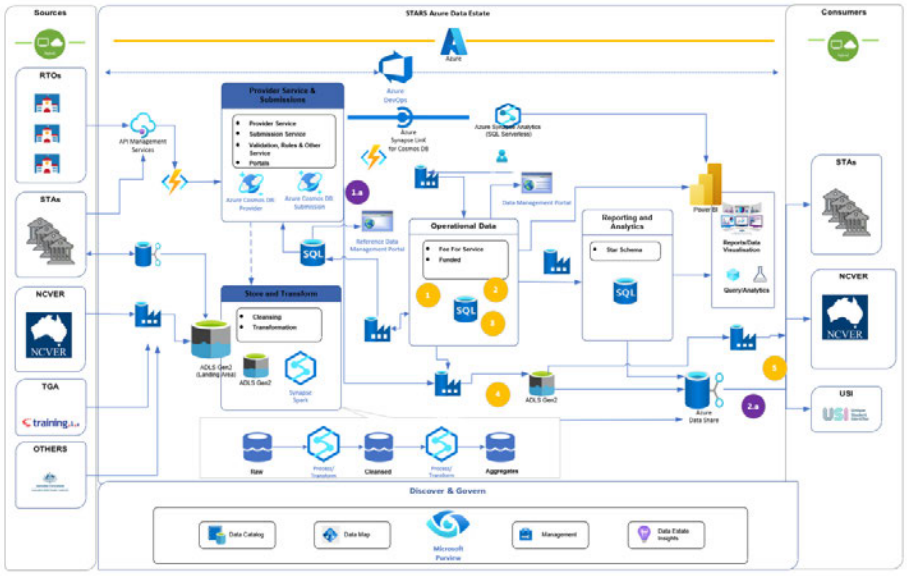


Figure 22: STARS Azure Data Estate

Table 11: Master Database Data Extract Steps

Step No	Description	Technologies
1	Refresh delta data for the current Collection period from the STARS Master database into the LZ schema.	Azure Data Factory, Azure SQL Database
2	Apply the delta to the ST schema of the data transformation zone of the STARS Master database.	Azure Data Factory, Azure SQL Database
3	Data transformations relevant to target data models (i.e., AVETMISS 8, STARS Data Packets) are applied within the TZ schema of the STARS Master database.	Azure Data Factory, Azure SQL Database
4	Generate extract files into Azure Data Lake from the LD schema.	Azure Data Factory, Azure SQL Database, Azure Data Lake
5	Transfer extract files to external consumers.	Azure Data Factory, Azure Data Share

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 798 of 1295

Table 22: Submission Database Data Extract Steps

Step No	Description	Technologies
1.a	Extract documents that have changed (identified by a 'LastModified' attribute stored within each document) since the last data extract for the current Collection period from the STARS Submission database into the Azure Data Lake where Load Method = 'Incremental'. Where Load Method = 'FULL' any existing version of the document will be overwritten. See Folder Structure and File Name Convention section of this document.	Azure Data Factory, Azure Cosmos Database, Azure Data Lake
2.a	Transfer extract files to external consumers.	Azure Data Factory, Azure Data Share

8.1 Data Target and Environment

This section covers the target environment details that will be required to enact transmission of data extracts to external consumers.

Table 33: Target DEV Environment

Source	Data Integration	Dev Details
NCVER	ADF – NCVER Data Target	To be provided
States	Azure Data Share	To be provided

Table 44: Target TEST Environment

Target	Data Integration	Test Details
NCVER	ADF – NCVER Data Target	To be provided
ASQA	Azure Data Share	To be provided
States	Azure Data Share	To be provided

Table 55: Target SIT Environment

Target	Data Integration	SIT Details
NCVER	ADF – NCVER Data Target	To be provided
ASQA	Azure Data Share	To be provided

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 799 of 1295

States	Azure Data Share	To be provided

Table 666: Target UAT Environment

Target	Data Integration	UAT Details
NCVER	ADF – NCVER Data Target	To be provided
ASQA	Azure Data Share	To be provided
States	Azure Data Share	To be provided

Table 777: Target PROD Environment

Target	Data Integration	Prod Details
NCVER	ADF – NCVER Data Target	To be provided
ASQA	Azure Data Share	To be provided
States	Azure Data Share	To be provided

8.2 Extract Data Integration

At the time of writing this document extract data will be transferred using the following methods:

- Site to Site transfer
- Azure Data Share

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 800 of 1295

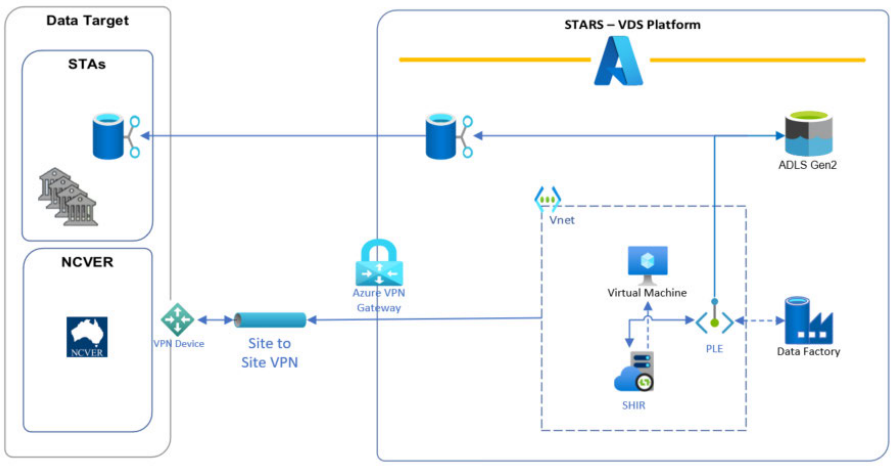


Figure 333: Data Extract Integration Technical Design

Site to Site data Integration:

Azure Data Factory is the logical and secure service to use for ETL operations in STARS. Azure Data Factory will connect to external data sources through a feature called Self-Hosted Integrated Runtime (SHIR). SHIR is a fully managed and serverless Azure-based integration runtime that will move data and process data activities to be performed between on-premise clients and STARS.

To connect to an NCVER data source using SHIR, an integration runtime (IR) will be created in Azure Data Factory (as shown in the above diagram, “Extract Data Integration Technical Design”). The IR will act as a bridge between Azure data factory and the on-premise data source, and it’s responsible for performing the data movement and processing activities.

Once the SHIR is created, it will be configured to connect to the on-premise data target. Security settings will be configured, such as enabling SSL encryption and setting up a firewall to restrict access to the external data target and STARS public IP range address will be whitelisted on the on-premise side.

Once the SHIR is configured, SHIR will be used to transfer data to the on-premise data target using Azure Data Factory by creating pipelines and data flows. The SHIR will handle the secure transfer of data and will provide monitoring and logging capabilities so that the progress of the data movement and processing activities can be monitored.

Data Share data Integration:

An option for data sharing includes the use of the Azure Data Share feature (as shown in the above diagram, “Extract Data Integration Technical Design”). This feature will provide secure, automated data replication to authorised consumers (e.g., STA) via the Azure Data share capability.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 801 of 1295

8.2.1 Extract Data Refresh Interval

Extract data sets will be refreshed into the STARs data lake as follows:

Table 888: STARS Extract Data Refresh Interval

Data Source	Frequency
Mastered Data	Daily
Unmastered Data	Daily

8.2.2 Full/ Incremental

The initial STARS data extract will be full load and then incremental.

8.2.3 Data Volume

Table 999: STARS Reference Data Volume

Data Set	Full/Incremental	Jurisdiction	Data Volume
Mastered Data	Full	National	Up to approx. 10GB
Mastered Data	Full	State	Up to approx. 10GB
Mastered Data	Incremental	National	TBD
Mastered Data	Incremental	State	TBD
Unmastered Data	Full	National	TBD
Unmastered Data	Full	State	TBD
Unmastered Data	Incremental	National	TBD
Unmastered Data	Incremental	State	TBD

8.3 ADF Pipeline – Data Refresh Process

Below is the high-level solution diagram to extract delta data sets for year-to-date data for the current Collection Period from either the STARS Master or Submission Databases to the Data Transformation Schema (LZ) or STARS Data Lake respectively using Azure Data Factory pipeline.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 802 of 1295

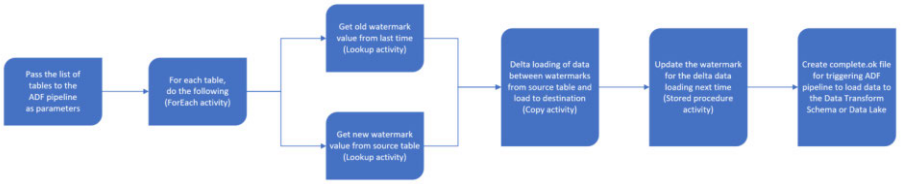


Figure 444: ADF Pipeline steps to refresh data

8.3.1.1 Steps to create this solution:

1. Select the watermark column: Last Modified Time will be selected in the source tables, which will be used to identify the new or updated records/documents for every run.
2. Store the Watermark value: This value will be stored in the data transformation load schema of the Azure SQL Master database.
3. Create ADF pipeline as per the above solution shown in the diagram.
    a. Create a ForEach activity that will iterate through a list of source table names that is passed as a parameter to the pipeline. For each source table, it will invoke the following activities to perform delta loading for that table.
    b. Create two lookup activities. Use the first Lookup activity to retrieve the last watermark value. Use the second Lookup activity to retrieve the new watermark value. These watermark values will be passed to the Copy activity.
    c. Create a Copy activity that will copy rows/documents from the source data store with the value of the watermark column greater than the old watermark value and less than the new watermark value. Then, it will copy the delta data from the source data store to Data Transformation Schema for Mastered Data and the Data Lake for Unmastered Data.
    d. Create a Stored Procedure activity that will update the watermark value for the pipeline that runs next time.
    e. Create complete.ok file to kick off the next ADF pipeline to pull data from the Azure SQL Master database and load it into the Data Transformation Schema or from the Cosmos Submission database into the STARS Data Lake.

8.3.1.2 Watermark table structure:

Table 10101~~0~~: Table name - etl.Watermark

Field Name	Data type

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 803 of 1295

WatermarkKey	Integer, IDENTITY (1, 1) NOT NULL
TableName	NVARCHAR(255)
WatermarkValue	DATETIME
LastStartTime	DATETIME
LastEndtime	DATETIME
LastProcessStatus	NVARCHAR(25)

Sample SQL script:

```
CREATE TABLE etl.watermark
(
    WatermarkKey integer IDENTITY (1, 1) NOT NULL,
    TableName nvarchar(255),
    WatermarkValue datetime,
    LastStartTime datetime,
    LastEndTime datetime,
    LastProcessStatus nvarchar(255),
);
```

8.4 Data Transformation Staging Schemas

Data transformation activities will be performed within database schemas implemented within the STARS Master database. Please see the Data Transformation section of this document.

8.5 ADF Data Extraction – Data Transformation Schema to Azure Data Lake

This section covers the data extraction from STARS Master Database data transformation Load (LD) schema to the STARS Azure Data Lake in the STARS Data Platform.

8.5.1 ADF Pipeline – Generate Files

Below is the high-level solution diagram to extract year to date data for the current Collection Period from the STARS Master Database data transformation Load (LD) schema to the STARS Azure Data Lake using Azure Data Factory pipeline.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 804 of 1295

Figure 555: ADF Pipeline steps to load into Azure Data Lake

8.5.1.1 Steps to create this solution:

1. Create ADF pipeline as per the above solution shown in the diagram.
    a. Create a ForEach activity that will iterate through a list of source table names that is passed as a parameter to the pipeline. For each source table, it will invoke the following activities to perform delta loading for that table.
    b. Create a Copy activity that will copy the data from the source data store to Azure Data Lake as a new file.
    c. Create a Stored Procedure activity that will update the watermark value for the pipeline that runs next time.
    d. Create complete.ok file to kick off the next ADF pipeline to pull data from the data transformation load schema of the Azure SQL Master database and generate an associated file in the Data Lake extract zone.
2. Once all files have been generated a 'ALL_complete.ok' file will be created to indicate that the ADF Data Transfer process can begin.

8.5.2 Raw Data Storage

The extracted STARS data will be stored in Azure Data Lake Gen2 as it has a Windows Azure Storage Blob over Secured Sockets layer (SSL) providing connectors to most windows.

8.5.3 Hierarchical Namespace enabled ADLS Gen2

ADLS Gen2 will be enabled with hierarchical namespace over blob data for atomic directory manipulation.

8.5.4 Folder Structure and File Name Convention

The folder structure for raw data layer will have 8 hierarchical levels as shown in the diagram. After the main folder (Data Integration), subsequent levels will organise the data extracts into their relevant Data Model, Data Source, Project and Load Method as described below:

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 805 of 1295

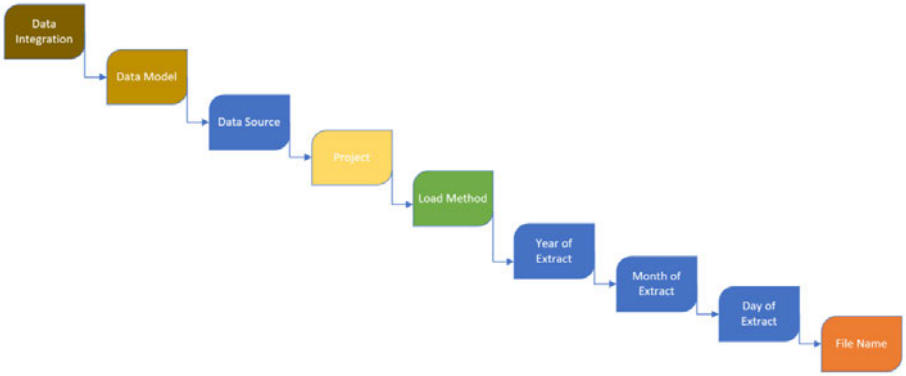


Figure ~~66~~: Data Lake Folder Structure

1. **Root Folder for Layer Identification:** The root folder identifies the logical layer or the data tier.
   E.g., RDL (Raw Data Layer)/DIL (Data Integration Layer)/DEL (data Enrichment Layer)/DAL (Data Access Layer) or Bronze/Silver/Gold
2. **Data Model Identification:** The second level folder will indicate which data model the extract conforms to I.e., VET IS, Data Packet or AVETMISS 8.
3. **Data Source:** This is to identify the system of records and/or the authenticated source system. This will be STARS Submission Service or STARS Master Data Service.
4. **Project:** This is to identify the project that has extracted the data, it will be STARS Data Extract (SDE).
5. **Load Method:** This will either be either:
   a. 'Full', files within this folder will contain the entire year to date data for the collection period.
   b. 'Incremental', files within this folder will contain only records that have changed since the last equivalent extract file was generated.
   c. 'Changed RTO Full', files within this folder will contain the entire year to date data for the collection period belonging to RTOs whose data sets have changed since the last set of extract files was generated.
6. **Data Partition Folders:** Based on the frequency of data transfer Year, Month and Day are the required folders for intraday extract files.
7. **File Name Convention**: It is best practice to prefix/postfix the context to the files that are ingested and created. Example: for raw files, if the target file is "~~NAT00010A~~Person", naming will be:

   DIL_ AVETMISS8_STARSMasterDataService_SDE_ ~~NAT00010A~~Person_20230214

   To preserve backwards compatibility with legacy ingestion processes, where the Load Method is 'Changed TO Full' AVETMISS NAT files will be extracted retaining their original names (i.e., NAT00010A.txt). The naming convention of the compressed archive (i.e., .zip) that contains these files will contain the associated RTO and the DateTime that the data was extracted as per follows:

   <RTO>_20230214_1123

Formatted: Indent: Left: 1.27 cm, First line: 0 cm

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 806 of 1295

8. **The full folder path will be as follows**:

   <data-layer>/<data-model>/<data-source>/<project-name (Optional)>/<year>/<month>/<day>/ <file_name>

   Example:

   DIL/AVETMISS8/STARSMasterDataService/SDE/2022/02/14/DIL_AVETMISS8_STARSMasterDataService_SDE_ NAT00010A_20230214

8.5.5 File Format for Storage

Files will be of the formats described in the File Format section of this document. File Sets will be compressed as .zip archives.

8.6 ADF Data Transfer – STARS Azure Data Lake to Client Stie-to-Site VPN

The following section covers data transfer from Azure Data Lake to external data consumers using an Azure Data factory pipeline via Site-to-Site VPN.

8.6.1 ADF Pipeline – data load

ADF Pipeline will be created to transfer data from Azure Data Lake to external data consumers. This pipeline will be triggered when the 'ALL_complete.ok' file is created as part of section ADF Pipeline – Generate Files.

ADF pipeline will loop through all the file extracts that were created on the latest run (Watermark table) in Azure Data Lake and will transfer the data extract files to the external data consumer.

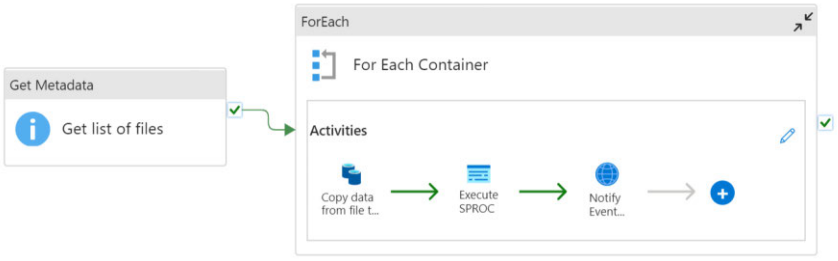


Figure 777: ADF Pipeline Data Transfer

The For each container will loop through all generated files for a client for the current extract set and then a Web activity task will notify the STARS Event service upon successful transfer of files. Configuration will be stored relating to a clients particular Load Method, Data Transformation Output and Data Source (i.e., Mastered, Unmastered data)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 807 of 1295

8.7 ADF Logging

8.7.1 Logs

Azure Data Factory (ADF) keeps the pipeline-run-data for only 45 days by default. Azure Monitor (AM) will be integrated with ADF to keep the logs data for a longer time and route diagnostic logs for further analysis. AM will provide higher visibility on the performance and health metrics of the workloads which will; be used for monitoring and trouble shooting.

Examples: Count of – pipeline failures, succeeded activities, failed trigger runs, IR failures, failed activities due to insufficient or unavailable memory for IR, etc.,

8.7.2 Alerting and Notification

Alerts and notifications will be generated for STARS and Stakeholders by calling Notification Service API (User Notification) from ADF pipeline to add a new notification to the Notification Service. The notification will be sent via email (depending on the user's preferences) and can be retrieved via the Notification API.

For more details refer section 8.3.5 Detailed Design Document - Notification Service D.0.4.

8.7.3 Exception Handling

ADF Pipelines fail for variety of reasons. While this is expected, it is important that the system safely recover from errors, they will be logged appropriately and that the system administrators will be notified immediately for remediation actions.

The activities in ADF have 'activity dependency statuses' such as success, failure, completion and skipped (condition based skipped) built-in which will be used for exception handling scenarios.

8.7.4 Custom Logging

The STARS Downstream Consumer ETL Pipeline Service will maintain a custom pipeline log table for storing the ADF pipeline activity details as records in the log table for each run.

Sample SQL script to create log table in Azure SQL Master database:

```
CREATE TABLE [etl].[PipelineLog](
   PipelineLogKey bigint IDENTITY (1, 1) NOT NULL,
   DataFactoryName nvarchar(500) NULL,
   PipelineName nvarchar(500) NULL,
   RunId nvarchar(500) NULL,
   Source nvarchar(500) NULL,
   Destination nvarchar(500) NULL,
   TriggerType nvarchar(500) NULL,
   TriggerId nvarchar(500) NULL,
   TriggerName nvarchar(500) NULL,
   TriggerTime nvarchar(500) NULL,
   RowsCopied nvarchar(500) NULL,
   RowsRead int NULL,
   NumberOfParallelCopies int NULL,
   CopyDurationInSeconds nvarchar(500) NULL,
   EffectiveIntegrationRuntime nvarchar(500) NULL,
   SourceType nvarchar(500) NULL,
   SinkType nvarchar(500) NULL,
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 808 of 1295

```
    ExecutionStatus nvarchar(500) NULL,
    CopyActivityStartTime datetime NULL,
    CopyActivityEndTime datetime NULL,
    CopyActivityQueuingDurationInSeconds nvarchar(500) NULL,
    CopyActivityTimeToFirstByteInSeconds nvarchar(500) NULL,
    CopyActivityTransferDurationInSeconds nvarchar(500) NULL,
) ;
```

8.7.5 Metadata Extraction and Cataloguing

Microsoft Purview will be connected to the ADF for capturing the metadata lineage, and to discover and explore data assets across the STARS data platform. This will be covered in a separate document Purview – technical Detailed Design.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 809 of 1295

9 Data

9.1 Schemas

9.1.1 AVETMISS 8

As per the data schema defined within the DRAFT VET Info Std Mapping to NAT Files - Reverse Engineering R1_V1.0 document.

9.1.2 VET IS

As per the VET IS physical database data schema defined within the 'Detailed Design Document - Master Data Service.docx' document.

9.1.3 VET IS to STARS Collection Specification (Data Packet)

As per the data schema defined within the DRAFT VET Information Standard Logical Interchange Specification V2 R1_V1.0.xlsx document.

9.1.4 State Specific Data Transformation

State specific data transformation has been specified as per the following:

9.1.4.1 South Australia

As per the data schema defined within the 'NATFilesFormat - and SA NAT File Extract Requirement.xlsx' document.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 810 of 1295

10 Access Management

The Downstream Consumer ETL Pipeline Service will conform to the Data Governance Access and Privacy Protocol document and the IDAM detail design. Access to the Downstream Consumer ETL Pipeline Service will be controlled by Azure RBAC, with all calling services identified and granted access through Azure AD Managed Identities. For detail about accessing the Downstream Consumer ETL Pipeline Service please refer to IDAM detail design document.

10.1 Control of data

The Governance Framework sets out the key data governance roles surrounding VET Data Streamlining.

10.1.1 Data Owner

The Data Owner has authority for establishing rules surrounding the quality, protection and appropriate access to data. They individually and collectively have a responsibility to protect the personal and sensitive information provided by Data Providers. Data can be jointly or co-owned.

10.1.2 Data Custodian

The Data Custodians receive authorised data from Data Providers or from other Data Custodians. Data Custodians are authorised to collect and handle that data to perform their functions. They are responsible for ensuring that the personal information collected and stored is reasonably necessary for them to conduct their functions and activities in relation to the VET sector.

For detail about the data access and control please refer to the Data governance Access and Privacy protocol document.

10.2 Security

The STARS solution will be built to meet an existing Australian Government policies, frameworks, standards and design principles, including:

- Australian Government API Design Standards
- Digital Service Standard
- Protective Security Policy Framework (PSPF)
- Information Security Registered Assessors Program (IRAP)
- Australian Government Information Security Manual (ISM)
- the Open Web Application Security Project® (OWASP)
- Hosting Certification Framework.

All data within STARS will be handled in accordance with relevant privacy laws and record keeping requirements, including the Privacy Act, the *Archives Act 1983* (Cth) and any other applicable legislation (e.g. applicable State and Territory privacy legislation or archiving requirements).

The Downstream Consumer ETL Pipeline Service will conform to the Security Architecture defined for STARS in Data Governance Access and Privacy Control document and IDAM detail design document.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 811 of 1295

Access to the Downstream Consumer ETL Pipeline Service will be controlled by Azure RBAC, with all calling services identified and granted access through Azure AD Managed Identities.

For detail about security please refer to the following

- Data Governance Access and Privacy Control document
- IDAM detail design document

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 812 of 1295

11 Dependency on Other Services

The following are STARS services upon which Downstream Consumer ETL Pipeline Service has dependencies.

11.1 Event Service

The Downstream Consumer ETL Pipeline Service will depend upon the Event service for raising events relating to the status of a data transfer. Please refer to Event Service detail design document for details about the Event Service.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 813 of 1295

12 Acronyms

Table 111~~11~~ - Acronyms

Term/Acronym	Expanded Form
ACSC	Australian Cyber Security Centre
AD	Active Directory
ADF	Azure Data Factory
ADLS	Azure Data Lake (Gen2) Storage
AI	Artificial Intelligence
API	Application Programming Interface
APIM	Application Programming Interface Manager
ASQA	Australian Skills Quality Authority
AVETMISS	Australian Vocational Education and Training Management Information Statistical Standard
BAU	Business as usual
BI	Business Intelligence
CARMS	Classification and Reference Management System
CMS	Contract Master Schedule
DESE	Department of Education, Skills and Employment
DevOps	Development Operations
DEWR	Department of Employment and Workplace Relations
DIU	Data Integration Units
ETL	Extract, Transform and Load
FFS	Fee For Service
FA	Funded Activity
FR	Functional Requirement
Iaas	Infrastructure as a Service
IAM	Identity and Access Management

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 814 of 1295

Term/Acronym	Expanded Form
ICT	Information and Communication Technology
IRAP	Information Security Registered Assessors Program
ISM	Information Security Manual
ITIL	Information Technology Infrastructure Library
IVETS	Integrated Vocational Education and Training System (NSW Dept of Education)
JSON	JavaScript Object Notation
JWT	JSON Web Token
LAW	Log Analytics Workspace
ML	Machine Learning
MDM	Master Data Management
NCVER	The National Centre for Vocational Education Research
NCI	National Careers Institute
NESM	New Employment Services Model
NFR	Non-Functional Requirements
NSA	National Skills Agreement
NSC	National Skills Commission
NVA	Network Virtual Appliance
NVETR	National Vocational Education and Training Regulator
OIDC	OpenID Connect
PaaS	Platform as a Service
PIA	Privacy Impact Assessment
PoC	Proof of Concept
PMP	Project Management Plan
PR	Production Environment
PRISM	Provider Registration and International Student Registration System

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 815 of 1295

Term/Acronym	Expanded Form
PSPF	Protective Security Policy Framework
RAAIDD	Risk Actions, Assumptions, Issues, Decisions, and Dependencies
RBAC	Role Based Access Control
REST	Representational State Transfer
RG	Resource Group
RTO	Registered Training Organisation
SaaS	Software as a Service
SAML	Security Assertion Markup Language
SDK	Software Development Kit
SDLC	Software Development Life Cycle
SEE	Skills for Education and Employment
SHIR	Share-Hosted Integration Runtime
SKU	Stock Keeping Unit
SMS	Student Management System
SSO	Single Sign On
SSON	Skills Senior Officials' Network
STA	State Training Authority
STARS	Student Training & Activity Reporting System
TGA	Training.gov.au (National VET directory website)
TP	Training Provider
TLS	Transport Layer Security
TRA	Trades Recognition Australia
TVA	Total VET Activity
UAT	User Acceptance Testing
USI	Unique Student Identifier

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 816 of 1295

Term/Acronym	Expanded Form
VDP	Unique Student Identifier
VDS	VET Data Streamlining
VET	Vocational Education and Training
VIS	VET Information Standard
VM	Virtual Machine
VNet	Virtual Network
VSL	VET Student Loans
VVCT	VSL, VET Compliance and TRA
WCAG	Web Content Accessibility Guidelines

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 817 of 1295

Australian Government
Department of Employment and Workplace Relations

Style Definition: TOC 3: Tab stops: 2.33 cm, Left + 15.98 cm, Right,Leader: ...

Student Training and Activity Reporting System (STARS)

Detailed Design Document

Azure Infrastructure and Connectivity

1 Document Management

1.1 Document Control

Version	Change Description	Date
D.0.1	Initial draft	8/2/2023
D.0.2	Incorporate first round of feedback	23/2/2023
1.0	Approval	3/3/2023
1.1	Incorporated late feedback from oobe	17/4/2023

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This design is a living document and DXC will be required to update this document through the build phase where required.

1.2 Document Author

Role	Name
Solution Lead	s 47F(1)

1.3 Document Reviewers

Role	Name

1.4 Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

1.5 Related Documents

Reference	Document
A	STARS Solution Requirements Document – 1.0.docx
B	STARS Solution Design – 1.0.docx
C	VDS – Azure Naming Standards.xlsx
D	VDS – Perimeta for Azure – Solution Design v.1.2

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 819 of 1295

Contents

1 Document Management 2
1.1 Document Control 2
1.2 Document Author 2
1.3 Document Reviewers 2
1.4 Document Approval 2
1.5 Related Documents 2
Contents 3
2 Introduction 5
2.1 Major Design Topics 5
2.2 Purpose 6
2.3 Intended Audience 7
3 Scope and Exclusions 8
3.1 In Scope 8
3.2 Out of Scope 8
4 Design Methodology, Guidance and Patterns 9
4.1 Microsoft Frameworks 9
5 Infrastructure 11
5.1 Azure Landing Zones 11
5.2 Azure Landing Zone Guidance 12
5.3 STARS Hub-and-Spoke Configuration 14
5.3.1 Hub PaaS Services 14
5.3.2 Hub Key Vault 15
5.4 STARS High Availability 15
5.5 STARS Azure Landing Zone Design 17
5.5.1 Management Groups and Subscriptions (Oobe) 20
5.6 STARS Application Resource Groups 21
6 Connectivity 2524
6.1 STARS External Connectivity 2624
6.1.1 STARS to Partner Agency Connectivity 2624
6.1.2 STARS to Training Provider SMS Connectivity 2726
7 Requirements 3028
7.1 Non-Functional 3028
8 Appendices 3533

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 820 of 1295

Appendix 1. Resources 3533
Appendix 2. Acronyms..... 3634

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 821 of 1295

2 Introduction

This document covers the design and organisation of components of Azure infrastructure and connectivity for the STARS solution down to the resource group and network level. It does not go beyond these levels – topics such as design and configuration of Azure PaaS services are covered in each of the various STARS services design documents.

Note: This is a living document that will be updated as designs for various other STARS services are developed and evolve. Accordingly, this document will evolve to reflect and document infrastructure and connectivity as the STARS solution build progresses. For example, as tenant level Azure policies for VNet separation are updated, the changes will be reflected in this document.

[redacted]

Please note – this is not 'as-built' infrastructure documentation. Detailed as-built documentation will be provided during the project, including Azure ARM templates and deployment pipelines.

Additionally, this document is intended to be supplemental to the more detailed DEWR MSP provided infrastructure documentation. Its purpose is to provide the infrastructure context for other STARS services designs. Please refer to document D – VDS Perimeta for Azure Solution Design.

The design is informed by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework, which provide a comprehensive set of best practices and guidance for designing, deploying, and operating secure, high-performing, and resilient cloud infrastructure and applications.

One key aspect of the design is the use of Azure Landing Zones as the infrastructure and networking design and deployment framework, which provide a consistent and efficient way to implement an Azure environment with a focus on governance, security, and operations. This helps to ensure that the Azure infrastructure and networking components are deployed in a standardized manner and provide a foundation for ongoing management and evolution of the STARS solution.

Of note is that many elements of the infrastructure and networking design are based upon Microsoft recommendations for hub-and-spoke technical architecture.

~~Additionally, this document is intended to be supplemental to the more detailed DEWR MSP provided infrastructure documentation. Its purpose is to provide the infrastructure context for other STARS services designs. Please refer to document D – VDS Perimeta for Azure Solution Design.~~

2.1 Major Design Topics

The design covers subscription and resource group level designs and configuration that cover major topics:

1. Azure Active Directory (AD) for identity and access management.
2. Virtual networks, subnets, network security groups, and virtual network peering for network segmentation and secure communication within and between Azure virtual networks.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 822 of 1295

3. Load balancing options, such as Azure Load Balancer and Application Gateway, to distribute incoming traffic and ensure high availability.
4. Azure ExpressRoute for dedicated, private network connections to Azure from on-premises or other cloud locations.
5. Azure VPN Gateway for secure network connections over public infrastructure to Azure from on-premises and other cloud locations.
6. Azure Backup, Azure Site Recovery, and Azure Disaster Recovery for data protection and disaster recovery.
7. Azure Monitor, Azure Log Analytics, and Azure Security Centre for monitoring, logging, and security operations.
8. Connectivity with Microsoft SaaS services, such as Power Platform and Power BI, to enhance the STARS solution with additional functionality and data analysis capabilities.

High-Availability Configurations:

To ensure high availability and disaster recovery for the STARS solution, this design considers the use of multiple Azure regions and high-availability configurations. Azure provides several options for deploying highly available applications and services, including:

1. Azure Availability Zones, which provide physically separate and isolated locations within an Azure region to protect applications and data from datacentre failures.
2. Azure Region Pairs, which allow you to deploy active-active solutions across two Azure regions for cross-region resiliency.
3. Azure Geographies, which allow you to deploy solutions across multiple Azure regions within a geography for disaster recovery purposes.

This document provides a comprehensive understanding of the Azure infrastructure and networking components and their configurations, including high-availability options, for the STARS solution, helping to ensure its successful deployment and operation, as well as seamless integration with Microsoft SaaS services.

2.2 Purpose

The purpose of this detailed design document is to provide a technical description of the Azure infrastructure and networking components of the STARS solution in the context of the DEWR MSP configured and managed infrastructure. This document is supplemental to the DEWR MSP Perimeta for Azure Solution Design, in the sense of describing the STARS application landing zone patterns in the context of the overall DEWR MSP configured infrastructure.

The document is intended for technical stakeholders, including architects, software engineers, administrators, and operations teams, who are responsible for designing, developing, deploying, and operating the STARS solution in Azure.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 823 of 1295

The document also serves as a reference for ongoing management and evolution of the STARS solution and to ensure that the STARS solution is deployed and operated in accordance with the best practices and guidance provided by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework.

The document is intended to be a living document, updated as the STARS solution evolves and changes over time. It provides a clear and concise understanding of the Azure infrastructure and networking components and their configurations, helping to ensure the STARS solution is deployed, operated, and managed in a consistent and efficient manner.

2.3 Intended Audience

This detailed design document provides technical information and guidance to help technical stakeholders understand the Azure infrastructure and networking components of the STARS solution and how they integrate with each other. The document is expected to be used by technical stakeholders who have a good understanding of cloud computing concepts and are experienced in the Azure platform.

This document is not intended for non-technical readers, nor for readers who are not experienced in the design and configuration of Azure-based solutions; informational and educational resources regarding Azure are readily available at https://azure.microsoft.com

The primary audience for this document includes:

- Program technical staff: This includes architects, software engineers, administrators, and operations teams within the program responsible for the STARS solution.
- External technical stakeholders: This includes technical staff from other federal, state, and territory governments agencies, including VDS embedded team members.
- ICT Technical Staff: This includes a variety of technical roles involved during the VDS program and post-go live, including network engineers, systems administrators, database administrators, security engineers, and DevOps engineers.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 824 of 1295

3 Scope and Exclusions

The scope of this detailed design document is restricted to describing relevant Azure infrastructure and network connectivity configurations for the STARS solution, utilizing guidance from the Microsoft Cloud Adoption Framework (MCAF) and the Azure Well-Architected Framework, with reference to Azure Landing Zones (LZs).

Whilst the high-level infrastructure configuration of STARS LZs is the responsibility of the DEWR MSP~~Oobe~~, the application oriented LZs/resource groups are the responsibility of DXC.

Please note – this is not 'as-built' documentation. Detailed as-built documentation will be provided during the project, including Azure ARM ~~T~~templates and deployment pipelines.

3.1 In Scope

The scope of this document is limited to describing the relevant Azure configurations for the STARS solution, as well as the configurations provided by the DEWR MSP~~Oobe~~ and DXC, with a particular focus on the relevant Azure Landing Zones patterns, associated resource groups, the roles of the resource groups, and associated network configurations. The document provides a high-level overview of the DEWR MSP~~Oobe~~ configured infrastructure, while providing a ~~detailed~~ description of the DXC-configured LZs/RGs and networking services within the Application LZs/RGs.

The document scope includes a summary description of the DXC-configured Azure PaaS services within the Application LZs, but the detailed descriptions of these services are not included in this document.

3.2 Out of Scope

The detailed descriptions of DXC configured STARS Azure PaaS services are excluded from this document; they are covered in other design documents, such as the IDAM design document and the various STARS pipeline services design documents.

The detailed description of DEWR MSP configured infrastructure is found in document D – VDS Perimeta for Azure Solution Design.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 825 of 1295

4 Design Methodology, Guidance and Patterns

4.1 Microsoft Frameworks

This design document is informed by the following Microsoft frameworks. These are extensively documented online and cover many aspects. With regards to the scope of this document, the linkages listed below are restricted to infrastructure and networking:

Microsoft Cloud Adoption Framework (CAF)

- Provides guidance on designing, deploying, and managing Azure solutions
- Discusses various options for connecting Azure tenants, and Azure to on-premises datacentres
- Provides guidance on securing Azure solutions, including data in transit
- Discusses support for hybrid environments

Azure Landing Zones

- Provides a set of proven deployment patterns and best practices for designing and deploying Azure solutions
- Provides guidance on the network architecture, including IP addressing, subnets, routing, and firewall rules
- Provides guidance on the network design to meet bandwidth requirements
- Discusses ExpressRoute and VPN/Site-to-Site VPN as key components of Azure connectivity

Azure Well-Architected Framework

- Provides guidance on designing and operating secure, high-performing, resilient, and efficient Azure solutions
- Provides guidance on monitoring and troubleshooting techniques for Azure connectivity, including using Azure Monitor, Network Watcher, and other tools.

Linkages with frameworks

Expressed in terms of this document's scope, the linkages to specific subjects are:

1. Azure connectivity options: The CAF provides guidance on how to design, deploy, and manage Azure solutions. In the context of connectivity, the CAF provides guidance on the various options for connecting Azure tenants, Azure, and on-premises datacentres.
2. Security considerations: The CAF provides guidance on how to secure Azure solutions, including data in transit. In the context of connectivity, the CAF provides guidance on the security measures for protecting data in transit, such as encryption and authentication methods.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 826 of 1295

3. Network design: Azure Landing Zones provide a set of proven deployment patterns and best practices for designing and deploying Azure solutions. In the context of connectivity, Azure Landing Zones provide guidance on the network architecture, including IP addressing, subnets, routing, and firewall rules.
4. Bandwidth requirements: Azure Landing Zones provide guidance on how to design the network to meet the bandwidth requirements of different use cases.
5. Azure ExpressRoute: ExpressRoute is a key component of Azure connectivity and is discussed in the context of Azure Landing Zones and the CAF.
6. VPN and Site-to-Site VPN: VPN and Site-to-Site VPN are important components of Azure connectivity and are discussed in the context of Azure Landing Zones and the CAF.
7. Hybrid scenarios: The CAF provides guidance on how to support hybrid environments, including scenarios where on-premises resources are connected to Azure and scenarios where multiple Azure tenants are connected.
8. Monitoring and troubleshooting: The Azure Well-Architected Framework provides guidance on how to design and operate secure, high-performing, resilient, and efficient Azure solutions. In the context of connectivity, the Azure Well-Architected Framework provides guidance on monitoring and troubleshooting techniques for Azure connectivity, including using Azure Monitor, Network Watcher, and other tools.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 827 of 1295

5 Infrastructure

5.1 Azure Landing Zones

Azure Landing Zones is a reference architecture for building multi-cloud solutions on Microsoft Azure that follows best practices for security, compliance, governance, and operational excellence. The goal of Azure Landing Zones is to provide a repeatable and consistent foundation for Azure deployments.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 828 of 1295

5.2 Azure Landing Zone Guidance

The following LZ guidance has informed the design.

Table 11~~1~~: LZ Guidance

Topic	LZ Guidance (followed by design)
Networking: The network component of Azure Landing Zones provides guidelines for designing secure and scalable network topologies using Azure Virtual Networks (VNets), Azure Firewalls, and Azure ExpressRoute.	STARS network topology is designed to ensure that different workloads are isolated from each other and that only authorized traffic is allowed to flow between them. Azure Virtual Networks are used to logically isolate network traffic and provide a secure network perimeter for STARS workloads. Network security controls, such as Azure/FortiGate Firewalls, will be used to enforce network traffic policies and protect against external threats. Where possible, dedicated network connections to STARS Azure tenancies are made using Azure ExpressRoute to provide a high-speed, low-latency, and secure connection to the cloud. Failing this, secure IPSec site-site VPNs will be used.
Identity and Access Management: The identity and access management component of Azure Landing Zones provides recommendations for implementing Azure Active Directory (AD) and Azure AD Domain Services, as well as best practices for role-based access control (RBAC) and multi-factor authentication (MFA).	STARS will use Azure Active Directory to centralize and manage user identities and access to resources in the cloud. STARS will, if necessary, implement Azure AD Domain Services to provide domain authentication services for legacy applications that require it (e.g., NCVER systems.) STARS will use role-based access control (RBAC) to grant users, groups and applications access to resources based on their role and responsibilities.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

Topic	LZ Guidance (followed by design)
	STARS will implement multi-factor authentication (MFA) to provide an additional layer of security for users accessing sensitive resources in the cloud, notably MyGovID MFA for RTOs.
Governance: The governance component of Azure Landing Zones provides best practices for establishing a governance framework using Azure Policy, Azure Blueprints, and Azure Resource Manager templates.	STARS will utilise Azure Policy to enforce organisational and Australian Government policies and standards across all resources in the cloud. STARS will use Azure Blueprints to define and manage cloud environments using a combination of Azure Policy and Azure Resource Manager templates. STARS will use Azure Resource Manager templates to automate the deployment of resources in a consistent and repeatable manner.
Security: The security component of Azure Landing Zones provides guidelines for securing data and applications using Azure Security Centre, Azure Key Vault, and Azure Disk Encryption.	STARS will utilise Azure Security Centre to proactively monitor and protect resources from security threats.
Operations: Azure Landing Zones provides best practices for automating and streamlining operations.	STARS will utilise Azure Monitor, Azure Log Analytics, and Azure Resource Manager templates, along with Azure DevOps Pipelines for resource deployment.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

5.3 STARS Hub-and-Spoke Configuration

The STARS solution follows the Azure Landing Zones recommendations to implement a hub-and-spoke architecture in Azure. In this architecture, the "hub" resource group acts as a central shared infrastructure for multiple "spoke" resource groups, each of which contains application-specific resources.

The hub resource group will be configured with Azure infrastructure services that are shared across all spoke resource groups, including Azure Virtual Network (VNet), Azure Network Security Groups (NSGs), Azure ExpressRoute or VPN gateways, and Azure Load Balancer.

These components will be deployed and configured using ARM templates, which can be parameterized to allow for customization while still adhering to the policies and configurations defined in the hub resource group.

The spoke resource groups will contain the application-specific resources, including Azure PaaS services such as Azure App Services, Azure Functions, and Azure SQL databases. By utilizing the hub-and-spoke architecture, the STARS solution will be able to enforce consistent policies and configurations, centralise monitoring and logging, and reduce management overhead for the entire solution.

5.3.1 Hub PaaS Services

Table 222: Hub PaaS Services

Hub Azure Infrastructure Service	Purpose	How it will be used by STARS Application Spoke PaaS services
Azure Active Directory (AD)	Central identity management solution	Used for authentication and authorization of users and applications. Enforces role-based access control (RBAC) and authorization policies.
Azure Policy	Governance and compliance enforcement	Enforces governance and compliance policies for Azure Hub PaaS services. Maintains a secure and compliant environment.
Azure Monitor	Performance and health monitoring	Provides real-time monitoring and alerting for Azure Hub PaaS services. Diagnoses and troubleshoots issues within the environment.
Azure Log Analytics	Log data collection and analysis	Collects and analyses log data for Azure Hub and Spoke PaaS services. Provides insights into performance, health, and security.
Azure Security Centre	Security and threat protection management	Manages security and threat protection for Azure hub and spoke IaaS and PaaS services. Provides continuous security assessments, threat detection, and security recommendations.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 831 of 1295

Hub Azure Infrastructure Service	Purpose	How it will be used by STARS Application Spoke PaaS services
Azure Key Vault	Secret management	Stores secrets, such as passwords and certificates, used by Azure hub and spoke PaaS services. Ensures secrets are managed and protected in a secure and compliant manner.

5.3.2 Hub Key Vault

STARS will utilise Azure Key Vault is to create a centralized instance in the hub resource group(s) to store and manage secrets and certificates. This centralized Key Vault instance will be referenced by multiple spoke resource groups, allowing for consistent management and protection of secrets and certificates across the entire solution.

In the hub resource group, the Key Vault instance will be created using an ARM template and will be configured with access policies and policies for certificate management. The template will be parameterized to allow for customization of the Key Vault instance, such as specifying the virtual network, subnet, and network security group for Key Vault.

In the spoke resource groups, the Key Vault reference will be added to ARM templates for resources that require secrets, such as App Services, App Configuration, and Azure databases. This allows for secrets to be securely stored and managed in the central Key Vault instance while still allowing spoke resource groups to reference them as needed.

5.4 STARS High Availability

Each of the Azure resources included in the STARS will be deployed with appropriate configurations to ensure high availability and redundancy where required.

Microsoft regularly updates its SLA levels for all services, for the purposes of this document the SLAs will be considered the latest available at: https://www.microsoft.com/licensing/docs/view/Service-Level-Agreements-SLA-for-Online-Services[1]

During all build stages, the underlying Azure infrastructure will be configured and tested repeatedly in various failure modes to ensure that HA is achieved.

The STARS Azure platform will be configured with High Availability as per below:

Table 333: STARS High Availability Configurations

[1] A side effect of licencing Azure services is that Microsoft is not responsible for claims by solution developers for their products that their product SLAs exceed those of the underlying Microsoft services. Accordingly, DXC will not be able to deliver requirements that exceed the stated Microsoft SLAs.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 832 of 1295

Azure Service	HA Configuration
Azure Cosmos Database	Azure Cosmos DB in STARS will be configured to replicate data across two regions "Australia East" and "Australia Southeast". Azure Cosmos DB account in STARS will be configured with a single write region in "Australia East" and a read region in "Australia Southeast" with Service-Managed failover enabled. Enabling this Service-Managed failover option will allow Azure Cosmos DB to fail over to the second region with no user action should a region become unavailable.
Azure SQL Database	For business continuity and higher availability of Azure SQL Databases (in Elastic pool), the SQL databases will be configured in the auto-failover group across two regions "Australia East" (Primary region) and "Australia Southeast" (Secondary region). The Primary database will be configured in "Australia East" and the secondary database in "Australia Southeast" in the auto-failover group, this feature will manage the database replication and automatically recover databases in a secondary region after an unplanned event that results in full or partial loss of the SQL Database availability in the primary region.
Azure Data Lake Gen2	Data Lake is not a core element of STARS but is proposed as a storage service for raw data that may in future need to be analysed. If required, DXC recommends the GZRS option below: Locally redundant storage (LRS) – This is the least expensive replication option where data is replicated three times in the same data centre in the primary region. LRS provides 99.999999999% availability. Zone-redundant storage (ZRS) – This is recommended for high availability and performance; with ZRS data is replicated across three Availability zones in the primary region. However, ZRS by itself may not protect your data against a regional disaster where multiple zones are permanently affected. Geo-Zone redundant storage (GZRS) – This is recommended for high availability and disaster recovery, this uses ZRS in the primary region and geo-replicates your data to a secondary region.
Azure Service Bus	Azure Service Bus Premium will be configured, at least in production environments within Australia East. Availability zones will be utilised to reduce the risk of service outages at region level. Service Bus Geo-

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 833 of 1295

Azure Service	HA Configuration
	disaster recovery will be appropriately configured to cater for extended region-level failures.
Azure Event Grid	Azure Event Grid Premium will be configured, at least in production environments within Australia East. Availability zones will be configured to reduce the risk of service outages at region level. Event Grid region-level outages will not be critical to the operation of STARS . Microsoft does have guidance on replication strategies for Event Grid across regions, but DXC does not at this point believe they are worth implementing.
Azure Functions	Functions will be configured in both Australia East and Southeast regions in an active/passive formation, with the active functions within Australia East (along with all other 'active' services.) Failover between functions will be automatically enabled through either Azure App Gateway or Azure APIM. Premium Plans will be configured, at least in production environments. Dev/Test environments may be configured using consumption-based plans.

5.5 STARS Azure Landing Zone Design

The overall STARS Azure Landing Zone technical architecture has been designed and implemented by the DEWR MSP~~Oobe~~, including Management Groups, Subscriptions and several hub RGs, including Management and Identity. Detailed design and as-built documentation for these are provided separately by the DEWR MSP~~Oobe~~, however a summary description is provided below.

Within this Azure LZ-based technical framework, DXC are responsible for design and configuration of the application LZs and associated RGs. These are described, down to the RG level below. Please refer to the STARS Solution Design document and the other STARS services design documents for information below the RG level (e.g., the design of Azure Functions based APIs, Azure SQL schemas).

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 834 of 1295

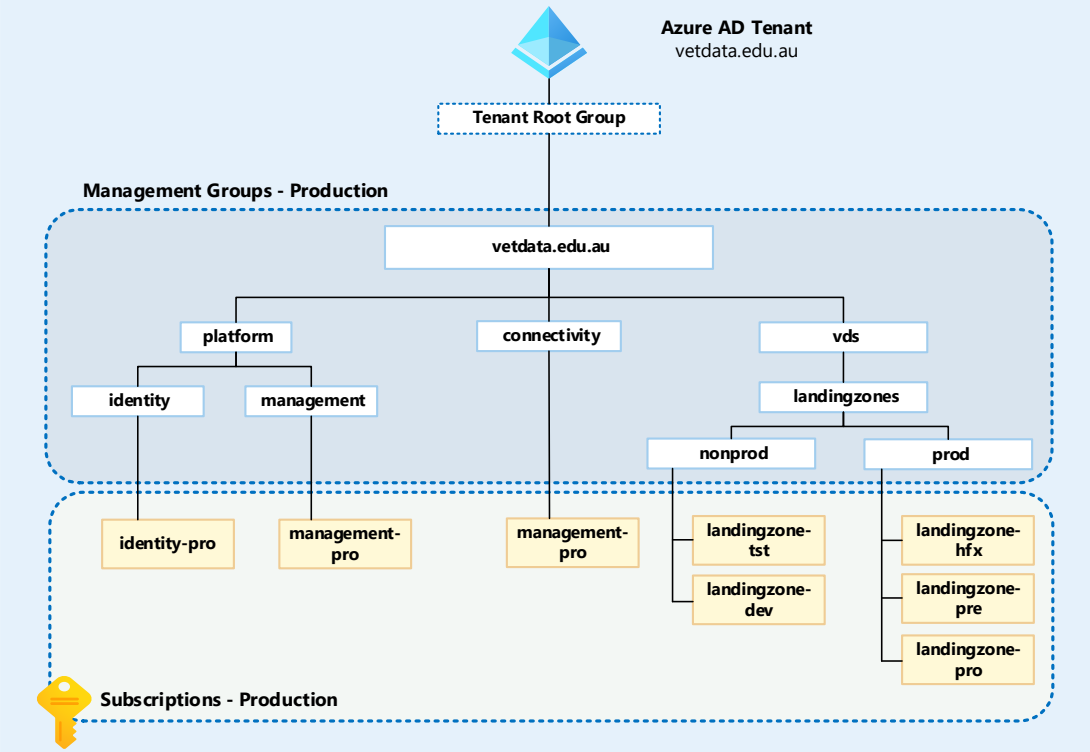


Figure 111: STARS Management Group and Subscriptions Hierarchy

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

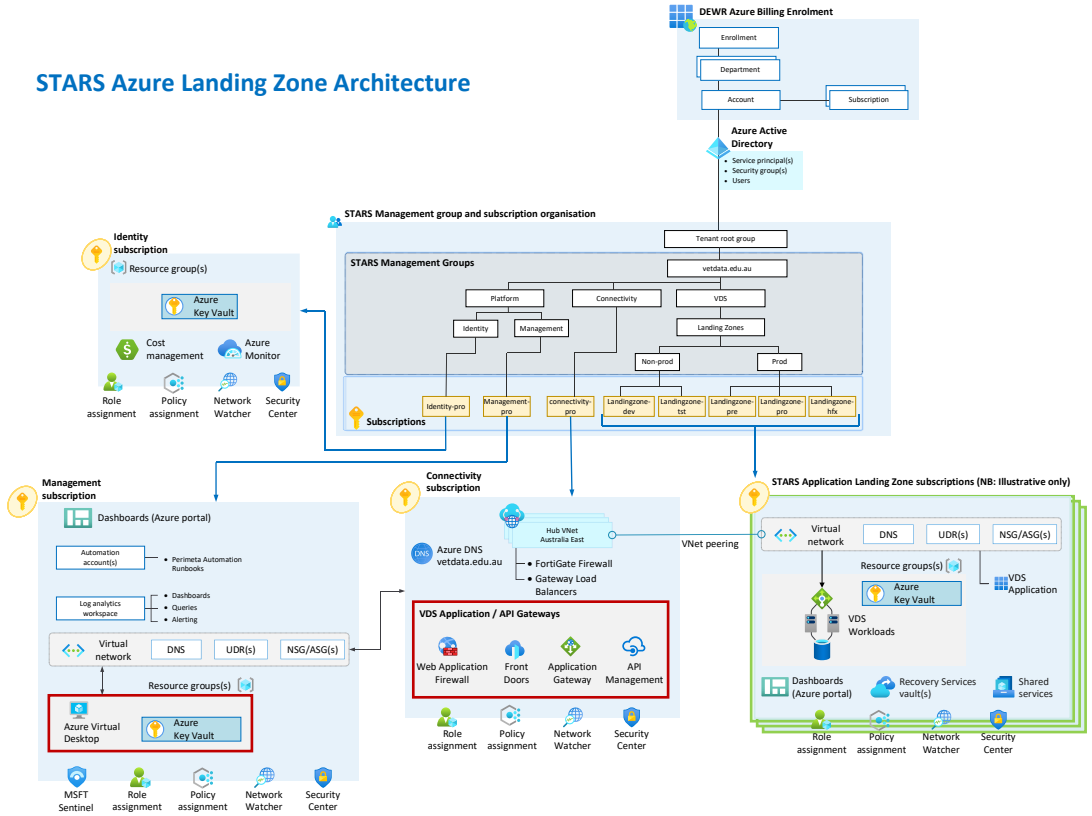


Figure 222: STARS Azure Landing Zone Architecture

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

5.5.1 Management Groups and Subscriptions (DEWR MSP~~Oobe~~)

The STARS Azure Tenant subscriptions are organised into several hierarchical management groups:

Table 444: STARS Management Group Hierarchy

Management Group Hierarchy	Subscriptions	Description
Platform		The Platform MG is the top-level container for most hub Azure infrastructure services.
Identity	Identity-pro	Contains (hub) resource groups with resources to manage IDAM, including AZ B2C. Hosts the vetdata.edu.au ADDC domain controls and Azure AD Connect synchronization servers. Also hosts the central KV for all platform certificates and secrets.
Management	management-pro	Contains (hub) resource groups that manage AZ Automation, Log Analytics, AVD, Security Centre, Policies.
Connectivity	connectivity-pro	Contains (hub) resource groups that manage STARS firewalls (e.g., FortiGate), load balancers, Application Gateway, APIM, WAF, VPN Gateway. This subscription, RGs and networking resources are the main point of connectivity between STARS, RTOs, STAs, NCVER, and other agencies. Hosts the "Cloud Gateway" consisting of 4x FortiGate network virtual appliances (2x AU East, 2x AU Southeast), the gateway management AVD, gateway backup, logging and supporting services.
STARS		The STARS MG is the top-level container for the STARS application RGs and associated resources. With the STARS MG, there are several 'environment' subscriptions (e.g., dev, prod).

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 837 of 1295

Management Group Hierarchy	Subscriptions	Description
		Each environment will be configured similarly, including the use, where relevant, of hub PaaS services (e.g., Service Bus).
STARS Non-Prod		Nonproduction environments.
	landingzone-dev	Development environment RGs
	landingzone-tst	Test environment RGs
STARS Prod		Production environments
	landingzone-pre	Preproduction/staging environment (NB: may be redundant – TBC)
	landingzone-pro	Production environment
	landingzone-hfx	Hotfix environment (NB: may be redundant, TBC. STARS may utilise App Services deployment slots instead.)

5.6 STARS Application Resource Groups

This section describes the expected resource groups that will be configured and deployed to contain and manage the various Azure services that support each of the STARS services. For specific details of each STARS service, please refer to the relevant service design documents and the STARS Solution Design documents.

Note that each resource group will be subject to Azure Policies, including those relating to security. These are out of the scope of this document, having been provided as part of the DEWR MSP~~Oobe~~ implementation.

Resource groups will also follow DEWR MSP~~Oobe~~ provided design patterns and inherited configurations for common monitoring, logging, and network services, e.g., VNets and NSGs.

Resource groups (and resources) will be named and tagged in alignment with the ~~Microsoft recommendations and STARS requirements (e.g., cost management), see: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-best-practices/naming-and-tagging~~ VDS Azure Naming Standards (refer to document C – Azure Naming Standards.xlsx).

~~Note ‘ENV’ will be replaced with one of ‘dev’, ‘tst’, ‘prd’, as applicable to the environment.~~

Table 5~~5~~: STARS Application Spoke Resource Groups

Formatted: Normal

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 838 of 1295

Resource Group Category	STARS Service	Azure PaaS Resources & Notes
~~rg-master-ENV~~Master	Master Data Service	• Function Apps • Logic Apps • Azure SQL • Cosmos DB • Service Bus
~~rg-reference-ENV~~Reference	Reference Data Service	• Function Apps • Logic Apps • Azure SQL • Cosmos DB • Service Bus
~~rg-provider-ENV~~Provider	Provider Service	• Function Apps • Logic Apps • Azure SQL • Cosmos DB • Service Bus
~~rg-submission-ENV~~Submission	Submission Service	• Function Apps • Logic Apps • Azure SQL • Cosmos DB • Service Bus
Staging~~rg-staging-ENV~~	Staging Service	• Function Apps • Logic Apps • Azure SQL • Cosmos DB • Service Bus
Release~~rg-release-ENV~~	Release Service	• Function Apps • Logic Apps • Azure SQL

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 839 of 1295

Resource Group Category	STARS Service	Azure PaaS Resources & Notes
		• Cosmos DB • Service Bus
Masteringrg-mastering-ENV	Mastering Service	• Function Apps • Logic Apps • Azure SQL • Cosmos DB
rg-rap-ENVReporting/Analytics	Reporting and Analytics Provisioning Service	• Azure Data Factory • Service Bus
Rulesrg-rules-EVN	Rules Service	• Function Apps • Logic Apps • Azure SQL • Cosmos DB • App Containers
Notificationsrg-notification-ENV	Notification Service	• Function Apps • Logic Apps • Azure SQL • Cosmos DB • Service Bus
Eventsrg-event-ENV	Event Service	• Function Apps • Logic Apps • Azure SQL • Cosmos DB • Service Bus • Event Grid
NCVER rg-integration-SYS-ENVIntegration	Data Consumption Services ETL Pipeline Services	Azure Integration Services resources, one RG for each downstream/upstream agency (SYS). e.g., rg-integration-ncver-ENV

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 840 of 1295

Resource Group Category	STARS Service	Azure PaaS Resources & Notes
STA Integrationrg-sta-ENV	STA Gateway Service	TBD – STA integration in Stage 3.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 841 of 1295

6 Connectivity

This section describes connectivity within the STARS Azure tenant, and between STARS and external agencies and training providers.

Azure Landing Zones provides a set of best practices for designing and implementing secure and scalable network topologies, including the use of Azure Virtual Networks, Azure Firewalls, and Azure ExpressRoute[2] for connectivity to on-premises networks.

In broad terms, this section of the design covers the following Azure LZ and Azure Well-Architected guidelines:

- Design your network topology to ensure that different workloads are isolated from each other and that only authorized traffic is allowed to flow between them.
- Use Azure Virtual Networks to logically isolate network traffic and provide a secure network perimeter for your workloads.
- Implement network security controls, such as ~~Azure F~~firewalls, to enforce network traffic policies and protect against external threats.
- ~~Establish a dedicated network connection to Azure using Azure ExpressRoute to provide a high-speed, low-latency, and secure connection to the cloud.~~

Specific guidance from the DEWR MSP includes:

- All networking is force tunneled to the central FortiGate firewalls. Any service built as part of ~~t~~the solution will follow a private routing path and navigate through these central FortiGate firewalls for Inbound/Outbound with the internet when possible. This includes~~ means~~:
  - Setting storage accounts up with Private Endpoints / Firewall enabled
  - Setting KeyVaults up with Private Endpoints / Firewall enabled
  - Configuring APIM to be private
  - Configuring ASE's to be private

  ~~Etc...~~
- ~~If possible,~~ FortiGate firewalls ~~are~~will to be used as part of the solution instead of Azure native service, such as Web Applications firewalls.
- If the FortiGates do not provide the functionality that an Azure native service does, and the Azure native service can be made private, it is to be made private and a VIP/NAT configured on the FortiGate to forward traffic to that service.
- ~~I~~~~Similar to the previous point, i~~f an Azure native service cannot be made private and is public only, where possible the FortiGates are to have a VIP/NAT configured as the entry point before traffic is sent to the Azure public endpoint.
- If none of the above is possible and the Azure native service must be the entry point, then full logging and auditing is turned on and that ~~particular service~~service is strictly called out as to why it cannot use the gateway. s 47F(1) ~~, please correct any of the above sections for clarity on networking here.~~
-

[2] Express Route is not currently required or supported in the DEWR MSP configuration and is only mentioned in this document for future consideration. ~~if or when partner agencies~~

Formatted: Space After: 10 pt, Line spacing: 1.5 lines

Formatted: Font: (Default) Segoe UI, 10.5 pt

Formatted

Formatted: Space After: 5 pt

Formatted: Font: (Default) Segoe UI, 10.5 pt

Formatted: Add space between paragraphs of the same style, Bulleted + Level: 1 + Aligned at: 0.63 cm + Indent at: 1.27 cm

Formatted: English (United States)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 842 of 1295

6.1 STARS External Connectivity

6.1.1 STARS to Partner Agency Connectivity

Azure provides several features for both inter-tenant Azure connectivity and connectivity between Azure and on-premises data centres.

As of this version of the document, no requirements have been created for external connectivity aside from NCVER. For NCVER an approach to use VPN connections to NCVER's data centres has been decided for Stage 1. Other agencies have indicated either the current or future use of Azure-Azure connectivity, with some agencies possibly needing short-term (1 year) connectivity through VPN. In any case, requirements and decisions will be gathered and made during Stage 1.

Accordingly, this design document outlines external connectivity approaches instead.

Inter-tenant Azure connectivity approaches:

- Azure Virtual Network (VNet) Peering: allows for secure and high-performance communication between Azure Virtual Networks, which can be in different tenants.

Azure and on-premises data centres connectivity approaches:

- Azure Site-to-Site VPN: allows for secure and encrypted connections between an on-premises network and an Azure virtual network.
- Azure ExpressRoute: provides dedicated and secure connections between Azure and on-premises data centres or colocation facilities.

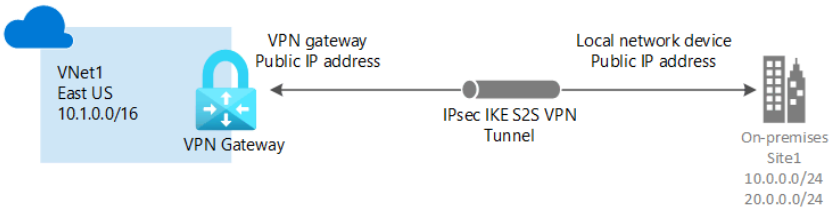


Figure 333: Azure VPN Gateway Example (Site-Site)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 843 of 1295

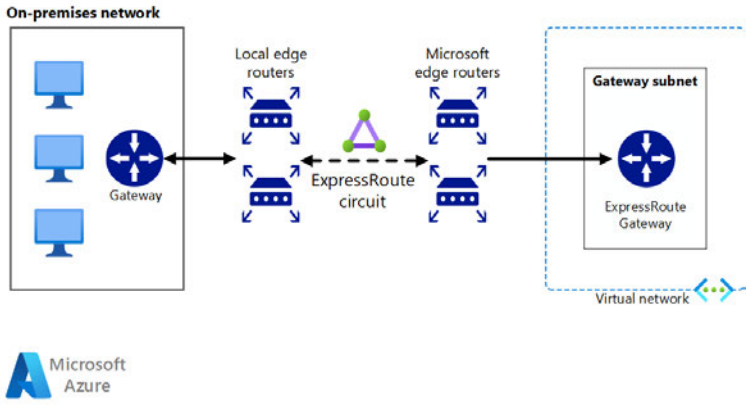


Figure 444: Azure Express Route Example

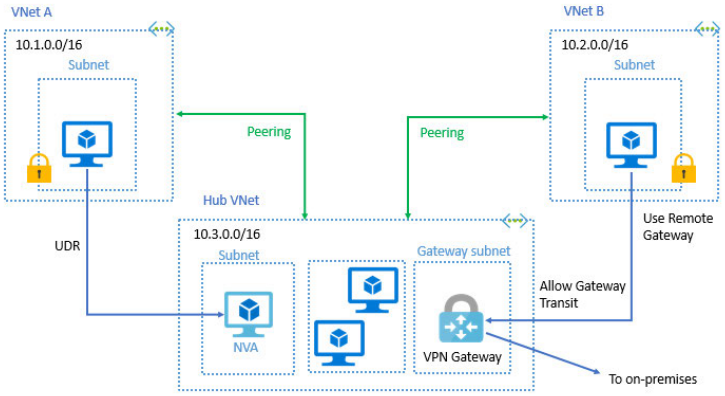


Figure 555: Azure VNet Peering Example

6.1.2 STARS to Training Provider SMS Connectivity

STARS to training provider connectivity is discussed in the context of system-to-system data exchange, i.e., Providers' SMS to STARS API. There are broadly two scenarios:

1. The SMS is located within the provider's own data centre/premises. In this scenario connectivity is over the public internet and will mediated by the services managed within the Connectivity subscription, including App Gateway and FortiGate.
2. The provider uses a cloud-based SMS service (managed by the SMS vendor), possibly hosted within Azure or another cloud provider. In this scenario, connectivity may be achieved through 'private' high-speed cloud networking, e.g., VNet peering, Express Route, etc.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 844 of 1295

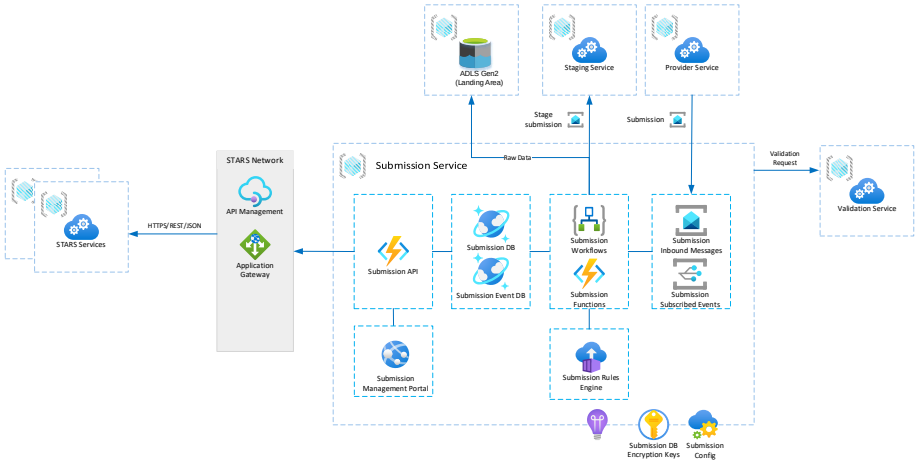
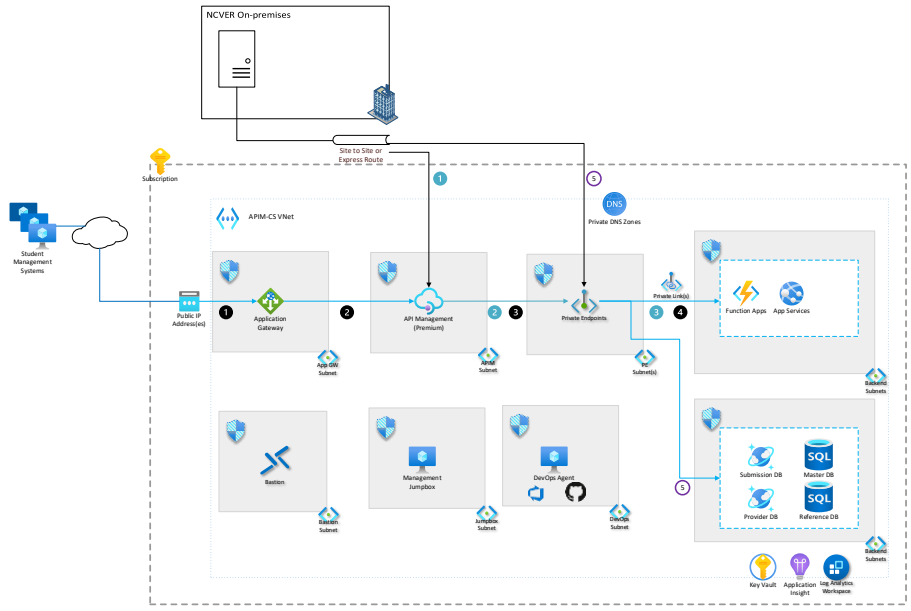


Figure ~~66~~: STARS Internet Connectivity

Figure 66: STARS Internet Connectivity~~Figure 6: 'STARS Internet Connectivity~~' shows the public internet-based connectivity. The STARS Network resources (APIM, AG) are in the Connectivity subscription (managed by the DEWR MSP~~Oobe~~.)

Cloud-based SMS connectivity is in most respects the same as partner agency connectivity, however details will depend on the technology stack of the cloud SMS vendor.



Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 845 of 1295

Figure 777: STARS Public/Private Connectivity

Figure 77: STARS Public/Private Connectivity~~Figure 7: 'STARS Public/Private Connectivity~~' shows the Azure resources involved for both public and private networking. The networking resources will be located in the Connectivity subscription.

Note that partner agencies (e.g., NCVER) will have direct access to certain Azure PaaS services, such as Azure SQL. This will require the configuration of Azure Private Endpoints.

NB: Details of private direct connectivity will be added to this document as, for example, NCVER connectivity is established.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 846 of 1295

7 Requirements

7.1 Non-Functional

NB: Acceptance Criteria have not been supplied by the VDS Program for any NFRs.

NFRs and traceability in the context of this networking and infrastructure document is restricted to general features of Azure and the configurations discussed in this document. NFRs as the apply to other STARS components and their underlying Azure services are discussed in their respective design documents.

It is very important to note that all the infrastructure related features of the solution are implemented through Infrastructure-as-Code (IaC) principles, and that the relevant NFRs, in this context, are primarily met through appropriate configurations of IaC templates – namely Azure Resource Manager (ARM) Templates. In other words, the NFRs are delivered through code artefacts.

To give an example, NFR001a scalability, will be delivered by setting the relevant set of SKUs for a PaaS service plans within one or more ARM templates. To trace this requirement, a link between the ARM template code artefact, the deployment pipeline artefact within the STARS source code repositories and the relevant Azure DevOps NFR work items and PBIs will be made, as the solution is implemented and delivered.

Noting that detailed item-level tracing will be made as templates and pipelines are implemented, the matrix below discusses how the NFRs will be met at a high level through the configuration guidelines established above.

NFR No	Requirement Name	Description	Traceability
NFR001a	Use Case scalability	The VDS System will be able to scale elastically in response to demand and remain functional and available to users regardless of dynamic changes in volume, processes, and number of users leveraging the VDS System. • The VDS System will be designed to achieve scalability of	Elastic scaling is a feature of all PaaS services selected in the solution, with levels determined by choice of, for example, compute plan. These are all in turn, just parameters in an ARM template.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

NFR No	Requirement Name	Description	Traceability
		additional use cases as needed • The VDS System will support both horizontal and vertical scaling	Please see discussion in section 5.4 for configurations chosen. In general, SKUs and scale-out/up ARM parameters will be adjusted to meet the numeric parameters given below.
NFR001b	Concurrent users scalability	• The VDS System will be designed, developed, tested and deployed to accommodate increased concurrent users, volumes and workloads specifically: • Real time scalability to accommodate increases in the number of concurrent users. • Consistent response times (e.g. user interface, data submission, application workflow, and data consumption) with increased volumes and concurrent users.	Please see above.
NFR001c	Baseline user volumes	The VDS System supports the following baseline volumes: • Over **2,000** concurrent users across training organisations and STAs • **5,000** total users across training organisations and STAs	Please see above. STARS will elastically scale as user loads change.
NFR001d	Baseline transaction volumes	The VDS System supports the following baseline volumes: • Over **xxx** transactions for the submission of data by training organisations per week • Over **xxx** data submissions per second	Please see above. These values will be confirmed during the project.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

NFR No	Requirement Name	Description	Traceability
NFR002a	Minimum Uptime	The VDS System will support continual operations by being available and functioning with minimal downtime (High Availability). • The VDS System will support a minimum proposed uptime for the users, excluding scheduled downtime or outage	Please see above. Section 8 discusses HA configurations for various Azure services, subject to published Microsoft SLAs.
NFR002d	Production environment uptime	The VDS System, including all user interfaces, in the Production environment is available **99.86% of the scheduled hours of availability**, measured over a month. The system is scheduled to be available for all hours, except those in approved change windows (i.e. planned outages). This equates to **60 minutes** of unavailability in a 30 day month with no planned outages.	Please see above.
NFR002e	Non-Production environments uptime	The VDS System, including all user interfaces, in environments other than Production (per NFR11), is available **99.79% of the scheduled hours of availability**, measured over a month. The system is scheduled to be available for all hours, except those in approved change windows (i.e. planned outages). This equates to **90 minute**s of unavailability in a 30 day month with no planned outages.	Please see above.
NFR002f	Outage window for major releases	The VDS System will use a single planned outage window of **five (5) hours allowed in a three (3) month period** for major releases. The planned outage will allow for a full system outage.	The general design principles for STARS include SOA, with each service able to operate and be upgraded independently. The current solution

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

NFR No	Requirement Name	Description	Traceability
			design does not require planned complete system outages for upgrades, etc.
NFR002g	Outage window for minor releases	The VDS System will use a single planned outage window of **one (1) hour allowed in a one (1) month period** for minor releases. The planned outage will allow for a full system outage.	See above.
NFR003c	No performance impact on state-managed systems	The VDS System integrates with state-managed systems in such a way that the traffic does not negatively impact the performance or availability of these systems.	STARS networking connectivity is independent of STA systems.
NFR004a	WoG Hosting Strategy	The VDS System will be hosted in accordance with the Whole of Government Hosting Strategy. • The VDS System will meet whole of government sovereignty requirements as specified in the Whole of Government Hosting Strategy.	Microsoft Azure has been certified to meet the requirements of WGHS.
NFR004b	Minimise Supply chain risks	The VDS System will minimise supply chain risks in regards to the hosting arrangements, in accordance with the Whole of Government Hosting Strategy.	Microsoft Azure has been certified to meet the requirements of WGHS.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

NFR No	Requirement Name	Description	Traceability
NFR016a	Recovery Mechanisms	The VDS System will support disaster recovery for all hosting services and systems. Three major recovery mechanisms must include Geo-Replication, Geo-Recovery, and Geo-Restore, to warrant that fail over of primary services and/or databases to a secondary counterpart in the event of a datacentre or regions downtime: • During a failover process, primary services and databases must fail over to secondary services and databases in different regions. • Geo-Restore must be enabled to recover a database from automatically maintained geo-redundant backups. • Repatriation and Geo-Replication must be used to repatriate updated databases to the primary database in their original regions.	Please see discussion in section 5.4. HA configurations are discussed, these will be implemented through relevant ARM template configurations and deployments of subscription, resource groups and associated PaaS services.
NFR016b	Defined tier levels for disaster recovery	The VDS System will support off-site recoverability based on a defined tier level for each of the VDS System's core functions, e.g. to support transaction integrity and data loss within the target RPO (per NFR17).	Off-site recoverability is achieved through Azure regional separation.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

8 Appendices

Appendix 1. Resources

1. Microsoft Cloud Adoption Framework (CAF): https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/
2. Azure Landing Zones: https://docs.microsoft.com/en-us/azure/landing-zones/
3. Azure Well-Architected Framework: https://docs.microsoft.com/en-us/azure/architecture/framework/
4. Azure ExpressRoute: https://docs.microsoft.com/en-us/azure/expressroute/
5. Azure VPN: https://docs.microsoft.com/en-us/azure/vpn-gateway/
6. Azure Site-to-Site VPN: https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-howto-site-to-site-resource-manager-portal
7. Azure Monitor: https://docs.microsoft.com/en-us/azure/azure-monitor/
8. Azure Network Watcher: https://docs.microsoft.com/en-us/azure/network-watcher/

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 852 of 1295

Appendix 2. Acronyms

Acronym/ Abbreviation	Definition
API	Application Programming Interfaces
AVETMISS	Australian Vocational Education and Training Management Information Statistical Standard
BABOK	Business Analysis Body of Knowledge V3.0
BPMN	Business Process Modelling Notation 2.0
BR	Business Requirements
DEWR	Department of Employment and Workplace Relations
DPT	DEWR Program Team
DTO	Data Transfer Objects
DXC	STARS Solution Provider and all round rockstars
FFS	Fee For Service
FR	Functional Requirement
IAM	Identity and Access Management
IIBA	International Institute of Business Analysis
Error! Reference source not found.~~IT~~	Information Technology
Error! Reference source not found.~~ITIL~~	Information Technology Infrastructure Library
LoB	Line-of-Business
MDM	Master Data Management
NCVER	National Centre for Vocational and Education Research
NFR	Non-Functional Requirement
Error! Reference source not found.~~OCM~~	Organisational Change Management
OCM	Organisational Change Management
ODIA	Offer Definition Improvement Activity
RTO	Registered Training Organisation

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 853 of 1295

Acronym/ Abbreviation	Definition
SMS	Student Management System
SMS	Student Management System
SOA	Service Oriented Architecture
STA	State Training Authority
STARS	Student Training & Activity Reporting System
TGA	Training.gov.au
TO	Training Organisation
TVA	Total VET Activity
USI	Unique Student Identifier
UX	User Experience
VDS	VET Data Streamlining
VET	Vocational Education & Training
VET-IS	VET Information Standard
WoG	Whole of Government
RG	Resource Group
LZ	Landing Zone
MG	Management Group

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 854 of 1295

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 855 of 1295

Australian Government
Department of Employment and Workplace Relations

Detailed Design Document

Rules Service

1 Document Management

1.1 Document Control

Version	Change Description	Date
0.1	Initial Draft	18/4/2023
0.2	Frist Draft after initial socialisation and walk-through	26/4/2023
1.0		

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This is a living document and will be updated as required.

1.2 Document Author

Role	Name
Senior Technical Consultant	s 47F(1)

1.3 Document Reviewers

Role	Name

1.4 Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

1.5 Related Documents

Reference	Document
A	STARS Solution Requirements Document – v1.3.docx
B	STARS Solution Design – v1.0.docx
C	Rules Service Diagrams – D.0.2.vsdx
D	STARS Detailed Design Document – Audit and Logging Services – v.1.0.docx
E	Detailed Design Document - IDAM D.0.5.docx
F	Detailed Design Document - Validation Service v1.0.docx
G	Detailed Design Document - Notification Service V.1.0.docx
H	Detailed Design Document - Reference Data Service D.0.3.docx

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 857 of 1295

Reference	Document
I	Detailed Design Document - Provider Service D.0.4.docx
J	Detailed Design Document - Submission Service - D.0.3.docx
K	STARS Validation Service API v1.0.pdf

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 858 of 1295

Contents

1 Document Management 2
1.1 Document Control 2
1.2 Document Author 2
1.3 Document Reviewers 2
1.4 Document Approval 2
1.5 Related Documents 2
Contents 4
2 Open Design Items 6
3 Introduction 7
3.1 Purpose 7
3.2 Intended Audiences 7
4 Scope and Exclusions 8
4.1 In Scope 8
4.2 Out of Scope 8
5 Design Decisions 9
6 Feature Overview 10
6.1 Rules Engine Overview 10
6.1.1 Rules Engine Terminology and Usage 10
6.1.2 Managing the evolution of STARS validation rules 10
6.1.3 Auditing & traceability 11
6.1.4 Rules Configuration 11
6.1.5 Errors and Notifications 11
6.2 Rules Service API 12
6.3 Rules Engine Instances 12
6.4 Funding Arrangements Database 13
6.5 Reference Data Cache 13
6.6 Exemptions 13
6.7 Rules Management Portal 14
6.8 Rules Service Key Vault 14
7 Business Processes 15
7.1 Rules Management 15
8 API Design 16
8.1 Validations 16

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 859 of 1295

8.1.1 Validate Submission 16
8.2 Submissions 16
8.2.1 Determine Funding Arrangements 16
8.3 Notifications 17
8.3.1 Generate Reminders 17
8.4 Release 19
8.4.1 Determine Data Enrichment Requirements 19
8.4.2 Determine Release Requirements 20
8.5 Rule Data 21
8.5.1 Get rule data 21
8.5.2 Get category data 22
8.5.3 Get instance data 23
9 Technical Design 24
9.1 API App 24
9.2 KIE Execution Servers 24
10 Access Management 26
11 Security 27
12 Dependency on Other Services 28
12.1 Submission Service 28
12.2 Provider Service 28
12.3 Reference Data Service 28
12.4 Release Service 28
13 Functional Requirements 29
13.1 Fulfilled Requirements 29
13.2 Informing Requirements 33
14 Non-Functional Requirements 37
15 Acronyms 38

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 860 of 1295

2 Open Design Items

This section describes items arising from the design process that remain unresolved pending further work that will take place later during the project.

ID	Open Item	Owner	Due Date	Work Item Ref.
1	Determine the set of rules representations that will be supported (DRL, DSL, DMN, etc)	s 47F(1)		3591
2	Need to know the purpose of the requirement that rules owners can export the rules they own. There are several distinct ways to meet this	s 47F(1)		3591
3	There are some gaps between the rules management process and the capabilities of the Workbench, we need to determine where the process should move towards the Workbench and where the Workbench should move towards the business process	s 47F(1)		3591
4	Where is the review process managed from? Exemption rules can be managed through the Rules Management Portal, does it also need to be handle the full review workflow?	s 47F(1)		3591
5	Which roles will have access to view rules (National and STA) in the Rule Management Portal?	s 47F(1)		3591
6	What are the rules management process requirements for individual STAs?	s 47F(1)		3591

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

3 Introduction

This document covers the design for the STARS solution Rules Service.

3.1 Purpose

The purpose of this document is to capture the detailed design requirements for rules management and execution for various stake holders, systems and services in STARS.

This will provide guidance to developers about the structure and process of interfacing with the rules engines, as well as the administration of business rules within the service by STARS operators.

The design is informed by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework, which provide a comprehensive set of best practices and guidance for designing, deploying, and operating secure, high-performing, and resilient cloud infrastructure and applications.

3.2 Intended Audiences

This detailed design document provides technical information and guidance to help technical stakeholders understand the rules management and execution services used in the STARS solution and how they integrate with the solution components. The document is expected to be used by technical stakeholders who have a good understanding of cloud computing concepts and are experienced in the Azure platform.

This document is not intended for non-technical readers, nor for readers who are not experienced in the design and configuration of Azure-based solutions; informational and educational resources regarding Azure are readily available at https://azure.microsoft.com

The primary audience for this document includes:

- Program technical staff: This includes architects, software engineers, administrators, and operations teams within the program responsible for the STARS solution.
- External technical stakeholders: This includes technical staff from other federal, state, and territory governments agencies, including VDS embedded team members.
- ICT Technical Staff: This includes a variety of technical roles involved during the VDS program and post-go live, including network engineers, systems administrators, database administrators, security engineers, and DevOps engineers.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 862 of 1295

4 Scope and Exclusions

4.1 In Scope

The following are in-scope of this detailed design solution:

1. Design of a rules management service
2. Design of a rules execution engine
3. Design of a Rules Service API

4.2 Out of Scope

The following are out of scope for this design document:

4. The analysis and coding of actual STARS rules, aside from examples. The analysis and design of the rules themselves is a build-activity, covered during the Build stage.
5. The UI design of the Rules Management Portal, this will be covered in other UI/UX documents. NB: The Rules Management Portal will be based on the Drools Workbench (also known as Business Central, unrelated to the identically name Microsoft product), adapted to suit STARS requirements and VDS branding.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 863 of 1295

5 Design Decisions

The STARS Rules Engine will utilise a widely used open-source Rules Engine, Drools.

Please note that DXC is not developing the rules algorithms and engine as such but are using the open-source Drools rules-engine tools to build a complete rules-processing framework for submission validation, business rules execution etc.

Rules will be encoded in the Drools Rules Language (DRL), enhanced by domain specific language (DSL) definition to aid STARS operators. The rules will be written in DRL (Drools Rule Language), a custom programming language designed to express business rules. DRL also allows the use of standard Java code within rule definitions. As such, anyone tasked with designing and maintaining rules will need to be comfortable with the basics of writing in code/scripting. In addition, the rules projects are stored in a local Git repository. Although understanding this backing store is not essential to use the Business Central Workbench, a good grasp of version control would provide richer capabilities for managing change and versioning.

The STARS Rules Engine will be configuration driven, allowing flexibility for enabling/disabling rules, setting validity periods, and even determining which data elements or jurisdiction-specific information the rules apply to. Rules themselves will be managed as 'Rule Sets', with different rule sets applied for different requirements, such as STA-specific validation rules.

The Rules Engine itself will be running on Docker containers managed by Azure Container Apps. Each container will be a KIE Execution Server (Drools is part of a larger project known as KIE), which is essentially the Drools engine hosted on a Wildfly webserver, providing HTTP POST access to the standard Rules Engine commands.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 864 of 1295

6 Feature Overview

6.1 Rules Engine Overview

6.1.1 Rules Engine Terminology and Usage

The Drools Rules Engine utilises optimised rule processing algorithms based on the Rete Algorithm[1] , developed and maintained by experts in the field. The Rete algorithm can efficiently process many rules against large volumes of data, but to do so can require a lot of computer memory – at least the size of all the data being evaluated plus any lookup/reference data required and any results.

From a usage perspective, the Rules Engine maintains and executes rules, assembled as rule sets. Each rule is made up of two sections, a 'when' clause, and a 'then' clause. When a condition is matched, the rule specifies an action to be performed.

The Rules Engine maintains an area of memory known as 'working memory' to hold information while processing rules.

Data to be evaluated against the rules, known as 'facts' is loaded into the working memory. Any supporting information, such as reference lookup values (e.g., from the Reference Data Service), is also loaded into working memory in the global space – this ensures that the Rules Engine can operate at high speed without having to pause to access, e.g., external databases. Typically, items used for lookups are cached in memory by the Rules Engine upon start up.

When ready, the Rules Engine 'executes' the rule sets against the facts (data) within the working memory. The combination of working memory, facts, rule sets and the execution of rules and actions is known as a 'session'.

During the session, rules that match against facts in the working memory perform their associated 'then' actions. Even though it is possible for a rule action to perform some external task (e.g., update a record in a database), this is not recommended. Instead, it is better for rule actions to modify existing facts or 'assert' new facts that represent some outcome of matching existing facts back into working memory. For example, an invalid submission is recorded by either modifying the submission to flag it as invalid, or creating a new 'invalid submission' fact, information which is then available within the working memory.

When all rules and actions have completed, the rule engine can be queried to provide details about the facts, including new and modified facts, in the working memory. In the example above, the Rules Engine could provide all submissions with the invalid flag set.

6.1.2 Managing the evolution of STARS validation rules

Drools can use rules written in either in DRL code, or a Domain Specific Language (DSL)[2] that is used to create pseudo-English scripts that define the rules. These scripts (which would need to be created and maintained) are then compiled down to DRL. Both can be used in combination, with DRL rules

[1] See https://en.wikipedia.org/wiki/Rete_algorithm

[2] See https://docs.drools.org/7.73.0.Final/drools-docs/html_single/#_domain_specific_languages for details of the Drools rules DSL.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 865 of 1295

providing more flexibility and power with the trade-off of more technical training requirements, while DSL rules require less technical capability but cannot express some more complex fact manipulation capabilities of Drools. Drools also supports other techniques for representing rules (such as DMN diagrams), however whether these will be used within the system is yet to be determined.

DRL/DSL scripts for new or evolving rules can easily be created or adapted. Even though there might be many rules, the rules themselves are usually expressible in a small piece of code – more time is usually spent analysing and specifying the rules.

The rules code itself – and by implication the rules logic – will be managed by a source code management tool (Git, optionally replicated by Azure Repositories). The code and any changes and subsequent releases are all tracked and linked to requirements (i.e., User Stories/Product Backlog Items), allowing for full traceability and history of changes to rules, including approvals.

DRL/DSL-based rules can be rapidly developed, and automatically tested and deployed to UAT, staging and production environments – completely under the control of STARS Product Owners/SMEs/Administrators.

Note that rules will be applicable to certain scopes. This may include 1 or more versions of the VET-IS, or a particular STA, or a particular data element/entity. As such, they have the flexibility to manage the evolution of changes to the VET-IS or changing business requirements.

Some rules that require large-scale data aggregation or complex interactions with external services may be implemented as Azure Function Apps which will in turn be called by DRL/DSL rules. Such rules will require developer implementation and a more complex deployment process, dependent on the functional detail.

6.1.3 Auditing & traceability

The Rules Engine provides mechanisms to log and audit execution of rules. This will enable recording of which rule failed and why. STARS will then log these events for diagnostics purposes as well as to provide detailed reasons for validation failure, e.g., for reporting back to Training Organisations.

6.1.4 Rules Configuration

Rules are dynamically loaded at runtime by the Rules Engine. The dynamic loading feature, along with dynamic rules parameters (e.g., applicable data elements, allowed values, dates applicable) allows for the Rules Engine to be configured via settings and parameters stored in a database or through the Rules Management UI.

Each of the rules will have a Description associated with them, that will provide a business-readable representation of the rules. Rule owners will have the ability to view rules that they own, as well as being able to retrieve copies for external use.

6.1.5 Errors and Notifications

The Rules Engine has several facilities to monitor execution and troubleshoot issues with any rules. Typically, information surfaced by the monitoring functions are exposed through a logging framework.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 866 of 1295

The Rules Engine libraries will integrate with any of the Azure logging frameworks through patterns specified in the Audit and Logging Detailed Design document (D), allowing for detailed tracing through, say, Azure Monitor.

6.2 Rules Service API

The Rules Service API will be responsible for providing API access to the Rules Service for notifications, validations, and submissions. The details of these interactions can be found in the API Design section.

6.3 Rules Engine Instances

The Rules Service will need to provide many different Rules Engine instances to meet the different requirements of the service. Rules Engine instances can be broadly divided into two categories: types and variants. The type is dependent on the purpose of the instance (validating national rules, calculating notifications, etc). A variant is a specific version of the rules of a specific type that needs to remain available for future use. For example, when a new version of the national rules is implemented within STARS, it may still be necessary to validate some data against the previous version. Rather than modify all the existing rules to take into account the existence of the new version, a variant of the same type can be created then the desired variant can be selected on a per case basis (typically by the selector engine instance described below).

Table 11: Rules Engine Instances

Engine Type	Purpose	Variants
Validation – National	Validates submissions against the national rules	As rules change over time, any version which may need to be used going forwards will result in the creation of a new variant.
Validation – STA	Validates submissions against the rules of a specific STA	As rules change over time, any version which may need to be used going forwards will result in the creation of a new variant.
Validation – Selector	Determines which variant of the national/STA rules to perform validation with	None. Any changes to the selector should be to improve/adjust its capacity to select variants of the other validation engine instances.
Validation – Exemptions	Processes any failed validations to see if they are covered by an exemption	STA/National. The exemption process is tied to the validation process that was undertaken. If necessary, new variants will be created where exemptions should not be applied to historical data.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 867 of 1295

Notification	Used daily to determine which reminders the Notification Service will need to send	None
Data Enrichment - STA	Determine whether valid records require data enrichment by an STA	When or if rules determining data enrichment requirements change, new variants will be created if those changes should not be retroactive
Data Enrichment - Selector	Determines which variant of the data enrichment rules to use	None. Any changes to the selector should be to improve/adjust its capacity to select variants of the other data enrichment engine instances.
Release - STA	Determine whether mastered records require explicit release by an STA before they become reportable	When or if rules determining data release change, new variants will be created if those changes should not be retroactive
Release - Selector	Determines which variant of the release rules to use	None. Any changes to the selector should be to improve/adjust its capacity to select variants of the other release engine instances.
Validation – Reference Data	Reference Data ingested into STARS is to be checked for its validity, to ensure STARS reference data is of high quality.	None. Reference data should always be up to date with the most current rules.

6.4 Funding Arrangements Database

TBD: the structure and role of the funding arrangements database is dependent on the outcomes of the design process as outlined in 2 Open Design Items

6.5 Reference Data Cache

TBD: the Rules Engine will only need to access a subset of the reference data. The exact structure of the cache is going to be dependent on the rules, which have not yet been defined.

6.6 Exemptions

When a submission fails validation, the submitter may request a review. As part of that review process, an exemption may be added to the rules system to allow the submission to pass validation. There are two components to the exemption: a record of reason for the exemption (who, what, why, etc), and the actual rule(s) that define the exemption.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 868 of 1295

Exemption rules will be kept in a separate rules project and executed in a separate Rules Engine, and only submissions which fail validation will be routed through the exemption engine.

6.7 Rules Management Portal

The Rules Management Portal is operator-facing and allows operators to manage the full lifecycle of rules creation and management, as well as providing traceability on the development of the rules. This will be implemented using the Drools Workbench, visually customised to display the appropriate STARS branding, with integrated Azure authentication.

6.8 Rules Service Key Vault

The Submission data will be encrypted at rest, encryption keys to be held in Azure key vault.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 869 of 1295

7 Business Processes

The primary role of the Rules Service is to provide implementations of business rule logic for other services, which are in turn responsible for implementing the business processes.

7.1 Rules Management

It is a requirement that STARS operators can manage the creation and modification of rules used to validate submissions. The Business Central Workbench will be used to meet this requirement, allowing creation and testing of rulesets across multiple jurisdictions, backed by Git version control to provide powerful traceability and auditing capabilities.

The approval process for creating and modifying rules largely sits outside of the Rules Service, relying on communication between the STARS Operators and various NCVER Rule Owners. Capacity will be provided to allow any rule change to be extensively tested before being deployed into the production system.

TBD: those STAs with some, or all of their validation process hosted within STARS will need to have the process by which these rules are managed clearly defined, in terms of who has responsibility for rule creation, testing, etc. as well as the visibility of their rules to other jurisdictions. This question is outlined in 2 Open Design Items.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 870 of 1295

8 API Design

Unless explicitly noted otherwise, all request and response body data will be in the JSON format.

8.1 Validations

The validation service will make requests to the rules service to validate submissions against one or more rulesets. The rules service will return a validation result for each individual record in the request.

8.1.1 Validate Submission

POST rules/validations/validate

8.1.1.1 Request

Table 22: Validations Request details

Validation Request	
validationRequestId	Unique identifier of this request. Will be included in the response
submissionId/datasetId	Unique identifier of the submission or dataset (subset of a submission) to validate, used to retrieve full data from the source Service (e.g., Submission Service or Provider Service)
stage	Integer value used for tracking by the Validation Service, unused by Rules Service

8.1.1.2 Response

The details of the response for validation requests are defined in related document F.Related Documents

8.2 Submissions

The submissions service will make requests to the rules service API to determine the funding arrangements and validation rulesets associated with a submission record.

8.2.1 Determine Funding Arrangements

POST rules/submissions/fundingarrangements

8.2.1.1 Request

The request will consist of a complete Submission object, as defined in related document J (Detailed Design Document - Submission Service - D.0.3.docx).

8.2.1.2 Response

Table 33: Funding Arrangements Response details

Funding Arrangement	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 871 of 1295

submissionId	Unique identifier for the submission, will be the same as passed to the endpoint by the request
recordFundingArrangements	List of recordFundingArrangement objects, one for each record in the original submission
recordFundingArrangement.recordId	Unique identifier for the record
TBD: the rest of the format of this object is dependent on analysis of the current funding arrangement rules, as noted in 2 Open Design Items	

8.3 Notifications

The notification service will make daily requests to the rules service to determine which Training Providers require reminder emails to be sent. The requests will include all Training Providers along with relevant previous reminders (within the previous submission period), as well as the current submission data.

8.3.1 Generate Reminders

POST rules/notifications/calculate

8.3.1.1 Notification – Submission packets were rejected

Table 44: Submission packets were rejected Notification

Service	Data Required
Submission	IDs for all Submissions derived from the current Submission
Validation	ValidationResults where the related Submission is in the set above

A submission with partial failures is still completed, but when a resubmission of the failed packets is done a new submission is created (with a previousSubmissionId set to the original submissionId). This means that in order to find out if there are outstanding rejected packets, to determine whether a reminder needs to be sent, the rules service will have to:

1. Retrieve the necessary data described in the table above
2. Determine whether the most recent validation for each record is a success (it may be sufficient to determine whether there are any successes for each specific record)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 872 of 1295

3. If there are any failed records, and the time since the most recent submission in the retrieved list exceeds the defined number of days, a reminder is needed

8.3.1.2 Notification – Data is overdue for submission

Training providers are required to submit data at on a regular basis (defined per Training Provider). If the time between the current date and the most recent submission is greater than the defined time period for the Training Provider, the Training Provider is deemed overdue.

8.3.1.3 Notification – Data soon due for submission

Same basis as the of the overdue notification, except the notification will be sent N days before the time period is reached. For example, if a provider is expected to submit data at least once every 60 days, they may get a reminder on day 45 that they need to submit data (or make a submission to the effect that they have had no activity).

8.3.1.4 Notification – Data uploaded but not submitted

Table 55: Data uploaded but not submitted Notification

Service	Data Required
Provider	Records for the Training Provider that aren't associated with a Submission

The reminder will be sent after the records have been uploaded for a preconfigured amount of time.

8.3.1.5 Notification – Submission has expired

Table 66: Submission has expired Notification

Service	Data Required
Event Service	All events that begin external validation and data enrichment where there isn't a matching completion event with the same correlation ID

A reminder is required for any unresolved event where the time since the beginning event was raised exceeds a predefined period (defined in the application configuration).

8.3.1.6 Request

The request made to the Rules Service will include sufficient information for each Training Provider to provide feedback for all reminder types in a single request.

Table 77: Reminder request details

Reminder Data	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 873 of 1295

runDate	Date for which reminders are being created (typically the current date)
trainingProviders	List of training provider objects
trainingProvider.trainingProviderId	The due date of the period for which reminders are being generated (necessary to determine whether a submission is overdue, etc)
trainingProvider.mostRecentSubmissionId	A list of trainingProvider objects, which contain information about existing reminders and submissions which can be used to determine any new reminders

8.3.1.7 Response

Table 88: Reminder response details

trainingProviders	List of trainingProvider objects
trainingProvider.trainingProviderId	Training Provider unique provider
trainingProvider.notifications	List of notification objects detailing the reminders required for the training provider
notification.ruleId	Name of the specific rule that caused the notification to be required
notification.notificationType	Type of the notification
notification.message	Message to be shown to the user

8.4 Release

The Release Service will make requests to the Rules Service to determine when submissions require data enrichment from STAs, and when submissions must be explicitly released by an STA.

8.4.1 Determine Data Enrichment Requirements

POST rules/release/dataenrichment/{STA}

8.4.1.1 Request

The request made to the Rules Service will contain sufficient information to determine whether the submission requires data enrichment. It will be up to the release service to determine which STAs need to be checked.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 874 of 1295

Table 99: Data Enrichment request details

Record Data	
recordId	The unique identifier of the submission record being checked

8.4.1.2 Response

Table 1010: Data Enrichment response details

Response Data	
recordId	The unique identifier of the submission record being checked. Will match the value passed in by the request
isEnrichmentRequired	Boolean flag indicating whether enrichment is required
requiredEnrichments	list of Enrichment objects describing the enrichment that is to be performed
enrichment.ruleId	Unique identifier of the rule which triggered this enrichment
enrichment.message	Message displayed to the user describing the nature of the required data enrichment

8.4.2 Determine Release Requirements

POST rules/release/release/{STA}

8.4.2.1 Request

The request made to the Rules Service will contain sufficient information to determine whether the submission requires explicit release. It will be up to the release service to determine which STAs need to be checked.

Table 1111: Release request details

Record Data	
recordId	The unique identifier of the submission record being checked

8.4.2.2 Response

Table 1212: Release response details

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 875 of 1295

Response Data	
recordId	The unique identifier of the submission record being checked. Will match the value passed in by the request
isReleaseRequired	Boolean flag indicating whether explicit release is required (true) or not (false)

8.5 Rule Data

For analytics and rules management, as well as the Reference Data revalidation process, the Rules Service provides an API to query the structure and metadata of the rules and rules engine instances. Typical usage would be to use the category data and/or instance data endpoints to narrow down which rules need to be retrieved, then use the rule data endpoint to retrieve the relevant specific rules. Returned data will be restricted to the categories that the querying user has at least read access to.

8.5.1 Get rule data

GET rules/rules?querystring

8.5.1.1 Request

Table 1313: Rule Data request

Parameter Name	Mandatory?	Description
category	Y	Top level category to retrieve rule data for (e.g., National, specific STA, Notifications)
instance	N	The instance of the category to be retrieved. Category must also be specified. If instance is not specified, all currently active instances will be retrieved (not recommended)
ruleid	N	Only retrieve information for a single rule. Category and instance must also be specified
rulefile	N	This will typically align with a category, but may be required with larger categories

8.5.1.2 Response

Table 1414: Rule Data response

Response Data	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 876 of 1295

rules	List of Rule objects
Rule.ruleId	Identifier of the rule which contains reference data metadata. This will be unique across category + instance combination.
Rule.ruleFile	File in which the rule is defined
Rule.category	Category in which the rule is defined. This is equivalent to a Project within the Business Central Workbench
Rule.instance	Even when the instance parameter is not defined, this value will indicate which version of the rule definition is being returned (i.e. most recent)
Rule.metadata	List of metadata objects
Rule.Metadata.name	name of the metadata tag as defined in the @ part of the rule definition
Rule.Metadata.values	List of strings assigned to the metadata value in the rule definition

8.5.2 Get category data

GET rules/categories

Retrieves a list of all categories in the Rules Service. A category is equivalent to a Project within the Business Central Workbench.

8.5.2.1 Request

No parameters.

8.5.2.2 Response

Table 1515: Category Data response

Response Data	
categories	List of Category objects
Category.name	Name of the category
Category.activeInstances	List of strings of active instances for this category. Value can be used to retrieve instance data

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 877 of 1295

8.5.3 Get instance data

GET rules/instances/?queryString

Retrieves a list of currently active rules engine instances for the specified category. A rules engine instance is considered active if it has a Deployment Unit defined in the Business Central Workbench

8.5.3.1 Request

Table 1616: Instance Data request

Parameter Name	Mandatory?	Description
category	Y	Name of the category to retrieve instance information for

8.5.3.2 Response

Table 1717: Instance Data response

Response Data	
instances	List of instance objects
Instance.name	Name of the instance
Instance.ruleFiles	List of string values defining the DRL files that make up this instance

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 878 of 1295

9 Technical Design

The architecture of the Rules Service can be seen in the diagram below. While Rule Resolution Functions are currently indicated to be Azure Functions, this may change as specific rules are analysed in a future release.

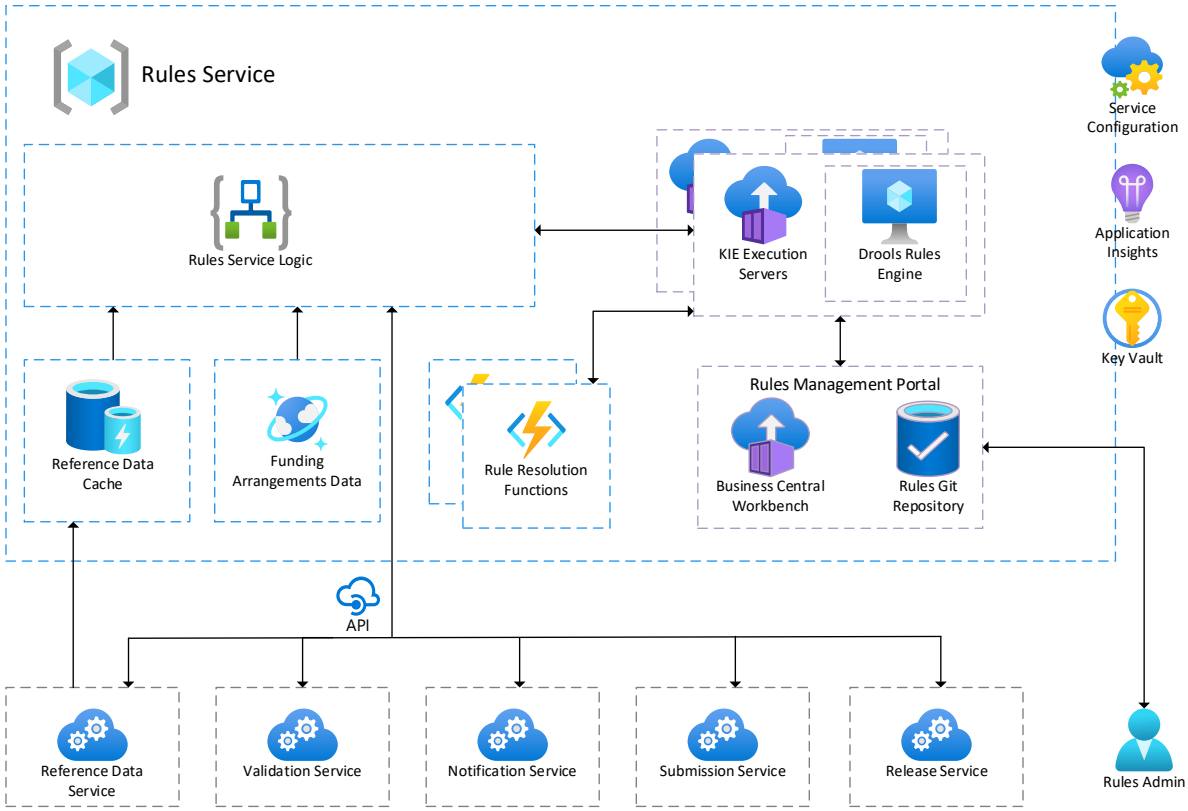


Figure 11: Rules Service Architecture (C1)

9.1 API App

The Rules Service API will be implemented in Azure API Management, with the backend implemented as one or more Azure API Apps. API Apps will support the implementation of OData APIs to provide consistent API interactions between STARS Services and for consumption by external services.

9.2 KIE Execution Servers

Azure Container Apps provide a streamlined approach to containerised applications, taking care of clustering, virtual network setup, and scaling automatically. The KIE Execution Server is provided as a Docker image – which can be customised as necessary to meet the STARS business requirements – that will in turn be deployed using Azure Container Apps.

The rules service will provide many different Rules Engines, both in terms of different categories (validation, notifications, submissions, funding arrangements) and variants (validation with a specific version of the standard, or for a specific point in time). While it is possible for multiple Rules Engines to be running on a single execution server, the rules service will keep each engine isolated on its own execution server, in its own container.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 879 of 1295

Selecting the category of Rules Engine will be the responsibility of the Rules Service API App, while selecting the appropriate variant will be the responsibility of a separate 'selector' Rules Engine whose sole task will be routing requests to the correct instance.

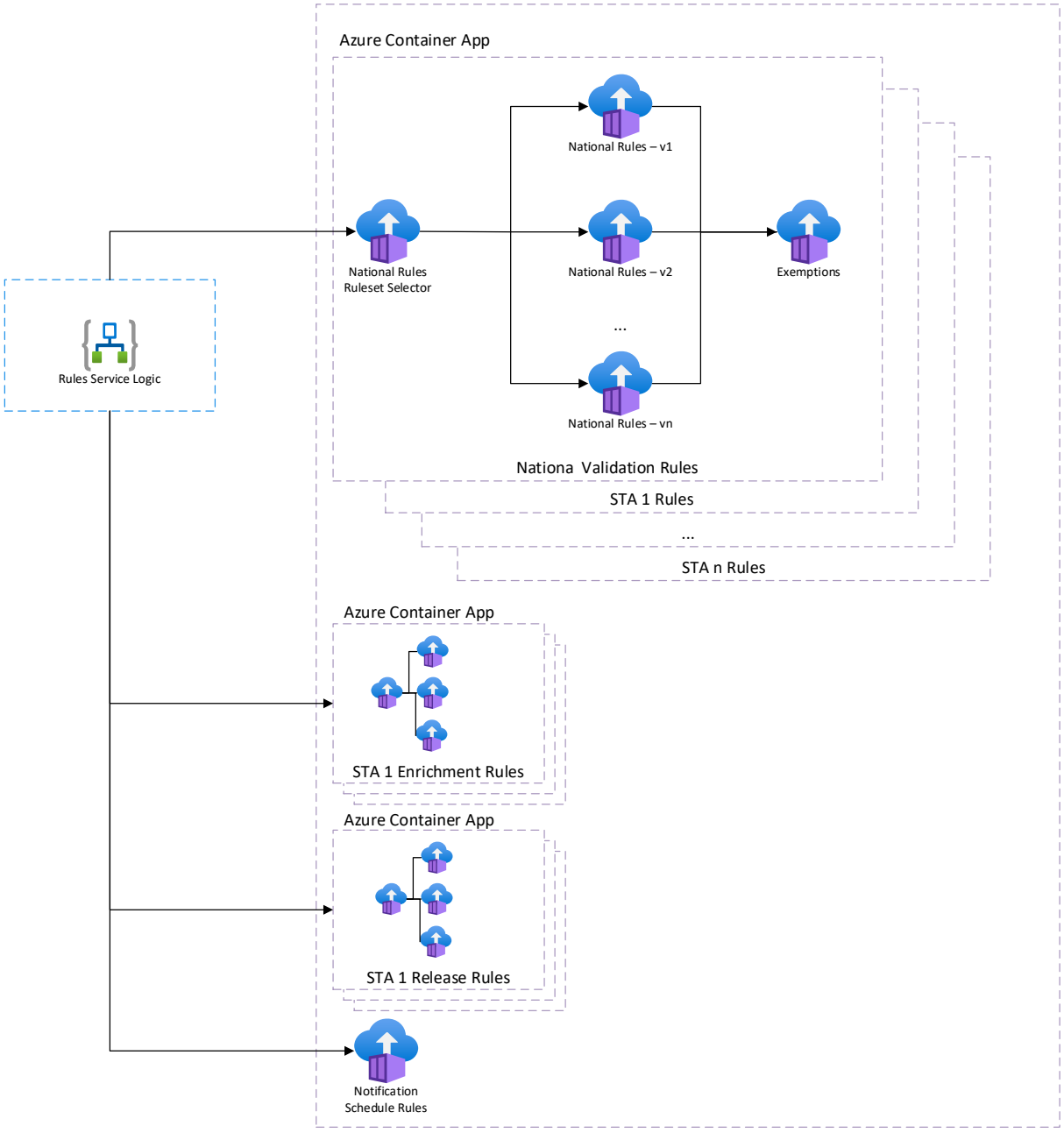


Figure 22: Rules Engine Architecture (C3)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 880 of 1295

10 Access Management

The System Operator role will be in charge of managing and configuring the Rules Service data and rules.

Designated operators in STAs with validation that is performed within the STARS system will also require access to the Rules Management Portal. This needs to be able to be configured as read only or read/write.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 881 of 1295

11 Security

STARS will be built to meet existing Australian Government policies, frameworks, standards and design principles, including:

- Australian Government API Design Standards
- Digital Service Standard
- Protective Security Policy Framework (PSPF)
- Information Security Registered Assessors Program (IRAP)
- Australian Government Information Security Manual (ISM)
- the Open Web Application Security Project® (OWASP)
- Hosting Certification Framework.

All data within STARS will be handled in accordance with relevant privacy laws and record keeping requirements, including the Privacy Act, the *Archives Act 1983* (Cth) and any other applicable legislation (e.g., applicable State and Territory privacy legislation or archiving requirements).

The Rules Service will conform to the Security Architecture defined for STARS in the Related Documents. Access to the Rules Service will be controlled by Azure RBAC, with all calling services identified and granted access through Azure AD Managed Identities.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 882 of 1295

12 Dependency on Other Services

12.1 Submission Service

When validations are requested, they will only include the identifier for the submission or dataset being validated. The Rules Service will need to make a call out to the Submission Service in order to retrieve the full data associated with the submission or dataset.

12.2 Provider Service

When pre-submission validations are requested, they will only include the identifier for the submission or dataset being validated. The Rules Service will need to make a call out to the Provider Service in order to retrieve the full data associated with the submission or dataset.

12.3 Reference Data Service

Many rules in the validation of training activity data are dependent on data provided by the Reference Data Service. Details on how this data will be accessed can be found in Related Documents.

12.4 Release Service

When the Rules Service receives requests for re-validation of records which have undergone data enrichment, the data enrichment information will have to be retrieved from the Release Service.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 883 of 1295

13 Functional Requirements

13.1 Fulfilled Requirements

Table 1818: Fulfilled Requirements

Requirement ID	Requirement Name	Requirements Statement	Acceptance Criteria	Traceability
FR026	FR026 - Test environment	The System Operator must be able to test a data classification, business or validation rule in a test environment.	Given a test environment When the system operator adds or modifies a data classification Then the new data classification can be tested in that environment Given a test environment When the system operator adds or modifies a business rule Then the new business rule can be tested in that environment Given a test environment When the system operator adds or modifies a validation rule Then the new validation rule can be tested in that environment	See 6.7 Rules Management Portal, 7.1 Rules Management
FR027	FR027 - Managed validation rules	A Systems Operator must be able to manage the National/Reference/Funder Data Validation Rules on behalf of rule owner (STA, NCVER etc): • Create • Read • Update • Delete • Copy • Activate • Deactivate	Given validation rules When a system operator requests a validation rule be created on behalf of the rule owner Then STARS will initiate a process to have that rule created Given validation rules When a system operator views the rules Then STARS will show all rules Given validation rules When a system operator requests a validation rule have its logic updated on behalf of the rule owner Then STARS will initiate a process to have that rule logic updated	See 6.7 Rules Management Portal, 7.1 Rules Management

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

			Given validation rules When a system operator requests a validation rule be deleted on behalf of the rule owner Then STARS will soft-delete that rule Given validation rules When a system operator requests a validation rule be copied on behalf of the rule owner Then STARS will copy that rule And allow its configuration to be updated And ensure no duplicates are added Given validation rules When a system operator activates a rule for a particular scope on behalf of the rule owner Then STARS will activate the rule for the scope defined Given validation rules When a system operator deactivates a rule for a particular scope on behalf of the rule owner Then STARS will deactivate the rule for the scope defined Given validation rules When a system operator changes rule configuration on behalf of the rule owner Then STARS will update the rule configuration	
FR028	FR028 - Test environment	A Custodian/STA must be able to test a business/validation rule in a testing environment.	Given a test environment When a custodian adds or modifies a business rule Then the new business rule can be tested in that environment Given a test environment When a custodian adds or modifies a validation rule Then the new validation rule can be tested in that environment Given a test environment When an STA adds or modifies a business rule	See 7.1 Rules Management

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

			Then the new business rule can be tested in that environment Given a test environment When an STA adds or modifies a validation rule Then the new validation rule can be tested in that environment	
FR031	FR031 - View and extract validation rules	The VDS system must allow the extraction of the National, Reference or STA/Funder Data Validation Rules. A STA/Funder will be able to view, distinguish, and extract the configured national and state-specific validation rules.	Given STARS internal validation rules When an STA/Funder views the validation rule configuration Then STARS will show all relevant validation rules Given STARS internal validation rules When an STA/Funder extracts the validation rule configuration Then STARS will generate an extract of all relevant validation rules with their current configuration values	See 6.7 Rules Management Portal
FR055a	FR055a - Business rules	The VDS system must have capability to create, configure, edit, delete business rules and capturing justifications for these actions • The VDS system will have the ability to add or update business rules that trigger notifications. • The VDS system manages a central register of rules, including a unique ID for each discrete rule, and classification of the rule to help define the rule type, to enable the easy identification of validation outcomes and analysis when providing support to training organisations to submit valid data, and when troubleshooting rule issues with the System Operator. • The VDS system will advise users of all additions and updates to business rules. • The VDS system will support users to download business rules and updates.	Given validation rules When a request to create a validation rule is received Then STARS will initiate a process to have that rule created And the rule will be given a unique ID And the rule will be given a Rule Type And the rule will include information to assist troubleshoot failures And STARS will notify appropriate users of the rule's creation And STARS will capture justifications for the new rule Given validation rules When a request to update the logic of a rule is received Then STARS will initiate a process to have that rule logic updated And STARS will notify appropriate users of the rule's modification And STARS will capture justifications for the modification Given validation rules When requests to delete a rule is received Then STARS will soft-delete that rule And STARS will capture justifications for the deletion Given validation rules When a request is received to activate a rule for a particular scope	See 6.7 Rules Management Portal, 7.1 Rules Management

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

			Then STARS will activate the rule for the scope defined And STARS will capture justifications for the activation Given validation rules When a request is received to activate a rule for a particular scope Then STARS will deactivate the rule for the scope defined And STARS will capture justifications for the deactivation Given validation rules When a request to change rule configuration is received Then STARS will update the rule configuration And STARS will capture justifications for the reconfiguration Given validation rules When a request to download the rules is received Then STARS will generate an export of the business rules And download them	
FR055b	FR055b - Updating business rules	The VDS system must allow business rules to be easily updated to reflect feedback from training organisations who identify rules that force submission of inaccurate data • The VDS system will have a process for data providers to report problems with the VIS that restrict high quality/accurate data.	Given validation results to training activity providers When the provider requests an update to a rule Then STARS will record the request And initiate a process to review the rule Given a rule review When the outcome requires an update to the rule Then STARS will update the rule And STARS will notify the provider of the outcome And the Review will be closed Given a rule review When the outcome requires does not require an update Then STARS will notify the provider of the outcome And the Review will be closed Given a rule review When the outcome requires an exemption Then STARS will record the exemption	7.1 Rules Management

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

			And STARS will notify the provider of the outcome And the Review will be closed	
FR141	FR141 - Access Control	The VDS Solution must allow for validations functionality to be disabled (put on hold) for individual training organisations at any point in time with specific criteria (eg specified validation rules, time turned off, specific training organisations/program). This is to support special exemption needs on contracted training organisations. Temporarily disabling validations must be approved and the system should capture the associated metadata (date & time; approver; requester; justification; associated records, threshold changes, related document links).	Given rules in STARS that are preventing submission of accurate data When a review is requested Then STARS will initiate an rule exemption review process And details of the exemption request will be recorded Given a rule exemption review When the outcome of the review is decided Then STARS will record the review outcome And notify the initiator of the review as to the outcome Given a completed rule exemption review When the outcome of the review is to grant an exemption Then STARS will update the rule configuration to apply the outcome And the outcome may affect multiple rules And the exemption may have specific start and end dates And the exemption may apply to an individual training organisation And the exemption may apply to a specific Program	See 6.6 Exemptions
FR142	FR142 - TSNSW - Program Specific Validations	Validation rules must include program specific validation rules that are only applied to contracted training organisations funded activity data specific to that program.	Given validation rules in STARS And a rule is specific to a particular Program When STARS validates training data for that program Then STARS will apply the program specific validation rules Given validation rules in STARS And a rule is specific to a particular Program When STARS validates training data for a different program Then STARS will not apply the validation rules for that particular program	7.1 Rules Management

13.2 Informing Requirements

Table 1919: Informing Requirements

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

Requirement ID	Requirement Name	Requirements Statement	Acceptance Criteria	Traceability
FR005a	FR005a - Application of validation rules based on dataset	The VDS system must validate data submitted by a training organisation against the VDS National/Reference Data rules. This includes: • Validation from a consistent rule set, regardless of which method was used to enter/upload the data	Given Training Activity submission When a Training Organisation submits data via the API Then the submission is validated against the relevant internal validation rules at both the National and STA level Given Training Activity submission When a Training Organisation submits data via the Web Interface Then the submission is validated against the relevant internal validation rules at both the National and STA level	See 6.3 Rules Engine Instances
FR005b	FR005b - Application of validation rules to prevent duplicate records	The VDS system must validate the data submitted by a training organisation against the VDS National/Reference Data rules. This includes: • The VDS system will be able to distinguish whether data submitted reflects a new or existing record in order to prevent duplicate records.	Given STARS contains a matching record(s) When a Training Organisation submits data Then STARS will identify the data as a match for the existing record(s) Given a Training Organisation training data submission has been accepted When the record is identified as being new Then a new entry will be recorded in STARS Given a Training Organisation training data submission has been accepted When the record matches existing data Then STARS will update the existing record(s)	See 8.1 Validations
FR006	FR006 - National rules error notification	The VDS system must notify a training organisation of the VDS National/Reference Data validation result (errors/warnings) in near real-time via: • Student Management System • Adaptive Web User Interface	Given Training Activity submission When a Training Organisation submits data via the SMS API Then the submission is validated against the relevant internal validation rules And the response via the API will include all errors, warnings and notifications relevant to the data validation result Given Training Activity submission When a Training Organisation submits data via the Web Interface Then the submission is validated against the relevant internal validation rules And the response in the Web Interface will include all errors, warnings and messages relevant to the data validation result	See 8.1 Validations

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR008a	FR008a - STA/Funder Validation	The VDS system must validate data submitted by a training organisation against the STA/Funder validation rules. This includes: • Validation from a consistent rule set, regardless of which method was used to enter/upload the data	Given funded Training Activity When the training activity records are submitted via SMS API Then STARS will validate the records against all relevant STA/Funder validation rules Given funded Training Activity When the training activity records are submitted via Web Interface Then STARS will validate the records against all relevant STA/Funder validation rules	See 8.1 Validations
FR008b	FR008b - STA/Funder Validation	The VDS system must validate the data submitted by a training organisation against the STA/Funder validation rules. This includes: • The VDS system will be able to distinguish whether data submitted reflects a new or existing record in order to prevent duplicate records.	Given STARS contains a matching record(s) When a Training Organisation submits funded data Then STARS will identify the data as a match for the existing record(s) Given a Training Organisation submission of funded activity has been accepted When the record is identified as being new Then a new entry will be recorded in STARS Given a Training Organisation training submission of funded activity has been accepted When the record matches existing data Then STARS will update the existing record(s)	See 8.1 Validations
FR009a	FR009a - Real - Time validation results	The VDS system must validate data submitted by training organisation against the STA/Funder Data validation and provide a result (warnings/errors) in near real-time via the interface the submitter used e.g.: • SMS • Adaptive Web User Interface • Mobile App (out of scope)	Given funded Training Activity submission When a Training Organisation submits data via the SMS API Then the submission is validated against the relevant STA/Funder validation rules And the response via the API will include all errors, warnings and notifications relevant to the data validation result Given funded Training Activity submission When a Training Organisation submits data via the Web Interface Then the submission is validated against the relevant STA/Funder validation rules	See 8.1 Validations

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

			And the response in the Web Interface will include all errors, warnings and messages relevant to the data validation result	
FR029	FR029 - Compliance Metrics	The System Operator must be able to configure compliance metrics in the VDS system on the behalf of the VET Regulator. E.g. thresholds for data integrity metrics to be defined in business rules	Given business rules with configured compliance metrics When the System Operator updates the configuration on behalf of the VET Regulator Then STARS will apply the configuration changes	See 6.5 Reference Data Cache

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

14 Non-Functional Requirements

Table 2020: Non-functional Requirements

Requirement ID	VDS NFR Category	Requirement Name	Requirements Statement
NFR006d	Compatibility	NFR006d - APIs are AVETMIS Standard agnostic	The VDS System's APIs are agnostic to which version of AVETMISS is used
NFR007a	Accessibility	NFR007a - Web Content Accessibility Guidelines (WCAG)	The VDS System will be compliant with the Web Content Accessibility Guidelines (WCAG) outlined in Digital Service Standard issued by the Digital Transformation Agency. • The VDS System will be compliant with or exceed the minimum Web Content Accessibility Guidelines (WCAG) Level 2.0 Level AA outlined in the Digital Service Standard and ensure ongoing adherence throughout the Agreement Term.
NFR013b	Audit	NFR013b - Generate business rules	The VDS System will be able to generate a version of the Business Rules.
NFR013c	Audit	NFR013c - Generate changes to business rules	The VDS System will be able to generate a record of all changes to Business Rules, including the specific changes that were made, the user that undertook the change, and the date and time the change was made.
NFR021f	User Experience	NFR021f - Browsers	The VDS System must support browsers such as Microsoft Edge, Google Chrome, Firefox and Apple Safari
NFR021n	User Experience	NFR021n - Validation rules	The VDS system should provide access to a rule set/ definitions that that it evaluates against. This needs to be as granular and up to date as possible.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

15 Acronyms

RAM	Relationship Authorisation Manager
RTO	Registered Training Organisation
STARS	Student Training and Activity Reporting System
TP	Training Provider
TDIF	Trusted Digital Identity Framework
SMS	Student Management System
SSO	Single Sign On
OIDC	OpenID Connect
STAs	State Training Authority
PoC	Proof of Concept
DEWR	Department of Employment and workplace Relations
VET	Vocation Education and Training
VDS	VET Data Streaming
Azure AD	Azure Active Directory
RBAC	Role Based Access Control
AAD	Azure Active Directory
IdP	Identity Provider
DRL	Drools Rules Language
DMN	Decision Model and Notation

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 893 of 1295

Australian Government
Department of Employment and Workplace Relations

Detailed Design Document

Identity & Access Management (IAM)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 894 of 1295

1 Document Management

1.1 Document Control

Version	Change Description	Date
0.1	Draft	10/02/2023
0.2	Address Feedback Comments	23/02/2023
0.3	Updated to include TDIF/AGDIS terms and figures and open items.	07/03/2023
0.4	Further updates to address feedback	17/03/2023
0.5	Further updates to address feedback, includes description of claims-based authentication.	05/04/2023
1.0	Acceptance	28/04/2023

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This is a living document and will be updated as required. If another component's design scope requires an update to this design document, then the appropriate change will be made and a new version issued.

1.2 Document Author

Role	Name
Technical Architect	s 47F(1)
Solution Architect	s 47F(1)

1.3 Document Reviewers

Role	Name

1.4 Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

1.5 Related Documents

Reference	Document
A	STARS Solution Requirements Document v1.3.docx
B	STARS Solution Design v1.0.docx
C	Detail Design Document - IAM – Diagrams v1.0.vsdx

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 895 of 1295

1.6 All STARS System Components

This list outlines the high-level service dependencies with **Identity and Access Management**. This is a reflection of what is known at the time of Acceptance and may be subject to change throughout the design process.

Green = This Design Document Grey = Other Service Dependencies

STARS Service
Azure Infrastructure and Networking
Identity & Access Management (IAM)
Event Service
Logging & Audit Service
Master Data Service
Reference Data Service
Data Governance Infrastructure
Data Quality Framework
Notification Service
Submission Service
Validation Service
Rules Service
Provider Service
Mastering Service
Downstream Consumer ETL Pipeline Services
Release Service
Downstream Consumer Reporting & Analytics
R & A Provisioning Service
STA Batch Submission Services
STA Gateway Services
USI Transcript Service
STA Bulk Data Consumption Service
SMS SDK Service

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 896 of 1295

2 Open Design Items

This section describes items arising from the design process that remain unresolved pending further work that will take place later during the project.

#	Open Item / Gap	Reason can't be resolved at initial acceptance	DevOps Action ID	Action Description	Owner	Target Date	Impact Date	Build Progress Impact (if any)
1	Analysis and design of Azure AD Federation with regards to access to STARS from partner agency users and federation.	Requirements for partner agency access control are to be determined, along with an assessment of related security considerations.	3793	Analysis and design of Azure AD Federation with regards to access to STARS from partner agency users and federation.	Requirements: VDS Program Design: DXC	TBD – pending program planning for partner agency access and associated requirements, during Stage 1.	Dec 2023	This item is not on the critical path for Stage 1 Capability Release build activities since there is no direct impact on the design or implementation of other STARS services. These services rely only on Azure AD provided access tokens, which allows for any future configuration-based changes to Azure AD that would include, say, adding a federated IDP

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

#	Open Item / Gap	Reason can't be resolved at initial acceptance	DevOps Action ID	Action Description	Owner	Target Date	Impact Date	Build Progress Impact (if any)
								from another agency. Future configuration changes in Azure AD control if a new user is granted a relevant token that is then used by STARS services. I.e., federation, identity and granting access is not a function of the service, this is delegated to Azure AD, rather, the services only checks that the user has been granted access, based on the token presented.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

#	Open Item / Gap	Reason can't be resolved at initial acceptance	DevOps Action ID	Action Description	Owner	Target Date	Impact Date	Build Progress Impact (if any)
2	Machine to machine IDM, choice of digital identity management system.	Discussions still underway as to suitable Australian Gov't approved M2M services, including PRODA and others. DXC is awaiting guidance/direction from program and DEWR.	3792 / 1857	Guidance/direction required from program and DEWR. Then, update the design	Requirements: VDS Program Design: DXC	28/4/2023 (Requirements/Guidance received from DEWR) 30/5/2023 (Design)	Dec 2023 (M2M established)	as per item 1, STARS services, including APIs, rely on tokens granted, not the actual M2M configuration. M2M is likely to be implemented through Azure B2B, noting this, M2M setup is mainly a configuration item once an approved M2M protocol/service is determined. As such, this does not impact development of capability services, but will need to be resolved prior to

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

#	Open Item / Gap	Reason can't be resolved at initial acceptance	DevOps Action ID	Action Description	Owner	Target Date	Impact Date	Build Progress Impact (if any)
								testing with external systems.
3	Define Training Organisation Onboarding and Registration Process	Onboarding process is not defined yet, around MyGovID	3824	Determine process for MyGovID	Requirements: VDS Program Design: DXC	28/4/2023	28/4/2023	Build can continue and then changes made as the process becomes clear.
4	Final access method for partner agencies for testing purposes once federation designs accepted.	Dependency on federation design.	3982	Set up temporary vetdata.edu.au accounts for nominated testers from partner agencies.	Requirements: VDS Program (Co)Design: DXC + partner agencies	30/6/2023 (confirmation from partner agencies on external users to be given STARS accounts)	31/12/2023 (federated accounts granted, min NCVER + DEWR; other agencies TBC)	Build can continue, with testing for external users made using vetdata.edu.au accounts. When federation design is accepted and implemented, external users can be granted access through their agency's AZAD federation configuration.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 902 of 1295

Contents

1 Document Management 2
1.1 Document Control 2
1.2 Document Author 2
1.3 Document Reviewers 2
1.4 Document Approval 2
1.5 Related Documents 2
1.6 All STARS System Components 3
2 Open Design Items 4
Contents 10
3 Introduction 12
3.1 Purpose 12
3.2 Intended Audiences 12
4 Scope and Exclusions 13
4.1 In Scope 13
4.2 Out of Scope 13
4.2.1 AGDIS Configuration 13
4.2.2 Azure Active Directory Federation 13
5 Design Decisions 14
6 STARS IAM Overview 15
6.1 Identity Management 15
6.2 Access Management 15
6.3 Azure Identity and Access Management 15
6.3.1 Claims-Based Authentication 16
6.3.2 Azure and Claims-Based Authentication 16
6.3.3 Azure API Management (APIM) and Securing STARS Services 16
7 APIs and SDKs 18
8 Identity and Access Management – STARS Internal 19
8.1 Authentication (Authn) 19
8.2 Authorization (Authz) 21
8.3 Azure AD Managed Identities 21
8.4 Azure Services IAM Configuration 23
9 Training Provider Authentication 27
9.1.1 Identity Exchange Service 28

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 903 of 1295

9.1.2 Azure B2C Mediated Sequence 30
10 Solution Security Matrix **35**
11 Requirements **36**
11.1 Functional 36
11.2 Non-Functional 36
12 Appendices **38**
Appendix 1. Acronyms 38
Appendix 2. System Administration Related ISM Security Controls 39

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 904 of 1295

3 Introduction

This document covers the design and organisation of IAM for the STARS solution down to the resource group and network level. Specific configuration of Azure-based IAM is covered in each of the various STARS services design documents.

3.1 Purpose

The purpose of this document is to capture the detailed design requirements for Identity and Access Management for various stake holders, systems and services in STARS.

This will provide a guidance to developers about the basic flow in relation to managing identity and accessing protected resources to develop various azure functions and services for STARS.

The design is informed by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework, which provide a comprehensive set of best practices and guidance for designing, deploying, and operating secure, high-performing, and resilient cloud infrastructure and applications.

With respect to expected RTO/SMS engagement activities, it is DXC's understanding based on the current requirements and solution design, noting that DXC have not yet conducted detailed consultation with the RTO/SMS Providers, that there is no component in this Deliverable that directly impacts RTO/SMSs. All direct impacts to RTO/SMSs are dealt with in the Provider Portal, Provider Service and/or SMS SDK Service as understood and agreed previously. Anything IAM related that does affect RTO/SMS will be added into the Provider Service design.

3.2 Intended Audiences

This detailed design document provides technical information and guidance to help technical stakeholders understand the Azure IAM services used in the STARS solution and how they integrate with the solution components. The document is expected to be used by technical stakeholders who have a good understanding of cloud computing concepts and are experienced in the Azure platform.

This document is not intended for non-technical readers, nor for readers who are not experienced in the design and configuration of Azure-based solutions; informational and educational resources regarding Azure are readily available at https://azure.microsoft.com

The primary audience for this document includes:

- Program technical staff: This includes architects, software engineers, administrators, and operations teams within the program responsible for the STARS solution.
- External technical stakeholders: This includes technical staff from other federal, state, and territory governments agencies, including VDS embedded team members.
- ICT Technical Staff: This includes a variety of technical roles involved during the VDS program and post-go live, including network engineers, systems administrators, database administrators, security engineers, and DevOps engineers.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 905 of 1295

4 Scope and Exclusions

4.1 In Scope

This document covers basic flow and common scenarios related to identity and access management for STARS and how it will be configured in different ways to suit service and user requirements.

4.2 Out of Scope

The following are out of scope for this document:

4.2.1 AGDIS Configuration

STARS will interface with the Australian Government Digital Identity System (AGDIS) Identity Exchange (currently homed within Service Australia), delegating identity provider and attribute provider enlistment, configuration and IAM services.

This document does not cover the relevant detailed AGDIS-based configuration, onboarding, etc., aside from noting where the STARS IAM interacts.

For example, the setup of RTO's Director as principal authority and other employees as authorised users in Relationship Authorisation Manager (RAM) and associating various digital IDs for RTO is internal process for RTOs and is not covered in this documentation (noting that this is likely covered by VDS program change management activities.)

4.2.2 Azure Active Directory Federation

A decision has been made that future external agency integrations in later stages, e.g., for STA API access, will be controlled through Azure Active Directory (AAD) Federation (between STARS and agency tenants). As requirements surrounding IAM for these integrations have not yet been established, the design of AAD Federation is (currently) out of scope for this document. Please note that as the STA integration requirements and associated IAM requirements are established, this design will be updated accordingly.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 906 of 1295

5 Design Decisions

This document outlines common Identity & Access Management (IAM) scenarios to consider when using Azure services and resources within STARS and provides a guidance based on methods indicated in STARS high level solution design document.

Certain technical solutions are outlined, along with common patterns to protect identity and manage access of Azure services and resources that will be used at the time of implementation.

The (to-be) accepted Solution Security Matrix will provide details to refine and map user roles with various business process at granular level within STARS.

The Solution Security Matrix will inform the design and creation of Azure AD groups for access controls across STARS.

The following design decisions apply for IAM within STARS:

1. Claims-based authentication, using OAuth/OpenID Connect (a.k.a. Modern Auth) as the main external Authn/Authz protocol.
2. Azure Active Directory (AAD) as the main IAM directory and will provide all IAM services for STARS internal services across the solution. AAD is synchronised with the vetdata.edu.au ADDC hosted within the Identity subscription; this ADDC is the primary source of internal STARS user accounts.
3. For external users and systems, STARS is designed to operate within the Trusted Digital Identity Framework (TDIF), using Australian Government Digital Identity Services (AGDIS) approved digital identity providers and attribute providers, via the Services Australia-managed Digital Identity Exchange service. Relationship attributes for individuals logging into the STARS web app will be handled via organisational relationships in Relationship Authorisation Manager (RAM)[1];
4. SMS authentication – ATO RAM M2M authentication will be leveraged here to ensure only authenticated SMS operators can use the Provider Portal and send communications via our API gateway.

Multifactor Authentication (MFA) may be handled by the approved digital identity provider, (e.g., myGovID smartphone app PIN, or biometric authentication) for training provider users or by Azure tenancy conditional access configuration with Microsoft Authenticator for Azure privileged users (e.g., partner agencies.)

The user authentication flow will be initiated by STARS, with referrals to identity Exchange using Azure Active Directory B2C[2] providing user email address and the business identifier (e.g., as configured in myGovID or Digital ID) and utilise the Australian Government Digital Identity Services[3].

The (to be finalised) Solution Security Matrix will map User Groups/Roles with business processes and provides details related to Access Control within the scope of functionalities and services provided by STARS.

[1] https://info.authorisationmanager.gov.au. NB: RAM is currently the only approved attribute provider.
[2] https://www.digitalidentity.gov.au/sites/default/files/2021-06/tdif-06b-openid-connect-1.0-profile-release-4-v1.6.pdf
[3] https://docs.microsoft.com/en-us/azure/active-directory-b2c/authorization-code-flow

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 907 of 1295

6 STARS IAM Overview

Azure AD is a cloud-based identity and access management solution by Microsoft in Azure. It will provide identity management and access control for STARS services.

6.1 Identity Management

Azure AD & Azure B2C will provide the following Authn services:

- User Management
- Multi-factor Authentication
- Application Integration
- Single Sign-On for STARS UIs

These features provide secure way of managing user access to resources and across STARS and play an important part of the overall security posture.

6.2 Access Management

Azure Role-Based Access Control (RBAC) will be used to manage access to STARS Azure resources by defining roles and assigning them to Azure AD users, managed & system identities or groups.

Note that access management, including administrative access, is subject to the ISM controls established for STARS. A subset of these controls, relating to administrative/privileged access, is listed in Appendix 2.

RBAC will be fine-tuned during build as access requirements to STARS Azure resources evolve.

STARS access for external users (e.g., training providers) will also be based on user relationships. For example, a user may be related to a particular STA, or a particular Training Provider. This relationship will define what the user can do and what data they can access, in conjunction with their role.

If a user is related to multiple Training Providers, the STARS user interface will allow the user to select which Relationship they are acting. Regulators from ASQA may add relationships for regulator staff to multiple Training Providers to finalise their submissions after a provider is disbanded.

Note: Further engagement required to understand details of how these relationships will be modelled in STARS. Ongoing discussions have prevented a final decision here, but the working assumption is that this will be managed through RAM.

6.3 Azure Identity and Access Management

In the STARS Azure-based solution, we employ claims-based identity and access management (IAM) to ensure the security and proper access control of Azure-based resources. This section introduces and describes claims-based authentication in the context of the key Azure Active Directory and Azure API Management (APIM) services

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 908 of 1295

6.3.1 Claims-Based Authentication

Claims-based authentication is a modern approach to authenticate and authorize users, applications, and services. It uses a set of claims, which are statements about the user or service, such as identity, role, or permissions, and packages them into a secure token. These tokens, a.k.a. JSON Web Tokens (JWT), are generated and signed by trusted identity providers (IdPs), using protocols like OAuth 2.0 or OpenID Connect.

6.3.2 Azure and Claims-Based Authentication

In the Azure ecosystem, Azure Active Directory (Azure AD) is the primary IdP that supports claims-based authentication. Azure AD is a multi-tenant, cloud-based directory and identity management service that offers a wide range of features, including single sign-on (SSO), multi-factor authentication (MFA), and role-based access control (RBAC). By integrating Azure AD with other Azure services, STARS will leverage claims-based authentication to provide secure access to STARS resources.

6.3.3 Azure API Management (APIM) and Securing STARS Services

Azure API Management is a service that enables the creation, publishing, and management of APIs for both external and internal consumers. It supports claims-based authentication through integration with external IdPs like Azure AD or other OAuth 2.0/OpenID Connect-compatible providers. We can use policy expressions in Azure APIM to validate JWT claims in incoming requests, and to authorize or restrict access based on these claims.

Reference: https://docs.microsoft.com/en-us/azure/api-management/api-management-key-concepts#security

STARS services that are fronted by APIM will be secured through a combination of mechanisms:

Authentication: By integrating APIM with Azure AD or other IdPs, STAR can require users, applications, and services to authenticate using claims-based authentication before accessing the APIs. This ensures that only authenticated entities can access the APIs.

Authorization: STARS will use policy expressions in APIM to inspect JWT claims and determine if the requester has the appropriate permissions to access the API. This provides fine-grained control over the access to APIs, based on the user or service's role, group membership, or other claim values.

IP Filtering: APIM allows us to restrict access to APIs based on the client's IP address. This will be used to allow or block specific IP ranges, ensuring that only trusted networks can access the APIs (e.g., external partner agencies.)

Rate Limiting and Quotas: STARS will configure rate limiting and quotas in APIM to control the number of requests per user or subscription over a specified time period. This helps to prevent abuse, protect backend services from excessive load, and maintain a fair usage policy among consumers.

Mutual SSL/TLS Authentication: APIM supports mutual SSL/TLS authentication, where both the client and the server present a certificate for authentication. This provides an additional layer of security,

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 909 of 1295

ensuring that only clients with valid certificates can access the APIs. This will be likely used for machine-to-machine integration for partner agency communications.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 910 of 1295

7 APIs and SDKs

There are no DXC built IAM-related APIs planned for this project; however, the solution will use Azure provided APIs to manage IAM programmatically.

Azure AD provides several built-in APIs for identity and access management (IAM) solutions that will be likely be utilised by various STARS services:

- Microsoft Graph API
- Azure AD B2C API
- Azure AD Graph API
- Azure AD Managed Identities for Azure Resources API
- Azure AD Managed Service Identity (MSI) API
- Azure AD Identity Protection API

All STARS code will use Microsoft provided SDKs that wrap and manage calls to the relevant APIs.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 911 of 1295

8 Identity and Access Management – STARS Internal

This section describes IAM in the context of internal (i.e., non-Training Provider) users and services within the STARS environment. External access for Training Providers is described below in section **Error! Reference source not found. Error! Reference source not found.**.

Azure Active Directory (AAD) is the main IAM directory and will provide all IAM services for the STARS solution. OAuth/OpenID Connect (a.k.a. Modern Auth) underlies Authn/Authz operations across Azure. In the main this is transparent for most development activities.

8.1 Authentication (Authn)

STARS services will use Azure AD to authenticate service principals, including both users and managed identities, to various Azure services and resources.

The Identity Verification flow with managed identity by using Azure AD is displayed below. The sequence of steps to obtain token and verify token follows similar patterns across most Azure services and resources and applies similarly for users.

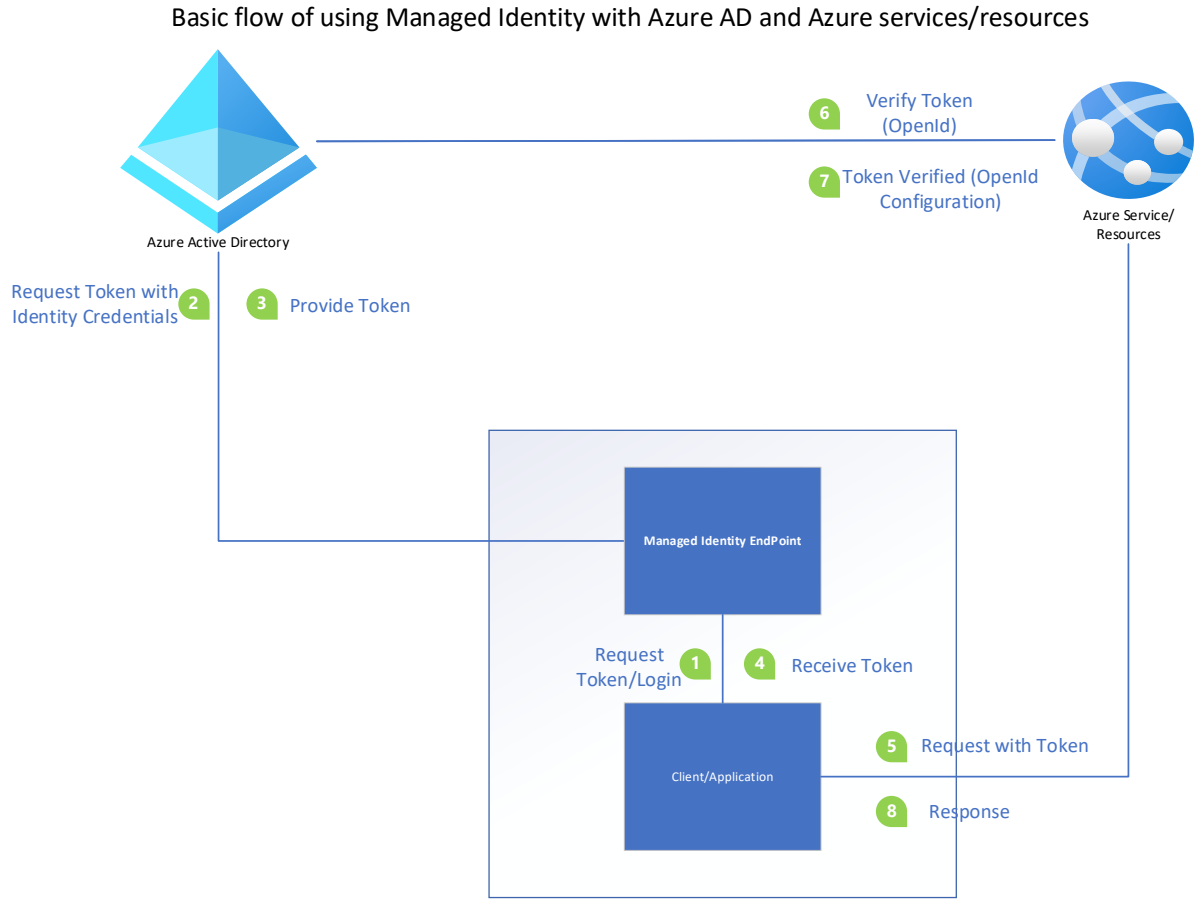


Figure 1: Identity Auth Flow – High Level

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 912 of 1295

Sequence diagram for authentication and token issuance in AAD

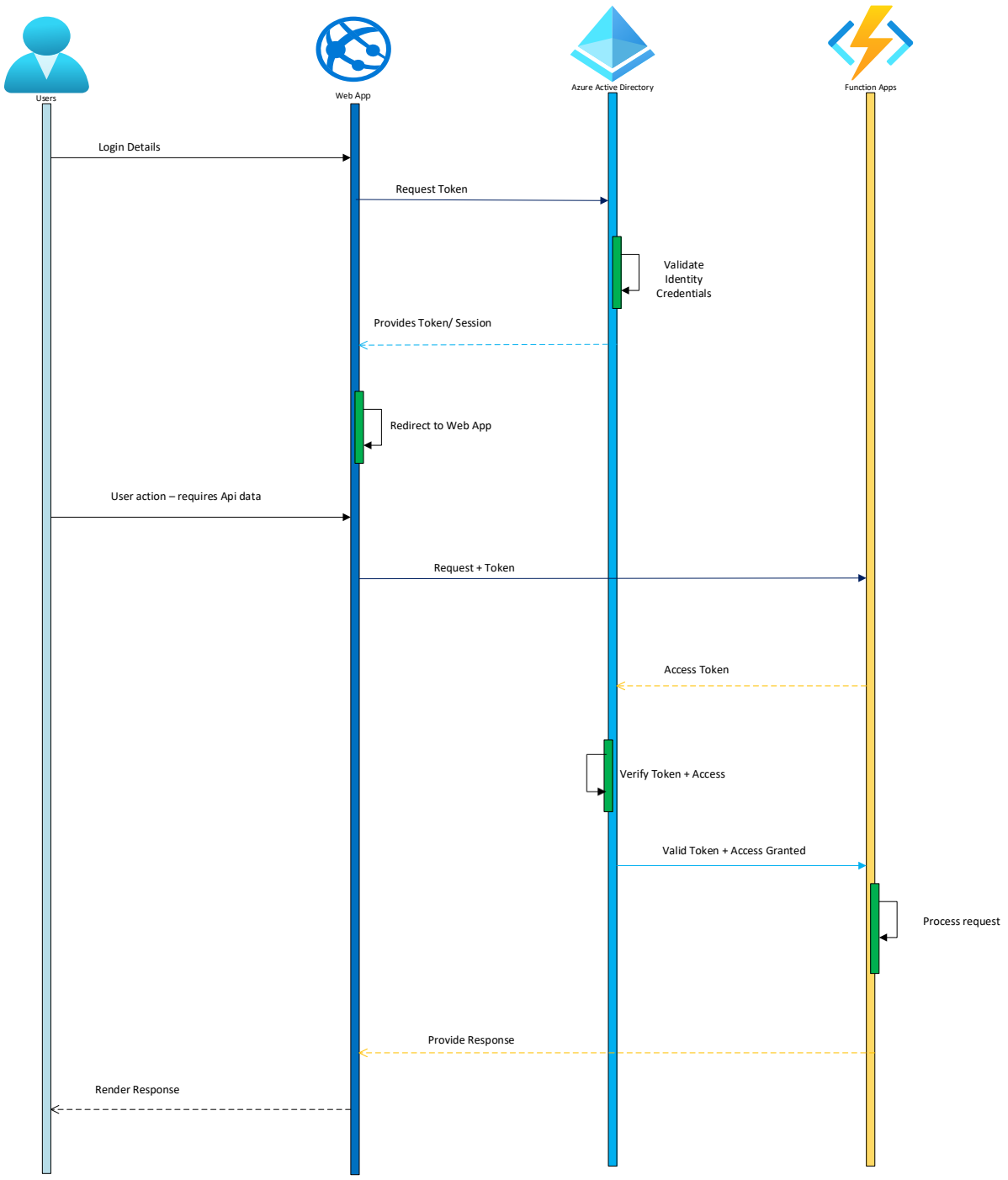


Figure 2: Auth Sequence Diagram (User interaction) (C2)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 913 of 1295

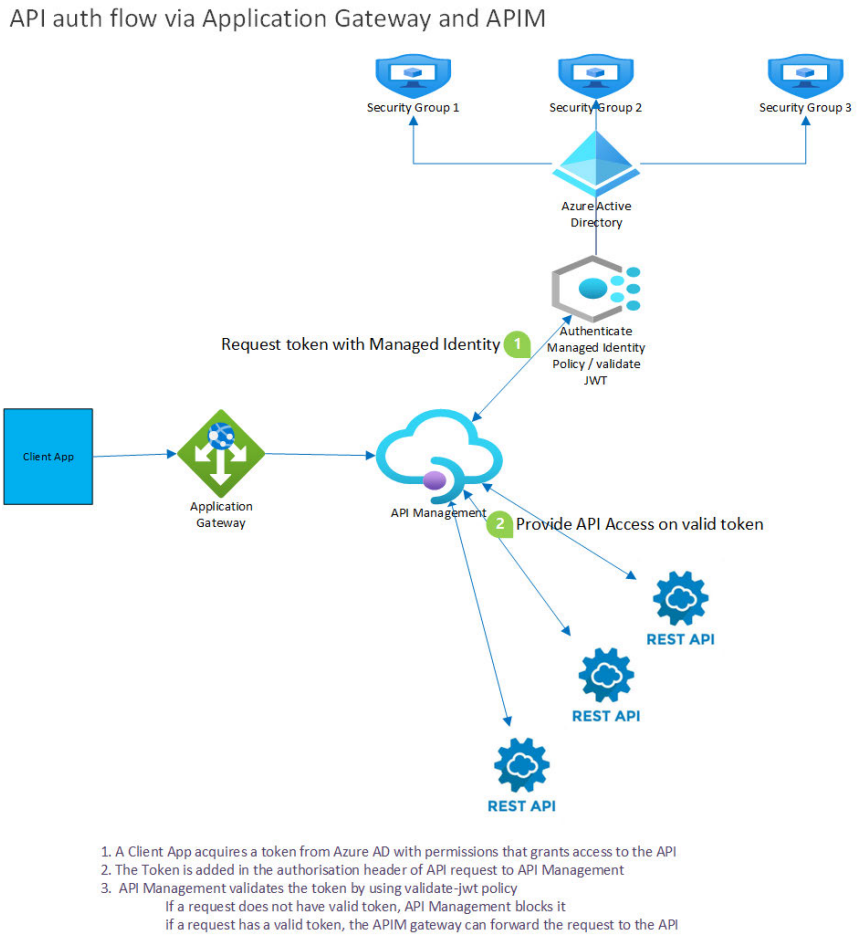


Figure 3: API Auth Flow (C3)

8.2 Authorization (Authz)

Azure Role-Based Access Control (RBAC) is a feature of Azure to manage access to Azure resources by defining roles and assigning them to users or groups. RBAC allows to control who can perform certain actions on specific resources or resource groups in Azure.

RBAC allows fine-tuning access to Azure resources, ensuring that only the right people have access to the right resources.

Roles are assigned to users, groups, service principals, or managed identities at a particular scope (Management Group, Subscription, Resource Groups, Resource).

8.3 Azure AD Managed Identities

A managed identity is a type of Azure Active Directory (AAD) security principal that is automatically created and managed by Azure. Managed Identity reduces risk of identity-related security threats, as well as reduce the complexity of managing authentication and authorization for applications and services.

Managed identities will be provisioned for all STARS Azure Services that support it.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 914 of 1295

Azure automatically creates an AAD managed identity service principals for the Azure services; these identities in turn will authenticate to other STARS Azure services and resources across the solution.

All managed identities for STARS services will be automatically provisioned through relevant ARM templates, deployed via Azure DevOps Pipelines

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 915 of 1295

8.4 Azure Services IAM Configuration

All STARS Azure services will be configured for IAM (Authn/Authz) as follows:

Table 1: Azure Services IAM Configuration

Azure Service	IAM Configuration
Azure Functions	All STARS Function Apps will use the Microsoft Identity Provider for authentication (i.e., Azure AD), and themselves be configured to be provisioned with a managed identity. Function App managed identities will be used for function access to other Azure resources.
Azure Logic Apps	All STARS Logic Apps will use the Microsoft Identity Provider for authentication, and themselves be configured to be provisioned with a managed identity. Logic App managed identities will be used for function access to other Azure resources. Azure Logic App managed identities will be used to authenticate to other Azure services through the relevant service connector (e.g., by providing the tenant id and client id of the managed identity in the connector configuration). For example, The basic flow of an Azure AD identity flow with an Azure Logic App. • A user/system sends a request to an Azure Logic App endpoint, for example, an HTTP request. • The Azure Logic App triggers the execution of a workflow. • The workflow invokes an Azure AD authentication action, which prompts the user/system to provide their credentials/secret. • The user/system presents their credentials/secret and Azure AD verifies them. • If the credentials are valid, Azure AD issues an access token to the user/system. • The workflow uses the access token to call other services or actions that require authentication. • The Logic App processes the request and sends a response to the user.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

Azure Service	IAM Configuration
	Please note that this is a general flow and will be configured in different ways to suit the Logic app workflows' needs.
Azure UIs and Portals (Angular PWA)	STARS portals such as Provider Portal will use Azure AD authentication with OAuth 2.0 token to access various Azure services and resources. (Please refer to section **Error! Reference source not found. Error! Reference source not found.** for a detailed example for Azure AD B2C Integration with AGDIS-approved identity providers to exchange claims and issue tokens for the provider portal.)
API Management	Azure API Management (APIM) will be provisioned with managed identity to authenticate to various Azure services and resources. For example, managed identity will be used to authenticate to an Azure Storage Account by setting the connection string to use the managed identity and the tenant ID of the AAD tenant that the Azure API Management instance is associated with.
Azure Service Bus	Service Bus will be configured to accept only connections from Azure service principals (not SSAL).
Azure Data Factory (ADF)	Azure Data Factory (ADF) will be configured to use a managed identity to authenticate to various Azure-based data sources and services. However, certain use cases, including integrations with external systems, may require alternative methods, e.g., API keys, IDs/passwords. In all instances, secrets for external systems will be held in Key Vault.
Azure Cosmos DB	Azure Cosmos DB allows the use of managed identity to authenticate to the service and access data. As a 'passive' service, it does not (usually) access other services and thus is not provisioned with its own managed identity.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

<table>
<tr><th>Azure Service</th><th>IAM Configuration</th></tr>
<tr><td></td><td>STARS services managed identities will be used authenticate to Cosmos DB. Granular access control to Cosmos DB data is managed through appropriate role-based access control (RBAC) permissions; these will be refined during build.</td></tr>
<tr><td>Power BI</td><td>Power BI will provide data visualisation capabilities across Azure SQL database and Azure Synapse (SQL Serverless). Power BI services may connect to these Azure resources through the methods below:

Table 2: Power BI Access Management Options
<table>
<tr><th>Azure Resource</th><th>Access Management</th></tr>
<tr><td>Azure SQL Database</td><td>SQL Authentication, Azure Active Directory</td></tr>
<tr><td>Azure Synapse (SQL Serverless)</td><td>SQL Authentication, Azure Active Directory</td></tr>
</table></td></tr>
<tr><td>Azure SQL database</td><td>Azure SQL Database will be provisioned with managed identities and will be configured to use Azure AD as its primary authentication mechanism. A decision is still outstanding as to the use of SQL Server identities.</td></tr>
<tr><td>Purview</td><td>To scan Azure data sources in Microsoft Purview securely, Microsoft Purview will be connecting to the Azure resources via Self-Hosted Integration runtime as public access to the data sources will be disabled. Microsoft Purview will scan the data sources that are behind a private endpoint as below.
<table>
<tr><th>Azure Resource/Services</th><th>Access Management</th></tr>
<tr><td>Azure SQL Database</td><td>Service Principal/ SQL Authentication</td></tr>
<tr><td>Azure Synapse Analytics</td><td>Service Principal/ SQL Authentication</td></tr>
<tr><td>Azure Cosmos</td><td>Account Key</td></tr>
</table></td></tr>
</table>

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

Azure Service	IAM Configuration	
	Azure Blob Storage	Service Principal/ Account Key
	Azure Data Lake Storage Gen2	Service Principal/ Account Key
	Power BI	Service principal/ Delegated Authentication

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

9 Training Provider Authentication

TDIF (Trusted Digital Identity Framework) is a set of standards and guidelines developed by the Australian government to establish a framework for trusted digital identity services in Australia, which are implemented and provided within the Australian Government Digital Identity System (AGDIS). It outlines a set of principles and technical requirements for digital identity providers, such as the need for strong identity verification, secure data storage, and data protection measures. AGDIS is designed to ensure that digital identity services in Australia are interoperable, secure, and user-centric, while also respecting privacy and protecting personal information. It provides a common framework for digital identity services across the Australian public and private sectors, making it easier for individuals and businesses to access online services.

DEWR have identified the AGDIS-compliant Services Australia Identity Exchange as a digital identity service to be supported by STARS, along with any TDIF-compliant and Identity Exchange approved and configured Identity Providers (e.g., MyGovID, Aust Post Digital iD) and Attribute Providers (e.g., ATO RAM).

Training providers will authenticate in the Provider Portal using Identity Exchange approved credentials, using OAuth/OIDC protocols, integrated with Azure AD and Azure AD B2C.

AGDIS provides personal identity verification. For STARS, this means the identity will be for a single user, not the organisation. It will identify the user's relationship to an organisation, but all users within the same organisation will have a different Id, rather than sharing an Id for the organisation.

The following diagram shows and example high-level Auth flow context diagram using Identity Exchange:

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 920 of 1295

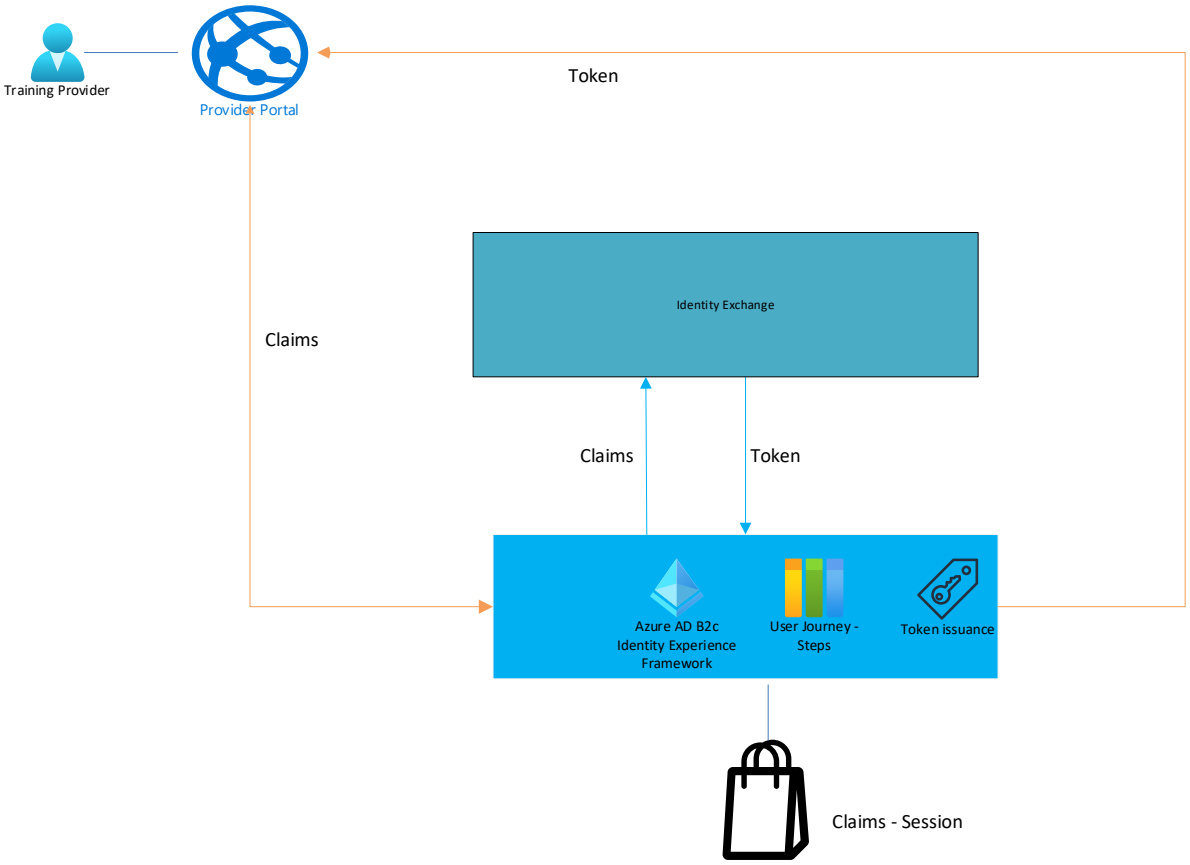


Figure 4: Example Auth flow context diagram with Identity Exchange – High Level

STARS will interoperate with Identity Exchange through Azure AD/B2C.

9.1.1 Identity Exchange Service

Training Providers authenticate via a TDIF compliant and AGDIS-approved identity provider. Identity Exchange service accepts login details/scopes via the OIDC Identity Exchange Authorization Endpoint over HTTPS to exchange identity details.

STARS will use the following 'Service Connection' pattern for connection to the AGDIS:

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 921 of 1295

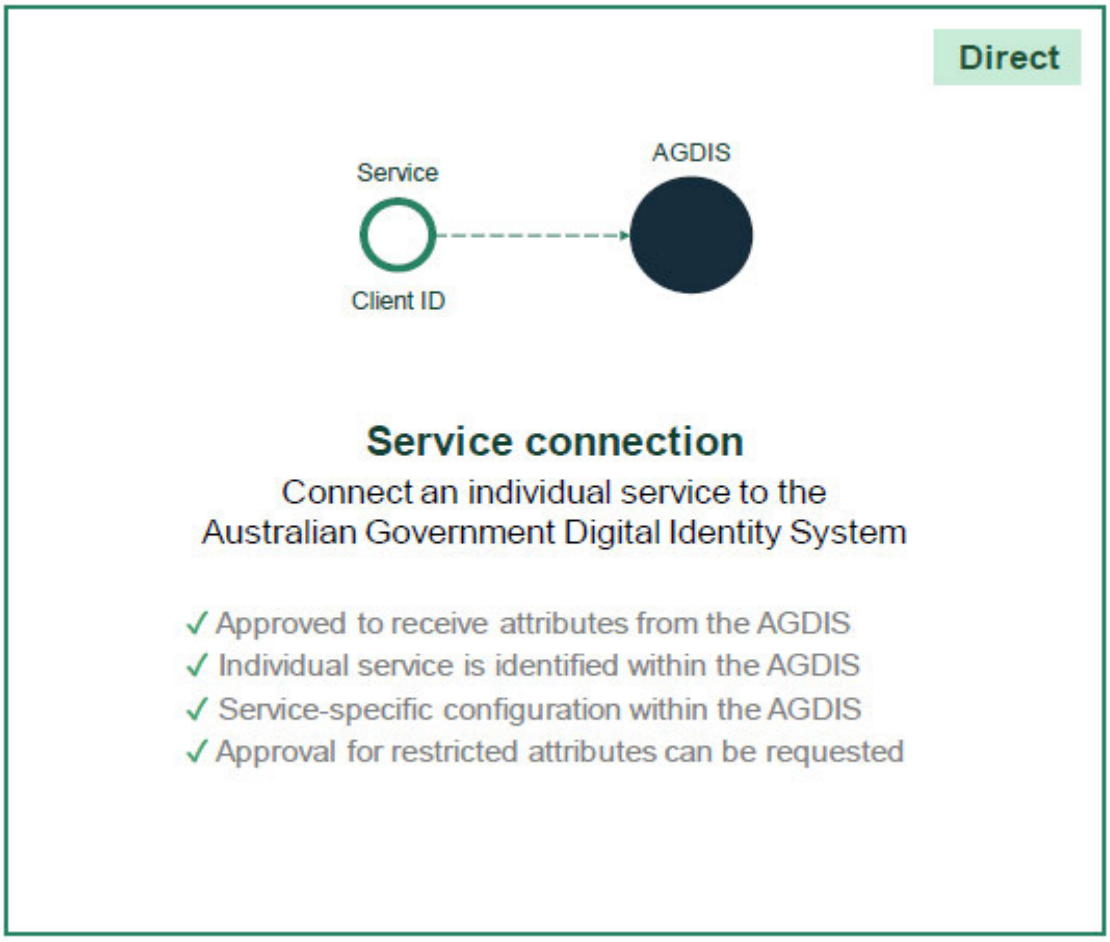


Figure 5: AGDIS Service Connection Pattern

With reference to the figure below, STARS acts as a 'Relying Service', delegating identity management to Identity Providers and relationships between parties to Attribute Service Providers:

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 922 of 1295

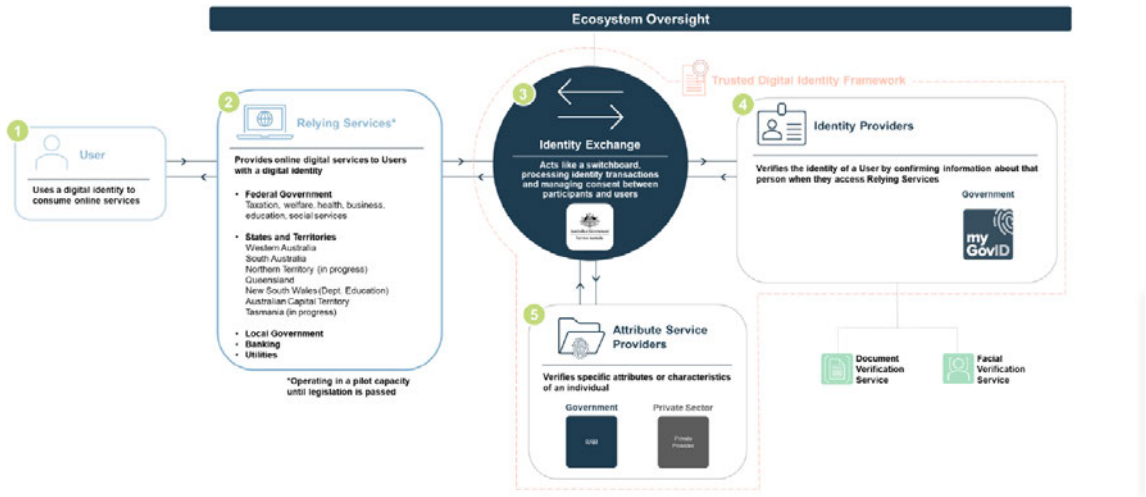


Figure 6: AGDIS ecosystem – high level

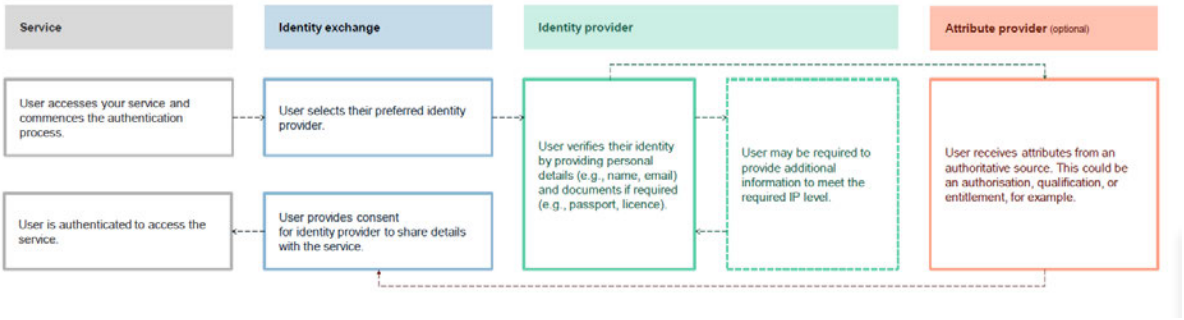


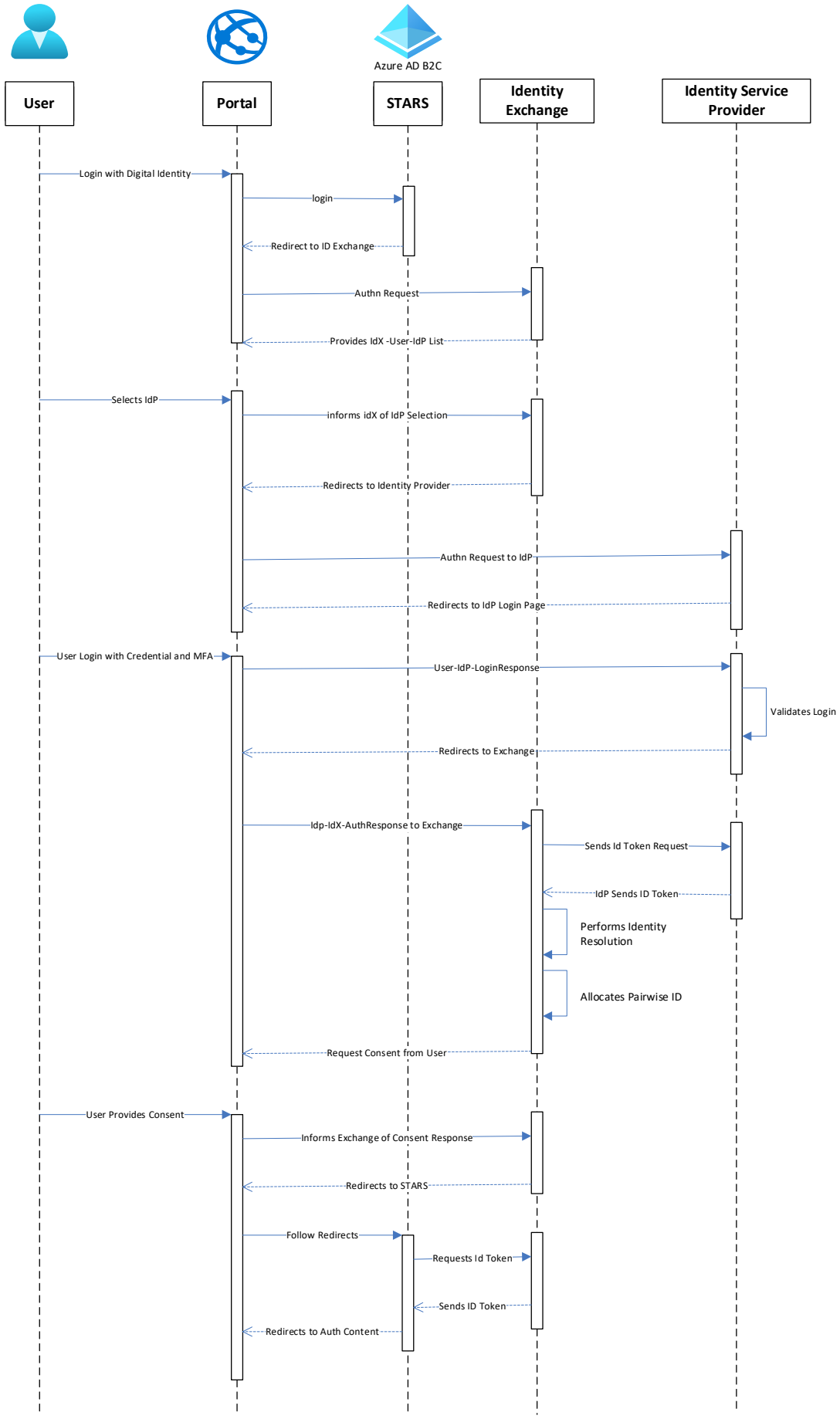
Figure 7: AGDIS IAM flow – high level

Figure 7: AGDIS IAM flow shows the general flow that will be followed by STARS Services (denoted as 'Service' above).

9.1.2 Azure B2C Mediated Sequence

A general sequence of Identity Verification with Identity Exchange and integration with Azure AD B2C acting as mediator is displayed below . The sequence of steps to obtain tokens and verify tokens follows similar patterns across all Azure-based STARS services and resources.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 923 of 1295



Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 924 of 1295

Figure 8: Azure B2C Identity Exchange sequence

9.1.3 M2M Setup for the Training Provider Client System

Training Providers can integrate their chosen SMS system and software to execute the functionality provided by the STARS API and Services.

To assert the SMS system's access within STARS, the Training Provider can obtain machine credentials from the Relationship Authorization Manager (RAM) for their SMS system, extract them to a public and private certificate, and provision the enterprise application within STARS.

The Software Development Kit (SDK) provided by DXC enables the SMS system's software to securely submit transactions and simplify the interaction with the STARS using certificates.

Below is the M2M Setup sequence for using the client credential flow between the Training Provider Client System and STARS.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 925 of 1295

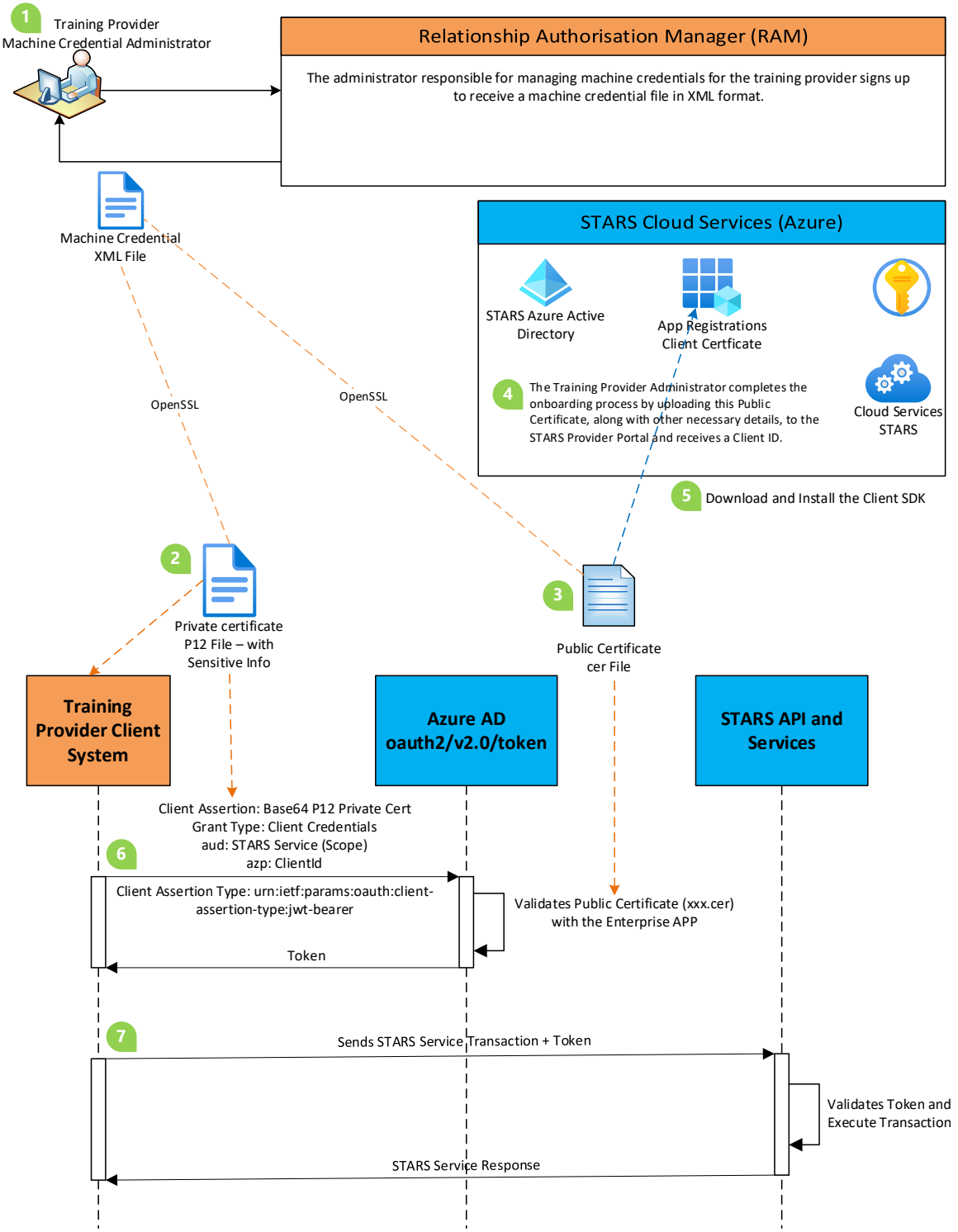


Figure 9: M2M Setup Sequence for the Training Provider Client System and STARS

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 926 of 1295

The STARS Client SDK Toolkit will simplify the interaction with STARS API and Services by providing functions for Step 6 and 7.

The steps for Training provider Client System to communicate with STARS are specified below:

1. Create Machine Credentials from the Relationship Authorization Manager (RAM).
   - Note down the password used by the Machine Credential Administrator.
   - Store the Machine Credential File in a secure location.
2. Extract the public certificate from the XML Machine Credential File.
   - Generate the xxx.cer public certificate.
3. Extract the private key.
   - Generate the xxx.p12 certificate.
4. Provision the RAM Public Certificate in STARS.
   - Use the Training Provider Portal to upload the xxx.cer public certificate produced in STEP 2.
   - The Training Provider Portal will create a new enterprise application with appropriate authorisations, store training organisation related details and provides the new Client ID.
   - Note down the Client ID.
5. STARS Client SDK
   Download and Install STARS Client SDK provided by DXC.
6. Obtain a token to communicate with the STARS API and Services.
   - To execute functionality within STARS, use a method provided by the SDK toolkit to obtain the OAuth Token.
   - This OAuth Token is short-lived and will expire after a certain period. It will be automatically refreshed to extend the validity period.

   For Example:

   Method Name:

   GetClientToken(string privateCert, string machineCredentialPassword, string clientId).

   Input parameters:

   - privateCert: The location of the xxx.p12 file generated in Step 3.
   - machineCredentialPassword: Machine Credential Password used by the Machine Credential Administrator in Step 1.
   - clientId: Provided by STARS in Step 4.

7. Access the STARS API and Services.

   Use the OAuth Token obtained in previous step to securely communicate with the STARS API and Services by calling any method within the SDK toolkit that requires the token.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 927 of 1295

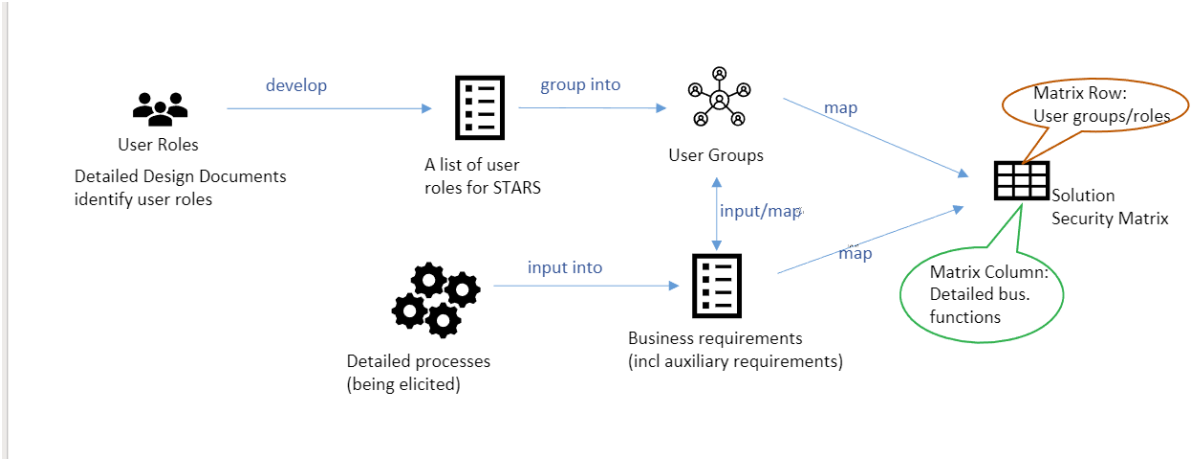
10 Solution Security Matrix

The Solution Security Matrix contains a detailed mapping of business functions and role-based access control. It informs the requirements and design of the IAM components of STARS

A development of Solution Security Matrix will be ongoing process that evolve with the detail design of other services and evolution of business requirements.

Security Matrix Process:

- Detailed design documents outline business processes in detail
- Detail business requirements to define and map user groups / Roles
- Fill and complete Solution Security Matrix



Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 928 of 1295

11 Requirements

11.1 Functional

FR No	Requirement Name	Description	Acceptance Criteria
FR072	FR072 - API authentication	The VDS system must allow API authentication by: • SMS and training organisations have different API IDs • SMS providers have an API token that allows them to submit data on behalf of their training organisations • training organisation can assign a token that to their SMS to validate and submit on their behalf	Given SMS access to STARS APIs And a recorded agreement for the SMS to submit data on behalf of the training organisation When an SMS makes requests to STARS APIs Then the SMS may be uniquely identified And the training organisation may be uniquely identified And API access will be determined by the training organisation's assigned access in conjunction with the SMS's assigned access Given SMS access to STARS APIs And no recorded agreement for the SMS to submit data on behalf of the training organisation When an SMS makes requests to STARS APIs Then an error response will be returned

11.2 Non-Functional

FR No	Requirement Name	Description
NFR005a	NFR005a - Digital Identity	The VDS System will fully support Digital Identity in accordance with the Trusted Digital Identity Framework (TDIF). • The VDS System will utilise the myGovID machine to machine authentication model.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Description
NFR005b	NFR005b - Partner identity federation	The VDS System enables users in each Partner organisation to authenticate using identity federation via single sign on (SSO). This includes: • AUSKey • Add specific requirement for each STA
NFR010f	NFR010f - Flexible authentication	The VDS System will support secure modern and flexible authentication and authorisations methods, including SAML federation with claims-based authorisation, OIDC etc.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

12 Appendices

Appendix 1. Acronyms

RAM	Relationship Authorisation Manager
RTO	Registered Training Organisation
STARS	Student Training and Activity Reporting System
TP	Training Provider
TDIF	Trusted Digital Identity Framework
SMS	Student Management System
SSO	Single Sign On
OIDC	OpenID Connect
STAs	State Training Authority
PoC	Proof of Concept
DEWR	Department of Employment and workplace Relations
VET	Vocation Education and Training
VDS	VET Data Streaming
Azure AD	Azure Active Directory
RBAC	Role Based Access Control
AAD	Azure Active Directory
IdP	Identity Provider

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 931 of 1295

Appendix 2. System Administration Related ISM Security Controls

Table 3 shows a subset of ISM controls relating to system administration. These controls are linked within DevOps to relevant functional requirements and associated STARS features and PBIs. The STARS solution features will be traceable through to relevant controls, supporting, for example, Authority to Operate documentation.

Table 3: VDS ISM Controls

DevOps ID	Title	Description
668	ISM-1388 - Administrative infrastructure	Only jump servers can communicate with assets requiring administrative activities to be performed.
669	ISM-1381 - Administrative infrastructure	Only privileged operating environments can communicate with jump servers.
670	ISM-1387 - Administrative infrastructure	Administrative activities are conducted through jump servers.
671	ISM-1386 - Administrative infrastructure	Network management traffic can only originate from administrative infrastructure.
672	ISM-1750 - Administrative infrastructure	Administrative infrastructure for critical servers, high-value servers and regular servers is segregated from each other.
673	ISM-1385 - Administrative infrastructure	Administrative infrastructure is segregated from the wider network.
1134	ISM-1587 - Annual reporting of system security status	System owners report the security status of each system to its authorising officer at least annually.
1021	ISM-0854 - Control of Australian systems	AUSTEO and AGAO data can only be accessed from systems under the sole control of the Australian Government that are located within facilities authorised by the Australian Government.
1022	ISM-0078 - Control of Australian systems	Systems processing, storing or communicating AUSTEO or AGAO data remain at all times under the control of an Australian national working for or on behalf of the Australian Government.
1023	ISM-1715 - Emergency access to systems	Break glass event logs are centrally stored and protected from unauthorised modification and deletion, monitored for signs of compromise, and actioned when cyber security events are detected.
1024	ISM-1613 - Emergency access to systems	Use of break glass accounts is logged.
1025	ISM-1615 - Emergency access to systems	Break glass accounts are tested after credentials are changed.
1026	ISM-1614 - Emergency access to systems	Break glass account credentials are changed by the account custodian after they are accessed by any other party.
1027	ISM-1612 - Emergency access to systems	Break glass accounts are only used for specific authorised activities.
1028	ISM-1611 - Emergency access to systems	Break glass accounts are only used when normal authentication processes cannot be used.
1029	ISM-1610 - Emergency access to systems	A method of emergency access to systems is documented and tested at least once when initially implemented and each time fundamental information technology infrastructure changes occur.
1042	ISM-1652 - Privileged access to systems	Privileged account and group change event logs are centrally stored and protected from unauthorised modification and deletion, monitored for signs of compromise, and actioned when cyber security events are detected.
1043	ISM-1651 - Privileged access to systems	Privileged access event logs are centrally stored and protected from unauthorised modification and deletion, monitored for signs of compromise, and actioned when cyber security events are detected.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

1044	ISM-1650 - Privileged access to systems	Changes to privileged accounts and groups are logged.
1045	ISM-1509 - Privileged access to systems	Use of privileged access is logged.
1046	ISM-0445 - Privileged access to systems	Privileged users are assigned a dedicated privileged account to be used solely for tasks requiring privileged access.
1047	ISM-1649 - Privileged access to systems	Just-in-time administration is used for administering systems and applications.
1048	ISM-1653 - Privileged access to systems	Privileged service accounts are prevented from accessing the internet, email and web services.
1049	ISM-1175 - Privileged access to systems	Privileged user accounts are prevented from accessing the internet, email and web services.
1050	ISM-1508 - Privileged access to systems	Privileged access to systems and applications is limited to only what is required for users and services to undertake their duties.
1051	ISM-1733 - Privileged access to systems	Requests for privileged access to data repositories are validated when first requested.
1052	ISM-1507 - Privileged access to systems	Requests for privileged access to systems and applications are validated when first requested.
1040	ISM-0447 - Privileged access to systems by foreign nationals	Foreign nationals, excluding seconded foreign nationals, do not have privileged access to systems that process, store or communicate AGAO data.
1041	ISM-0446 - Privileged access to systems by foreign nationals	Foreign nationals, including seconded foreign nationals, do not have privileged access to systems that process, store or communicate AUSTEO or REL data.
1135	ISM-1526 - Protecting systems and their resources	System owners monitor each system, and associated cyber threats, security risks and controls, on an ongoing basis.
1136	ISM-0027 - Protecting systems and their resources	System owners obtain authorisation to operate each system from its authorising officer based on the acceptance of the security risks associated with its operation.
1137	ISM-1636 - Protecting systems and their resources	System owners ensure controls for each system and its operating environment are assessed to determine if they have been implemented correctly and are operating as intended.
1138	ISM-1635 - Protecting systems and their resources	System owners implement controls for each system and its operating environment.
1139	ISM-1634 - Protecting systems and their resources	System owners select controls for each system and tailor them to achieve desired security objectives.
1140	ISM-1633 - Protecting systems and their resources	System owners determine the type, value and security objectives for each system based on an assessment of the impact if it were to be compromised.
1032	ISM-0407 - Recording authorisation for personnel to access systems	A secure record is maintained for the life of each system covering: • all personnel authorised to access the system, and their user identification • who provided authorisation for access • when access was granted • the level of access that was granted • when access, and the level of access, was last reviewed • when the level of access was changed, and to what extent (if applicable) • when access was withdrawn (if applicable).
674	ISM-1689 - Separate privileged operating environments	Privileged accounts (excluding local administrator accounts) cannot logon to unprivileged operating environments.
675	ISM-1688 - Separate privileged operating environments	Unprivileged accounts cannot logon to privileged operating environments.
676	ISM-1687 - Separate privileged operating environments	Privileged operating environments are not virtualised within unprivileged operating environments.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

677	ISM-1380 - Separate privileged operating environments	Privileged users use separate privileged and unprivileged operating environments.
1033	ISM-1734 - Suspension of access to systems	Privileged access to data repositories is automatically disabled after 12 months unless revalidated.
1034	ISM-1647 - Suspension of access to systems	Privileged access to systems and applications is automatically disabled after 12 months unless revalidated.
1035	ISM-1716 - Suspension of access to systems	Access to data repositories is automatically disabled after 45 days of inactivity.
1036	ISM-1648 - Suspension of access to systems	Privileged access to systems and applications is automatically disabled after 45 days of inactivity.
1037	ISM-1404 - Suspension of access to systems	Unprivileged access to systems and applications is automatically disabled after 45 days of inactivity.
1038	ISM-1591 - Suspension of access to systems	Access to systems, applications and data repositories is removed or suspended as soon as practicable when personnel are detected undertaking malicious activities.
1039	ISM-0430 - Suspension of access to systems	Access to systems, applications and data repositories is removed or suspended on the same day personnel no longer have a legitimate requirement for access.
1062	ISM-0435 - System access requirements	Personnel receive any necessary briefings before being granted access to a system and its resources.
1063	ISM-0434 - System access requirements	Personnel undergo appropriate employment screening and, where necessary, hold an appropriate security clearance before being granted access to a system and its resources.
1064	ISM-0432 - System access requirements	Access requirements for a system and its resources are documented in its system security plan.
678	ISM-1211 - System administration processes and procedures	System administrators document requirements for administrative activities, consider potential security impacts, obtain any necessary approvals, notify users of any disruptions or outages, and maintain system and security documentation.
679	ISM-0042 - System administration processes and procedures	System administration processes, and supporting system administration procedures, are developed and implemented.
1141	ISM-1525 - System ownership and oversight	System owners register each system with its authorising officer.
1142	ISM-1071 - System ownership and oversight	Each system has a designated system owner.
1030	ISM-0443 - Temporary access to systems	Temporary access is not granted to systems that process, store or communicate caveated or sensitive compartmented information.
1031	ISM-0441 - Temporary access to systems	When personnel are granted temporary access to a system, effective controls are put in place to restrict their access to only data required for them to undertake their duties.
1055	ISM-1714 - Unprivileged access to systems	Unprivileged access event logs are centrally stored and protected from unauthorised modification and deletion, monitored for signs of compromise, and actioned when cyber security events are detected.
1056	ISM-1566 - Unprivileged access to systems	Use of unprivileged access is logged.
1057	ISM-0405 - Unprivileged access to systems	Requests for unprivileged access to systems, applications and data repositories are validated when first requested.
1053	ISM-0411 - Unprivileged access to systems by foreign nationals	Foreign nationals, excluding seconded foreign nationals, do not have access to systems that process, store or communicate AGAO data unless effective controls are in place to ensure such data is not accessible to them.
1054	ISM-0409 - Unprivileged access to systems by foreign nationals	Foreign nationals, including seconded foreign nationals, do not have access to systems that process, store or communicate AUSTEO or REL data unless effective controls are in place to ensure such data is not accessible to them.
1058	ISM-0420 - User identification	Where a system processes, stores or communicates AUSTEO, AGAO or REL data, personnel who are foreign nationals are identified as such, including by their specific nationality.
1059	ISM-1583 - User identification	Personnel who are contractors are identified as such.
1060	ISM-0415 - User identification	The use of shared user accounts is strictly controlled, and personnel using such accounts are uniquely identifiable.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

1061	ISM-0414 - User identification	Personnel granted access to a system and its resources are uniquely identifiable.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

Document 14

Australian Government
Department of Employment and Workplace Relations

Detailed Design Document

Notification Service

DXC TECHNOLOGY

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 936 of 1295

1 Document Management

1.1 Document Control

Version	Change Description	Date
V.0.1	Draft Version	2023-02-01
V.0.2	General comment tidy up for improved presentation	2023-02-09
V.0.3	Added more Introduction. Updated diagrams for consistency. Additional changes from ongoing workshop feedback.	2023-02-10
V.0.4	Added missing relevant NFRs. Added missing notification types around notifying STAs of submissions. Updated terminology around "rule exemptions". Changes from feedback.	2023-02-21
V.0.5	Minor feedback changes (remove references to Release process). Updated developer implementation details to align with recent decisions (API Apps instead of Function Apps and Logic Apps, move to API and API Operations sections).	2023-03-01
V1.0	Submitted for Acceptance	2023-04-03
V2.0	Added All STARS System Components table and RTO/SMS engagement activities statement. Accepted	2023-04-28

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This is a living document and will be updated as required. If another component's design scope requires an update to this design document, then the appropriate change will be made and a new version issued.

1.2 Document Author

Role	Name
Design Lead	s 47F(1)

1.3 Document Reviewers

Role	Name

1.4 Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

1.5 Related Documents

Reference	Document
A	STARS Solution Requirements Document – V.1.3.docx

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 937 of 1295

Reference	Document
B	STARS Solution Design – V.1.0.docx
C	STARS Solution Design - Attachment 3 - API Design - V.1.0.docx
D	Notification Service Diagrams - V1.0.vsdx
E	STARS Solution Design - Attachment 1 - Collaboration Diagrams - V.1.0.vsdx

1.6 All STARS System Components

This list outlines the high-level service dependencies with **Notification Service**. This is a reflection of what is known at the time of Acceptance and may be subject to change throughout the design process.

Green = This Design Document Grey = Other Service Dependencies

STARS Service
Azure Infrastructure and Networking
Identity & Access Management (IAM)
Event Service
Logging & Audit Service
Master Data Service
Reference Data Service
Data Governance Infrastructure
Data Quality Framework
Notification Service
Submission Service
Validation Service
Rules Service
Provider Service
Mastering Service
Downstream Consumer ETL Pipeline Services
Release Service
Downstream Consumer Reporting & Analytics
R & A Provisioning Service
STA Batch Submission Services
STA Gateway Services

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 938 of 1295

STARS Service
USI Transcript Service
STA Bulk Data Consumption Service
SMS SDK Service

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 939 of 1295

1.7 Open Design Items

This section describes items arising from the design process that remain unresolved. This document is accepted on the proviso that an updated version of this design is provided after the below items have been completed.

#	Open Item / Gap	Reason can't be resolved at initial acceptance	DevOps Action ID	Action Description	Owner	Estimated Target Date	Impact Date	Build Progress Impact (if any)
1	Specific data provision rules around the due date of training activity.	Cannot finalise the design until the specific rules are provided. The DPR needs to be confirmed to provide context on the requirement.	3962	Action: Requirement Clarification and confirm DPRs to provide full context on the requirement (VDS PMO) Action: Update the design where required to reflect the information provided (DXC)	VDS PMO - s 47F(1) DXC	28/4/2023	2/6/2023	Expected to be just a configuration, we can assume a default value initially and update it at a future point
2	Information required on use cases regarding nil training activity submissions.	We do not have the use case information to confirm how it needs to be implemented	3890 1882	Action: PMO/Policy to provide further context and clarification on the requirement/use cases regarding nil training activity requirements Action: Update Notification Service where required	VDS PMO - s 47F(1) DXC	28/4/2023	28/4/2023	Additional Functionality (not rework) – DXC will leave any build items related to this function until details are available

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

1 Document Management 2
1.1 Document Control 2
1.2 Document Author 2
1.3 Document Reviewers 2
1.4 Document Approval 2
1.5 Related Documents 2
1.6 All STARS System Components 3
1.7 Open Design Items 5
2 Introduction 8
2.1 Purpose 8
2.2 Intended Audience 9
3 Scope and Exclusions 9
3.1 In Scope 9
3.2 Out of Scope 10
4 Design Decisions 10
5 Functional Requirements 11
6 Non-functional requirements 17
7 Feature Overview 19
7.1 Notification Categories 20
7.2 Notification Types 21
7.3 Data and Templating 26
7.4 Rule-based Reminders / Scheduled Notifications 27
7.5 User Configuration 27
8 API Design 28
8.1 Notification Config 28
8.2 Notification Event DB 29
8.3 Notification DB 29
8.4 Notification API 38
8.5 Notification API Operations 64
8.6 Notification Events 75
8.7 Notification Messaging 75
9 Technical Design Patterns 76
10 Other Service Dependencies 79
10.1 Rules Service 79
10.2 Submission Service 79

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 941 of 1295

10.3 Master Data Service 79
10.4 Provider Service 79
10.5 Identity and Access Management Services 79
11 Access Management 80
12 Appendices 81
12.1 STARS Solution Design Diagrams 81
12.2 Example Notification Contents 82
13 Acronyms 84
14 Glossary 84
15 Terminology 84

Figure 1: STARS Support Services 19
Figure 2: Notification Sources (D:4) 21
Figure 3: Template process (D:1) 26
Figure 4: Schedule Process (D:1) 27
Figure 5: User Configuration Overview (D:1) 27
Figure 6: API Design Overview (D:1) 28
Figure 9: Send User Notification (D:5) 64
Figure 10: Send User Notification from Schedule (D:6) 65
Figure 12: Get Broadcast Notification List (D:7) 66
Figure 13: Create Entity (General) (D:8) 70
Figure 14: Update Entity (General) (D:9) 71
Figure 15: Delete Entity (General) (D:10) 72
Figure 16: Get List of Entities (General) (D:11) 73
Figure 17: Get Entity by ID (General) (D:12) 74
Figure 18: Appendix:Reference Notifications Service (E:12) 81
Figure 19: Appendix: Notification Service Implementation 81

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 942 of 1295

2 Introduction

The purpose of this document is to capture the detailed design requirements for the STARS Notification service solution.

The Notification service manages and distributes Notifications raised by any of the other services. Notifications are raised as events through the STARS Event Service or via API calls. As a Support Service, it provides a "cross-cutting" system-wide service to support the operation of most other services in STARS.

The service will be responsible for sending out notifications using the following methods:

- Emails to individual people
- Website announcement messages
- Providing an API for SMS APIs to retrieve relevant notifications from

> Note: This is a living document that will be updated as designs for various other STARS services are developed and evolve. Accordingly, this document will evolve to reflect and document notification components as the STARS solution build progresses.

The design is informed by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework, which provide a comprehensive set of best practices and guidance for designing, deploying, and operating secure, high-performing, and resilient cloud infrastructure and applications.

2.1 Purpose

The purpose of this detailed design document is to provide a technical description of the Azure notification components of the STARS solution. The document is intended for technical stakeholders, including architects, software engineers, administrators, and operations teams, who are responsible for designing, developing, deploying, and operating the STARS solution in Azure.

The document also serves as a reference for ongoing management and evolution of the STARS solution and to ensure that the STARS solution is deployed and operated in accordance with the best practices and guidance provided by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework.

The document is intended to be a living document, updated as the STARS solution evolves and changes over time. It provides a clear and concise understanding of the Azure notification components and their configurations, helping to ensure the STARS solution is deployed, operated, and managed in a consistent and efficient manner.

With respect to expected RTO/SMS engagement activities, it is DXC's understanding based on the current requirements and solution design, noting that DXC have not yet conducted detailed consultation with the RTO/SMS Providers, that there is no component in this Deliverable that directly impacts RTO/SMSs. All direct impacts to RTO/SMSs are dealt with in the Provider Portal, Provider Service and/or SMS SDK Service as understood and agreed previously.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 943 of 1295

2.2 Intended Audience

This detailed design document provides technical information and guidance to help technical stakeholders understand the Azure notification components of the STARS solution and how they integrate with each other. The document is expected to be used by technical stakeholders who have a good understanding of cloud computing concepts and are experienced in the Azure platform.

This document is not intended for non-technical readers, nor for readers who are not experienced in the design and configuration of Azure-based solutions; informational and educational resources regarding Azure are readily available at https://azure.microsoft.com

The primary audience for this document includes:

- Program technical staff: This includes architects, software engineers, administrators, and operations teams within the program responsible for the STARS solution.
- External technical stakeholders: This includes technical staff from other federal, state, and territory governments agencies, including VDS embedded team members.
- ICT Technical Staff: This includes a variety of technical roles involved during the VDS program and post-go live, including network engineers, systems administrators, database administrators, security engineers, and DevOps engineers.

3 Scope and Exclusions

3.1 In Scope

Notifications will be sent to email addresses. They will also appear as portal announcement banners. Relevant notifications can be retrieved via the Notification API (including by any SMS API).

Notifications fall under a category that individual users may choose to opt out of i.e., unsubscribe from. Some of these categories will be considered process-critical and can't be opted out of. If a notification category is opted out of, emails won't be sent, and banners won't be shown in the relevant portals.

System Operators will be responsible for notification contents and updating templates as required.

Users will be able to choose how often they will receive notifications via email – one email per notification, or a daily or weekly summary email.

The full list of notification categories and notification types can be found in the Feature Overview section below.

Note: Any references to validation errors / warnings / messages are relevant only for asynchronous validation. The Validation Service will be responsible for producing the messages used in these notifications. The Notification Service will only provide a notification that the validation results are now available.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 944 of 1295

3.2 Out of Scope

Infrastructure-related alerts (e.g., API app exceptions) are not part of this service.

When the Validation Service produces synchronous results, notifications will not be sent out. The Notifications Service will not be involved in synchronous validation results.

The Notification Service doesn't define every notification that gets sent out. General notifications can be sent from other services that use a generic Notifications template.

Notifications are not considered State-dependent, and so can be stored together without separation. When, for example. a list of notifications is requested, there will be no issue with returning an unfiltered list to an authorised user.

The Notification Management Portal is out of scope for this document. The services to support the portal will be described in this document, but any Management Portal interactions will be the responsibility of the Notification Management Portal Detailed Design Document.

4 Design Decisions

STARS will have a single service that consolidates all notification sending. Notifications should be sent automatically where applicable.

This service will be accessible by all other services to send general notifications as required. Lists of relevant notifications can be retrieved. See the section 8.4 Notification API for more details.

The service will provide the "backend" for the Notification Management Portal via API endpoints. This portal will be used to manage and maintain notification messages, as well as user notification preferences.

This service will allow STARS System Operators to modify notification type templates. It will also allow operators to view the history and status of notifications.

For a broad overview, see Reference B: STARS Solution Design. The high-level diagrams relevant to the Notifications Service are also included in the Appendices section of this document.

Note that FR055b and FR141 refer to "exemption requests". In this design, these are referred to as "validation outcome review requests".

Note that FR055a refers to "relevant users". In this case, this will be the users of the training provider that created the validation outcome review request.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 945 of 1295

5 Functional Requirements

Note: Some of these FRs are "owned" by this service, and others are owned by different services but are still relevant for referencing. "Owned" functional requirements are highlighted in green.

Table 1: Functional Requirements

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR006	(related) National rules error notification	See 7.2 Notification Types	Send error notification to Provider and prompt to resubmit The VDS system must notify a training organisation of the VDS National/Reference Data validation result (errors/warnings) in near real-time via: • Student Management System • Adaptive Web User Interface	Given Training Activity submission When a Training Organisation submits data via the SMS API Then the submission is validated against the relevant internal validation rules And the response via the API will include all errors, warnings and notifications relevant to the data validation result Given Training Activity submission When a Training Organisation submits data via the Web Interface Then the submission is validated against the relevant internal validation rules And the response in the Web Interface will include all errors, warnings and messages relevant to the data validation result
FR010	(related) Reject data	See 7.2 Notification Types	The VDS system must accept records that have passed validation checks, and reject only records that have failed validation checks, during submission. • The VDS system will notify validation errors/warnings/messages in near real time at the time of submission, where possible	-

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR011	(related) Submit data	See 7.4 Rule-based Reminders / Scheduled Notifications	A training organisation must authorise the submission of data into the VDS system. • If the time limit for submission has been reached, the VDS system will notify the Provider if data has completed a validation check, but has not been authorised for submission.	Given Training Organisation has entered or uploaded data into STARS And has not submitted that data When the time limit for a submission has been reached Then STARS will notify the Training Organisation
FR014c	(related) Release data to Master DB	See 7.4 Rule-based Reminders / Scheduled Notifications	A STA/Funder must be able to authorise the release of a successfully validated and quality assured dataset into the Master Database. • If the time limit for release has been reached, the VDS system will notify the STA/Funder	Given submissions undergoing data quality checks by an STA/Funder When the STA/Funder notifies of failed data quality checks Then STARS will notify the submitter of the data And prevent the submission from being mastered Given submissions undergoing data quality checks by an STA/Funder And the submission is not configured to auto-release When the time limit for release has been reached Then STARS will notify the STA/Funder
FR016b	(related) Output Edits Action	See 7.2 Notification Types	The VDS system must perform output edits on the aggregated data in the Master Database when the system detects data inconsistency a) Identify the inconsistency b) Notify the Provider (fee for service) and the STA/Funder (funded) of the inconsistency for resolution through an agreed process	Given inconsistencies in mastered data When data quality checks are performed Then STARS will identify the inconsistencies And notify the provider of the inconsistent data
FR036	Manage data in Master DB	See 7.2 Notification Types	The VDS System must provide the ability to view and trigger a request for update to the authorised source for data committed to the Master Database, in accordance with FR032a and FR032b.	Given a request to have mastered data updated When the request is confirmed Then STARS will notify the source of the data And provide details from the request for update

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR040	Notifications	See 7.5 User Configuration	The VDS system must have the ability to add, update and change the notifications to users (error, validation, output edits, reminders): • Content of the notification • Delivery type of the notification • Frequency of the notification	Given error, validation, data quality check, and reminder notifications When the system operator is managing notifications Then STARS shall enable the system operator to customise the content, delivery type and frequency of these notifications
FR055a	(related) Business rules	See 7.2 Notification Types	The VDS system must have capability to create, configure, edit, delete business rules and capturing justifications for these actions • The VDS system will have the ability to add or update business rules that trigger notifications. • The VDS system manages a central register of rules, including a unique ID for each discrete rule, and classification of the rule to help define the rule type, to enable the easy identification of validation outcomes and analysis when providing support to training organisations to submit valid data, and when troubleshooting rule issues with the System Operator. • The VDS system will advise users of all additions and updates to business rules. • The VDS sytem will support users to download business rules and updates.	Given validation rules When a request to create a validation rule is received Then STARS will initiate a process to have that rule created And the rule will be given a unique ID And the rule will be given a Rule Type And the rule will include information to assist troubleshoot failures And STARS will notify appropriate users of the rule's creation And STARS will capture justifications for the new rule Given validation rules When a request to update the logic of a rule is received Then STARS will initiate a process to have that rule logic updated And STARS will notify appropriate users of the rule's modification And STARS will capture justifications for the modification

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR055b	(related) Updating business rules	See 7.2 Notification Types	The VDS system must allow business rules to be easily updated to reflect feedback from training organisations who identify rules that force submission of inaccurate data • The VDS system will have a process for data providers to report problems with the VIS that restrict high quality/accurate data.	Given a rule review When the outcome requires an update to the rule Then STARS will update the rule And STARS will notify the provider of the outcome And the Review will be closed Given a rule review When the outcome requires does not require an update Then STARS will notify the provider of the outcome And the Review will be closed Given a rule review When the outcome requires an exemption Then STARS will record the exemption And STARS will notify the provider of the outcome And the Review will be closed
FR056a	Notifications for due or overdue records	See 7.4 Rule-based Reminders / Scheduled Notifications	The VDS system must identify records that are due or overdue to provide a notification to the training organisation and other users where such an entry is missing from the submitted data (e.g. Open Commitment data in IVETS).	Given STARS has notifications enabled And time-based rules are defined to detect overdue records When a record is due or overdue Then STARS will send a notification to the Training Organisation/STA users
FR056b	Notifications to user groups	See 7.2 Notification Types	The VDS system will e3.1nable configuration of notification rules to notify specific user groups (e.g. a STA/Funder or other users) based on the validation outcome.	Given STARS has notifications enabled When a validation rule is run Then notification will be sent to the stated user group, based on the validation rule configuration
FR056c	Message exchange	See 7.3 Data and Templating	The VDS System must enable message exchange for errors, warnings, system notifications, etc.	Given STARS has human readable text for each error warning or system notification When an authorised user wants to customise the human readable text Then STARS will enable the customisation of the error message

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR056d	Notifications to training organisation	See 7.2 Notification Types	The VDS system must notify the user when data entered is pending validation and notify the user again when the data has been validated.	Given to Training Organisation submitting training activity When validation has been initiated Then a status notification will be provided
FR073	(related) Reporting requirements where no activity has occurred	See 7.4 Rule-based Reminders / Scheduled Notifications	The VDS system must provide a mechanism to identify when data hasn't been submitted/updated within an expected timeframe.	Given STARS has notifications enabled When a record is due or overdue Then STARS will send a notification to the Training Organisation/STA users
FR091	(related) Exceptional Data Handling (STA User handling the request for review)	See 7.2 Notification Types – Validation Outcome Reviews will be notified to STAs (for their funded data), so they can retrieve such a list an aid with resolution.	As a STA User, I want to be able to retrieve the list of records and errors/warnings that were requested for review by training organisation so that I can identify the issue, resolve the problem and re-process the data	Given an active review for a validation error When a resubmission is received for the source submission of the review Then STARS will resolve the review And notify the initiator of the review And record details of the resolution Given an active review for a validation error And a submission that was the source of the review And no re-submission related to the submission has been received When the review is resolved Then STARS will revalidate the submission And proceed the submission through the pipeline if it passes validation And notify the provider if validation fails
FR118	Notifications for Validations Completed during Transition Period	See 7.2 Notification Types	An email must be sent to the training organisation (person who initiated the upload) when validations are complete for a particular upload. The message must be sent to the email address recorded against the user's access profile.	Given the Training Organisation submitting training activity data When they have successfully uploaded the validated data Then a notification will be sent to the submitting user

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Acceptance Criteria
FR119	(related) Notifications for Changes in Fees during Transition Period	See 7.2 Notification Types	The training organisation must be informed of all changes in student fees. (All fee changes for students attending their training organisation under Smart and Skilled). Note: The training organisation must be alerted to the fee change at the time the fee is known to have changed. (Not at payment authorisation time. This is because fees can change when no payment is due).	Clarification required
FR135	Notifications for pending actions	See 7.4 Rule-based Reminders / Scheduled Notifications	The VDS system must be able to send notifications to training organisations when there are pending actions required. For example, the training organisation must receive a reminder X days after valid data has been received for a training organisation, and no subsequent valid files have been uploaded. This email does not detail individual students. The training organisation must receive a reminder X. X depends on the Program Type and relevant contracts with training organisations	Given time-based rules are configured in STARS When the conditions for the rule are met Then STARS will action the result of the rule And this may include notifications to contacts for the data the rule has run against
FR141	(related) Access Control	See 7.2 Notification Types	The VDS Solution must allow for validations functionality to be disabled (put on hold) for individual training organisations at any point in time with specific criteria (e.g. specified validation rules, time turned off, specific training organisations/program). This is to support special exemption needs on contracted training organisations. Temporarily disabling validations must be approved and the system should capture the associated metadata (date & time; approver; requester; justification; associated records, threshold changes, related document links).	Given a rule exemption review When the outcome of the review is decided Then STARS will record the review outcome And notify the initiator of the review as to the outcome

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

6 Non-functional requirements

Table 2: Non-functional requirements

FR No	Requirement Name	Traceability	Description / Comments	Comments
NFR002b	Downtime notification	See 7.2 Notification Types	The VDS System will have a notification process to notify all system users of any scheduled or unscheduled downtime or outage, including to address critical issues (e.g. denial of service attack). For scheduled downtime, users will be notified three (3) business days prior to the downtime. For unscheduled downtime, users will be notified as soon as the outage is identified.	Note: "Unscheduled downtime" will only be notified of if the notifications service itself isn't affected. Unscheduled downtime notifications may also appear after the service is back up, as post-mortem information. NCVER will be in charge of sending out unscheduled downtime notifications.
NFR002c	Availability – Scheduled downtime	NFR002c - Scheduled downtime	The VDS System will have an agreed scheduled downtime or outage (if applicable) within business hours and/or outside business hours	
NFR002f	Availability	NFR002f - Outage window for major releases	The VDS System will use a single planned outage window of five (5) hours allowed in a three (3) month period for major releases. The planned outage will allow for a full system outage.	
NFR002g	Availability	NFR002g - Outage window for minor releases	The VDS System will use a single planned outage window of one (1) hour allowed in a one (1) month period for minor releases. The planned outage will allow for a full system outage.	
NFR002h	Availability	NFR002h - Outage window for emergency changes	The VDS System will use an outage window for emergency changes, targeting installation within a 30 minute window. The outage will be scheduled between 2:00 am and 4:00 am AEST on any day. The planned outage will allow for a full system outage.	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

FR No	Requirement Name	Traceability	Description / Comments	Comments
NFR021i	Notifications	See 7.5 User Configuration	The VDS System enables users to customise the frequency of system notifications.	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

7 Feature Overview

The Notification Service is a Support Service in STARS responsible for the configuration and delivery of Notifications to STARS users and SMS applications. Support Services provide 'cross-cutting' system-wide services to support the operation of other services in STARS.

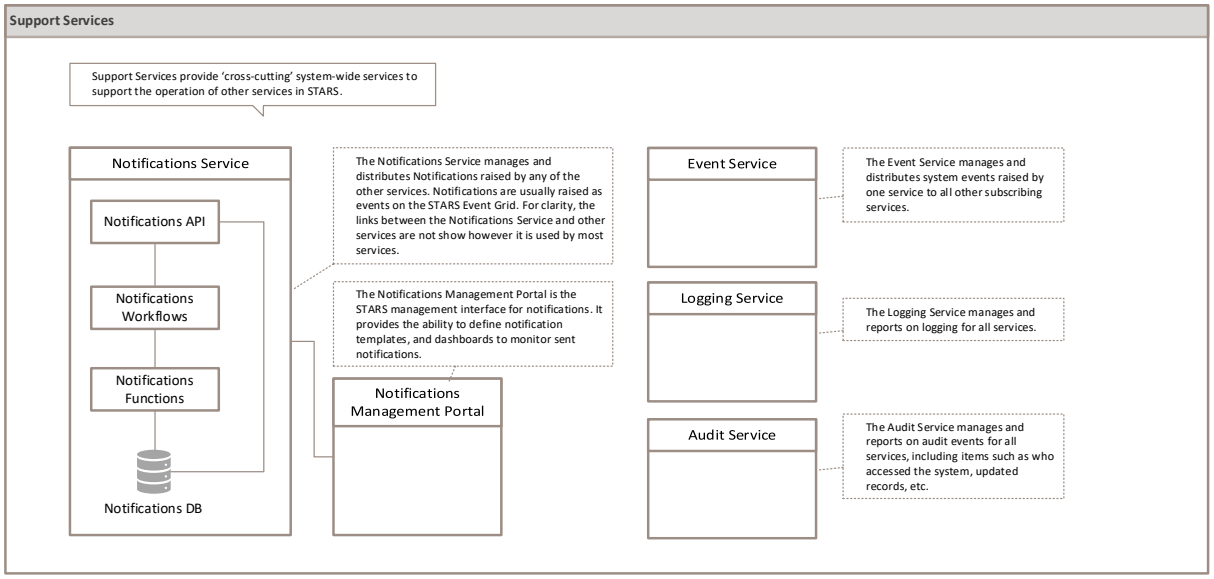


Figure 1: STARS Support Services

Each STARS Service that raises Notifications will be responsible for initiating those Notifications. The Notification Service will provide an API and operations to facilitate delivery of these Notifications to the appropriate audience, through the appropriate channel(s). The API presented also allows the Provider Service, and other Management Portals to present notifications in user interfaces, and to Student Management Systems.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 954 of 1295

7.1 Notification Categories

Notifications are divided into distinct categories. Users can choose to subscribe / unsubscribe from notifications at the category level.

Table 3: Notification Categories

Notification Category	Description	Can be unsubscribed from?	Can receive in summary email instead?
Data submission reminders	Reminder notifications sent when data is due or overdue for submission.	No	No
Validation result updates	Notifications raised when asynchronous validation results become available.	No	Yes
Validation outcome review request updates	Notifications raised in relation to validation outcome review requests.	No	Yes
Changes made to submitted data	Notifications sent to TPs when submitted data is updated by non-TPs.	No	Yes
Master data changes	Raised when master data are changed.	Yes	Yes
Reference data changes	Raised when reference data are changed.	Yes	Yes
Business rule changes	Raised when business rules are changed.	Yes	Yes
Student fee changes	Raised when student fees are changed.	Yes	Yes
Downtime notifications	Notifications relevant to scheduled and unscheduled STARS maintenance / downtime.	No	No
TP actions	Notifications sent to STAs based on activity by TPs.	No	Yes

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 955 of 1295

7.2 Notification Types

The following diagram and table describe all known notification types, based on the requirements listed above.

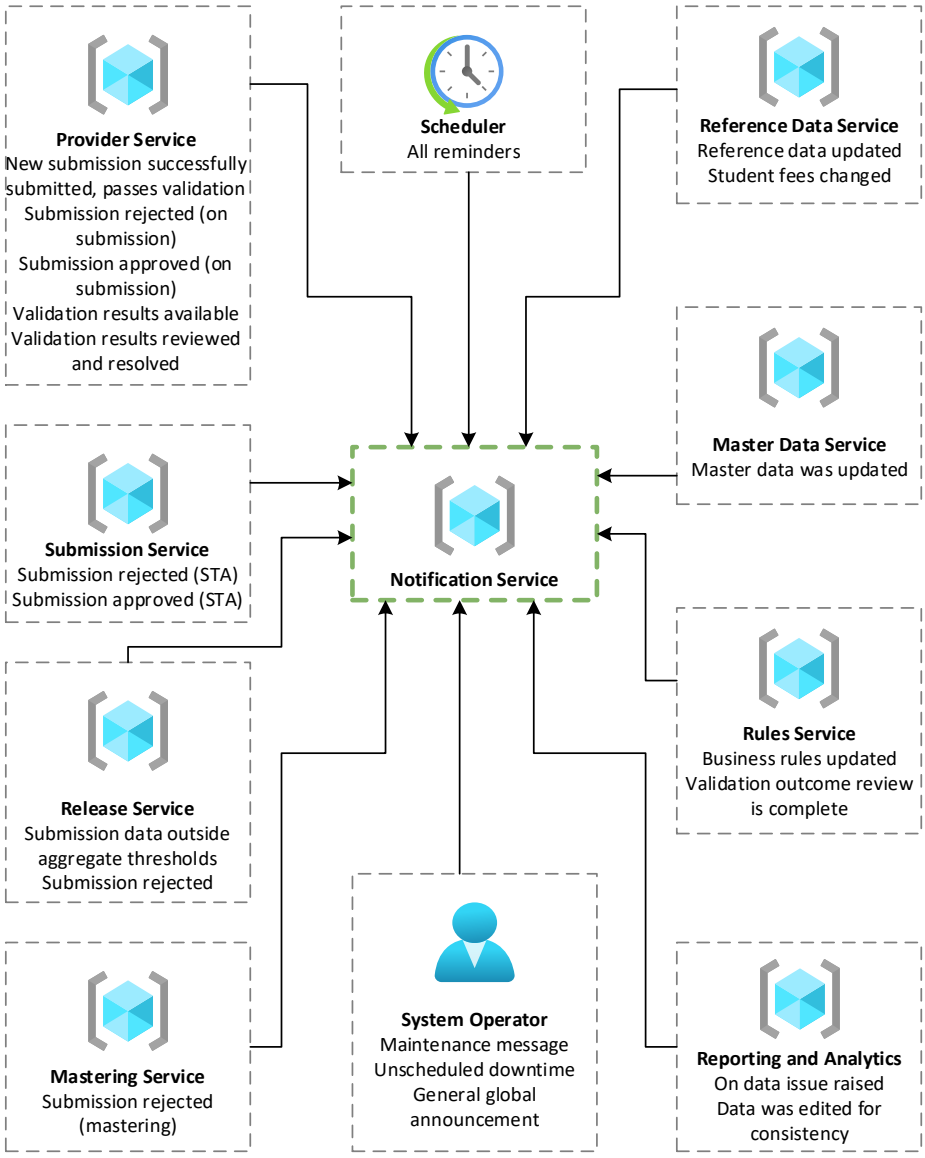


Figure 2: Notification Sources (D:4)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 956 of 1295

Table 4: Notification Types

Notification Type	Category	Source	Sent to	Relevant Requirements	Template Values	Comments
New submission successfully submitted and packets pass validation	TP actions	Submission Service	STAs	-	STA Name, TP Name	
Submission packets were rejected (on submission)	Validation result updates	Submission Service	Data provider / source of data (TPs)	FR014c	TP Name	Raised by the Submission service.
Submission packets were approved (on submission)	Validation result updates	Submission Service				NOTE: NSW will require a state-specific template for this notification.
Submission packets were rejected (on mastering)	Validation result updates	Mastering Service	Data provider / source of data (TPs)	FR014c	TP Name	Raised by the Mastering service.
On data issue raised	Master data changes	Reporting and Analytics	Data provider / source of data (TP), STA(s)	FR036	TP Name	Raised by STARS administrators as part of Reporting, Analytics and Aggregate Data Quality Checks.
Reminder: Data soon due for submission	Data submission reminders	(Scheduled)	Training Providers	FR056a, FR135	TP Name	Fixed timing value configured in the associated Rule Service rule.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

Notification Type	Category	Source	Sent to	Relevant Requirements	Template Values	Comments
Reminder: Data is overdue for submission	Data submission reminders	(Scheduled)	Training Providers	FR056a, FR135	TP Name	Sent on a scheduled basis to TPs and STAs when data submission is overdue. Fixed timing value configured in the associated Rule Service rule. NOTE: NSW will require a state-specific template for this notification.
Reminder: Data uploaded but not submitted	Data submission reminders	(Scheduled)	Training Providers	FR011, FR073	TP Name	Fixed timing value configured in the associated Rule Service rule.
Reminder: Data quality checks failed	Validation result updates	(Scheduled)	Training Providers	FR014c	TP Name	Fixed timing value configured in the associated Rule Service rule.
Reminder: Submission packets were rejected	Validation result updates	(Scheduled)	Training Providers	FR014c	TP Name	Fixed timing value configured in the associated Rule Service rule. NOTE: NSW will require a state-specific template for this notification.
Reminder: Submission has expired	Validation result updates	(Scheduled)	Training Providers	FR014c	TP Name	Fixed timing value configured in the associated Rule Service rule.NOTE: NSW will require a state-specific template for this notification.
Reminder: Pending warnings in submission	Validation result updates	(Scheduled)	Training Providers	-	TP Name	Data may be accepted with warnings but the TP will need to be reminded of outstanding warnings that have not been rectified. Fixed timing value configured in the associated Rule Service rule.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

Notification Type	Category	Source	Sent to	Relevant Requirements	Template Values	Comments
Data was edited for consistency	Changes made to submitted data	Reporting and Analytics	Data provider / source of data (TP)	FR016b	TP Name	The VDS system must perform output edits on the aggregated data in the Master Database when the system detects data inconsistency. Details will be included in the Reporting and Analytics design document.
Reference data was updated	Reference data changes	Reference Data Service	STAs, Training Providers, NCVER	FR036		Raised by the Reference Data Service.
Master data was updated	Master data changes	Master Data Service	STAs, regulators, NCVER			Only for a subset of fields. For example, the Enrolment changes from FFS to Funded, but not if the phone number changes.
Validation results are available (asynchronous results)	Validation result updates	Provider Service	Data submitter (TP)	FR006, FR040, FR056c, FR056d	Submitter Name	
Validation results are reviewed and resolved	Validation result updates	Provider Service	Data submitter (TP)	FR091, FR118	Submitter Name	
Business rules updated	Business rule changes	Rules Service	Portal users	FR055a, FR055b	N/A	
Student fees changed	Student fee changes	Reference Data Service	Training Providers	FR119	N/A	NOTE: NSW will require a state-specific template for this notification.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

Notification Type	Category	Source	Sent to	Relevant Requirements	Template Values	Comments
Validation outcome review is complete	Validation outcome review request updates	Rules Service	Training Providers	FR141	Submitter Name	Sent to all the relevant TP's users.
Maintenance message	Downtime notifications	(System Operator)	Portal users/SMS	NFR002b	DateStart, DateEnd	Used to communicate upcoming scheduled maintenance. The DateStart and DateEnd values describe the scheduled maintenance period (period that this broadcast is shown to users is set separately to templating values). See NFR002 Availability requirements for details on when this maintenance message would be used.
Unscheduled Downtime message	Downtime notifications	(System Operator)	Portal users/SMS	NFR002b	ExtraDowntimeInfo	Used to communicate ongoing unscheduled downtime of other systems, and post-mortem details of past unscheduled downtime.
General global announcement	Downtime notifications	(System Operator)	Portal users	NFR002b	CustomAnnouncement	An empty template for creating simple one-off notifications for a general purpose. Useful when a template isn't required for a simple notification that another service will provide the contents for.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

7.3 Data and Templating

Each notification type will have a corresponding template which defines the common content for that type. This template will be used to produce the final "plain English" notifications shown to users.

Some notifications will require slight variations depending on the receiving party, e.g. the relevant TP's name in reminder emails. A mechanism is required for marking parts of the template to swap out with relevant data on a per-user basis. When a request to send a notification is received (either via the API or via Event Grid), the request will also send the required template data for that notification. If the values aren't provided, the notifications will not be sent and an error will be logged.

When a timed reminder notification needs to be sent, the Notification Service will be in charge of providing the template data.

For broadcast notifications (where per-user notifications are never stored), this templating process will run when notifications are retrieved from the API, using the requesting user as the context for setting the notification content.

The process of replacing parts of the template with required data is detailed in the Function Apps section.

Some template values will always be considered available. This will include NotificationSendDate and Severity.

Any new Notification templates will be added in a Non-Production environment and tested before being applied to a Production environment. This will allow appropriate testing, including Accessibility testing, to be conducted.

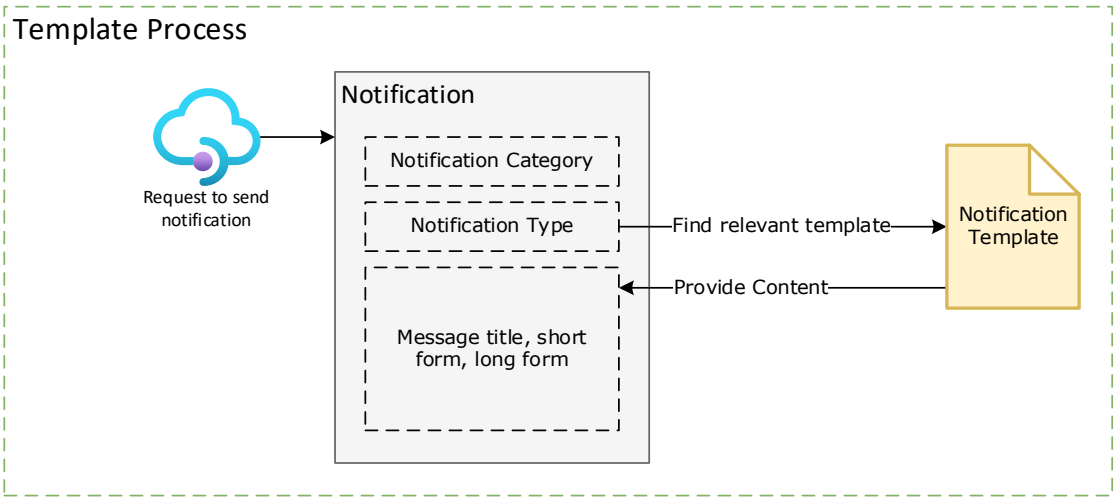


Figure 3: Template process (D:1)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 961 of 1295

7.4 Rule-based Reminders / Scheduled Notifications

Some notifications are sent automatically based on a schedule. These are reminders for upcoming actions or overdue actions. For example, if data submission from a TP for this month is due in 7 days and the submission hasn't come through yet, the TP should be notified.

The Rules Service will define notification-related rules for determining if a notification is required or not. This will include checking to see if a required reminder has already been sent out. These time period rules for reminders will also need to provide a description of the timing, for use in notification templating.

Once a day, a scheduler will check for any reminder notifications that need to be sent out. Scheduled notifications will only need to be created if past the relevant due date, and haven't already been created.

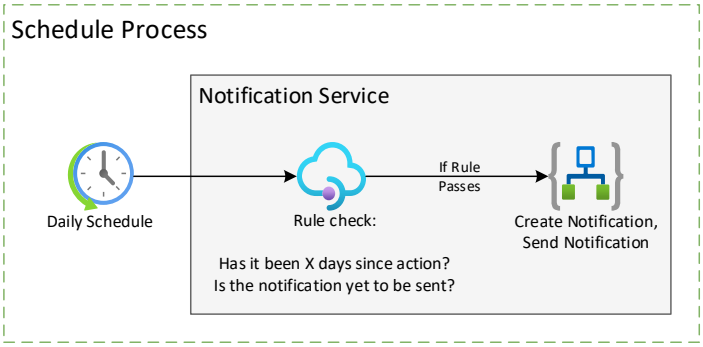


Figure 4: Schedule Process (D:1)

7.5 User Configuration

Users have control over how notifications are presented to them. They can choose to subscribe / unsubscribe from some notification categories, to control whether emails are sent or banners are shown.

Users can choose how frequently they receive notification emails. By default, every notification will be sent as a separate email. Users can choose to opt for a daily summary of notifications, where appropriate for each notification type.

The Notification Service will track when a user has dismissed a portal broadcast notification.

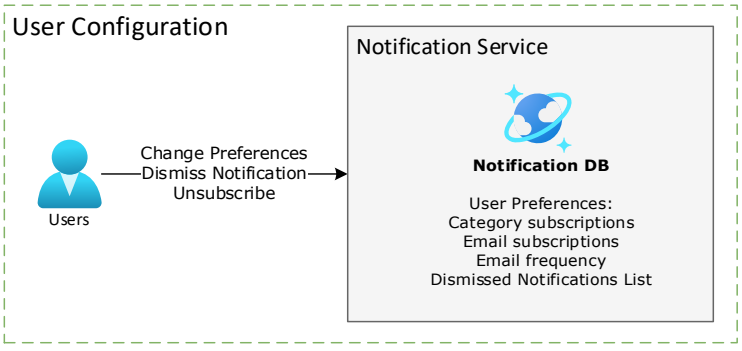


Figure 5: User Configuration Overview (D:1)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 962 of 1295

8 API Design

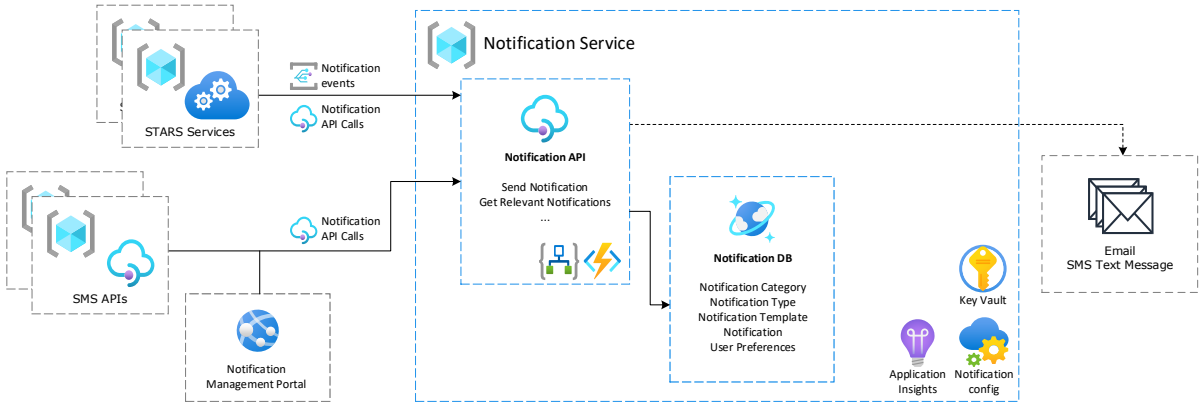


Figure 6: API Design Overview (D:1)

8.1 Notification Config

The notification Config will store values for configuring the service, as well as configuring the notifications. Note that config values that every service will require (env variables, resource group name) will be defined in a separate, common location.

Table 5: Notification Service Config

Key	Value	Comments
DB:ConnectionString	-	Stored in Key Vault.
EventDB:ConnectionString	-	Stored in Key Vault.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 963 of 1295

8.2 Notification Event DB

Events will be managed and stored by the Event Service.

8.3 Notification DB

The Notification Database will be implemented in Azure Cosmos DB, storing non-relational (noSQL) data. It will utilise a standardised document-oriented JSON schema to ensure consistency in the management and presentation of notification data. The file schemas stored in this database are described below.

8.3.1 Reference Entities Schema

8.3.1.1 Notification Category (NotificationCategory)

See 7.1 Notification Categories for more details.

Table 6: Notification Category Fields

Field	Comments
ID	Unique identifier.
Name	Human-readable descriptor.
CanOptOutOf	Boolean. If true, users can set a preference that disables notifications in this category.

8.3.1.1.1 Data

Table 7: Notification Category Data

ID	Name	CanOptOutOf
DataSubmissionReminders	Data submission reminders	false
ValidationResultUpdates	Validation result updates	false
ValidationOutcomeReviewRequests	Validation outcome review request updates	false
ChangesMadeToSubmittedData	Changes made to submitted data	false
MasterDataChanges	Master data changes	true
ReferenceDataChanges	Reference data changes	true
BusinessRuleChanges	Business rule changes	true
StudentFeeChanges	Student fee changes	true
DowntimeNotifications	Downtime notifications	false

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 964 of 1295

8.3.1.2 Notification Type (NotificationType)

See 7.2 Notification Types for more details. This is the comprehensive list of all known notifications that will be sent out as part of the STARS solution.

Table 8: Notification Type Fields

Field	Comments
ID	Unique identifier.
CategoryID	Key to Notification Category. Used for sorting into user subscription groups.
Name	Human-readable descriptor.

8.3.1.2.1 Data

Table 9: Notification Type Data

ID	CategoryID	Name
ReminderDataSubmissionUpcoming	DataSubmissionReminders	Reminder: Data soon due for submission
ReminderDataSubmissionOverdue	Data submission reminders	Reminder: Data is overdue for submission
ReminderDataUploadedNotSubmitted	Data submission reminders	Reminder: Data uploaded but not submitted
ReminderDataQualityChecksFailed	Validation result updates	Reminder: Data quality checks failed
SubmittedDataEdited	Changes made to submitted data	Data was edited for consistency
MasterDataUpdated	Master data changes	Master data was updated
AsyncValidationResultsAvailable	Validation result updates	Validation results are available (asynchronous results)
ValidationResultsReviewedAndResolved	Validation result updates	Validation results are reviewed and resolved
BusinessRulesUpdated	Business rule changes	Business rules updated
StudentFeesChanged	Student fee changes	Student fees changed
ValidationOutcomeReviewComplete	Validation outcome review request updates	Validation outcome review is complete
BroadcastMaintenance	Downtime notifications	Maintenance message
BroadcastUnscheduledDowntime	Downtime notifications	Unscheduled Downtime message
BroadcastGeneral	Downtime notifications	General global announcement

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 965 of 1295

8.3.1.3 Schedule Type (ScheduleType)

When a schedule is processed, this type will determine the party that the notification will be relevant to. Reminder notifications are either sent out to people relevant to an existing submission, or all applicable TPs that have not submitted any data.

Table 10: Schedule Type Fields

Field	Comments
ID	Unique identifier.
Name	Human-readable descriptor.

8.3.1.3.1 Data

Table 11: Schedule Type Data

ID	Name
CheckAllTPs	Check All TPs
CheckAllSubmissions	Check All Data Submissions

8.3.1.4 Schedule Frequency Type (ScheduleFrequencyType)

Note that each frequency type listed here is associated with a separate Logic App schedule. Each of the schedules will filter the list of relevant Notification Schedules based on this reference value. If a new Schedule Frequency Type is required, a new value is added to this table and a new scheduler will need to be set up in the infrastructure.

Based on the current requirements, reminder notifications will only need to be checked / sent out daily.

Table 12: Schedule Frequency Type Fields

Field	Comments
ID	Unique identifier.
Name	Human-readable descriptor.

8.3.1.4.1 Data

Table 13: Schedule Frequency Type Data

ID	Name
Daily	Every 24 hours at 5:00am AEST.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 966 of 1295

8.3.1.5 Notification Status (NotificationStatus)

Although user notifications can't be dismissed / acknowledged if sent to email, it's possible that they will appear as a website banner, like broadcast notifications. In this case, the notification should track if the user has dismissed the banner.

Table 14: Notification Status Fields

Field	Comments
ID	Unique identifier.
Name	Human-readable descriptor.

8.3.1.5.1 Data

Table 15: Notification Status Data

ID	Name
Unacknowledged	Notification which hasn't been dismissed.
Acknowledged	Notification has been dismissed i.e. acknowledged.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 967 of 1295

8.3.2 Notification Template (NotificationTemplate)

A notification "skeleton" used to create notifications for events. An identifier is used for use by other services that want to use a particular template. The various template sections (NotificationTitle, NotificationContentLongForm, NotificationContentShortForm) will contain placeholder text snippets as required (an identifier surrounded by curly braces, e.g. "{SubmissionName}"). When the actual notifications are created from the template, these placeholders must be replaced from relevant information.

Table 16: Notification Template Fields

Field	Type
ID	Unique identifier
TemplateName	String (human readable descriptor)
NotificationTypeID	Ties a particular notification type to this template. The template type value will be used by other services when they want to use a particular template.
NotificationTitle	String. For e.g. email subject. Raw, unformatted text.
NotificationContentLongForm	String. For long-form content like email (detailed form). This field will contain simple HTML including links, styling information, base64-encoded images... features that modern email clients will be able to parse.
NotificationContentShortForm	String. For any short form messages, should the requirement for them arise. This will be raw, unformatted text.
SoftDeleted	Flag to keep track of soft deletion.

8.3.2.1 Example file

```
{
  "ID": "8d0b90b3-bd2c-4929-abe6-a96046f38dcf",
  "TemplateName": "Scheduled maintenance",
  "NotificationTypeID": "broadcast",
  "NotificationTitle": "Upcoming scheduled maintenance",
  "NotificationContentLongForm": "<p>Please note there is upcoming
scheduled maintenance planned. The system will not be available from
{StartTime} to {EndTime}.</p>",
  "NotificationContentShortForm": "STARS will be down for maintenance
from {StartTime} to {EndTime}",
  "SoftDeleted": false
}
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 968 of 1295

8.3.3 Reminder / Notification Schedule (NotificationRuleSchedule)

A Notification Schedule will describe a rule that should be checked automatically on a schedule. Notifications will be sent out depending on the rule result. The rule will check if a certain amount of time has passed, and that this notification hasn't already been set out to the particular recipient.

Based on the current requirements, these schedules will each be run daily.

Table 17: Reminder / Notification Fields

Field	Type
ID	Unique identifier
ScheduleName	String (human readable descriptor)
ScheduleFrequencyTypeID	Key to Schedule Frequency Type. Determines how frequently this schedule should be run. Likely this will always be "daily".
ScheduleTypeID	Key to Schedule Type. "Will this rule apply to each STA, or to each TP?".
TemplateID	Key to Template to use when notifications need to be created.
RuleID	Key to the Rule to check from the Rules Service.
SoftDeleted	Flag to keep track of soft deletion.

8.3.3.1 Example file

This example schedule will check a rule each day. If the rule returns a pass result, notifications will be sent to all relevant contacts (depending on each user's notification preferences).

```
{
 "ID": " 23927518-0b77-46d6-9fda-006dbbf141ff",
 "ScheduleName": "Notify NSW STAs on overdue records",
 "ScheduleFrequencyTypeID": "Daily",
 "ScheduleTypeID": "AllTPs",
 "TemplateID": "DataSubmissionDue",
 "RuleID": "NotificationRule_101_ShouldNotifyPT",
 "SoftDeleted": false
}
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 969 of 1295

8.3.4 Broadcast Notification / Announcements (BroadcastNotification)

A Broadcast Notification is not tied to a specific user. They will appear as a banner announcements in the relevant portals. User-specific data (for example, timezone-aware dates) will be applied to broadcast notifications when retrieved by a user.

Table 18: Broadcast Notification Fields

Field	Type
ID	String (identifier / code for reference)
TemplateID	Key to relevant Template for this notification. This also determines the Notification Type and Notification Category for this broadcast, and ultimately the groups of users that will see this broadcast.
StartDate	The date at which the announcement should first appear.
EndDate	The date at which the announcement should no longer appear.
SoftDeleted	Flag to keep track of soft deletion.

8.3.4.1 Example file

This notification has been broadcast to all users in the System Operator group. All notification template sections will be filled out on a per-user basis when a user requests a list of notifications. The template used here will include placeholder values for "MaintenanceStartTime" and "MaintenanceEndTime", which will be contextualised by the user's configured timezone when retrieved by the user.

```
{
  "ID": "440fbaa9-ee61-4af3-b09c-e8ace2ae6629",
  "TemplateID": "298555eb-33c5-4f3c-b2f3-9e76ed20c9db",  "Status":
"Active",
  "StartDate": "2022-01-01T08:30:00",
  "EndDate": "2022-01-01T17:30:00"
}
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 970 of 1295

8.3.5 User Notification (UserNotification)

A User Notification is sent to a specific user. They will contain personalised content for a particular user if defined by the associated template. The notification will be sent via email (depending on the user's preferences) and can be retrieved via the Notification API.

Note that Notifications do not have a SoftDeleted flag field. As notifications are sent externally, marking an internal Notification entity as deleted would be misleading.

Table 19: User Notification Fields

Field	Type
ID	String (identifier / code for reference)
TemplateID	Key to relevant Template for this notification. This also determines the Notification Type and Notification Category for this broadcast.
NotificationTitle	Corresponds to Template's NotificationTitle, but with template data filled.
NotificationContentLongForm	Corresponds to Template's NotificationContentLongForm, but with template data filled.
NotificationContentShortForm	Corresponds to Template's NotificationContentShortForm, but with template data filled.
UserID	Key to the user that this notification has been sent to. This can be used to look up the relevant SMS, TP or STA for this user.
MarkedAsRead	Boolean. If true, the notification has been acknowledged by the user via the portal. This flag does not relate to sent emails.
DateSent	Date. Used to ensure notifications sent a long time ago can be filtered out on the portal.

8.3.5.1 Example file

This notification has been sent via email to the chosen recipient. The notification has been marked as delivered. All notification template sections have been filled out as required.


```
{
  "ID": "9cf9cab9-92f5-4e8e-8212-ca622b64a393",
  "TemplateID": "298555eb-33c5-4f3c-b2f3-9e76ed20c9db",
  "SentByService": "Submission Service",
  "NotificationTitle": "Your VET-IS submission My Submission Test did not
pass validation checks",
  "NotificationContentLongForm": "Hi Example Person, your submission did
not pass validation checks. See www.example.com/vetis/dac25bec for more
details.",
  "NotificationContentShortForm": null,
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 971 of 1295

```
  "UserID": "12345"
}
```

8.3.6 User Preferences

Table 20: User Notification Fields

Field	Type
ID	String (identifier / code for reference)
UserID	Key to relevant User.
NotificationTypeSubscriptionsEmail	List of notification types that this user should receive notifications for via email.
NotificationTypeSubscriptionsBanner	List of notification types that this user should receive notifications for via portal banners / alerts.
Emails	List of strings. Stored in Master DB.
SummaryEmailTime	Time. For summary emails and timed notifications, determines what time of the day they will be sent out.
Timezone	UTC offset. Contextualises the SummaryEmailTime set by the user. Used to contextualise e.g. downtime notifications.
DismissedBroadcastNotifications	List of broadcast notifications that have been dismissed by the user. Controls which banners will appear in the portal for a given user.

8.3.6.1 Example file

These example user preferences describe a user with a specific subscription to rule change notifications. They will also receive all notifications that fall under a "mandatory" category.

```
{
  "ID": "cf88a0d2-6942-460b-89af-c05132c96054",
  "UserID": "298555eb-33c5-4f3c-b2f3-9e76ed20c9db",
  " NotificationTypeSubscriptionsEmail": ["RuleChangedNotificationType"],
" NotificationTypeSubscriptionsBanner": ["RuleChangedNotificationType"],
  "Emails": [],
  "SummaryEmailTime": "17:15",
  "SummaryEmailTimezone": "GMT+05:00"
}
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 972 of 1295

8.4 Notification API

The Notification API will be implemented in Azure API Management, with the backend implemented as one or more Azure API Apps. API Apps will support the implementation of OData APIs to provide consistent API interactions between STARS Services and for consumption by external services. A scheduler will also call the Notification API on a regular basis.

The Notification Service API will be presented at the following path:

<STARS Base URL>/notificationService/

The API will allow other services to flexibly query and provide notification data in JSON format, including information on the notifications themselves and all the notification's lifecycle data, including transmission and receipt status. The API will also be used by the Management Portal user interfaces to service UI requests for notification management.

The endpoints will manage the creation and storage of notifications and supporting data, and transmission of notifications to the receiving parties using various delivery methods (e.g., via email).

The API will conform to the STARS Design Principles (reference B section 7) and the STARS API Design Document (reference C). The endpoints described below will be standard CRUD REST endpoints. See the STARS API Design Document for success / failure status codes and expected response headers. See Technical Design Patterns section below for general endpoint descriptions.

Filtering (including free-text search) will be handled using OData, as detailed below. There will be implicit filtering based on the calling user, to ensure only relevant notifications are returned.

8.4.1 Notification Service Endpoints

Notification API Endpoint details are listed here, with details of the implementations of these endpoints found in section 8.5 Notification API Operations below. The request header / response body details for these endpoints are also listed below.

Table 21: Notification Service API Endpoints

API Endpoint Name	Base Definition	Path
Send User Notification	-	POST <serviceurl>/userNotifications
Send User Notification from Schedule	-	POST <serviceurl>/scheduledUserNotifications
Update User Notification	Update Endpoint	PATCH <serviceurl>/userNotifications/[id]
Get User Notifications	Read Endpoint – List	GET <serviceurl>/userNotifications
Get User Notification by Id	Read Endpoint – Detail	GET <serviceurl>/userNotifications(“asdasd”)
Create Broadcast Notification	Create Endpoint	POST <serviceurl>/broadcastNotifications
Update Broadcast Notification	Update Endpoint	PUT <serviceurl>/broadcastNotifications/[id]

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 973 of 1295

Get Broadcasts for User	Read Endpoint – List	GET <serviceurl>/broadcastNotifications
Get Broadcast Notification by Id	Read Endpoint – Detail	GET <serviceurl>/broadcastNotifications/[id]
Get Notification Categories	Read Endpoint – List	GET <serviceurl>/notificationCategories
Get Notification Types	Read Endpoint – List	GET <serviceurl>/notificationTypes
Get Schedule Types	Read Endpoint – List	GET <serviceurl>/scheduleTypes
Get Schedule Frequency Types	Read Endpoint – List	GET <serviceurl>/scheduleFrequencyTypes
Get Notification Status Types	Read Endpoint – List	GET <serviceurl>/notificationStatusTypes
Create Notification Template	Create Endpoint	POST <serviceurl>/notificationTemplates
Update Notification Template	Update Endpoint	PUT <serviceurl>/notificationTemplates/[id]
Delete Notification Template	Delete Endpoint	DELETE <serviceurl>/notificationTemplates/[id]
Get Notification Templates	Read Endpoint – List	GET <serviceurl>/notificationTemplates
Get Notification Template by Id	Read Endpoint – Detail	GET <serviceurl>/notificationTemplates/[id]
Create Notification Schedule	Create Endpoint	POST <serviceurl>/notificationSchedules
Update Notification Schedule	Update Endpoint	PUT <serviceurl>/notificationSchedules/[id]
Delete Notification Schedule	Delete Endpoint	DELETE <serviceurl>/notificationSchedules/[id]
Get Notification Schedules	Read Endpoint – List	GET <serviceurl>/notificationSchedules
Get Notification Schedule by Id	Read Endpoint – Detail	GET <serviceurl>/notificationSchedules/[id]
Create Broadcast Notification	Create Endpoint	POST <serviceurl>/broadcastNotifications
Update Broadcast Notification	Update Endpoint	POST <serviceurl>/broadcastNotifications/[id]
Delete Broadcast Notification	Delete Endpoint	DELETE <serviceurl>/broadcastNotifications/[id]
Get Broadcast Notifications	Read Endpoint – List	GET <serviceurl>/broadcastNotifications
Get Broadcast Notification by Id	Read Endpoint – Detail	GET <serviceurl>/broadcastNotifications/[id]
Update User Notification	Update Endpoint	PUT <serviceurl>/userNotifications/[id]
Get User Notifications	Read Endpoint – List	GET <serviceurl>/userNotifications
Get User Notification by Id	Read Endpoint – Detail	GET <serviceurl>/userNotifications/[id]
Update User Preferences	Update Endpoint	POST <serviceurl>/userPreferences/[userid]
Get User Preferences	Read Endpoint – Detail	GET <serviceurl>/userPreferences/[userid]

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 974 of 1295

8.4.2 Send User Notification

POST <serviceurl>/odata/userNotification

8.4.2.1 Request Body

When creating a notification, a notification type must be supplied, along with a collection of data keys/values. The data keys will depend on the type of notification.

Table 22: Send User Notification Request Fields

Key	Comments
NotificationType	
RecipientUserId	The user that should receive this notification.
Data	Template fields. Used to fill the template associated with this notification type. This is a JSON object of keys and values.

8.4.2.2 Response Body

Table 23: Send User Notification Response Fields

Key	Comments
Id	
NotificationType	
NotificationTitle	The resulting value after templating is applied.
NotificationContentLongForm	The resulting value after templating is applied.
NotificationContentShortForm	The resulting value after templating is applied.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 975 of 1295

8.4.3 Send User Notification from Schedule

POST <serviceurl>/scheduledUserNotification

This endpoint will only be called by the scheduler.

8.4.3.1 Request Body

Table 24: Send User Notification from Schedule Request Fields

Key	Comments
ScheduleFrequencyType	Determines which schedules will be run as a result of this request.

8.4.3.2 Response Body

200 Success

8.4.4 Update User Notification

PATCH <serviceurl>/userNotifications/[id]

This endpoint will use the Update Entity operation. Note that this is a very restricted endpoint – notifications can only be updated to be marked as read.

8.4.4.1 Request Body

Table 25: Update User Notification Request Fields

Key	Comments
MarkedAsRead	Boolean.

8.4.4.2 Response Body

Table 26: Update User Notification Response Fields

Key	Comments
Id	
TemplateId	
NotificationTitle	
NotificationContentLongForm	
NotificationContentShortForm	
MarkedAsRead	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 976 of 1295

8.4.5 Get User Notifications

GET <serviceurl>/userNotifications

8.4.5.1 Filters

Table 27: Get User Notifications Filters

Key	Comments
NotificationType	Only return notifications for the given type.
StartDateBefore	Notification was created before this date.
StartDateAfter	Notification was created after this date.
MarkedAsRead	Boolean.
Query	Free-text search. Searches for matching text in NotificationTitle, NotificationContentLongForm, NotificationContentShortForm and Status.

8.4.5.2 Response Body – Notification

The results section of the response body will contain the following values:

Table 28: Get User Notifications Response Fields

Key	Comments
Id	
TemplateId	
MarkedAsRead	

8.4.6 Get User Notification by Id

GET <serviceurl>/userNotifications/[id]

8.4.6.1 Response Body

Table 29: Get User Notifications by ID Response Fields

Key	Comments
Id	
TemplateId	
MarkedAsRead	
NotificationTitle	
NotificationContentLongForm	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 977 of 1295

NotificationContentShortForm	

8.4.7 Create Broadcast Notification

POST <serviceurl>/broadcastNotifications

8.4.7.1 Request Body

When creating a broadcast notification, a notification type must be supplied, along with a start and end date (to determine the period that this notification is broadcast for).

Table 30: Create Broadcast Notification Request Fields

Key	Comments
TemplateId	
StartDate	
EndDate	
TemplateValues	Stored with the broadcast notification. Used later when the notification template is filled.

8.4.7.2 Response Body

Table 31: Create Broadcast Notification Response Fields

Key	Comments
Id	
TemplateId	
StartDate	
EndDate	
TemplateValues	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 978 of 1295

8.4.8 Update Broadcast Notification

PUT <serviceurl>/broadcastNotifications/[id]

8.4.8.1 Request Body

Table 32: Update Broadcast Notification Request Fields

Key	Comments
TemplateId	Change template.
StartDate	Change start of broadcast.
EndDate	Change end of broadcast.
TemplateValues	Change associated data.

8.4.8.2 Response Body

Table 33: Update Broadcast Notification Response Fields

Key	Comments
Id	
TemplateId	
StartDate	
EndDate	
TemplateValues	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 979 of 1295

8.4.9 Get Broadcasts for User

GET <serviceurl>/broadcastNotifications

8.4.9.1 Filters

Table 34: Get Broadcasts for User Filters

Filter	Comments
UserId	Implicit depending on the calling user.
NotificationType	
Status	

8.4.9.2 Response Body

Table 35: Get Broadcasts for User Response Fields

Key	Comments
Id	
TemplateId	
StartDate	
EndDate	
Status	
NotificationTitle	
NotificationContentLongForm	
NotificationContentShortForm	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 980 of 1295

8.4.10 Get Broadcast Notification by Id

GET <serviceurl>/broadcastNotifications/[id]

8.4.10.1 Response Body

Table 36: Get Broadcast Notification by Id Response Fields

Key	Comments
Id	
NotificationType	
StartDate	
EndDate	
Status	
NotificationTitle	
NotificationContentLongForm	
NotificationContentShortForm	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 981 of 1295

8.4.11 Get Notification Categories

GET <serviceurl>/notificationCategories

8.4.11.1 Filters

Table 37: Get Notification Categories Filters

Key	Comments
Query	Free-text search. Searches for matching text in ID and Name.

8.4.11.2 Response Body

The results section of the response body will contain the following values:

Table 38: Get Notification Categories response Fields

Key	Comments
Id	
Name	

8.4.12 Get Notification Types

GET <serviceurl>/notificationTypes

8.4.12.1 Filters

Table 39: Get Notification Types Filters

Key	Comments
Query	Free-text search. Searches for matching text in ID, Category Name and Name.

8.4.12.2 Response Body

The results section of the response body will contain the following values:

Table 40: Get Notification Types Response Fields

Key	Comments
Id	
CategoryId	
Name	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 982 of 1295

8.4.13 Get Schedule Types

GET <serviceurl>/scheduleTypes

8.4.13.1 Filters

Table 41: Get Schedule Types Filters

Key	Comments
Query	Free-text search. Searches for matching text in ID and Name.

8.4.13.2 Response Body

The results section of the response body will contain the following values:

Table 42: Get Schedule Types Response Fields

Key	Comments
Id	
Name	

8.4.14 Get Schedule Frequency Types

GET <serviceurl>/scheduleFrequencyTypes

8.4.14.1 Filters

Table 43: Get Schedule Frequency Types Filters

Key	Comments
Query	Free-text search. Searches for matching text in ID and Name.

8.4.14.2 Response Body

The results section of the response body will contain the following values:

Table 44: Get Schedule Frequency Types Response Fields

Key	Comments
Id	
Name	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 983 of 1295

8.4.15 Get Notification Status Types

GET <serviceurl>/notificationStatusTypes

8.4.15.1 Filters

Table 45: Get Notification Status Types Filters

Key	Comments
Query	Free-text search. Searches for matching text in ID and Name.

8.4.15.2 Response Body

The results section of the response body will contain the following values:

Table 46: Get Notification Status Types Response Body

Key	Comments
Id	
Name	

8.4.16 Create Notification Template

POST <serviceurl>/notificationTemplates

8.4.16.1 Request Body

Table 47: Create Notification Template Request Fields

Key	Comments
TemplateName	
NotificationType	
NotificationTitle	
NotificationContentLongForm	
NotificationContentShortForm	

8.4.16.2 Response Body

Table 48: Create Notification Template Response Fields

Key	Comments
Id	
TemplateName	
NotificationType	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 984 of 1295

NotificationContentTitle	
NotificationContentLongForm	
NotificationContentShortForm	

8.4.17 Update Notification Template

PUT <serviceurl>/notificationTemplates/[id]

8.4.17.1 Request Body

Table 49: Update Notification Template Request Fields

Key	Comments
TemplateName	
NotificationType	
NotificationTitle	
NotificationContentLongForm	
NotificationContentShortForm	

8.4.17.2 Response Body

Table 50: Update Notification Template Response Fields

Key	Comments
Id	
TemplateName	
NotificationType	
NotificationTitle	
NotificationContentLongForm	
NotificationContentShortForm	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 985 of 1295

8.4.18 Delete Notification Template

DELETE <serviceurl>/notificationTemplates/[id]

8.4.18.1 Response Body

Table 51: Delete Notification Template Response Fields

Key	Comments
Id	
TemplateName	
NotificationType	
NotificationTitle	
NotificationContentLongForm	
NotificationContentShortForm	

8.4.19 Get Notification Templates

GET <serviceurl>/notificationTemplates

8.4.19.1 Filters

Table 52: Get Notification Templates Filters

Key	Comments
Query	Free-text search. Searches for matching text in TemplateName, NotificationType, NotificationTitle, NotificationContentLongForm and NotificationContentShortForm.

8.4.19.2 Response Body – Notification Template

The results section of the response body will contain the following values:

Table 53: Get Notification Template Response Fields

Key	Comments
Id	
NotificationType	
NotificationTitle	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 986 of 1295

8.4.20 Get Notification Template by Id

GET <serviceurl>/notificationTemplates/[id]

8.4.20.1 Response Body

Table 54: Get Notification Template by Id Request Fields

Key	Comments
Id	
NotificationType	
NotificationTitle	
NotificationContentLongForm	
NotificationContentShortForm	

8.4.21 Create Notification Schedule

POST <serviceurl>/notificationSchedules

8.4.21.1 Request Body

Table 55: Create Notification Schedule Request Fields

Key	Comments
ScheduleName	
ScheduleFrequencyType	
ScheduleType	
TemplateId	
RuleId	

8.4.21.2 Response Body

Table 56: Create Notification Schedule Response Fields

Key	Comments
Id	
ScheduleName	
ScheduleFrequencyType	
ScheduleType	
TemplateId	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 987 of 1295

RuleId	
SoftDeleted	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 988 of 1295

8.4.22 Update Notification Schedule

PUT <serviceurl>/notificationSchedules/[id]

8.4.22.1 Request Body

Table 57: Update Notification Schedule Request Fields

Key	Comments
ScheduleName	
ScheduleFrequencyType	
TemplateId	
RuleId	

8.4.22.2 Response Body

Table 58: Update Notification Schedule Response Fields

Key	Comments
Id	
ScheduleName	
ScheduleFrequencyType	
ScheduleType	
TemplateId	
RuleId	
SoftDeleted	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 989 of 1295

8.4.23 Delete Notification Schedule

DELETE <serviceurl>/notificationSchedules/[id]

8.4.23.1 Response Body

Table 59: Delete Notification Schedule Response Fields

Key	Comments
Id	
ScheduleName	
ScheduleFrequencyType	
ScheduleType	
TemplateId	
RuleId	
SoftDeleted	

8.4.24 Get Notification Schedules

GET <serviceurl>/notificationSchedules

8.4.24.1 Filters

Table 60: Get Notification Schedules Filters

Key	Comments
ScheduleFrequencyTypeId	Filter by frequency type (daily, frequent).
TemplateId	Filter by template.
RuleId	Filter by rule.
RelevantToPersonId	Filter by user.
Query	Free-text search. Searches for matching text in ScheduleName, FrequencyTypeName, TemplateName, RuleName.

8.4.24.2 Response Body – Notification Schedule

The results section of the response body will contain the following values:

Table 61: Get Notification Schedules Response Fields

Key	Comments
Id	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 990 of 1295

ScheduleName	
ScheduleFrequencyType	
ScheduleType	
TemplateId	
RuleId	
SoftDeleted	

8.4.25 Get Notification Schedule by Id

GET <serviceurl>/notificationSchedules/[id]

8.4.25.1 Response Body

Table 62: Get Notification Schedules by Id Response Fields

Key	Comments
Id	
ScheduleName	
ScheduleFrequencyType	
TemplateId	
RuleId	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 991 of 1295

8.4.26 Create Broadcast Notification

POST <serviceurl>/broadcastNotifications

8.4.26.1 Request Body

Table 63: Create Broadcast Notification Request Fields

Key	Comments
TemplateId	
StartDate	
EndDate	

8.4.26.2 Response Body

Table 64: Create Broadcast Notification Response Fields

Key	Comments
Id	
TemplateId	
StartDate	
EndDate	
SoftDeleted	

8.4.27 Update Broadcast Notification

PUT <serviceurl>/broadcastNotifications/[id]

8.4.27.1 Request Body

Table 65: Update Broadcast Notification Request Fields

Key	Comments
TemplateId	
StartDate	
EndDate	

8.4.27.2 Response Body

Table 66: Update Broadcast Notification Response Fields

Key	Comments

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 992 of 1295

Id	
TemplateId	
StartDate	
EndDate	
SoftDeleted	

8.4.28 Delete Broadcast Notification

DELETE <serviceurl>/broadcastNotifications/[id]

8.4.28.1 Response Body

Table 67: Delete Broadcast Notification Response Fields

Key	Comments
Id	
TemplateId	
StartDate	
EndDate	
SoftDeleted	

8.4.29 Get Broadcast Notifications

GET <serviceurl>/broadcastNotifications

8.4.29.1 Filters

Table 68: Get Broadcast Notifications Filters

Key	Comments
TemplateId	Filter by template.
StartDateBefore	
StartDateAfter	
EndDateBefore	
EndDateAfter	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 993 of 1295

8.4.29.2 Response Body

Table 69: Get Broadcast Notifications Response Fields

Key	Comments
Id	
TemplateID	
StartDate	
EndDate	
SoftDeleted	

8.4.30 Get Broadcast Notification by Id

GET <serviceurl>/broadcastNotifications/[id]

8.4.30.1 Response Body

Table 70: Get Broadcast Notification by Id Response Fields

Key	Comments
Id	
TemplateID	
StartDate	
EndDate	
SoftDeleted	

8.4.31 Update User Notification

PUT <serviceurl>/userNotifications/[id]

8.4.31.1 Request Body

Table 71: Update User Notification Request Fields

Key	Comments
TemplateId	
UserId	
Data	Template placeholder data values.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 994 of 1295

8.4.31.2 Response Body

Table 72: Update User Notification Response Fields

Key	Comments
Id	
TemplateId	
NotificationTitle	
NotificationContentLongForm	
NotificationContentShortForm	
UserId	
MarkedAsRead	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 995 of 1295

8.4.32 Get User Notifications

GET <serviceurl>/userNotifications

8.4.32.1 Filters

Table 73: Get User Notifications Filters

Key	Comments
TemplateId	Filter by template.
DateBefore	
DateAfter	
Query	Free text search on NotificationTitle, NotificationContentLongForm, NotificationContentShortForm.

8.4.32.2 Response Body

Table 74: Get User Notifications Response Fields

Key	Comments
Id	
TemplateId	
NotificationTitle	
NotificationContentLongForm	
NotificationContentShortForm	
UserId	
MarkedAsRead	

8.4.33 Get User Notification by Id

GET <serviceurl>/userNotifications/[id]

8.4.33.1 Response Body

Table 75: Get User Notification by Id Response Fields

Key	Comments
Id	
TemplateId	
NotificationTitle	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 996 of 1295

NotificationContentLongForm	
NotificationContentShortForm	
UserId	
MarkedAsRead	

8.4.34 Update User Preferences

PUT <serviceurl>/userPreferences/[userid]

8.4.34.1 Request Body

Table 76: Update User Preferences Request Fields

Key	Comments
NotificationTypeSubscriptionsEmail	
NotificationTypeSubscriptionsBanner	
SummaryEmailTime	
Timezone	
DismissedBroadcastNotifications	

8.4.34.2 Response Body

Table 77: Update User Preferences Response Fields

Key	Comments
Id	
UserId	
NotificationTypeSubscriptionsEmail	
NotificationTypeSubscriptionsBanner	
SummaryEmailTime	
Timezone	
DismissedBroadcastNotifications	

8.4.35 Get User Preferences

GET <serviceurl>/userPreferences/[userid]

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 997 of 1295

8.4.35.1 Response Body

Table 78: Get User Preferences Response Fields

Key	Comments
Id	
UserId	
NotificationTypeSubscriptionsEmail	
NotificationTypeSubscriptionsBanner	
SummaryEmailTime	
Timezone	
DismissedBroadcastNotifications	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 998 of 1295

8.5 Notification API Operations

The Notification Service internal processes are facilitated through API operations defined in the Notification API. Details of these operations are as follows:

8.5.1 Send User Notification

Various other services can send notifications via the API. Based on the template data sent, a notification is created and stored in the Notification DB. Finally, the notification will be sent out via email.

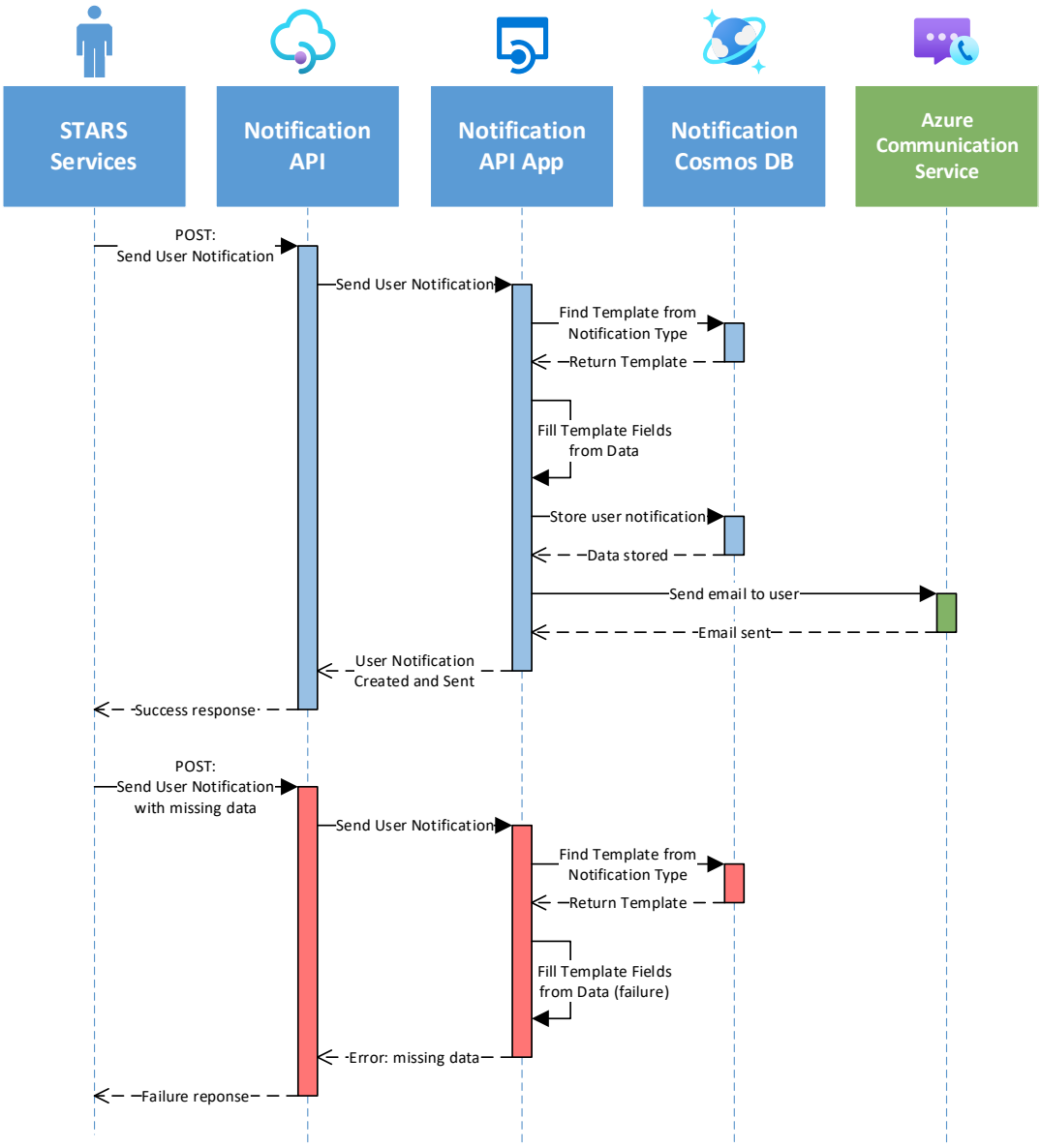


Figure 7: Send User Notification (D:5)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 999 of 1295

8.5.2 Send User Notification from Schedule

Reminder emails are sent out daily for the notification types listed above. The scheduler will check for reminders due to be sent out once a day, and will check that the associated rule passes and that the notification has not already been sent out.

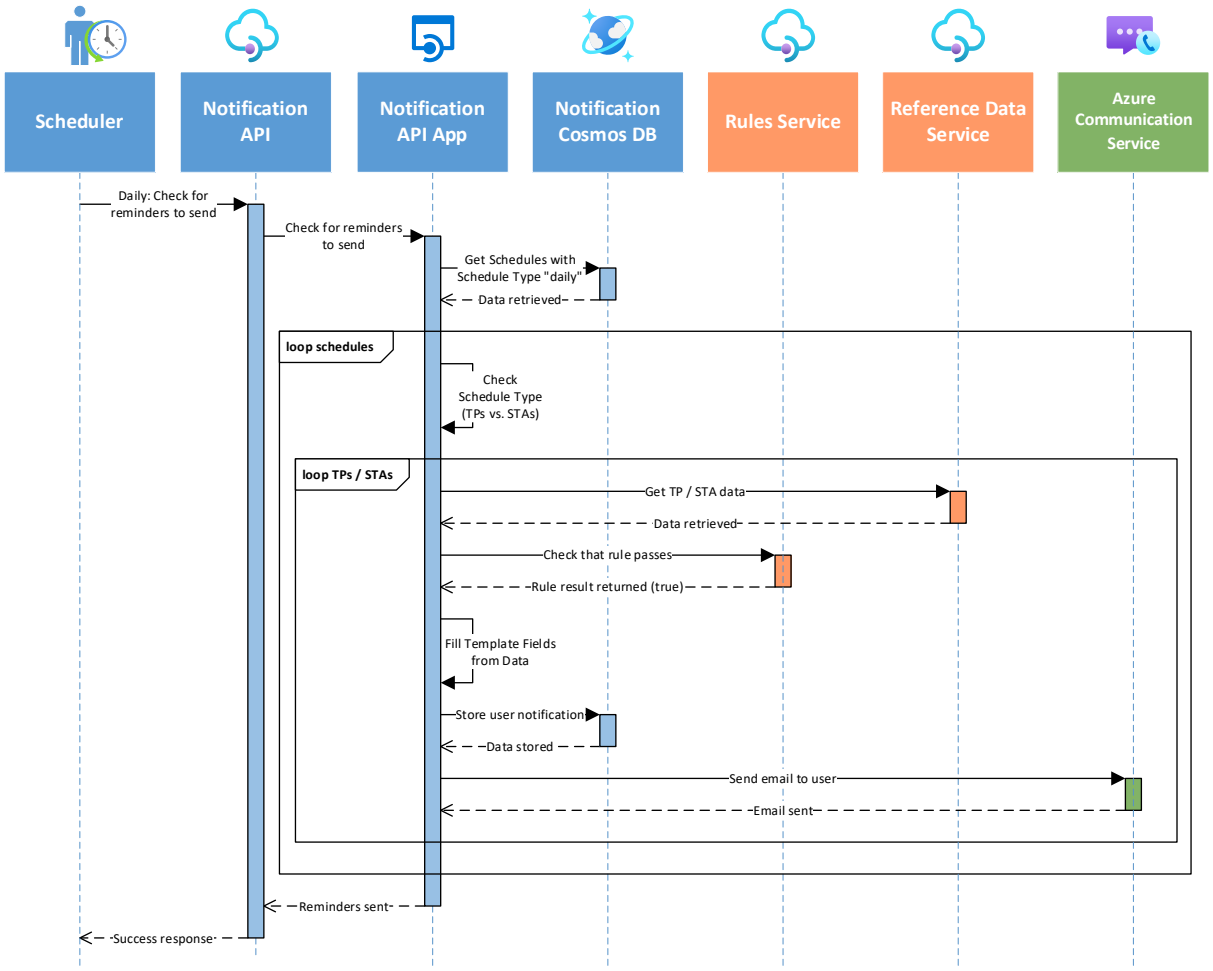


Figure 8: Send User Notification from Schedule (D:6)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1000 of 1295

8.5.3 Get Broadcast Notification List

Broadcast notifications are stored as a general description. When a user wants to see relevant broadcast notifications, this endpoint is used and the user-specific contents are generated and included in the response. The user-specific data will include timezone-aware dates. Users can only retrieve notifications that are relevant to them.

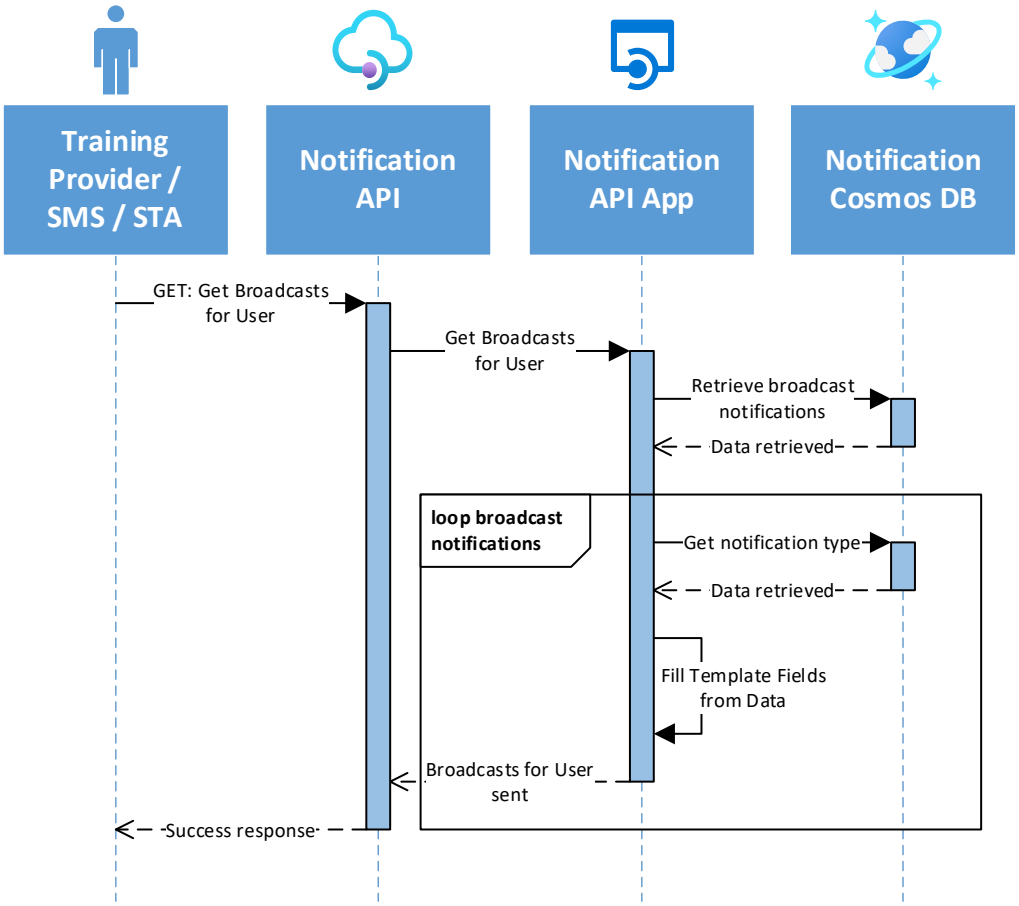


Figure 9: Get Broadcast Notification List (D:7)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1001 of 1295

8.5.4 Fill Template Fields from Data

Part of the Send User Notification, Send User Notification from Schedule and Get Broadcast Notification List operations.

1. Notification Type and Notification data will be passed in
2. Find Notification Template from Notification Type
3. Check that the Notification data provides all the necessary data for the chosen Template
    a. Check NotificationTitle, NotificationContentLongForm and NotificationContentShortForm fields for any text between "{{" and "}}" markers
    b. Trim whitespace from the beginning and end of this text to find the placeholder identifier
    c. Produce a list of these placeholder identifiers
    d. Ensure all placeholder identifiers have associated data keys in the Notification data that was passed in
    e. If not, log an error and do not continue
    f. If Notification data passed in values that aren't used by this template, log a warning detailing the fields that are being ignored
4. Copy the base template parts (NotificationTitle, NotificationContentLongForm and NotificationContentShortForm) into the final notification content to return
5. For each of these fields
    a. Search for a placeholder value starting at "{{" and ending with "}}"
    b. Replace content (including "{{" and "}}" markers) with the relevant data
        i. Do not replace or trim whitespace on the outside of these markers
6. Return the finalised NotificationTitle, NotificationContentLongForm and NotificationContentShortForm fields

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1002 of 1295

8.5.5 Send Email to User

Part of the Send User Notification and Send User Notification from Schedule operations.

Note: Consider SendGrid for templating and sending emails.

1. User Notification entity will be passed in including contents and UserID
2. Get the SMTP details for sending emails from the Notification Config
    a. Retrieve SMTP server, SMTP username, SMTP password
3. If Notification UserID is not subscribed to this event category, do not continue
    a. Log that the message was not delivered via email
4. Get the list of relevant emails for this UserID
5. For each email address for this user, produce an email
    a. Set email subject to Notification's NotificationTitle
    b. Set email body to Notification's NotificationContentLongForm
    c. Set "To" field to User's email address
6. Set "From" field to relevant SMTP email address
7. Send each email
8. Log that each email was sent

For technical details on sending emails using Azure, see: https://learn.microsoft.com/en-us/azure/communication-services/quickstarts/email/send-email?pivots=programming-language-csharp

8.5.5.1 Code Snippet

See the below C# example from the link above for creating and sending an email.

```
// create the message
var msg = new MailMessage();
msg.From = new MailAddress("info@YourWebSiteDomain.com");
msg.To.Add(strTo);
msg.Subject = strSubject;
msg.IsBodyHtml = true;
msg.Body = strMessage;

// configure the smtp server
var smtp = new SmtpClient("YourSMTPServer")
{
  Credentials = new
System.Net.NetworkCredential("YourSMTPServerUserName",
"YourSMTPServerPassword")
};

// send the message
smtp.Send(msg);
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1003 of 1295

8.5.6 Check Rule Results for Each Relevant Entity

Used by the Send User Notification from Schedule operation.

Produces a set of notification data entities for each relevant TP / submission.

Consider the example where the scheduler fails to run one day. The next day, we need to send out notifications relevant to the previous day. Rule checks should ask "is it now less than 7 days?", not "is it exactly 7 days?". Duplicate notifications are avoided by having the rule also check "have we already sent out a notification for this event?".

1. A Notification Schedule will be passed in, including the relevant RuleID and NotificationTypeID
2. If the Schedule Type is CheckAllTPs
    a. Retrieve a list of all TPs from Master Data API
    b. For each TP
        i. Get TP identifier
        ii. Get all relevant contact users for this submission
        iii. Run the associated Rule, passing in the specific TP information for the rule to check
        iv. If the rule does not pass, log info that the TP users were not notified and do not continue
        v. If rule passes, return the Notification Type, Rule result and TP data as Notification data
3. If the Schedule Type is CheckAllSubmissions
    a. Retrieve a list of all submissions from Submission API
    b. For each submission
        i. Get the submission identifier
        ii. Get all relevant contact users for this submission
        iii. Run the associated Rule, passing in the specific submission information for the rule to check
            1. If the rule does not pass, log info that the submission users were not notified and do not continue
        iv. If true, return the Notification Type, Rule result and submission data as Notification data
4. If the Schedule Type is any other value
    a. Log an error – schedule type not supported

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1004 of 1295

8.5.7 Create Entity (General)

As a "General" operation, the implementation of this will cover all entities with Create endpoints described in the 8.4.1 Notification Service Endpoints section above.

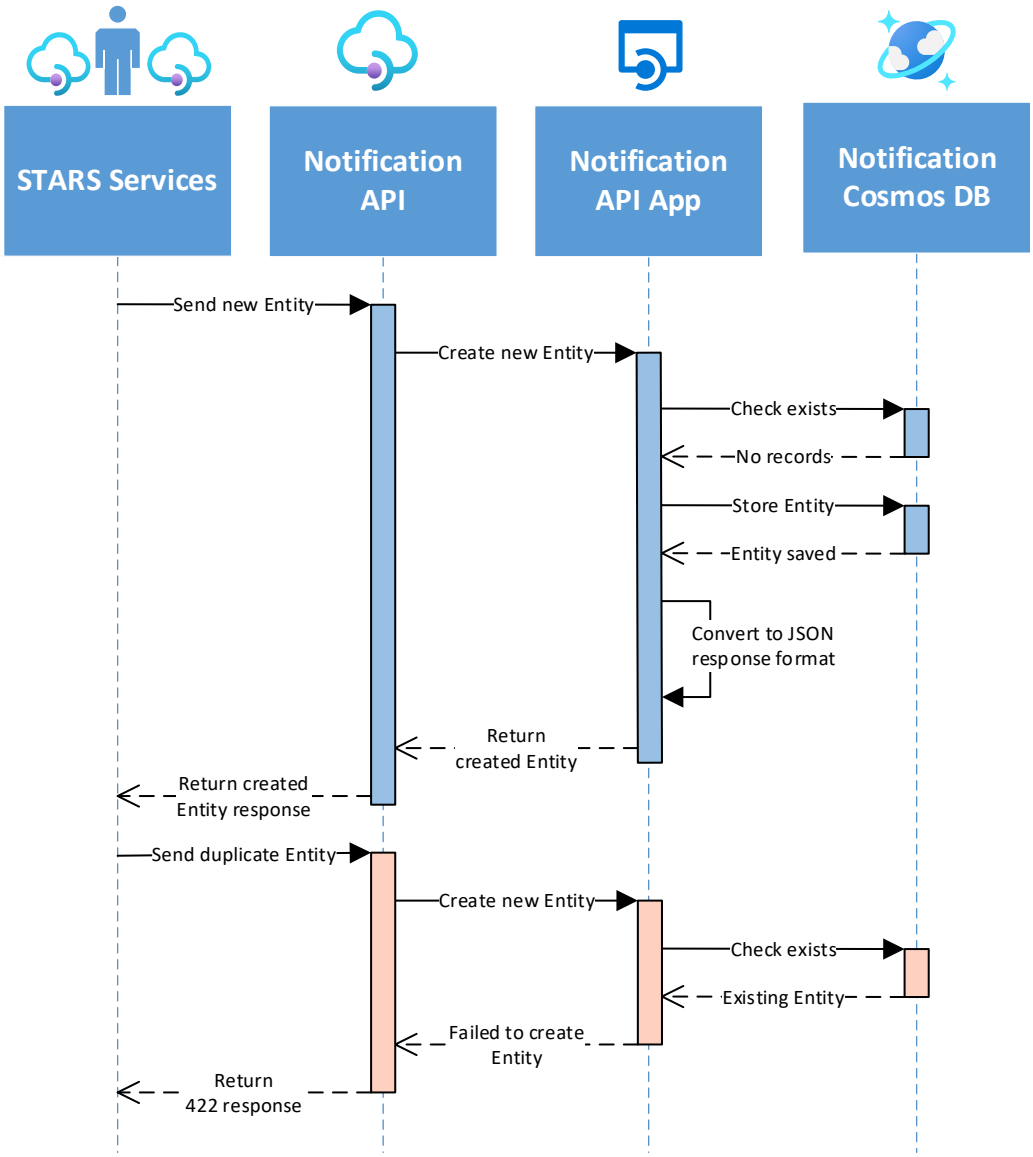


Figure 10: Create Entity (General) (D:8)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1005 of 1295

8.5.8 Update Entity (General)

As a "General" operation, the implementation of this will cover all entities with Update endpoints described in the 8.4.1 Notification Service Endpoints section above.

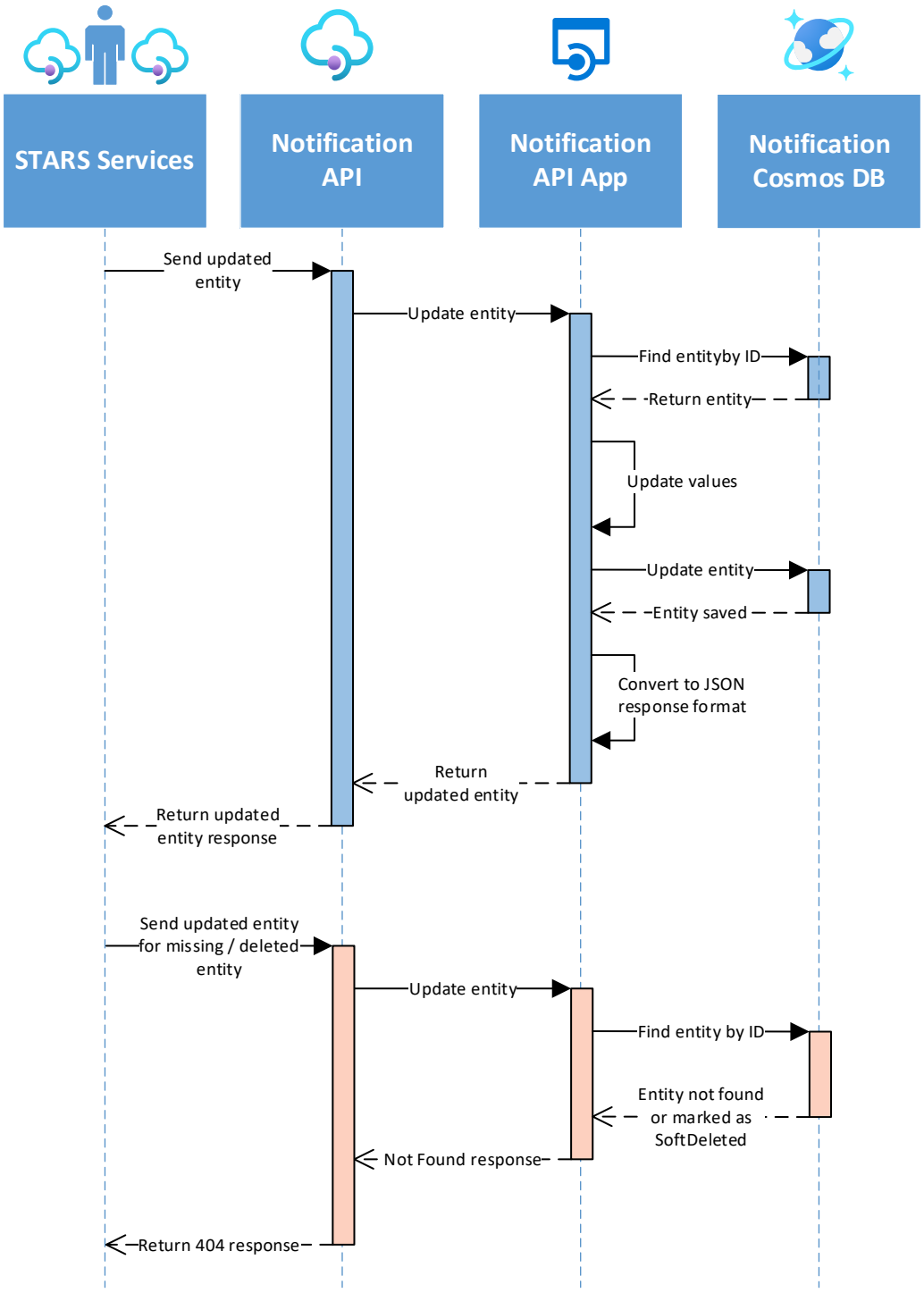


Figure 11: Update Entity (General) (D:9)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1006 of 1295

8.5.9 Delete Entity (General)

As a "General" operation, the implementation of this will cover all entities with Delete endpoints described in the 8.4.1 Notification Service Endpoints section above.

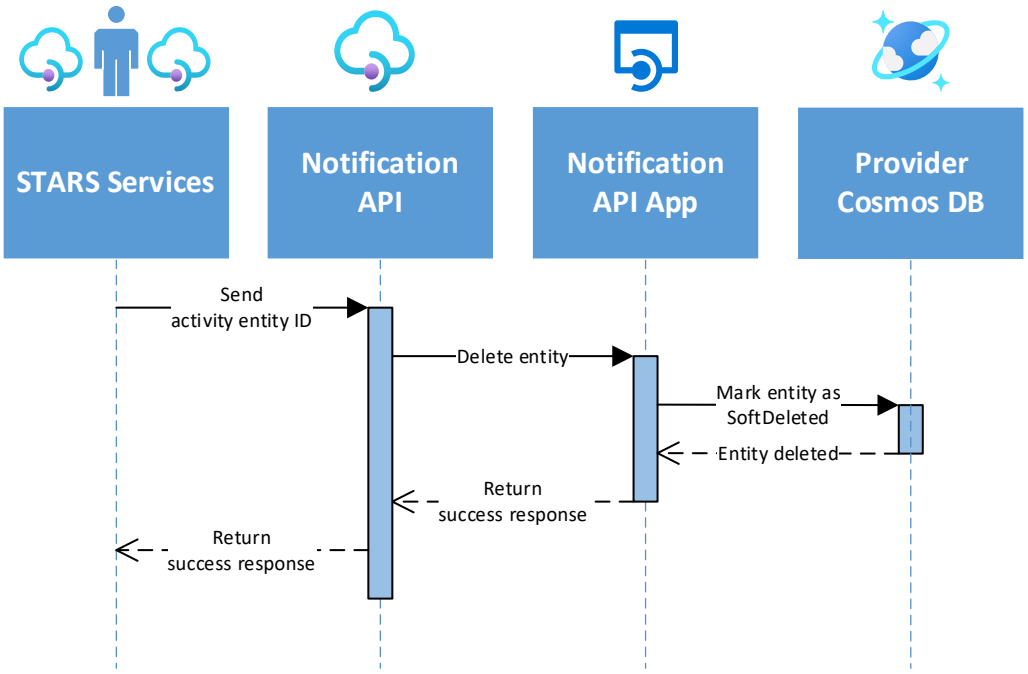


Figure 12: Delete Entity (General) (D:10)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1007 of 1295

8.5.10 Get List of Entities (General)

As a "General" operation, the implementation of this will cover all entities with Get Entities endpoints described in the 8.4.1 Notification Service Endpoints section above.

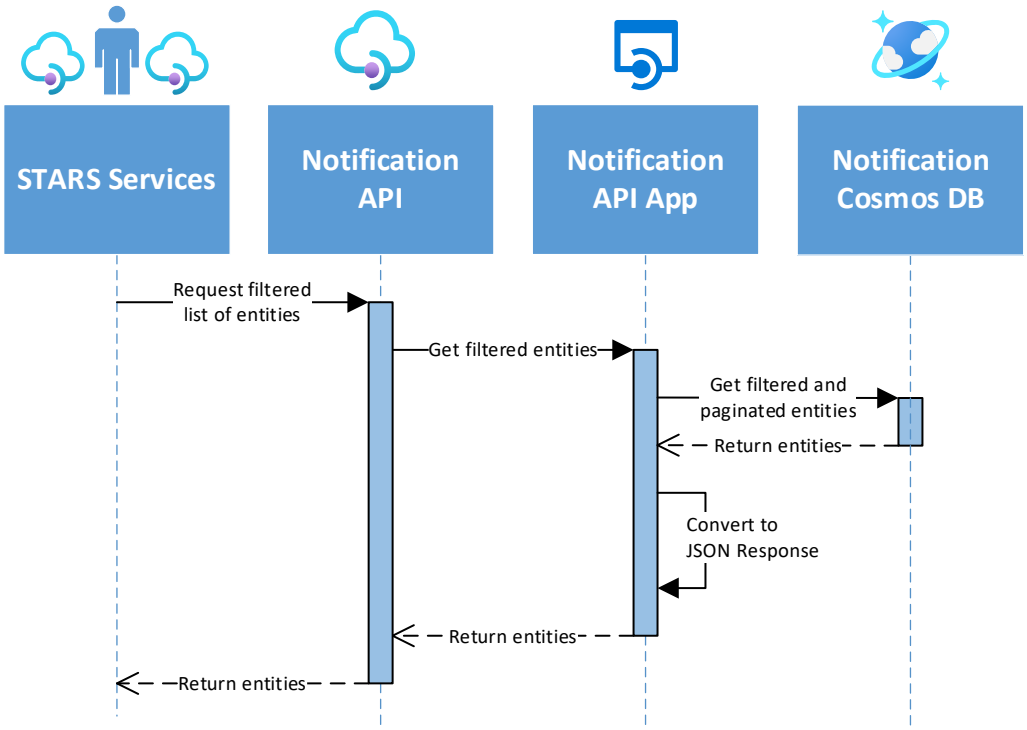


Figure 13: Get List of Entities (General) (D:11)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1008 of 1295

8.5.11 Get Entity by ID (General)

As a "General" operation, the implementation of this will cover all entities with Get by ID endpoints described in the 8.4.1 Notification Service Endpoints section above.

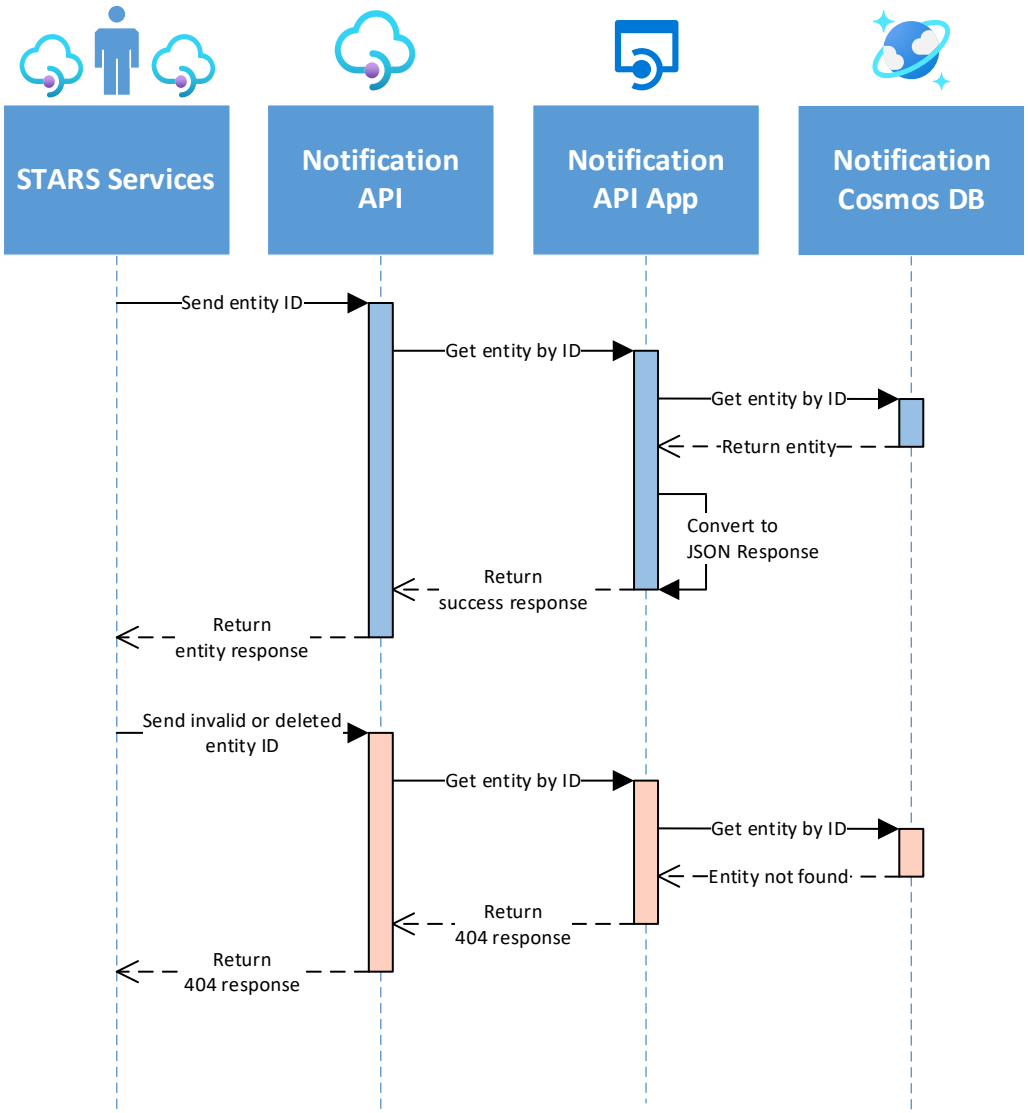


Figure 14: Get Entity by ID (General) (D:12)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1009 of 1295

8.6 Notification Events

Basic notifications that do not require template contents can be sent as determined by certain events, with recipient data attached. If template contents need to be set, the Notification API will be used instead.

Note: The Notification Service is a low-level service with few opinions on how notification data should be produced. Because of this, listening for events (e.g. On Data Submitted) and producing relevant notification is not the approach that will be taken. Instead, the Notification Service will rely on API endpoints that other services can call and provide data to. Using the API also allows the Notification Service to respond with a failure case e.g. if not enough data was provided to fill a template.

Table 79: Notification Events

Event Name	Comments
On Business Rule Changed	-
On Reference Data Changed	-
On Student Fees Changed	-

8.7 Notification Messaging

See 8.6 Notification Events. Service Bus messages will not be used by the Notification Service – the processing order of notifications will be independent.

SMSs will want to retrieve relevant notifications, but they will be able to do this via API calls, not via a message queue.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1010 of 1295

9 Technical Design Patterns

9.1.1 General Endpoint Descriptions

9.1.1.1 Read Endpoint – List

A GET endpoint that will return a list of entities with optional filtering. OData will be used for filtering and paginating results. Client-driven paging will be used for pagination.

9.1.1.1.1 Request Parameters

The parameters are used to control pagination and to filter the list of results.

Table 80: OData List Request Query Parameters

Query parameter	Comments
$top	Limits the number of results returned for this query.
$skip	The number of records to "skip over" when retrieving results. Used for pagination. If not included, default to 20.
$format=json	How the format of the response data. Should be set to "json".
$inlinecount=allpages	Ensures odata.count is included in the response with the total number of relevant results, regardless of pagination. Set to "allpages".
(filters)	Varies per endpoint. Values used to filter results to return.

9.1.1.1.2 Request Body

None.

9.1.1.1.3 Response Body

A JSON response is returned, with describing the number of results and how to retrieve adjacent pagination pages. The page of results is also returned.

Table 81: OData Response Fields

Field	Comments
odata.count	Total number of results (regardless of pagination).
value	List of relevant entities. Entity structure depends on individual endpoints.

```
{
  "odata.count": 155,
  "value": [
    {
      "entityID": 101,
      "field1": "value"
    }]}
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1011 of 1295

9.1.1.2 Read Endpoint – Detail

A GET endpoint for retrieving a specific entity.

9.1.1.2.1 Request Parameters

The parameters are used to control pagination and to filter the list of results.

Table 82: OData Detail Request Query Parameters

Query parameter	Comments
id	The unique identifier for the desired entity.

9.1.1.2.2 Request Body

None.

9.1.1.2.3 Response Body

A JSON object describing the relevant entity. The fields will depend on the entity relevant to each specific endpoint.

```
{
  "entityID": 101,
  "field1": "value"
}
```

9.1.1.3 Create Endpoint

A POST endpoint for creating a specific entity.

9.1.1.3.1 Request Parameters

None.

9.1.1.3.2 Request Body

The request body will be a JSON object that matches part of the relevant entity schema, containing the initial object values.

```
{
  "field1": "value",
  "field2": "another value"
}
```

9.1.1.3.3 Response Body

A JSON object describing the newly created entity. The fields will depend on the entity relevant to each specific endpoint.

```
{
  "entityID": 101,
  "field1": "value",
  "field2": "another value"
}
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1012 of 1295

9.1.1.4 Update Endpoint

A POST endpoint for updating a specific entity.

9.1.1.4.1 Request Parameters

None.

9.1.1.4.2 Request Body

The request body will be a JSON object that matches part of the relevant entity schema, containing the fields that need to be updated. The ID field is also provided to match a single entity.

```
{
  "ID": 101,
  "field1": "updated value",
  "field2": "updated value 2"
}
```

9.1.1.4.3 Response Body

202 Accepted response, with a JSON object describing the now-updated entity. The fields will depend on the entity relevant to each specific endpoint.

```
{
  "entityID": 101,
  "field1": "updated value",
  "field2": "updated value 2"
}
```

9.1.1.5 Delete Endpoint

A DELETE endpoint for deleting a specific entity (soft delete). This will ultimately set a "Deleted" flag on the entity which will no longer be returned in GET queries.

9.1.1.5.1 Request Parameters

None.

9.1.1.5.2 Request Body

The request body will be a JSON object that matches part of the relevant entity schema, containing the entity ID for the entity to delete.

```
{
  "ID": 101
}
```

9.1.1.5.3 Response Body

A 202 Accepted response, with a JSON object describing the deleted entity. The fields will depend on the entity relevant to each specific endpoint.

```
{"ID": 101, "field1": "value"...}
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1013 of 1295

10 Other Service Dependencies

10.1 Rules Service

The Rules Service will require an endpoint that the Notifications service can use to retrieve a list of all relevant Notification-based rules. It will also require an endpoint to check a rule result and return the pass / fail status. As the notification templates will be dependent on the reminder timing described by the rule, the Rules Service will also need to return a description of the timing for use in populating template placeholders.

The following table describes the required rules.

Table 83: Rule Dependencies

Rule Name
Reminder notification – Data soon due for submission
Reminder notification – Data is overdue for submission
Reminder notification – Data uploaded but not submitted
Reminder notification – Data quality checks failed
Reminder notification – Submission was rejected
Reminder notification – Submission has expired
Reminder notification – Pending warnings in submission

10.2 Submission Service

The Submission Service will be able to return a list of submitted data, which the Notification Service can use to call the relevant “overdue” rules on. This is how the Notification Service will know of the relevant people to contact – their preferences can then be linked up.

10.3 Master Data Service

Used to retrieve a list of relevant Trading Providers to send reminder emails to. Used to look up Training Provider email addresses and other relevant contact details.

10.4 Provider Service

SMSs will want to retrieve lists of relevant notifications via the Provider Service. The Provider Service will need to be able to query the Notification API.

10.5 Identity and Access Management Services

The Notification Service will store user preferences on how often each user should be contacted, and with which types of notification. These will need to be associated with users in Azure AD.

The Notification Service will need to know about user roles as defined in Identity and Access Management Services.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1014 of 1295

11 Access Management

The System Operator role will be in charge of managing and configuring the Notification Service data.

Every user will be able to manage their own notification preferences.

SMSs will only be able to access the Notification API via the Provider Service.

Users will only be able to retrieve notifications that are relevant to them. For example, users from an SMS can only retrieve notifications relevant to that SMS.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1015 of 1295

12 Appendices

12.1 STARS Solution Design Diagrams

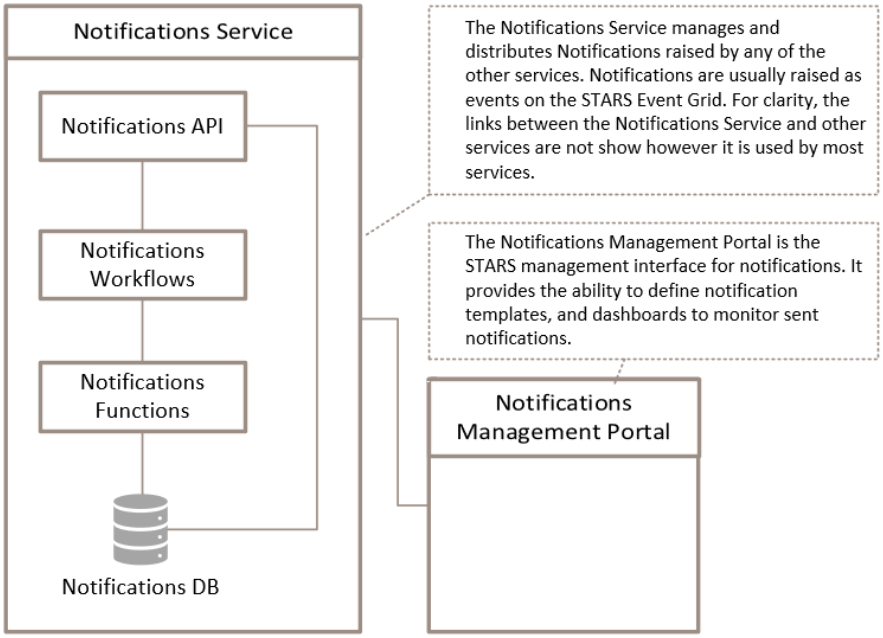


Figure 15: Appendix:Reference Notifications Service (E:12)

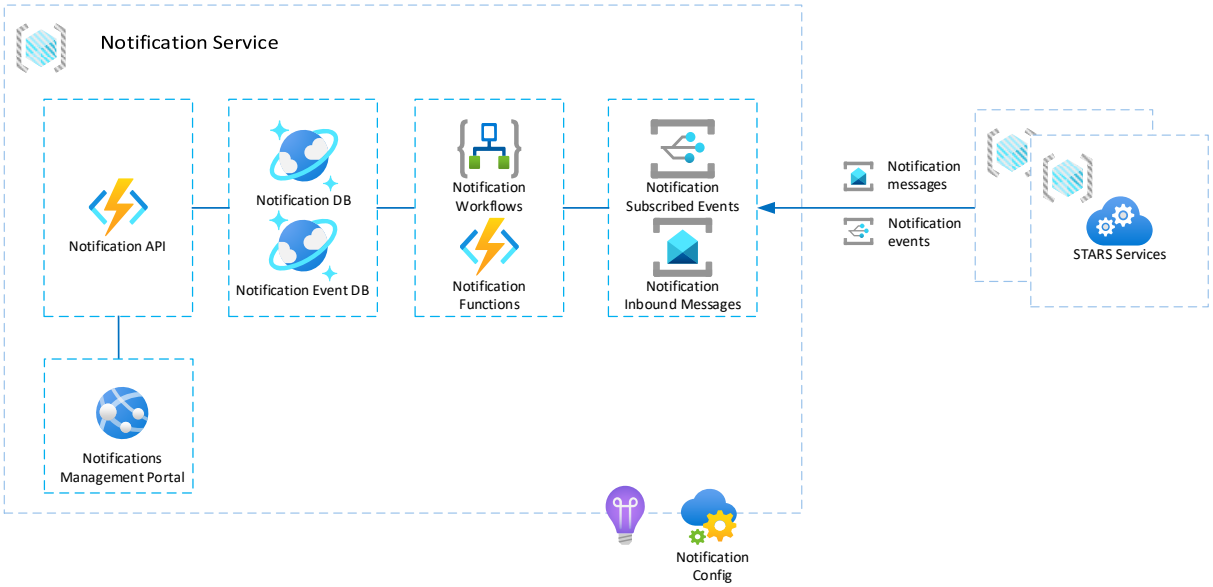


Figure 16: Appendix: Notification Service Implementation

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1016 of 1295

12.2 Example Notification Contents

12.2.1 Broadcast / Announcement

Upcoming scheduled maintenance
Please note that STARS will be unavailable from 2022-01-01 8:00pm to 2022-01-03 3:00am.

Unexpected service outage
Please note that STARS is experiencing unexpected downtime.

Outage report
STARS was unavailable from 2022-01-01 8:00pm to 2022-01-03 3:00am. The details of this outage are described below: {{OutageDetails}}

12.2.2 Data Submission Due

[STARS] Data submission due
Data submission for {{TP}} is now due. Please visit {{STARSUploadUrl}} to submit data. If no data submission is required for this period, send a nil to report submission at {{STARSNilReportUrl}}.

[STARS] Data submission overdue
Data submission for {{TP}} is now 7 days overdue. Please visit {{STARSUploadUrl}} to submit data. If no data submission is required for this period, send a nil to report submission at {{STARSNilReportUrl}}.

12.2.3 Asynchronous Submission Result

[STARS] Your submission has completed validation with errors
Data submission for {{TP}} The submission completed with errors and could not be submitted.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1017 of 1295

Please visit {{STARSValidationUrl}} for details. If a validation outcome review is required, submit a request at {{STARSValdationOutcomeReviewRequestUrl}}.

[STARS] Your submission has completed successfully, with warnings
Data submission for {{TP}} The submission completed without errors and has been submitted. There were warnings associated with this submission. Please visit {{STARSValidationUrl}} for details.

[STARS] Your submission has completed successfully
Data submission for {{TP}} The submission completed without errors and has been submitted. Please visit {{STARSValidationUrl}} for details.

12.2.4 General

[STARS] {{MessageHeader}}
{{MessageContents}}

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1018 of 1295

13 Acronyms

Table 84: Acronyms

Acronym	Definition
API	Application Programming Interface
CRUD	Create Read Update Delete
REST	Representational State Transfer
FR	Functional Requirement
NFR	Non-functional Requirement
STARS	Student Training & Activity Reporting System
VET	Vocational Education Training
VDS	VET Data Streamlining
IVETS	Integrated Vocational Education and Training System
DB	Database
DDS	Data Definition Language
SQL	Structured Query Language
TP	Training Provider
STA	State Training Authority
SMS	Student Management

14 Glossary

15 Terminology

The term “Training Organisation” has an identical meaning to “Training Provider”. For consistency, Training Provider (TP) is used in this document unless referred to as the former in Requirements text.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1019 of 1295

Australian Government
Department of Employment and Workplace Relations

Detailed Design Document

Validation Service

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1020 of 1295

1 Document Management

1.1 Document Control

Version	Change Description	Date
D.0.1	Initial Draft	17/02/2023
D.0.2	Updated Draft following stakeholder feedback	03/03/2023
1.0	Submitted for Acceptance	4/4/2023
2.0	Added All STARS System Components table and RTO/SMS engagement activities statement Accepted	28/04/2023

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This is a living document and will be updated as required. If another component's design scope requires an update to this design document, then the appropriate change will be made and a new version issued.

1.2 Document Author

Role	Name
Design Lead	s 47F(1)

1.3 Document Reviewers

Role	Name

1.4 Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

1.5 Related Documents

Reference	Document
A	STARS Solution Requirements V1.3.docx
B	STARS Solution Design - V.1.0.docx
C	STARS Solution Design - Attachment 1 - Collaboration Diagrams - V.1.0.vsdx
D	STARS Solution Design - Attachment 3 - API Design - V.1.0

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1021 of 1295

Reference	Document
E	Validation Service reference diagrams v1.0.vsdx
F	STARS Validation Service API v1.0.pdf
G	ValidationAPI v1.0.json
H	RelavantRequirementsForValidation v1.0.xlsx

1.6 All STARS System Components

This list outlines the high-level service dependencies with **Validation Service**. This is a reflection of what is known at the time of Acceptance and may be subject to change throughout the design process.

Green = This Design Document Grey = Other Service Dependencies

STARS Service
Azure Infrastructure and Networking
Identity & Access Management (IAM)
Event Service
Logging & Audit Services
Master Data Service
Reference Data Service
Data Governance Infrastructure
Data Quality Framework
Notification Service
Submission Service
Validation Service
Rules Service
Provider Service
Mastering Service
Downstream Consumer ETL Pipeline Services
Release Service
Downstream Consumer Reporting & Analytics
R & A Provisioning Service
STA Batch Submission Services
STA Gateway Services
USI Transcript Service

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1022 of 1295

STARS Service
STA Bulk Data Consumption Service
SMS SDK Service

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1023 of 1295

2 Contents

1 Document Management 2
1.1 Document Control 2
1.2 Document Author 2
1.3 Document Reviewers 2
1.4 Document Approval 2
1.5 Related Documents 2
1.6 All STARS System Components 3
2 Contents 5
3 Introduction 7
3.1 Purpose 7
3.2 Intended Audience 8
4 Scope and Exclusions 9
4.1 In Scope 9
4.2 Out of Scope 9
5 Design Decisions 10
6 Requirements 11
7 Feature Overview 12
7.1 Validation Service 13
7.2 Validation API 13
7.3 Validation Workflow 13
7.4 Validation Functions 14
7.5 Validation Rules Engine 14
7.6 Validation DB 14
8 Business Processes 15
8.1 Validation Request 15
8.2 Validation Data Query 15
9 Technical Architecture 16
9.1 Validation API 16
9.2 Validation Workflow 17
9.3 Validation Rules Engine 17
9.4 Validation DB 17
9.5 Validation Config 17
10 API Design 18

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1024 of 1295

10.1 Validation Request Endpoint 18
10.2 Validation Result Endpoint 18
10.3 Validation Dataset Endpoint 18
11 Validation Service Workflows 19
11.1 Fully Integrated Workflow 19
11.2 API Validation Request 21
11.3 Workflow Validation Request 22
11.4 API Validation Result 23
11.5 Workflow Collation Request 24
11.6 Validation Event Service Messages 25
12 Data 26
13 Security and Access Management 27
14 Acronyms 28

Figure 1: Decision Services (E4) 12
Figure 2: The Validation service business process (E3) 15
Figure 3: Validation Service Architecture (E2) 16
Figure 4: Technical Validation Flow (E1) 19
Figure 5 Validation request API workflow (E1) 21
Figure 6 Validation request workflow for validation workflow (E1) 22
Figure 7 Validation API result endpoint workflow (E1) 23
Figure 8 Validation collation request workflow (E1) 24

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1025 of 1295

3 Introduction

The purpose of this document is to capture the detailed design requirements for the STARS Validation service solution.

The Validation service is responsible for:

- Processing validation requests from any of the other internal STARS solution services as well as from external sources (through the Provider service) and providing them with Validation results.
- Providing an interface for external validation providers to integrate their validation results into the STARS solution
- Allowing the monitoring of validation request progression and querying of validation requests and validation results
- Maintaining a record of all validations processed by the service

Validation Requests are put in through API calls and the API is also responsible for querying progression of current and historical API calls and results. It uses the Event service to notify other services when a result is ready.

As a Decision Service, it provides the means for Submission pipeline services and their embedded workflows to control the flow of data within the system.

Note: This is a living document that will be updated as designs for various other STARS services are developed and evolve. Accordingly, this document will evolve to reflect and document validation components as the STARS solution build progresses.

The design is informed by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework, which provide a comprehensive set of best practices and guidance for designing, deploying, and operating secure, high-performing, and resilient cloud infrastructure and applications.

3.1 Purpose

The purpose of this detailed design document is to provide a technical description of the Azure validation components of the STARS solution. The document is intended for technical stakeholders, including architects, software engineers, administrators, and operations teams, who are responsible for designing, developing, deploying, and operating the STARS solution in Azure.

The document also serves as a reference for ongoing management and evolution of the STARS solution and to ensure that the STARS solution is deployed and operated in accordance with the best practices and guidance provided by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1026 of 1295

The document is intended to be a living document, updated as the STARS solution evolves and changes over time. It provides a clear and concise understanding of the Azure Validation components and their configurations, helping to ensure the STARS solution is deployed, operated, and managed in a consistent and efficient manner.

With respect to expected RTO/SMS engagement activities, it is DXC's understanding based on the current requirements and solution design, noting that DXC have not yet conducted detailed consultation with the RTO/SMS Providers, that there is no component in this Deliverable that directly impacts RTO/SMSs. All direct impacts to RTO/SMSs are dealt with in the Provider Portal, Provider Service and/or SMS SDK Service as understood and agreed previously.

3.2 Intended Audience

This detailed design document provides technical information and guidance to help technical stakeholders understand the Azure validation components of the STARS solution and how they integrate with each other. The document is expected to be used by technical stakeholders who have a good understanding of cloud computing concepts and are experienced in the Azure platform.

This document is not intended for non-technical readers, nor for readers who are not experienced in the design and configuration of Azure-based solutions; informational and educational resources regarding Azure are readily available at https://azure.microsoft.com

The primary audience for this document includes:

- Program technical staff: This includes architects, software engineers, administrators, and operations teams within the program responsible for the STARS solution.
- External technical stakeholders: This includes technical staff from other federal, state, and territory governments agencies, including VDS embedded team members.
- ICT Technical Staff: This includes a variety of technical roles involved during the VDS program and post-go live, including network engineers, systems administrators, database administrators, security engineers, and DevOps engineers.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1027 of 1295

4 Scope and Exclusions

4.1 In Scope

- Validation service API specification
- Validation service workflow
- Validation service Data repository and data schema
- Interaction of validation service with STARS and external services

4.2 Out of Scope

- Rules service and rules workflow, this will be covered by the Rules service detailed design document
- Specification of any service that validation service is dependent on or any service that is reliant on the validation service. Only interaction mechanisms such as data transfer objects will be defined. Other services will be defined in respective detailed design documents.
- Definition of external validation provider results. Only the specifications for them to be integrated into the validation report will be specified.
- User interface elements based on results produced by the validation service. These will have their respective detailed design documents.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1028 of 1295

5 Design Decisions

STARS will have a single service that is responsible for all validation request so that request and responses to those requests are consistent, maintainable and scalable. The design tries to be as efficient as possible while still maintaining modularity and focuses on providing as much information as early as possible.

This service will be accessible by all other services to request validations based on a submission and a ruleset with the optional ability specify a rule to select a subset of the submission data. All validation history is saved by the service, and they can be queried using the API.

The service will also be able to interact with external validation providers to keep the validation solution as flexible as possible. Keeping with philosophy of providing as much information as possible early on the service will provide partial results as soon as it had them available.

External providers will be provided access to this service through the provider service.

For a broad overview, see Reference B: STARS Solution Design. The high-level diagrams relevant to the Validation Service are also included in the Appendices section of this document.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1029 of 1295

6 Requirements

The list of requirements relevant to the Validation Service detailed design can be found in related document H Requirements relevant to Validation Service.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1030 of 1295

7 Feature Overview

The validation service is part of Decision services module of the STARS solution. The two main components of Decision services are the Rules Service and the Validation service. Please refer to the To Be STARS services in Reference C (Collaboration diagrams) for a view of where Decision services is placed in the whole of the STARS solution.

Note: That the Validation service applies to all records and data submitted into STARS regardless of source i.e., RTO through SMS, RTO direct, STA or external reference data.

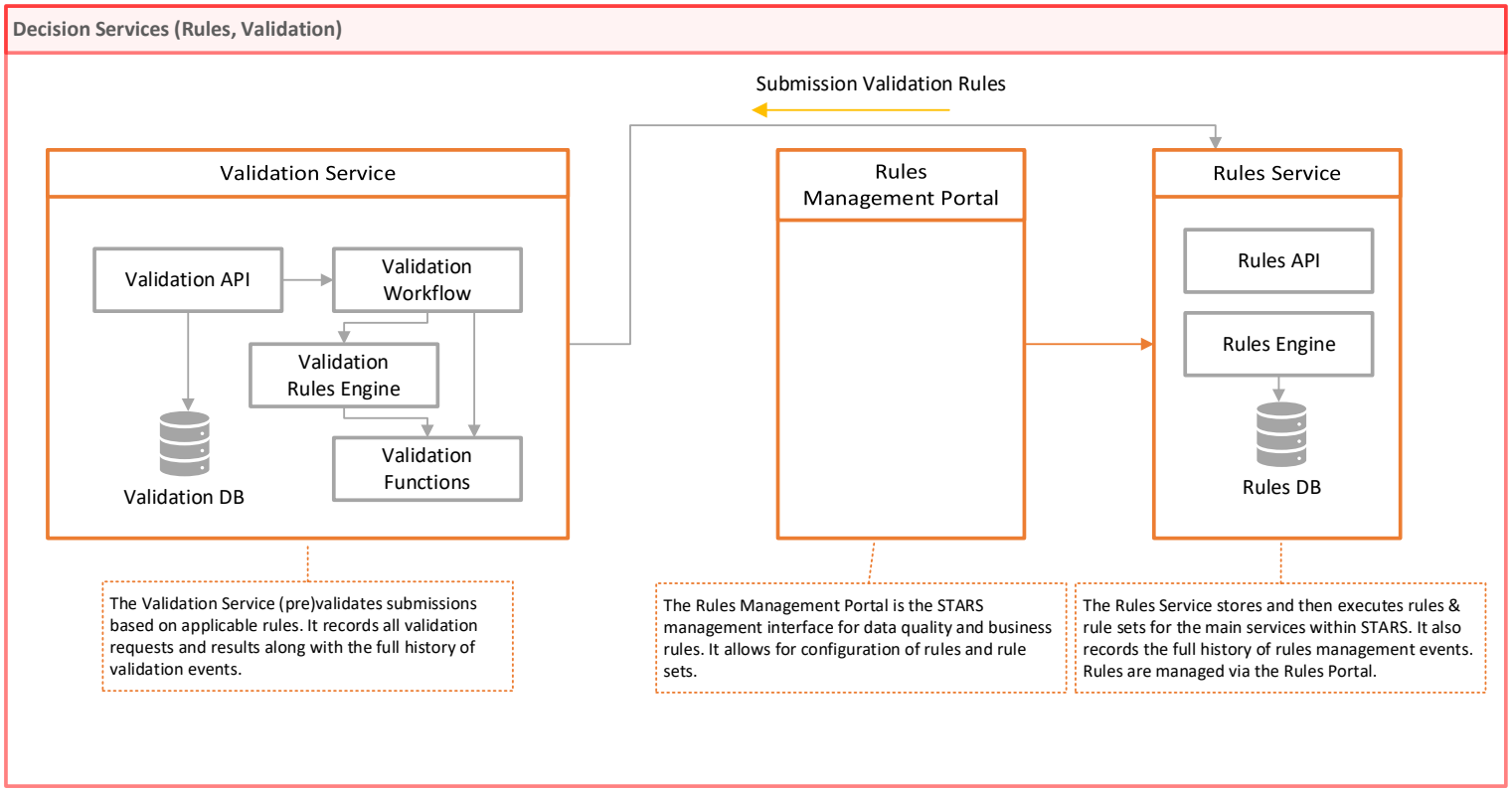


Figure 1: Decision Services (E4)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

7.1 Validation Service

The Validation Service uses the rules engine in the rules service to run relevant validation rules to (pre)validate submissions based on applicable rules (sourced from and run by the Rules Service). It records all validation requests and results along with the full history of validation events.

The Validation Service validates submissions using the STARS Rules Engine but can also call out to external STA-based validation services. External validation is controlled by the Validation Workflows, with specific workflows configured to cater for different STA external validation requirements.

The validation service can also be used by entities external to the STARS solution using the Provider services as a mediator.

Note that Validation Service managed external validation is separate and distinct from STARS internal validation by the rules engine in the rules service. The Validation Service external STA validation scenario only applies when validation requirements cannot be translated into rules for the internal rules engine in the rules service. The internal rules engine in the rules service will attempt to provide as close to real-time results as possible, however, the external validation providers are by definition external to STARS and therefore may not be able to do so.

7.2 Validation API

The API will allow other services to present submission and rule information in the form of a Validation Request object in a JSON format for submission validation. The information will include the identifiers of the relevant data held within either the Submission Service or the provider service, to allow retrieval of the submission data for validation processing.

It will also have endpoints to allow external and internal validation providers to submit validation result data.

Monitoring of progress of current validation requests and querying of historical validation data will be facilitated though further read only endpoints.

The API will conform to the STARS Design Principles and the STARS API Design Document.

7.3 Validation Workflow

The Validation Service internal processes will be controlled by Azure Logic App workflows.

Upon receipt of a validation request by the Validation API, a workflow will be triggered. The workflow will examine the metadata within the request to determine the nature of the request and decide which national and/or STA validation workflows apply.

The validation workflows will route the request through the appropriate data quality and business rule sets, managed and executed by the Rules Service. During this process, the workflows will retrieve the relevant submission data from the Submission Service, passing the data into the Rules Service via the Rules API.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1032 of 1295

When the Rules Service has finished and returned the rules execution results, the workflows collate and format the results in a form suitable for reporting back to the calling service.

When required, the validation workflows can also route submission data to STA validation web services, via the STA Service Gateway. Returned STA validation results will also be collated with the results of any STARS-hosted rules.

When relevant, STAs will validate submissions using their own internal systems. Validation results will be transmitted back to the staging service, utilising the STA Gateway Service.

The STA Gateway Service will call the Validation API (below) to inform the Validation Service of STA validation results. The gateway will have a consistent interface on the STARS side (i.e., it sends and receives messages using STARS schemas, transports and protocols), but it has various adaptors on the STA side to facilitate data and message exchange with external STA validation systems.

7.4 Validation Functions

The Validation Service code-based functionality will be implemented as Azure Function Apps. These functions will support the operation of the service's workflows.

7.5 Validation Rules Engine

The Validation Rules Engine, and any associated rules, will assist with the operation of the Validation Workflows. Please note that this does not include submission validation rules – these are handled by the Rules Service directly. The validation rules engine role is to apply any business rules associated with the validation workflow itself.

7.6 Validation DB

All data relating to validation requests, validation results and validation datasets will be recorded in the Validation Cosmos DB.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1033 of 1295

8 Business Processes

8.1 Validation Request

The flow diagram below represents the validation service business process of processing a validation request. The validation service uses the rules engine in the rules service to run relevant validation rules to validate the data specified in the request using the rules specified in the request. It stores both the request and the result, corelates them to form a validation report and then returns that to the requester.

It should be noted that the rules engine in the rules service is responsible for running rules that are required for data validation or in the case of external validation providers the external validation service is required to validate the data and return a response in the format specified in the API details validationResults endpoint as specified in related document F (STARS Validation Service API D.0.2.pdf) depending on the details of the request.

Validation requests can come in from other services in the STARS solution or from external sources through the Provider Service.

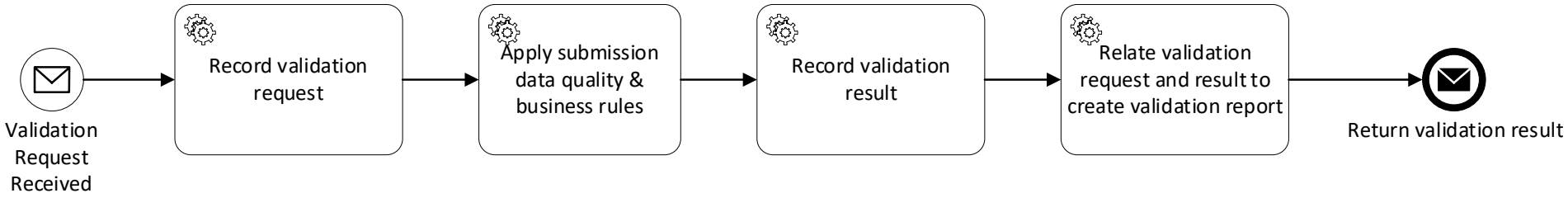


Figure 2: The Validation service business process (E3)

8.2 Validation Data Query

This is a much simpler process where the validation service returns data on validation requests and results according to specified query parameters. This also serves record keeping needs.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

9 Technical Architecture

The technical architecture of the Validation service is depicted in the diagram below. Note the interaction with the Rules Service, Application config and App insights for the STARS solution.

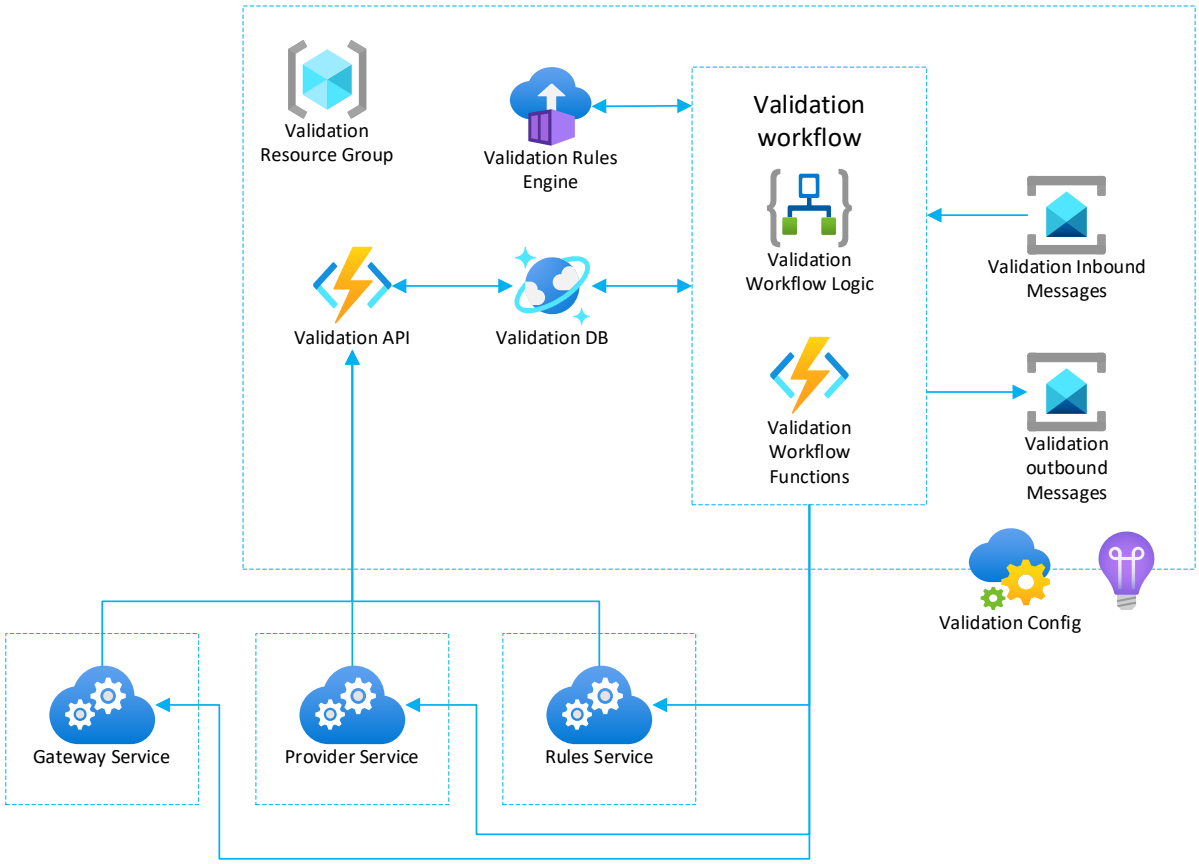


Figure 3: Validation Service Architecture (E2)

All incoming requests to be handled by the validation service comes in through the Validation API. External requests to the Validation API are mediated through the Provider Service.

The validation workflow makes calls to the rules service to validate submission data and calls to external validation providers are mediated through the STA gateway service. The rules service also provides the Validation service with validation workflow rules.

9.1 Validation API

The Validation API will be implemented as an Azure Function App, utilising HTTP endpoints. It will be responsible for

- Queuing incoming validation requests on to the Event Service to be consumed and processed by the Validation workflow
- Accepting validation results posted to it by validation providers

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1035 of 1295

- Responding to queries relating to validation requests, validation results and validation datasets

9.2 Validation Workflow

The Validation workflow component will be implemented using an Azure Logic app with use of Azure functions for some more complicated functionality.

9.3 Validation Rules Engine

The validation rules engine will be used by the validation workflow logic app to evaluate business rules related to the validation processes used by the validation workflow. It will use the Drools rule engine and be hosted in an azure container.

9.4 Validation DB

The validation DB will store all incoming validation requests and related validation results. It will also store identifiers to identify exactly what subset of the submission dataset the specified validation rules were executed on. It will be implemented using Azure Cosmos DB

9.5 Validation Config

The Validation Config (Key Vault) will store values for configuring the service. Note that common config values that every service will require will be defined in a separate, common location.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1036 of 1295

10 API Design

A Detailed API specification and Open API definition can be found in related documents F and G. JSON object definitions are contained in related document F

Here we will highlight the primary endpoints and their functions. Please note that the API will be implemented using OData. Please refer to reference document D for details on STARS API implementation.

10.1 Validation Request Endpoint

This endpoint is responsible for receiving validation requests and allowing the querying of pending and completed validation requests.

- It receives validation requests through an OData post action with a CreateValidationRequest JSON object.
- It allows for querying using OData and returns ValidationRequest JSON objects for pending and completed validation requests

10.2 Validation Result Endpoint

This endpoint is responsible for receiving validation requests and allowing the querying of collated and uncollated validation results.

- It receives validation results through an OData post action with a ValidationResultSubmission JSON object.
- It allows for querying using OData and returns ValidationResult JSON objects for all Validation Results received

10.3 Validation Dataset Endpoint

This endpoint is responsible for allowing the querying of validation datasets.

- It allows for querying using OData and returns ValidationDataSet JSON objects for all Validation datasets received

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1037 of 1295

11 Validation Service Workflows

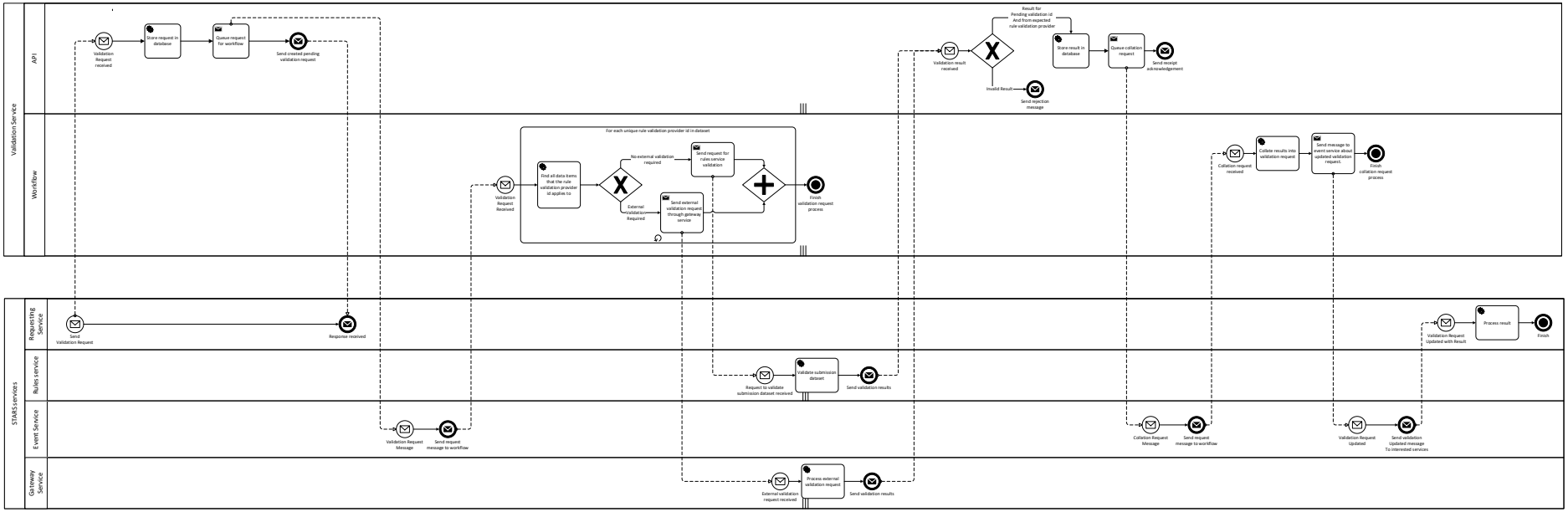


Figure 4: Technical Validation Flow (E1)

11.1 Fully Integrated Workflow

The fully integrated workflow can be found in related document E which is part of the detailed design pack. Please refer to it for a visual representation of the following steps. The sequence of events in the fully integrated workflow are as follows please

- A HTTP request is received at the Validation API
    - The API stores the request to the database separating the dataset in the request for efficiency purposes.
    - The API queues a request with Validation Workflow using the event service
    - The API sends back a ValidationRequest with a validation Id to the calling service for the service to query results later

- The Message from the Event Service is received by the Validation Workflow component
    - The workflow then batches out the dataset by validation providers (whether internal or external) and sends them a request to validate.
    - The request to the external provider is mediated by the STA Gateway service and the details of this will be provided in the gateway service detailed design.
- Once a validation provider is done with processing a request it sends a HTTP post with a ValidationResult object to the API result endpoint. External validation providers will have this mediated thought the STA Gateway service.
    - The API checks if the result is one that it was expecting
        - It expects results that are in the expectedResultValidationProviders array without received being true.
        - If the result wasn't expected return a bad request result
    - The API stores the ValidationResult in the database
    - The API creates a collation request message in the event queue
    - The API returns an accepted response
- When the collation request is received by the workflow it
    - collates the result by adding the result id to the Validation Request and setting the collated field in the Validation result to true
    - creates a result updated event in the event service

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

11.2 API Validation Request

Please refer to fully integrated workflow in document E1 before this section about an individual workflow component.

The workflow below is the followed by the Validation API on receiving a Validation Request at the API post endpoint validations/request. The format of the request can be found in document F.

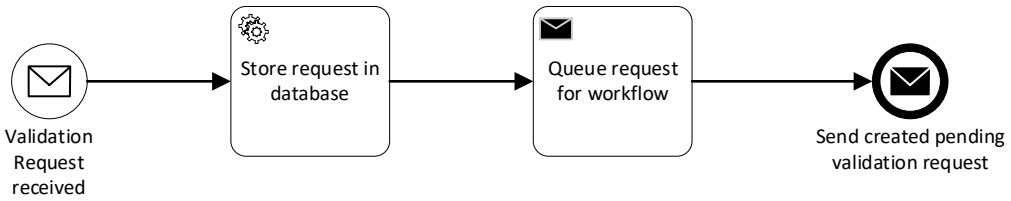


Figure 5 Validation request API workflow (E1)

Points to note:

- ValidationCreateRequest object is separated into a ValidationRequest Object and the ValidationDataset object before being stored into the DB. This is to reduce the amount of data exchanged while monitoring a request.
- The queued message for workflow uses the Event Service to queue it.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

11.3 Workflow Validation Request

Please refer to fully integrated workflow in document E1 before this section about an individual workflow component.

This is the workflow for a validation request message being processed by the workflow component of the validation service.

Note that:

- As many requests are sent at there are validation providers in the request
- The same data items may be validated by different rules by different validation providers

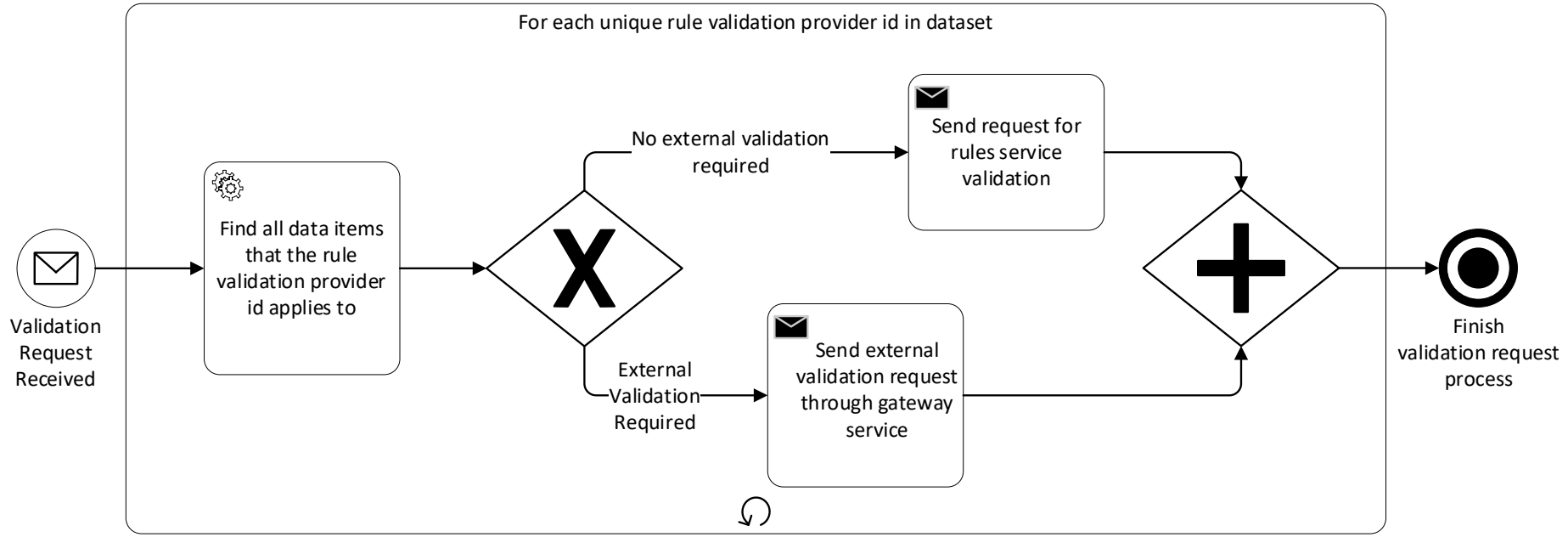


Figure 6 Validation request workflow for validation workflow (E1)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

11.4 API Validation Result

Please refer to fully integrated workflow in document E1 before this section about an individual workflow component.

This is workflow for when a validation result is returned by a validation provider to the Validation API validation/result post endpoint.

Note that:

- Request is only accepted if it is an expected result. An expected result is one that is from a validation provider that a result is expected from and if the result is still not received from that provider.
- Event Service is responsible for queueing the collation request

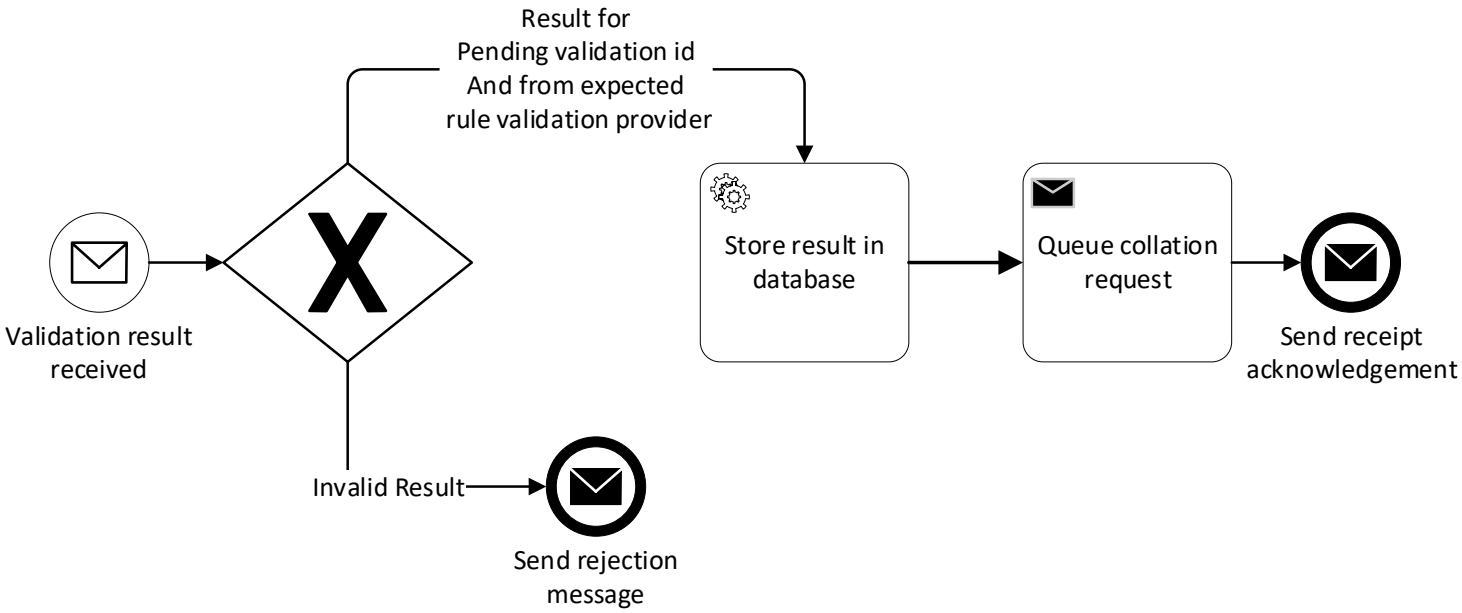


Figure 7 Validation API result endpoint workflow (E1)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

11.5 Workflow Collation Request

Please refer to fully integrated workflow in document E1 before this section about an individual workflow component.

This is the collation request workflow for the validation service.

Note that:

- Collation adds the result id to the list of results in the validation request that the result is related to.
- It is also responsible for setting the ValidationRequest\ expectedResultValidationProviders\{validationRuleProviderId}\received value to true to indicate that result was received from that Validation provider, and it is not pending.

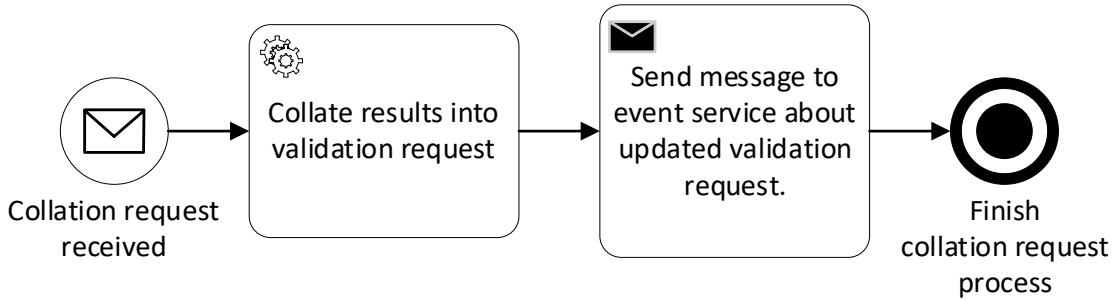


Figure 8 Validation collation request workflow (E1)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

11.6 Validation Event Service Messages

Below is a list of Validation service messages to the event service

Name	Purpose
Validation Request	Message to request new validation from validation service
Collate Validation Result Request	Message from validation service to itself to collate a validation result into a validation request
Validation Result Updated	Message to any services interested in the result of a particular validation request

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1044 of 1295

12 Data

Please refer to document F for all data schemas and examples.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1045 of 1295

13 Security and Access Management

The validation service is an internal service. Access to the Validation service from systems external to the STARS solution will be mediated through other services such as the Provider service and the Gateway Service.

The service will conform to the Security Architecture. Access to the Validation Service will be controlled by Azure RBAC, with all calling services identified and granted access through Managed Identities.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1046 of 1295

14 Acronyms

Acronym	Definition
API	Application Programming Interface
CRUD	Create Read Update Delete
REST	Representational State Transfer
FR	Functional Requirement
NFR	Non-functional Requirement
STARS	Student Training & Activity Reporting System
VET	Vocational Education Training
VDS	VET Data Streamlining
IVETS	Integrated Vocational Education and Training System
DB	Database
DDS	Data Definition Language
SQL	Structured Query Language
TP	Training Provider
STA	State Training Authority
SMS	Student Management

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1047 of 1295

Australian Government
Department of Employment and Workplace Relations

Detailed Design Document

Audit & Logging Services

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1048 of 1295

1 Document Management

1.1 Document Control

Version	Change Description	Date
0.1	First Draft	09-02-2023
0.2	Addressed Feedback Comments	23-02-2023
1.0	Submitted for Acceptance	03-04-2023
2.0	Added All STARS System Components table and RTO/SMS engagement activities statement Accepted	28-04-2023

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This is a living document and will be updated as required. If another component's design scope requires an update to this design document, then the appropriate change will be made and a new version issued.

1.2 Document Author

Role	Name
Design Lead	s 47F(1)

1.3 Document Reviewers

Role	Name

1.4 Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

1.5 Related Documents

Reference	Document
A	STARS Solution Requirements Document – V.1.3.docx
B	STARS Solution Design – V.1.0.docx
C	STARS Solution Design - Attachment 3 - API Design - V.1.0.docx
D	Event , Logging and Audit v1.0.vsdx

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1049 of 1295

1.6 All STARS System Components

This list outlines the high-level service dependencies with **Audit and Logging Service**. This is a reflection of what is known at the time of Acceptance and may be subject to change throughout the design process.

Green = This Design Document Grey = Other Service Dependencies

STARS Service
Azure Infrastructure and Networking
Identity & Access Management (IAM)
Event Service
Logging & Audit Services
Master Data Service
Reference Data Service
Data Governance Infrastructure
Data Quality Framework
Notification Service
Submission Service
Validation Service
Rules Service
Provider Service
Mastering Service
Downstream Consumer ETL Pipeline Services
Release Service
Downstream Consumer Reporting & Analytics
R & A Provisioning Service
STA Batch Submission Services
STA Gateway Services
USI Transcript Service
STA Bulk Data Consumption Service
SMS SDK Service

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1050 of 1295

Contents

1 Document Management 2
1.1 Document Control 2
1.2 Document Author 2
1.3 Document Reviewers 2
1.4 Document Approval 2
1.5 Related Documents 2
1.6 All STARS System Components 3
Contents 4
2 Introduction 6
2.1 Purpose 6
2.2 Intended Audience 6
3 Scope and Exclusions 8
3.1 In Scope 8
3.2 Out of Scope 8
4 Design Decisions 9
5 Requirements 10
5.1 Functional 10
5.2 Non-Functional 10
6 Feature Overview 12
7 Audit Service Design 14
7.1 Portal and Middleware Audit Features 14
7.1.1 Access Logs 14
7.1.2 Custom Logs with Application Insights 14
7.1.3 Business Rules Engine (Drools) Audit Logs 17
7.2 Data Audit Features 18
7.3 User Access and Security Audit Features 18
8 Azure Log Analytics Reporting and Analysis 19
9 Performance Impacts and Service Limits 20
10 Technical Design Patterns 21
10.1 Audit Service Api EndPoint Description 22
11 Access Management 24
12 Appendix 25
12.1 Default Dimensions for customEvents table 25

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1051 of 1295

12.2 Default Dimensions Front Door Access Logs 27
12.3 Design Tasks and Decisions 31
13 Acronyms 32

Figure 1: Custom Telemetry Data Collection using App Insights 21

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1052 of 1295

2 Introduction

2.1 Purpose

The purpose of this document is to describe the features within Azure that will be enabled or customized to fulfill the Audit and Logging related requirements for the STARs system services.

The Audit and Logging capabilities will allow the system to capture details related to various user actions and functions within the system. The logs captured shall be immutable and retained for the purpose of analysis or responding to any security or audit related incidence. These logs shall also provide insights into the systems overall performance and help uncover any issues that may have a detrimental impact on the overall system performance and responsiveness.

Note: This is a living document that will be updated as designs for various other STARS services are developed and evolve. Accordingly, this document will also evolve to reflect any review feedback related changes.

With respect to expected RTO/SMS engagement activities, it is DXC's understanding based on the current requirements and solution design, noting that DXC have not yet conducted detailed consultation with the RTO/SMS Providers, that there is no component in this Deliverable that directly impacts RTO/SMSs. All direct impacts to RTO/SMSs are dealt with in the Provider Portal, Provider Service and/or SMS SDK Service as understood and agreed previously.

2.2 Intended Audience

This detailed design document provides technical information and guidance to help technical stakeholders understand the Azure IDAM services used in the STARS solution and how they integrate with the solution components. The document is expected to be used by technical stakeholders who have a good understanding of cloud computing concepts and are experienced in the Azure platform.

This document is not intended for non-technical readers, nor for readers who are not experienced in the design and configuration of Azure-based solutions; informational and educational resources regarding Azure are readily available at https://azure.microsoft.com

The primary audience for this document includes:

- Program technical staff: This includes architects, software engineers, administrators, and operations teams within the program responsible for the STARS solution.
- External technical stakeholders: This includes technical staff from other federal, state, and territory governments agencies, including VDS embedded team members.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1053 of 1295

- ICT Technical Staff: This includes a variety of technical roles involved during the VDS program and post-go live, including network engineers, systems administrators, database administrators, security engineers, and DevOps engineers.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1054 of 1295

3 Scope and Exclusions

3.1 In Scope

The following are in-scope of this detailed design solution:

1. Portal and Middleware Audit Logging Capabilities.
2. Data Layer Audit.
3. User and Security Audit is only for informational purpose. Implementation is beyond scope of this document as it will be managed by OOBE.

3.2 Out of Scope

The following are out of scope:

1. Details about infrastructure related logs that are enabled to measure system efficiency or alert system admins
2. Details about ISM controls as they relate to Security and Infrastructure.
3. Network Security and Audit.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1055 of 1295

4 Design Decisions

The following decisions have been agreed upon

1. The audit and logging services will be providing the framework for other services to consume. Core STARs services will be leveraging the framework to log details about appropriate events. The decision around which events need to be logged shall be driven by the functional requirements and these are addressed by the respective services.
2. The service shall be designed to leverage capabilities within Azure Monitor. There is no intent to build a service from the ground up, but instead to ensure existing fit for purpose services are leveraged within the Azure space for delivering the capability.
3. Given that Audit and Logging is a requirement for many areas within the solution, it is not intended to cover every aspect of those requirements in this design. Instead, this design focuses primarily on the portal, middleware and data layer audit capabilities.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1056 of 1295

5 Requirements

5.1 Functional

This document provides the framework for addressing any audit and logging requirements for the portal, middleware and data layer components. Any business or functional requirements to log specific events rest with those specific services and shall be covered in the scope of those detailed designs.

5.2 Non-Functional

FR No	Requirement Name	Description
NFR013a	Audit	The VDS System will be able to log activities and provide audit information as needed. • The VDS System will be able to generate all audit information supporting all historical logging for a specified user function.
NFR013b	Audit	• The VDS System will be able to generate a version of the Business Rules.
NFR013c	Audit	The VDS System will be able to generate a record of all changes to Business Rules, including the specific changes that were made, the user that undetraining organisationok the change, and the date and time the change was made.
NFR013d	Audit	• The VDS System will be able to generate an audit log describing when a user accessed functionality with the integrity of such audit logs maintained to prevent any deletion or modification.
NFR013e	Audit	• The VDS System will be able to generate an audit log describing what information was accessed by who, when and what action was undertaken.
NFR013f	Audit	The VDS System will allow for • Systematic sampling • Program monitoring • Segment level compliance reports for programs

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611

6 Feature Overview

Azure Audit and Logging provides a centralized repository for logs and audit data generated by various Azure services and resources. The data collected includes activiy logs, resource logs, diagnostic logs, and platform logs, which can be used to monitor, diagnose, and troubleshoot issues, as well as meet compliance and regulatory requirements.

Activity logs: This type of log records information about the operations performed on resources in Azure, such as creating or deleting a virtual machine, or updating the configuration of a storage account. Activity logs provide information on the who, what, when, and where of operations.

Resource logs: These logs are generated by resources within Azure, such as virtual machines, storage accounts and databases. The logs contain detailed information about the state and operation of those resources. Resource logs can be used to troubleshoot issues with specific resources.

Diagnostic logs: Diagnostic logs are used to diagnose issues with Azure services and resources, and contain detailed information about the configuration, performance, and health of those services and resources.

Platform logs: Platform logs provide information about the underlying infrastructure of Azure, including system events and data about the health and performance of the Azure platform.

Azure Audit and Logging data can be analysed and visualized using Azure Monitor, which provides a centralized dashboard for monitoring, analysing, and alerting on log data. The logs can also be exported to other platforms or storage solutions for further analysis, archiving, or backup.

Overall, Azure Audit and Logging provides a comprehensive solution for collecting, analysing, and visualizing logs and audit data generated by Azure services and resources. These tools make it easier to monitor, diagnose, and troubleshoot issues and to meet compliance and regulatory requirements.

The table below table provides a summarised set of logs that will be collected. Please note there will be additional logs that will be collected but are not within the scope of this document.

System Resource	Type of Logs	Logging Solution
Azure Functions	Function Application Logs Metrics App Insights Telemetry Logs	Azure Monitor Logs App Insights Integration
Azure Logic Apps	Custom Logs Diagnostic Logs	Azure Monitor Logs App Insights Integration
Azure Web Apps	Custom Logs Diagnostic Logs	Azure Monitor Logs App Insights Integration

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1059 of 1295

Azure Frontdoor	FrontDoor Access Logs	Azure Monitor Logs
Azure Gateway	Azure Gateway Access Logs	Azure Monitor Logs
Azure Api Management Gateway	APIM Access Logs Request/Response Logs	Application Insights Integration
Azure Cosmos DB	Diagnostic Logs	Azure Monitor Logs
Azure Keyvaults	Diagnostic Logs	Azure Monitor Logs
Azure Data Factory	Diagnostic Logs	Azure Monitor Logs
Azure SQL Database	Diagnostic Logs	Azure Monitor Logs
Synapse Spark	Diagnostic Logs	Azure Monitor Logs
Azure Data Lake Gen2 Storage	Diagnostic Logs	Azure Monitor Logs
Azure Synapse SQL Serverless	Diagnostic Logs	Azure Monitor Logs

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1060 of 1295

7 Audit Service Design

To achieve a wholistic approach for implementing the STARS audit capabilities, audit data needs to be collected across system layers namely infrastructure, application, data and security. This document focuses primarily on the application and data.

7.1 Portal and Middleware Audit Features

7.1.1 Access Logs

The system is expected to create an audit trail of various user to system interactions and system to system interactions. This is to address requirement **NFR13e.**

All the interactions of users with STARs services shall be abstracted away using the API Management based gateway solution. APIM Gateway shall have Applications Insights based monitoring enabled. Logging of front end and backend request and response shall be enabled to capture request response data. It is to be noted that this has a limit of 8192 bytes. In case the request or response exceeds the limit, the data is truncated.

Along with the above, Access logs shall be enabled for Azure Frontdoor and Azure Application Gateway. The access logs provide information regarding which area of the portal/application was accessed by the end user. Please refer to the table below in section 12.2, captured as part of FrontDoor Access Logs.

7.1.2 Custom Logs with Application Insights

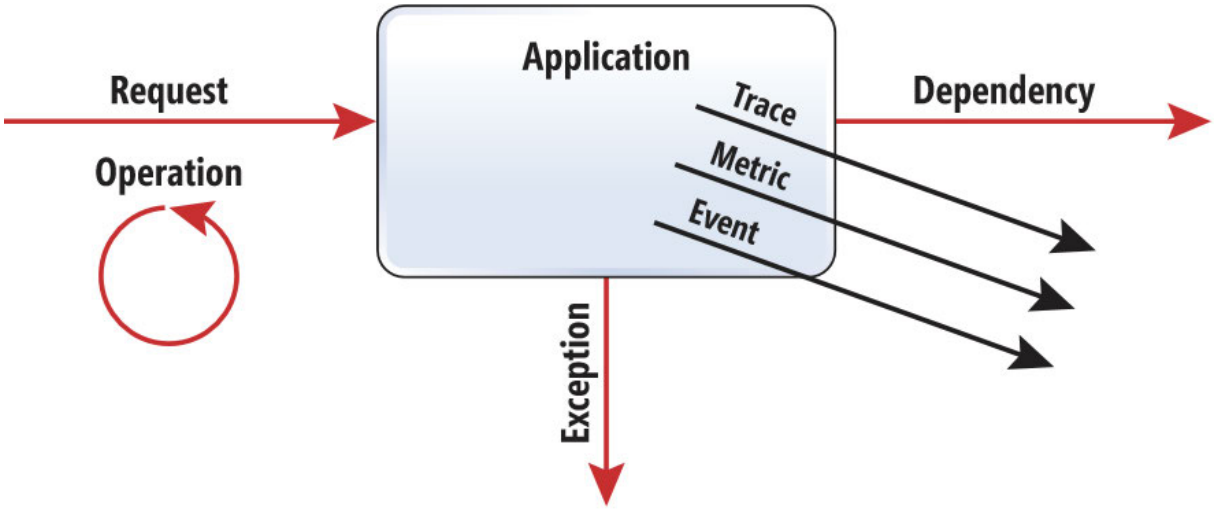
The core services for the STARs system shall be leveraging Azure functions and Logic Apps for the purpose of building custom middleware layer. Most of the user to system interactions and system to system interactions shall be orchestrated via this layer. To achieve this, we shall be using App Insights integration which shall help capture the following telemetry.

1. Requests
2. Dependencies
3. Exceptions
4. Performance counters
5. Heartbeats
6. Logs

Data collected by Application Insights models this typical application execution pattern.

Diagram that shows an Application Insights telemetry data model.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1061 of 1295



The following types of telemetry are used to monitor the execution of your app. Three types are automatically collected by the Application Insights SDK from the web application framework:

Request: Generated to log a request received by your app. For example, the Application Insights web SDK automatically generates a Request telemetry item for each HTTP request that your web app receives.

Exception: Typically represents an exception that causes an operation to fail.

Dependency: Represents a call from your app to an external service or storage, such as a REST API or SQL. In ASP.NET, dependency calls to SQL are defined by System.Data. Calls to HTTP endpoints are defined by System.Net.

Application Insights maintains a set of tables within the Log Analytics workspace to ingest data based on the type of Telemetry that is being collected. To capture functional Audit data by any bespoke STARs, service a data record shall be created in the **customEvents** table. This service does not intend to define various Events that need to be logged for Audit purpose. Rather this service defines the framework that can be leveraged by various services that need to retain a log of business-critical user action or data change log that needs to be retained for the purpose of auditing.

It is to be noted that Application Insights apply adaptive sampling to filter out data. This behaviour is configurable. Hence it is recommended that the following type of Telemetry be

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1062 of 1295

exempted from sampling so that the information is retained for the purpose of auditing and is not lost.

a) Events
b) Exceptions
c) Requests

Various STARs services shall be interacting and making changes to the data in an asynchronous fashion. To maintain traceability, various transaction logs would need to be co-related to each other using a common id. Using APIM policy the system will impose a constraint on all the transactions brokered via the APIM to inject a correlation ID. This ID shall be added as a HTTP header to the incoming requests and shall be used to trace the lifecycle of the transaction. Apart from this if a transaction is originating from a portal due to a user action, a **USER ID** HTTP header shall also be appended to the request that are being passed back to the APIM.

All the services will then be able to access these headers and log them as custom dimensions when creating the audit log entries in the **customEvents** table. These dimensions shall help correlate a series of audit log entries that are logged for facilitating a particular user/system transaction.

For a list of default dimensions please refer to the table in section 12.1 in the appendix.

The service allows to capture custom data properties as part of logging an Audit log entry. This information is stored in field called custom dimensions. At a minimum the below mentioned dimensions should be captured.

1. Correlation ID – This id helps track the lifecycle of changes that a data element was subject to.
2. User ID – This field helps identify if the transaction was user initiated.
3. App Name – The service that is initiating the transaction.
4. EventType – This describes the type of log that is being captured. Possible values include Audit, UserAction etc. This can be a configurable set of values which can be extended based on business requirements.

Apart from the ones mentioned above additional data specific to the event being audited can be stored. It needs to be ensured that the total payload does not exceed 8192 bytes.

The following table describes the interfaces that shall be exposed to the Azure Resources for the purpose of capturing custom Audit log records.

Acronym	Definition
Function Apps	App Insights SDK exposed via Helper Functions

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1063 of 1295

Azure Web Apps	App Insights SDK
Logic Apps	App Insights SDK

For the Logic Apps to be able to log a custom Audit record a custom api end point will be exposed. External Apps shall be able to send requests for the purpose of custom audit data collection.

7.1.3 Business Rules Engine (Drools) Audit Logs

Auditing in Drools allows you to track and record events related to rule execution, such as the activation and firing of rules. The audit information can be used to understand the behaviour of the rules and to troubleshoot issues.

1. Rule activation: triggered when a rule is activated, i.e., added to the agenda.
2. Rule firing: triggered when a rule is fired, i.e., executed.
3. Object insertion: triggered when an object is inserted into the working memory.
4. Object deletion: triggered when an object is deleted from the working memory.
5. Object modification: triggered when an object is modified in the working memory.
6. Agenda group activation: triggered when an agenda group is activated, i.e., made the current active agenda group.
7. Agenda group deactivation: triggered when an agenda group is deactivated.
8. Ruleflow group activation: triggered when a ruleflow group is activated.

Logging in Drools is a feature that allows you to record information about the internal workings of the engine, such as rule activation and deactivation, fact insertion and removal, and other key events. The key events that will be logged are listed below

1. Rule execution: information about the rules that were executed, including the rule names, the facts that triggered them, and the results of their execution.
2. Fact insertion, modification, and deletion: information about the objects that were inserted, modified, or deleted in the working memory, including the object type, the time of the change, and the values of the object's attributes.
3. Debugging information: information that can help with debugging, such as error messages, stack traces, and detailed information about the execution of the rules.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1064 of 1295

7.2 Data Audit Features

In STARS, data submitted by RTOs in Azure Cosmos DB will be ingested, transformed, and mastered using Azure Data Factory into Master database and then to Reporting database for reporting. Both databases will be implemented on Azure SQL database.

Master Data will be held in temporal tables, for audit and time-based analysis. Temporal tables (SYSTEM_VERSIONING) will be used to retain history on applicable objects in the Azure SQL Master database. Changes to or audit on data will be tracked using this mechanism.

Reporting database sources data from Master database, changes to data in Reporting database is tracked using Slowly Changing Dimensions.

All reference data sets from external sources that are required for validation and mastering data (lookup activity) will be ingested via ADF and will be stored in STARS Reference database (Azure SQL Database) with a history of change (Effective Start/End Dates). Effective Dates will be as per scheduled pull from source.

Azure Data Factory will be used for ingestion and data movement in STARS. A log table will be maintained in Azure SQL database that will store and capture the pipeline activity as records in the table.

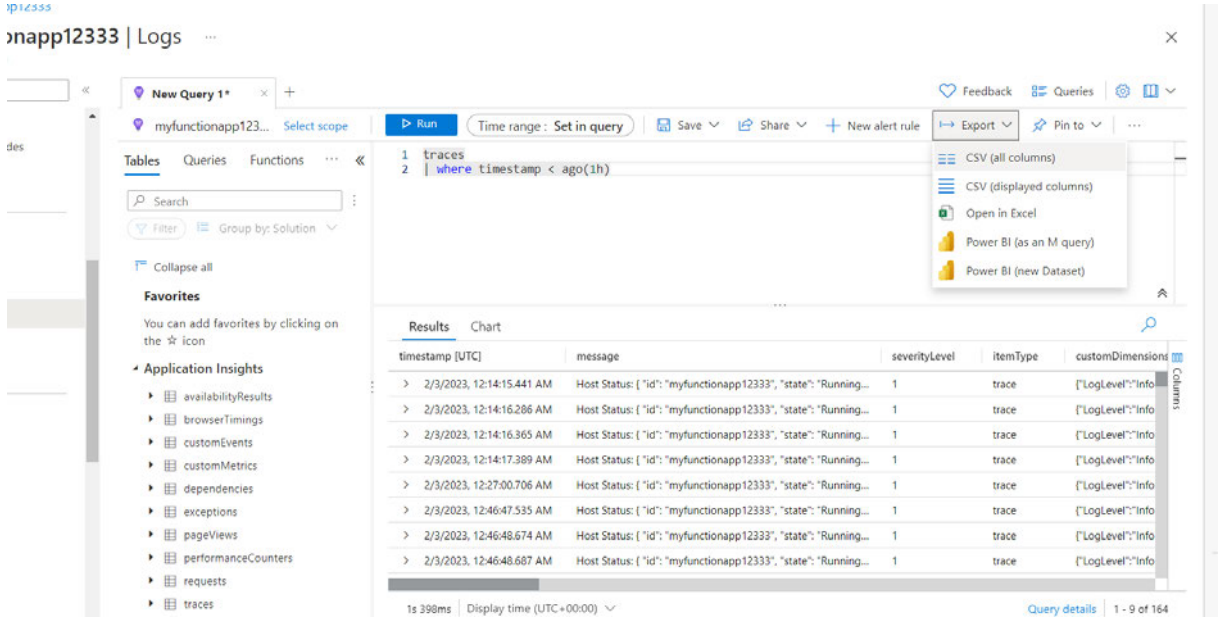
7.3 User Access and Security Audit Features

All the user management shall be handled via Azure AD. Users role management shall be controlled via Azure AD groups which will have pre-defined roles within the application domain. Azure AD Audit log data is ingested via Microsoft Sentinel for the purpose of monitoring and advanced threat management.

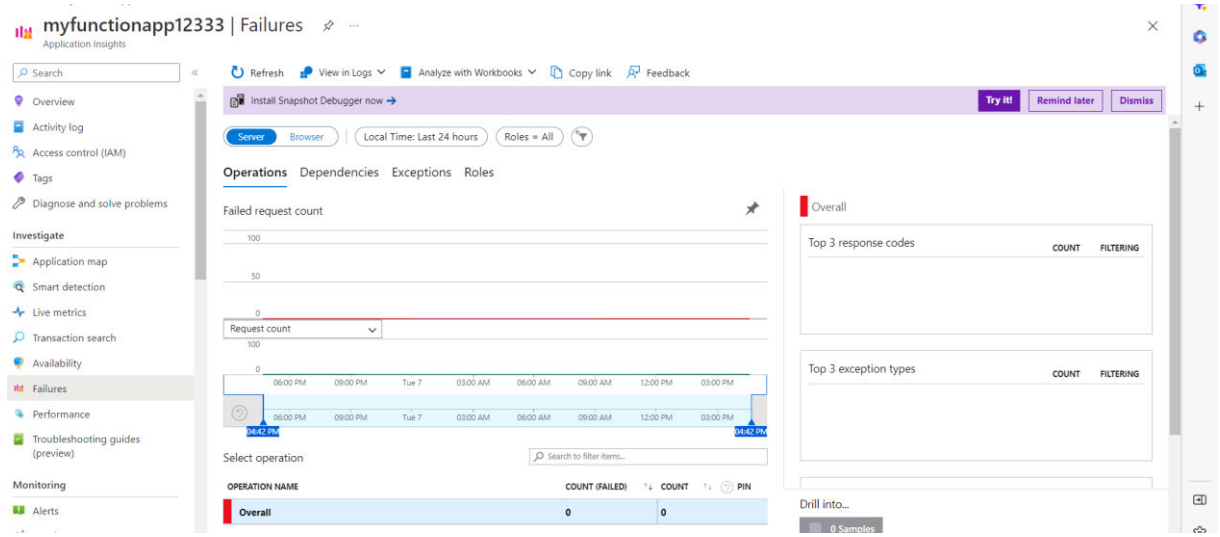
Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1065 of 1295

8 Azure Log Analytics Reporting and Analysis

The logs aggregated via log analytics workspace can be used to generate reports which can then be shared with the external parties. The logs can be filtered based on the various dimensions captured for the logs. The result sets can be exported to csv, excel or power bi dataset formats.



Apart from the reporting capabilities App insights also provide a range of monitoring capabilities like Failure tracking and Live Metrics. Live metrics directly hook on the app instance and provide live telemetry. This can be useful when tracking issues.



Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1066 of 1295

9 Performance Impacts and Service Limits

To use the Application Insights SDK a small instrumentation package is installed in the app and then instrument the web app, any background components, and JavaScript within the web pages. The app and its components don't have to be hosted in Azure. The instrumentation monitors the app and directs the telemetry data to an Application Insights resource by using a unique token. The effect on the app's performance is small, tracking calls are non-blocking and batched to be sent in a separate thread.

Detailed service limits for the Azure Monitor service are available for reference Azure Monitor Service Limits .

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1067 of 1295

10 Technical Design Patterns

The below diagram depicts the audit log capturing process for the portal and core STARs services.

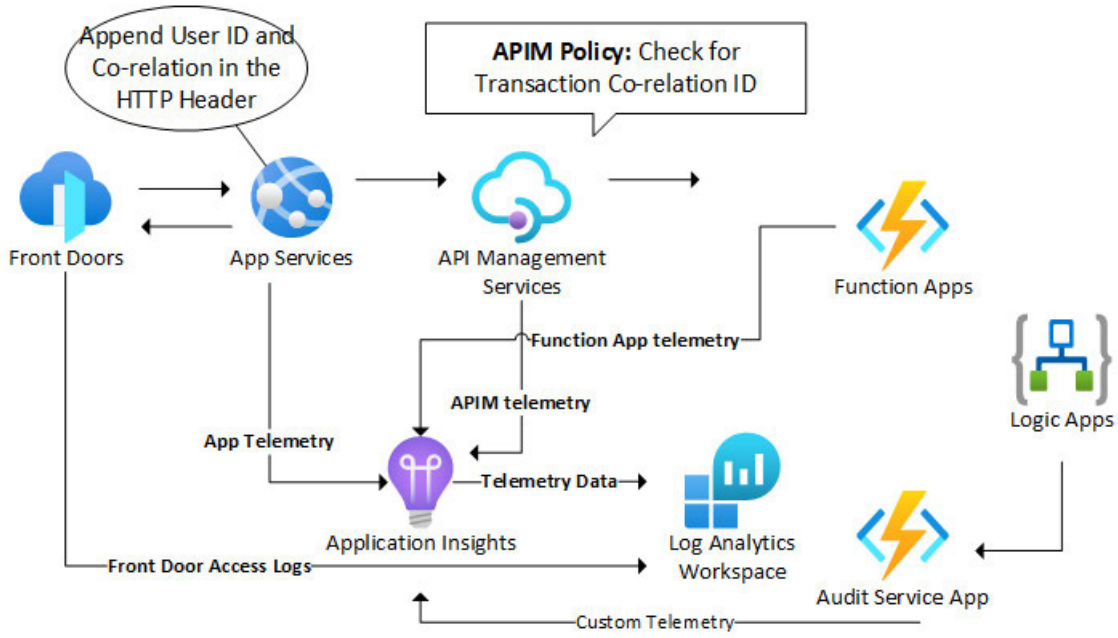


Figure 1: Custom Telemetry Data Collection using App Insights

1. Azure Web Apps and Function Apps will be able to send direct telemetry data using App Insights SDK integration. A common library will expose helper methods to standardize logging process from core STAR services function apps. This will also abstract away the plumbing required for log management from the functional codebase.
2. Azure Logic Apps will natively support App insights integration for diagnostic and exception logging. For logging any functional audit event it will need to consume a custom api endpoint, allowing it to post audit related data to be ingested by Application Insights.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1068 of 1295

10.1 Audit Service Api EndPoint Description

The Audit service Api will support the following operations.

Operation Name	Comments
Create	This endpoint allows a post message request to be sent along with custom metadata.

Create Operation Input Parameters

Operation Name	Data Type	Comments
correlationID	Guid	This is a guid which is used to link various audit log entries that are functionally linked to a transaction.
userID	String	A reference to user ID if the action has been initiated by a user and in case of a service it would be identity of the service.
appName	String	Name of the invoking service
eventType	String	This property defines what type of event is being logged i.e. Audit, user action etc.
customDimensions	Json	This object holds various properties that need to be logged along with the entry.

Request Body

```
{
    correlationID : { Guid },
    userID : { string} ,
    appName : { string } ,
     eventType : { string } ,
     customDimensions : { json Object }
}
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1069 of 1295

Response Body

```
{
  status :  {string}

}
```


Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1070 of 1295

11 Access Management

Access to system and audit logs shall be a purely administrative function. The system admins will have access to Azure Log Analytics Workspace Logs. They shall be able to query the logs.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1071 of 1295

12 Appendix

12.1 Default Dimensions for customEvents table

timestamp (datetime)
name (string)
itemType (string)
customDimensions (dynamic)
customMeasurements (dynamic)
operation_Name (string)
operation_Id (string)
operation_ParentId (string)
operation_SyntheticSource (string)
session_Id (string)
user_Id (string)
user_AuthenticatedId (string)
user_AccountId (string)
application_Version (string)
client_Type (string)
client_Model (string)
client_OS (string)]
client_IP (string)
client_City (string)
client_StateOrProvince (string)
client_CountryOrRegion (string)
client_Browser (string)
cloud_RoleName (string)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1072 of 1295

cloud_RoleInstance (string)
appId (string)
appName (string)
iKey (string)
sdkVersion (string)
itemId (string)
itemCount (int)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1073 of 1295

12.2 Default Dimensions Front Door Access Logs

Property	Description
TrackingReference	The unique reference string that identifies a request served by AFD, also sent as X-Azure-Ref header to the client. Required for searching details in the access logs for a specific request.
Time	The date and time when the AFD edge delivered requested contents to client (in UTC).
HttpMethod	HTTP method used by the request: DELETE, GET, HEAD, OPTIONS, PATCH, POST, or PUT.
HttpVersion	The HTTP version that the viewer specified in the request.
RequestUri	URI of the received request. This field is a full scheme, port, domain, path, and query string
HostName	The host name in the request from client. If you enable custom domains and have wildcard domain (*.contoso.com), hostname is a.contoso.com. if you use Azure Front Door domain (contoso.azurefd.net), hostname is contoso.azurefd.net.
RequestBytes	The size of the HTTP request message in bytes, including the request headers and the request body. The number of bytes of data that the viewer included in the request, including headers.
ResponseBytes	Bytes sent by the backend server as the response.
UserAgent	The browser type that the client used.
ClientIp	The IP address of the client that made the original request. If there was an X-Forwarded-For header in the request, then the Client IP is picked from the same.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1074 of 1295

SocketIp	The IP address of the direct connection to AFD edge. If the client used an HTTP proxy or a load balancer to send the request, the value of SocketIp is the IP address of the proxy or load balancer.
timeTaken	The length of time from the time AFD edge server receives a client's request to the time that AFD sends the last byte of response to client, in milliseconds. This field doesn't take into account network latency and TCP buffering.
RequestProtocol	The protocol that the client specified in the request: HTTP, HTTPS.
SecurityProtocol	The TLS/SSL protocol version used by the request or null if no encryption. Possible values include: SSLv3, TLSv1, TLSv1.1, TLSv1.2
SecurityCipher	When the value for Request Protocol is HTTPS, this field indicates the TLS/SSL cipher negotiated by the client and AFD for encryption.
Endpoint	The domain name of AFD endpoint, for example, contoso.z01.azurefd.net
HttpStatusCode	The HTTP status code returned from Azure Front Door. If a request to the origin times out, the value for HttpStatusCode is set to **0**.
Pop	The edge pop, which responded to the user request.
Cache Status	Provides the status code of how the request gets handled by the CDN service when it comes to caching. Possible values are: • HIT and REMOTE_HIT: The HTTP request was served from the Front Door cache. • MISS: The HTTP request was served from the origin. • PARTIAL_HIT: Some of the bytes from a request were served from the Front Door cache, and some of the bytes were served from origin. This status occurs in object chunking scenarios. • CACHE_NOCONFIG: Request was forwarded without caching settings, including bypass scenario. • PRIVATE_NOSTORE: No cache configured in caching settings by customers. • N/A: The request was denied by a signed URL or the rules engine.
MatchedRulesSetName	The names of the rules that were processed.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1075 of 1295

RouteName	The name of the route that the request matched.
ClientPort	The IP port of the client that made the request.
Referrer	The URL of the site that originated the request.
TimeToFirstByte	The length of time in milliseconds from AFD receives the request to the time the first byte gets sent to client, as measured on Azure Front Door. This property doesn't measure the client data.
ErrorInfo	This field provides detailed info of the error token for each response. Possible values are: • NoError: Indicates no error was found. • CertificateError: Generic SSL certificate error. • CertificateNameCheckFailed: The host name in the SSL certificate is invalid or doesn't match. • ClientDisconnected: Request failure because of client network connection. • ClientGeoBlocked: The client was blocked due geographical location of the IP. • UnspecifiedClientError: Generic client error. • InvalidRequest: Invalid request. It might occur because of malformed header, body, and URL. • DNSFailure: DNS Failure. • DNSTimeout: The DNS query to resolve the backend timed out. • DNSNameNotResolved: The server name or address couldn't be resolved. • OriginConnectionAborted: The connection with the origin was disconnected abnormally. • OriginConnectionError: Generic origin connection error. • OriginConnectionRefused: The connection with the origin wasn't established. • OriginError: Generic origin error. • OriginInvalidRequest: An invalid request was sent to the origin. • ResponseHeaderTooBig: The origin returned a too large of a response header. • OriginInvalidResponse: Origin returned an invalid or unrecognized response. - OriginTimeout: The timeout period for origin request expired. - ResponseHeaderTooBig: The origin returned a too large of a response header. - RestrictedIP: The request was blocked because of restricted IP. - SSLHandshakeError: Unable to establish connection with origin because of SSL hand shake failure. - SSLInvalidRootCA: The RootCA was

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1076 of 1295

	invalid. - SSLInvalidCipher: Cipher was invalid for which the HTTPS connection was established. - OriginConnectionAborted: The connection with the origin was disconnected abnormally. - OriginConnectionRefused: The connection with the origin wasn't established. - UnspecifiedError`: An error occurred that didn't fit in any of the errors in the table.
OriginURL	The full URL of the origin where requests are being sent. Composed of the scheme, host header, port, path, and query string. **URL rewrite**: If there's a URL rewrite rule in Rule Set, path refers to rewritten path. **Cache on edge POP** If it's a cache hit on edge POP, the origin is N/A. **Large request** If the requested content is large with multiple chunked requests going back to the origin, this field will correspond to the first request to the origin. For more information, see Caching with Azure Front Door.
OriginIP	The origin IP that served the request. **Cache hit on edge POP** If it's a cache hit on edge POP, the origin is N/A. **Large request** If the requested content is large with multiple chunked requests going back to the origin, this field will correspond to the first request to the origin. For more information, see Caching with Azure Front Door
OriginName	The full DNS name (hostname in origin URL) to the origin. **Cache hit on edge POP** If it's a cache hit on edge POP, the origin is N/A. **Large request** If the requested content is large with multiple chunked requests going back to the origin, this field will correspond to the first request to the origin. For more information, see Caching with Azure Front Door

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1077 of 1295

12.3 Design Tasks and Decisions

Task No	Task Link
1843	W11b - Action #01 - DXC Service leads need to provide confirmation. | Due Date: Varies for each Service
1845	W11b - Action #02 - Microsoft to work with DXC to verify the best approach for our requirements.
1847	W11b - Action #04 - ISM Controls Conformance - Outstanding Decision require to be confirmed by DEWR
1848	W11b - Action #05a - ISM Controls Conformance - Nathan to provide Nick with the control breakdown.
1849	W11b - Action #05b - ISM Controls Conformance - Nick to provide feedback on the accuracy of this breakdown.
1850	W11b - Action #06a - OOBE needs to confirm if they will provide a separate document covering the audit capabilities implemented for infrastructure
1851	W11b - Action #06b - DXC Data team needs to confirm if they will be providing inputs on how they are addressing audit requirements at the data layer level.
1944	W11b - Action #06c - Follow up with DEWR about security assessment
1852	W11b - Action #07 - Creating and Managing Alerts - OOBE to provide confirmation if they will be managing this capability.
1853	W11b - Action #08 - Backend Audit Reporting - DEWR needs to confirm the understanding.
1854	W11b - Action #09 - DEWR Confirmation required if any sensitive information in the payloads that should not be persisted.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1078 of 1295

13 Acronyms

Acronym	Definition
API	Application Programming Interface
CRUD	Create Read Update Delete
REST	Representational State Transfer
FR	Functional Requirement
NFR	Non-functional Requirement
STARS	Student Training & Activity Reporting System
APIM	API Management Gateway
VDS	VET Data Streamlining
ADF	Azure Data Factory

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1079 of 1295

Document 17

Australian Government
Department of Employment and Workplace Relations

Data Governance Tool – MS Purview Technical Design

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1080 of 1295

1 Document Management

1.1 Document Control

Version	Change Description	Date
0.1	Initial draft version	03/04/2023
0.2	Added more detail in Environments and Security sections	18/04/2023
0.2	Updated document for reviewers meeting	01/05/2023
0.3	Added Screenshots from MS Purview – relevant to STARS plus Extra information on Purview functionality.	05/05/2023

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This is a living document and will be updated as required.

1.2 Document Author

Role	Name
Data Governance Lead	s 47F(1)

1.3 Document Reviewers

Role	Name
Data Architect (DEWR – Project)	s 47F(1)

1.4 Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

1.5 Related Documents

Reference	Document
A	STARS Solution Requirements Document v1.3.docx
B	STARS Solution Design v1.0.docx
C	Governance Framework for Vocational Educational and Training Student and Training Activity Data – Draft - 2023
D	VET Data Access and Privacy Protocol – Draft - 2023
E	Detail Design Document - Data Governance Infrastructure D.0.1.docx

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1081 of 1295

Reference	Document
F	MS Purview Diagrams – V0.1
G	DEWR TSD – Azure Service Design for Purview

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1082 of 1295

Contents

1 Document Management 2
1.1 Document Control 2
1.2 Document Author 2
1.3 Document Reviewers 2
1.4 Document Approval 2
1.5 Related Documents 2
Contents 4
2 Introduction 6
2.1 Purpose 6
2.2 Intended Audience 6
2.3 Scope and Exclusions 7
2.3.1 In Scope 7
2.3.2 Out of Scope 7
3 MS Purview – An Overview 8
3.1 Functionality of Purview – An Overview 8
3.2 Data Map 8
Data Map Creation – Steps in MS Purview: 9
3.3 Data Catalogue 10
Data Catalogue Creation in MS Purview: 11
3.4 Data Classification 12
3.5 Data Lineage 13
Data Lineage Creation – Steps in MS Purview: 13
3.6 Data Estate Insights 13
3.7 Purview – Design 15
3.7.1 Collections Architecture 15
3.7.2 Purview Authorization Model 16
3.7.3 Environments 18
4 Defence in Depth 20
4.1 Networking 20
5 Security 22
5.1 Identity Governance and Access Management (Authentication and Authorization) 22
6 Requirements 23
6.1 Functional 23

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1083 of 1295

6.2 Non-Functional 23
7 Appendix 24
7.1 Acronyms 24
7.2 Databases and Repositories 25

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1084 of 1295

2 Introduction

The Detail Design Document - Data Governance Infrastructure D.0.1 document described how STARS Reporting Data Governance was managed from a People and Process perspective.

This document describes in detail how the Data Governance tool of choice for STARS Reporting – MS Purview, would be embedded into the Data Governance Infrastructure, ensuring

- Improved Business glossary
- Superior data classification
- Advanced Data Lineage

All of the above leading to superior data quality and strengthened security.
This document is Data Governance from a Technology perspective, thus completing the triage of **People + Process + Technology**.

Note: This is a living document that will be updated as designs for various other STARS services are developed and evolve. Accordingly, this document will evolve to reflect and document notification components as the STARS solution build progresses.

2.1 Purpose

The purpose of this document is to explain how MS Purview will help:

1. Embed holistic and complete Data Governance into the STARS Reporting solution – based on the DEWR/NCVER Governance and Metadata Policies and Frameworks
2. Ensure compliance through MS Purview – using business glossaries, data classifications, roles and persona access and data lineage (for compliance).
3. Create synergy in STARS – using business glossary through a metadata repository

This approach promotes data efficiency through a holistic approach ensuring all aspects of Data Governance are covered through the triage of **People + Process + Technology**.

2.2 Intended Audience

The intended audience for MS Purview is all those that interact with the STARS Reporting System. And all these Persona interactions are deemed as the intended audience.

Thus, the primary audience for this document includes:

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1085 of 1295

- The Policy and Compliance Teams – who are very close to creating Data Governance Police and Frameworks
- The DEWR and NCVER Enterprise Architects, Data Architects and Solutions Architects
- All Personas that are part of the (future) Data Governance Council(s)
- The Executive Team – who need to endorse this policy
- Any Team that is responsible for creating the Data Management Policies

2.3 Scope and Exclusions

2.3.1 In Scope

The following are in-scope of this detailed design solution:

1. Installation of the Data Governance Tool:
   - MS Purview
2. Ensuring that the Data Governance and Data Management Frameworks are complete and cover all aspects of data – and that there are no missing gaps which can leave data vulnerable.

The MS Purview tool – once installed – will have the following aspects of Data Governance covered:

- Data Map (Data Assets, Data Classifications, Business Glossary)
- Data Catalogue
- Data Lineage
- Data Estate Insights
- Metadata reports (Data Assets; Classifications; Sensitivity Labels)

2.3.2 Out of Scope

The following are out of scope:

Creating Policy documents and maintenance of these documents:

Some but not all are listed below:

- Data Governance Policy, Data Governance Framework
- Data Management Policy, Data management Framework
- Metadata Management Documentation
- Data Literacy Programme – to onboard Data Governance Framework Personas
- Data Governance Council(s) Framework Establishment
- Data Governance/Data Persona establishment (Data Champion, Data Adept, Data Citizen).
- Data Sharing and Data Policy functionality are out of scope as not available in this release of MS Purview

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1086 of 1295

3 MS Purview – An Overview

Microsoft Purview is a unified data governance service that helps manage and govern on-premises, multi-cloud and software-as-a-service (SaaS) data.

It will help create a holistic, up-to-date map of The STARS Reporting data landscape with automated data discovery, sensitive data classification and end-to-end data lineage – from Capture to Consumption layer(s).

The aim of implementing a data governance tool like MS Purview is to empower the data personas in the STARS Data Governance Framework and enable them to find valuable, trustworthy data in a centralized application, reducing time around business glossary definitions rework and creating organisational synergy and efficiency through

- **A Centralized business glossary** - Viewing data definitions/data classifications
- **A Data Catalogue** - Sharing assets location
- **A Data Lineage view** – Tracking data (Ingestion to Consumption layers)

3.1 Functionality of Purview – An Overview

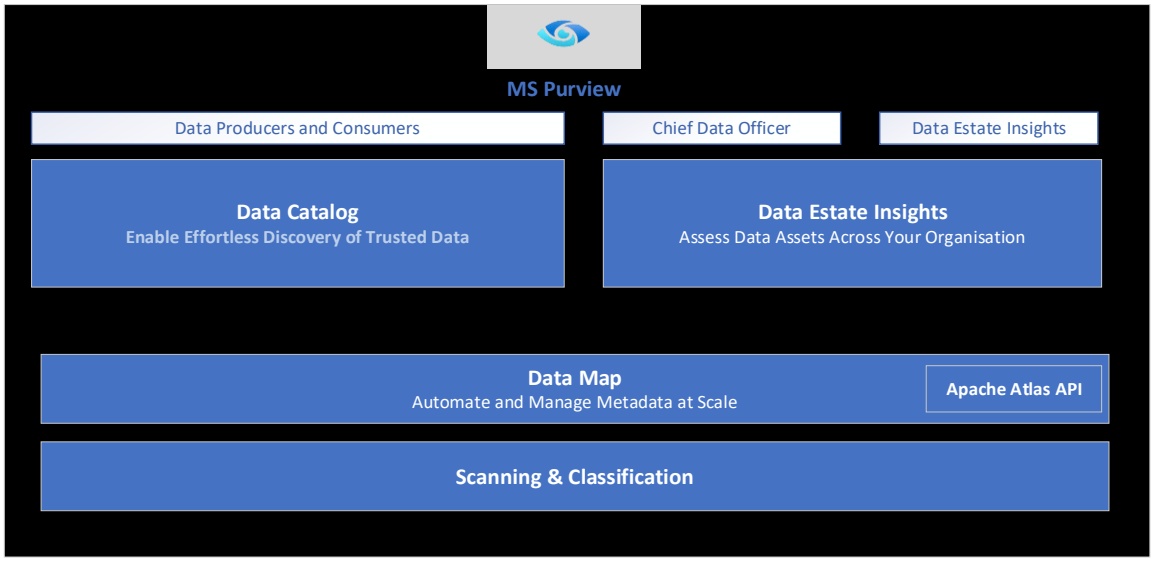


Figure 1: MS Purview Overview (F1)

3.2 Data Map

Audience: All data governance personas that have permission for scanning STARS metadata from different data sources.

Information risk can be reduced by performing the following:

a. Identification and classification of sensitive data assets in STARS
b. Location of sensitive data throughout the STARS ecosystem
c. Determining how each asset in STARS can be protected

For STARS Reporting, Microsoft Purview will perform all the above by

- Automating metadata discovery scheduled through metadata scans
- Classification for data assets across the STARS data estate

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1087 of 1295

Metadata and descriptions of discovered data assets will be integrated into an integrated map of STARS data estate.
Microsoft Purview Data Map provides the foundation for metadata discovery and data governance.

Data Map Creation – Steps in MS Purview:

The STARS Data Map – is created through metadata ingestion and capture from the following STARS data sources.
Azure Cosmos DB; Azure Data Lake; Azure Data Factory; Azure SQL Server; Azure Blob Storage; Azure Synapse. (Please see below).

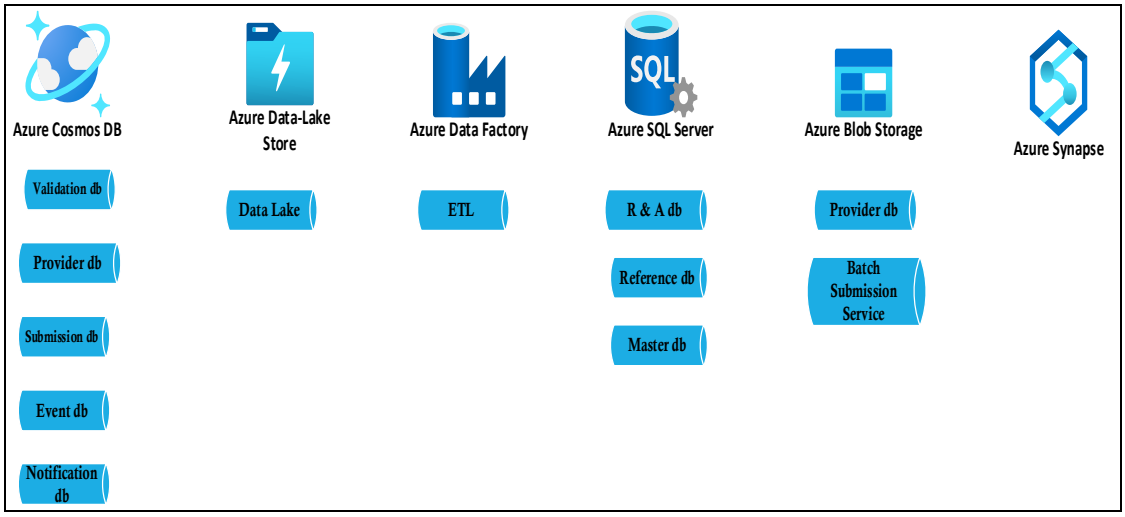


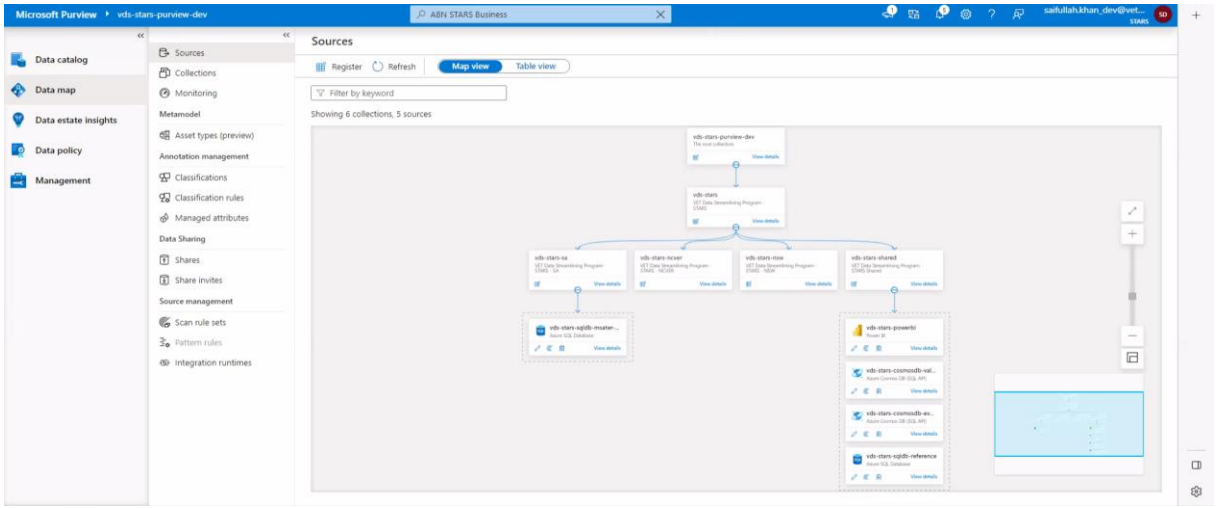
Figure 2: MS Purview Data Sources (F2)

Azure Cosmos DB, Azure Data Lake, Azure Data Factory, Azure SQL Server, Azure Blob Storage and Azure Synapse

Steps in creating a Data Map:

Map View:

In the Data Map, you can see all your data sources in one place, edit your data sources, schedule scans of your data sources, and organize your data sources visually into collections, as in the example below.



Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1088 of 1295

Figure 3: MS Purview Map View (F3)

Table View:

If you prefer a list view of your data sources, you can switch to "Table view" using the slider, and your Data Map will look like this:

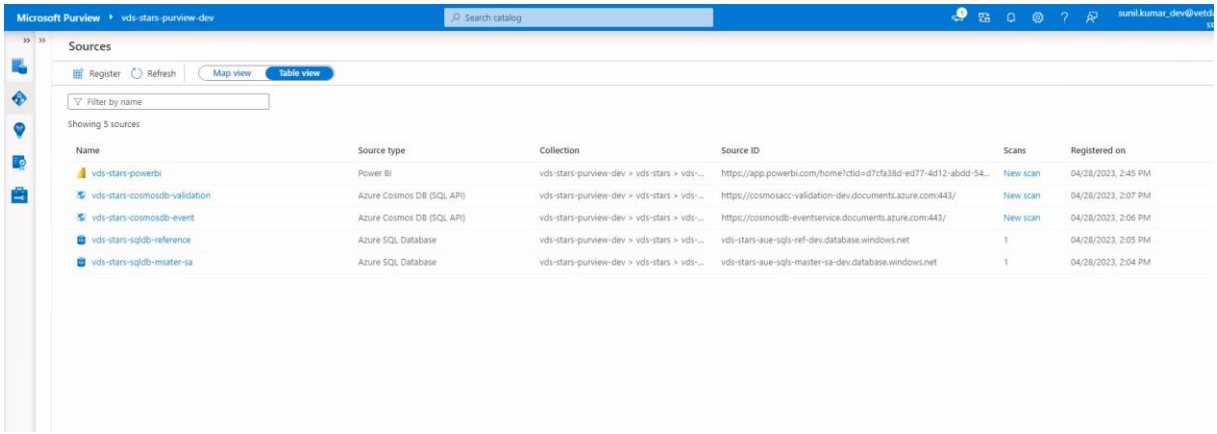


Figure 4: MS Purview Table View (F4)

MS Purview Data Map has two components –

- Metadata Storage (inbuilt MS-SQL Server Repository)
- Operation throughput as a capacity unit (elastic growth based on usage)

The data map account is elastic, and growth is based on consumption and usage through the auto-scale feature, with no default limit on storage.

It can include any Create, Read, Write, Update, and Delete operations on metadata stored in the Data Map.

The STARS Data Map will be automatically updated – through built-in automated scanning and classification system, maintained through scheduling.

3.3 Data Catalogue

Audience: All data governance personas that have permission for scanning and sharing a data catalogue.

The Data Catalogue functionality of MS Purview

- Searches for all of STARS data assets
- Aligns data assets with friendly business terms and data classification

All of the above leading to a comprehensive identification of data sources.

Using the MS Purview Data Catalog Application, STARS business and technical personas will easily find relevant data locations using a search experience with filters based on lenses such as business glossary terms, data classifications and sensitivity labels.

Data Sensitivity labels will be based on the Data Security classification as per the DEWR Data Governance Framework.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1089 of 1295

Identified data - data which relates to personal an identified individual or organisation, such as an RTO.
(High Confidentiality)

De-identified or Identifiable data - data which, in some circumstances, could enable the identification of an individual or organisation.
(Medium Confidentiality)

Confidentialised data - data provided in a manner that is unlikely to enable the identification of an individual or organisation.
(Low Confidentiality)

Figure 5: Data Confidentiality (Sensitivity Labels) (F5)

Business SME's, Data Governance Personas (Data Custodians, Data stewards, Data Officers) will use the MS Purview Data Catalog and its meta curation features such as STARS business glossary management and the manual/auto tagging ability of data assets with glossary terms.

Data Catalogue Creation in MS Purview:

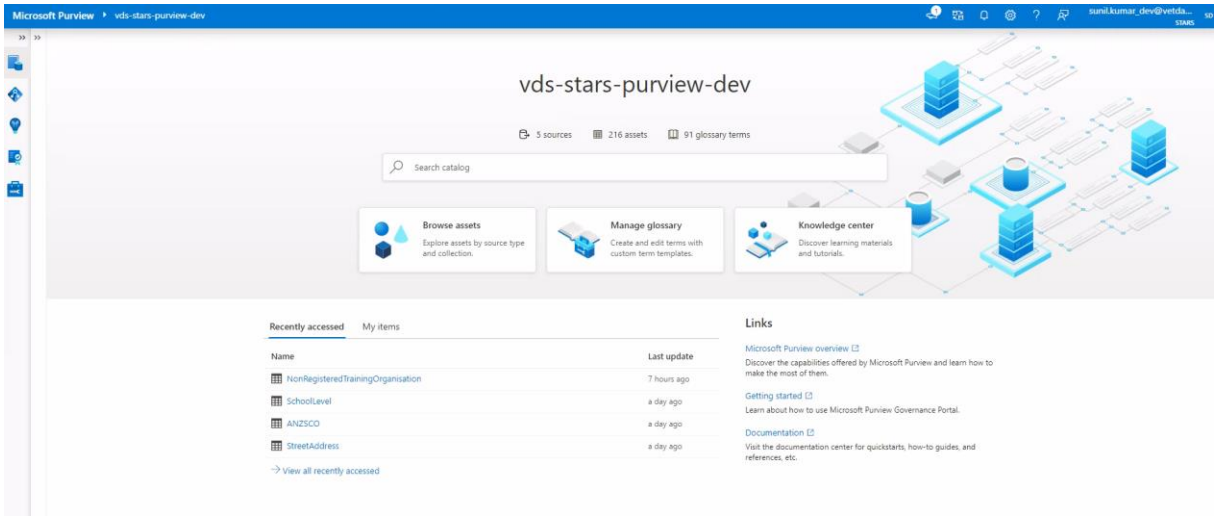


Figure 6: Data Catalog (F6)

The above is the MS Purview Data Catalog that has been created for STARS.
The Data Catalog is created after the creation of the Data Map – where the metadata is ingested from the different sources.
The Stars Data Catalog – as shown – has a

- Search bar for browsing assets by source type
- Ability to create and edit business glossaries and glossary terms

Purview supports the following out-of-the-box attributes for any glossary term:

- Name
- Definition
- Data Stewards

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1090 of 1295

- Data Experts
- Acronym
- Synonyms
- Related Terms

Below is an Business Glossary example of 'Non-registered Training Organisation' entity in STARS:

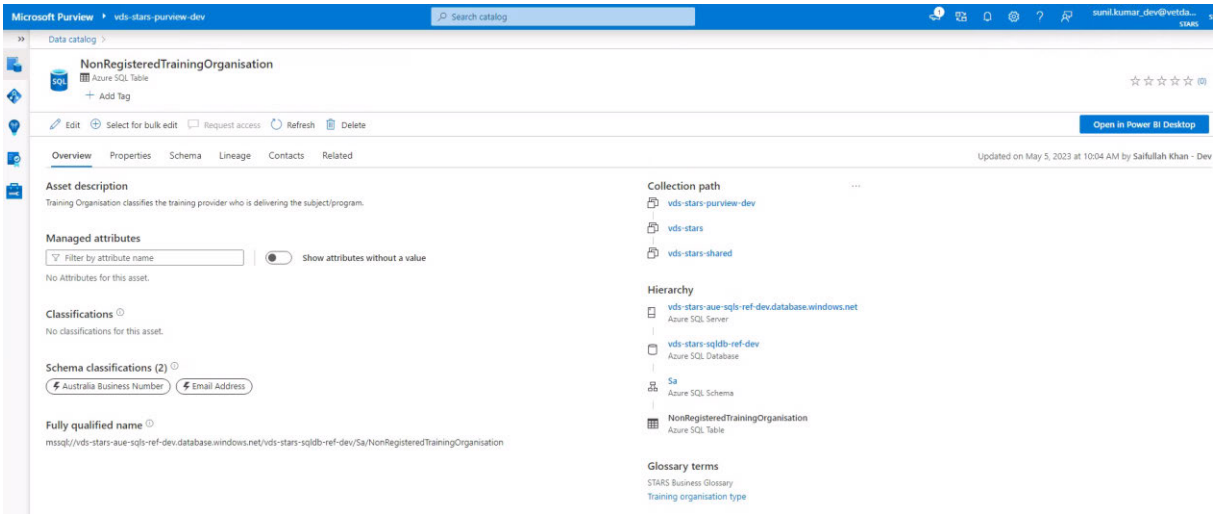


Figure 7: Business Glossary (F7)

3.4 Data Classification

Classification is the process of organizing data into logical categories that make the data easy to retrieve, sort, and identify for future use. This can be important for data governance. Among other reasons, classifying data assets is important because it helps you:

- Narrow down the search for data assets that you're interested in.
- Organize and understand the variety of data classes that are important in your organization and where they're stored.
- Understand the risks associated with your most important data assets and then take appropriate measures to mitigate them.

The Microsoft Purview governance portal supports both system and custom classifications.

- **System classifications**: 200+ system classifications supported are out of the box
- **Custom classifications**: You can create custom classifications when you want to classify assets based on a pattern or a specific column name that's unavailable as a system classification.

Custom classification rules can be based on a ***regular expression*** **pattern or** ***dictionary*****.**

For an example of STARS data Classification please see diagram above

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1091 of 1295

3.5 Data Lineage

Lineage is broadly understood as the lifecycle that spans the data's origin, and where it moves over time across the data estate.
Microsoft Purview will capture lineage for data in different parts of STARS reporting data estate, and at different levels of preparation including:

- Raw data staged from various platforms (Capture)
- Transformed and prepared data (Ingest/Curate)
- Data used by visualization platforms (Consumed)

Data Lineage tracking for STARS will be available for Data Producers and Data Consumers – as it enables tracing lineage of data assets - **visually** – from the consumption layer (e.g. Power BI Report) back tracking to the Capture/Ingestion layer for the origins of a specific entity being tracked.

Data Lineage in Purview is critical as it

- Tracks data as it moves across the STARS data estate
- Performs Up-steam and Downstream Change Impact analysis – thus reducing risk
- Captures the changes of data residency throughout the data life cycle

MS Purview will thus enable lineage of data as it travels through STARS – using metadata as the guide.

Data Lineage Creation – Steps in MS Purview:

In the example below – a sample lineage flow in a Purview demo environment with Cosmos, fed by ADF, then output to Synapse.

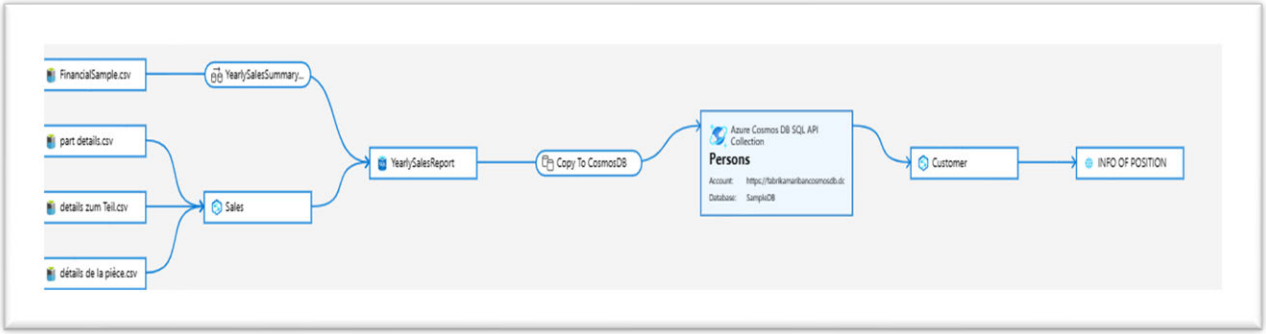


Figure 8: Data Lineage (F8)

Note: A 'STARS demo' example cannot be produced in this document – due to the unavailability of data in the data sources plus development in STARS has just begun – thus not enough complexity to track lineage. Lineage will be added in the next version of this document.

3.6 Data Estate Insights

Audience: CDO and Governance Stakeholders – who want STARS metadata estate visibility.

The Data Estate Insights application gives an overview of STARS data estate to help discover data categories and their location in the STARS Data Estate.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1092 of 1295

With the Microsoft Purview Data Estate Insights, the chief data officers and other governance stakeholders and governance personas will get a bird's eye view of their data estate and gain actionable insights/highlight governance gaps for STARS.

The Data Estate Insights application will be primarily used for governance personas focused on data management, compliance, and data use.

The Data Estate Insights application provides actionable insights STARS data estate, catalogue usage, adoption, and processes.
As the MS Purview Data Map is created through metadata scans, the Data Estate Insights application will automatically extract valuable governance gaps and highlight them in its top metrics for STARS.
It also provides a drill-down experience that enables STARS stakeholders, such as data owners, data stewards, Custodians to take appropriate action to close the gaps.
All the reports within the Data Estate Insights application are automatically generated and populated, so governance stakeholders and personas can focus on the information itself, rather than building the reports.
The pre-built dashboards and reports available within Microsoft Purview Data Estate Insights will be categorized in three sections:

- Health of Data Estate
- Inventory and Ownership
- Curation and Governance

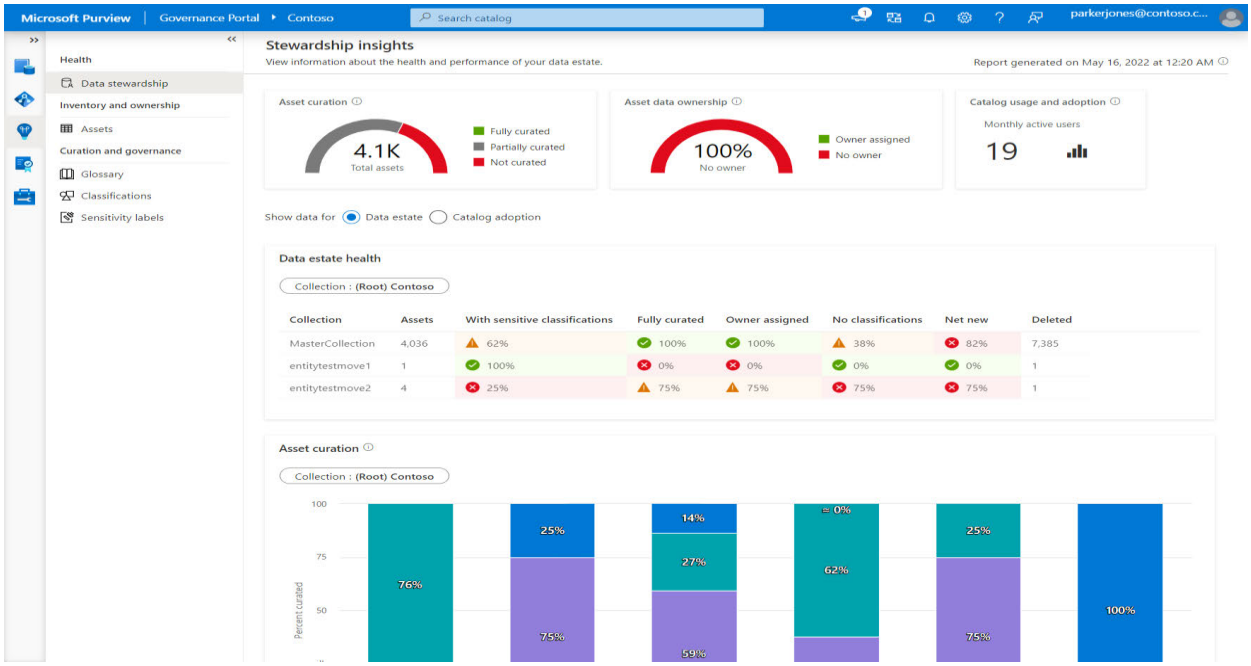


Figure 9: Health of Data Estate (F9)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1093 of 1295

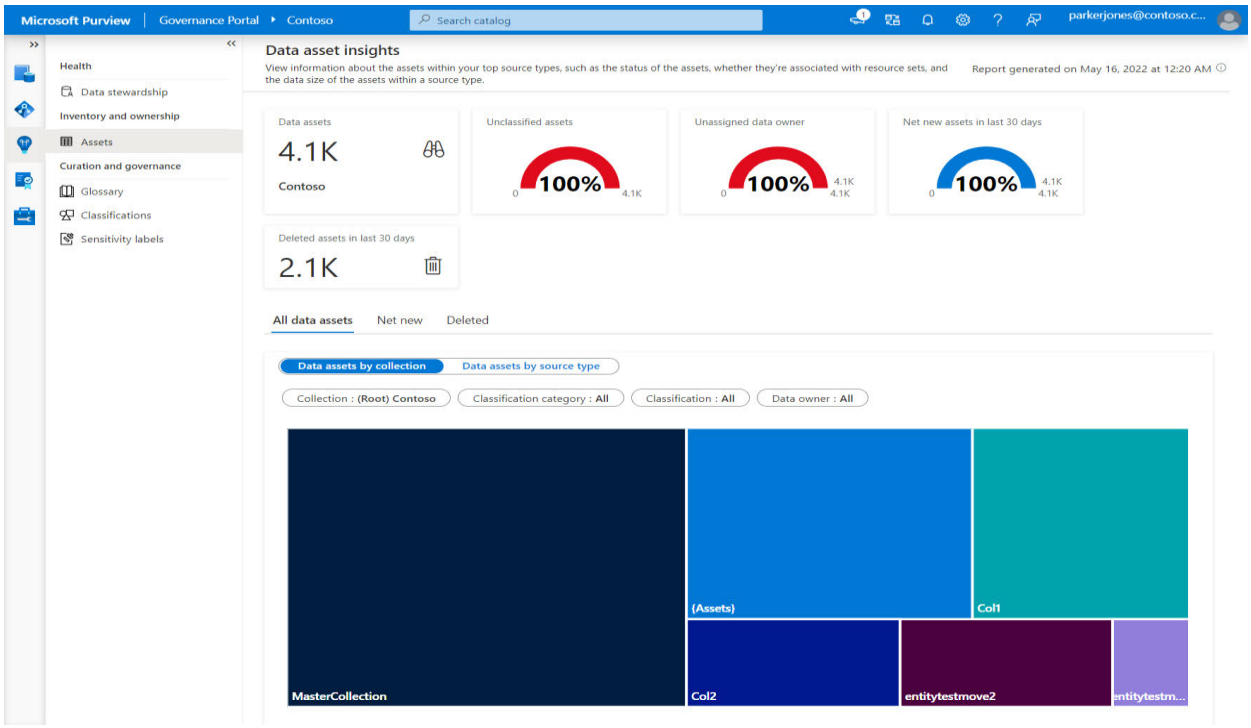


Figure 10: Inventory and Ownership (F10)

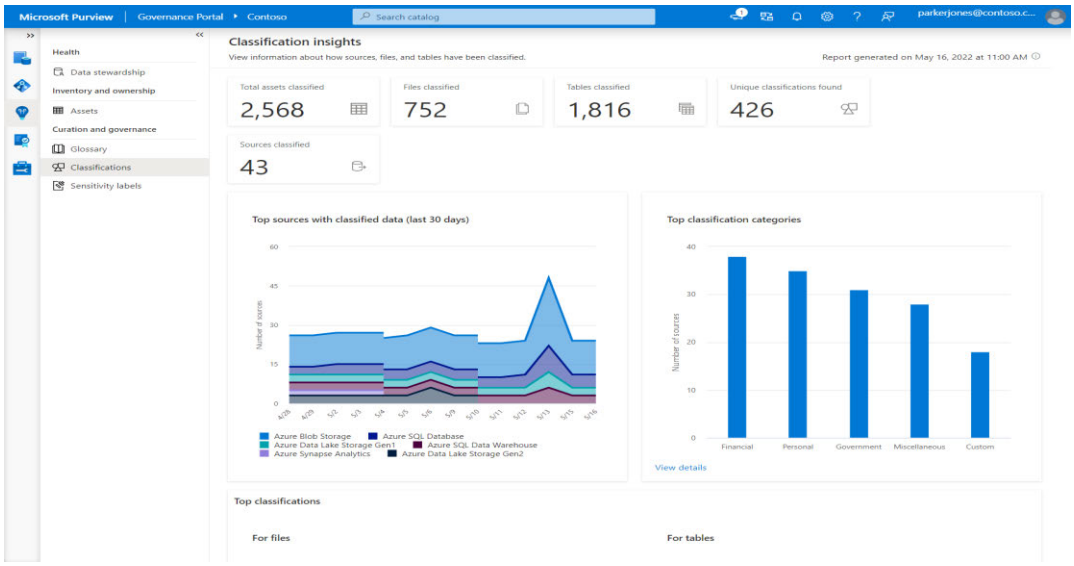


Figure 11: Curation and Governance (F11)

Note: The 3 dashboard examples above are generic examples only – as at the time this document was written – there was no data available and thus resulting reports would be quite incomplete.

3.7 Purview – Design

3.7.1 Collections Architecture

The design of MS Purview is based on the **Collection Architecture**.

The Collection Architecture is created to manage

- Ownership
- Access Control across assets, sources and information

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1094 of 1295

For STARS – the Collections design is based on a combination of Data repositiories+ Application + State.
For e.g. Master DB: Collection will be based on Master DB for each State Database (NSW, QLD, SA, WA, NT, ACT, Commonwealth, NCVER).
Below is an suggested example of collections design which will be finalised after access to a proper Purview environment.

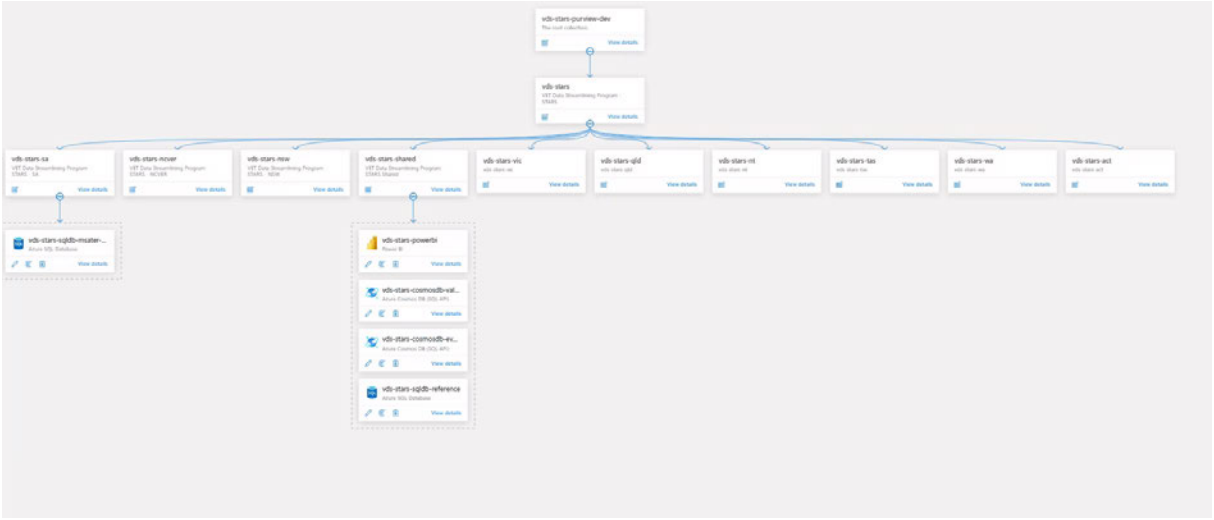


Figure 12: Purview Collections Design (F12)

3.7.2 Purview Authorization Model

The authorisation model of Purview – for STARS – is based on the pre-defined roles and responsibilities – as per the Purview Application.

The pre-defined roles and responsibilities in Purview are:

Collection administrator - a role for users that will need to assign roles to other users in the Microsoft Purview governance portal or manage collections. Collection admins can add users to roles on collections where they're admins. They can also edit collections, their details, and add subcollections. A collection administrator on the root collection also automatically has permission to the Microsoft Purview governance portal. If your root collection administrator ever needs to be changed, you can follow the steps in the section below.

Data curators - a role that provides access to the data catalog to manage assets, configure custom classifications, set up glossary terms, and view data estate. Data curators can create, read, modify, move, and delete assets. They can also apply annotations to assets.

Data readers - a role that will provide read-only access to data assets, classifications, classification rules, collections and glossary terms.

Data source administrator - a role that will allow a user to manage data sources and scans. If a user is granted only to Data source admin role on a given data source, they can run new scans using an existing scan rule. To create new scan rules, the user must also be granted as either Data reader or Data curator roles.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1095 of 1295

Insights reader - a role that provides read-only access to insights reports for collections where the insights reader also has at least the Data reader role.

Workflow administrator - a role that allows a Workflow admin defined for a collection to create self-service workflows and bind these workflows to the collections they have access to. *A Workflow admin defined for any collection can create approved workflows for the business.*

	Annotate Glossary, classification, classification rules	Collections Access control, lineage connections	Data Catalog Search, browse, and asset details	Data Estate Insights All reports	Data Map Sources, scan rule sets, credentials, integration runtime	Policies Data policies	Shares Data sharing	Workflows Access to workflow authoring information
Collection admin CollectionRead, CollectionWrite	✗	✓	✗	✗	✗	✗	✗	✗
Data curator DataRead, DataWrite, CollectionRead, InsightRead	✓	READ-ONLY	✓	READ-ONLY *	✗	READ-ONLY	✗	✗
Data reader DataRead, CollectionRead	READ-ONLY	READ-ONLY	READ-ONLY	✗	✗	✗	✓	✗
Data source admin ScanRead, ScanWrite, SourceRead, SourceWrite, CollectionRead	READ-ONLY	READ-ONLY	✗	✗	✓	READ-ONLY	✗	✗
Insights reader InsightRead, CollectionRead	✗	READ-ONLY	✗	READ-ONLY	✗	✗ **	✗	✗
Policy author PolicyRead, PolicyWrite, SourceRead, CollectionRead, DataRead	✗	READ-ONLY	READ-ONLY	✗	✗	✓	✗	✗
Workflow admin CollectionRead, WorkflowRead, WorkflowWrite, WorkflowBind	READ-ONLY	READ-ONLY	✗	✗	✗	✗	✗	✓
Account creator Access to data map and data catalog	✓	✓	✓	✗	✓	✗	✓	✗

Figure 13: Pre-Defined Purview Roles & Responsibilities (F13)

The Roles and Responsibilities as per the DERW Data Governance Framework are shown in the diagram below:

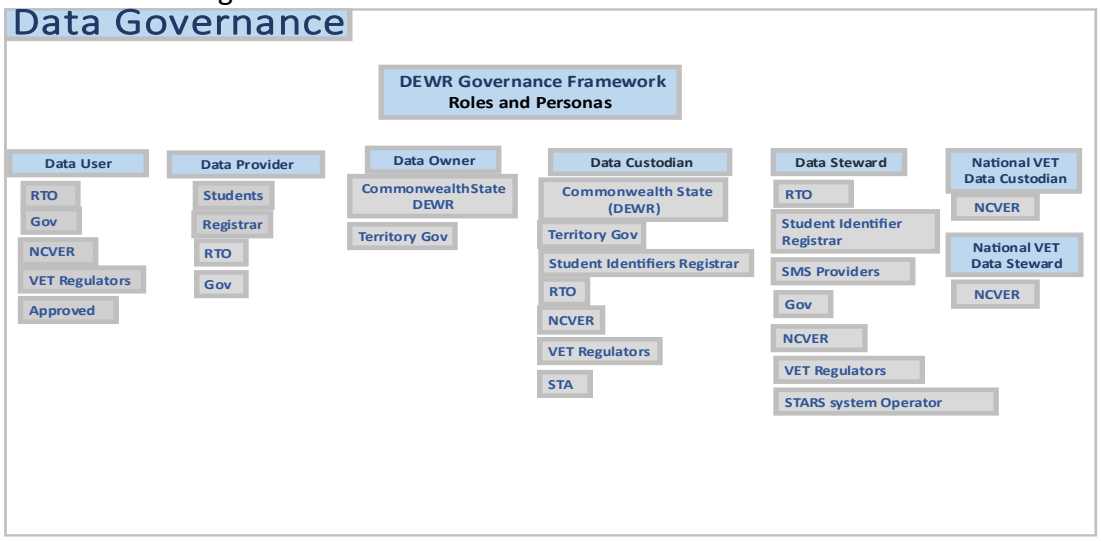


Figure 14: DEWR Governance Framework Roles (F14)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1096 of 1295

#	Purview In-Built Roles →	Collections Admin	Data Curator	Data Reader	Data Source Admin	Insights Reader	Workflow Admin	Account Curator
1	VET Data Roles							
	↓							
2	National VET Data Custodian (NCVER)		X	X	X		X	X
3	National Collections Admin; STA Collections Admin	X						
4	SMS Vendors			X				
5	Commonwealth and State Data Readers			X		X		
6	STARS Systems Operator				X			
7	VET Regulators (AQSA, State)			?		?		

Figure 15: Governance Roles Mapped (F15)
DEWR Data Governance Personas Mapped to Purview pre-defined roles

3.7.3 Environments

There will be two/three Separate Environments of Microsoft Purview accounts.

- Development
- UAT
- Production

All Environments will need separate respective instances of STARS data.

STARS Dev → Purview Dev

STARS UAT → Purview UAT

STARS Prod → Purview Prod

Thus, STARS Dev, UAT and Prod data sources will be registered and scanned inside their corresponding Microsoft Purview instances.

PS. Purview UAT is optional – and will only be made available after discussing with Project Team.
Aim is to keep project ongoing costs to a minimal.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1097 of 1295

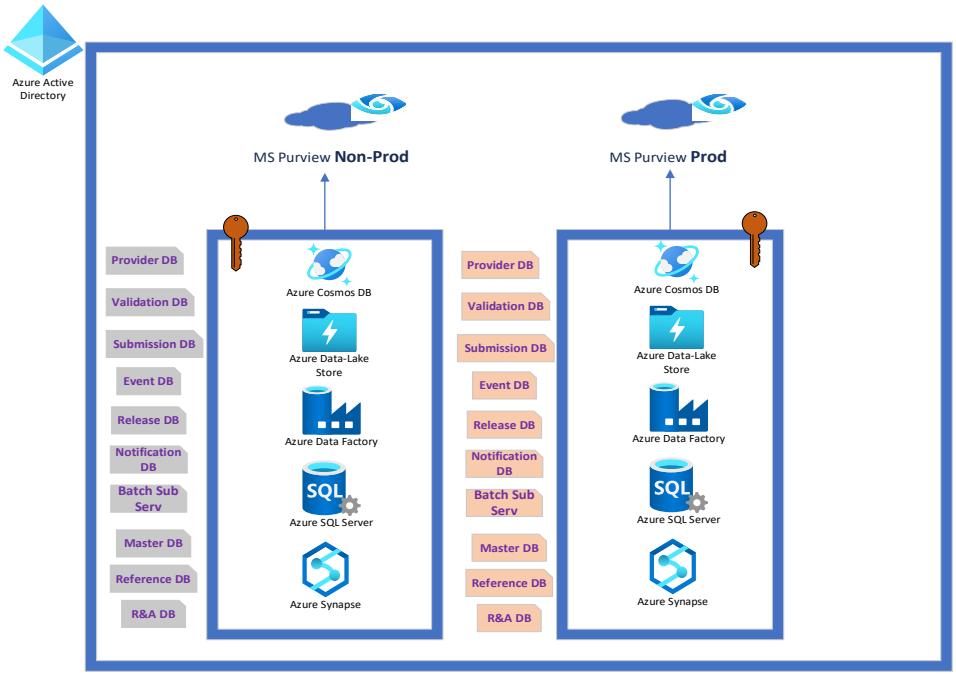


Figure 16: MS Purview Environments(F16)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1098 of 1295

4 Defence in Depth

To ensure Purview access is secured from all aspects – ensuring it is impenetrable and secure from all cloud and online threats, the '**Defence in Depth**' model is adopted – with 4 layers of protection:

I. **Networking**
II. **Security**
III. **Threat Protection**
IV. **Information Protection**

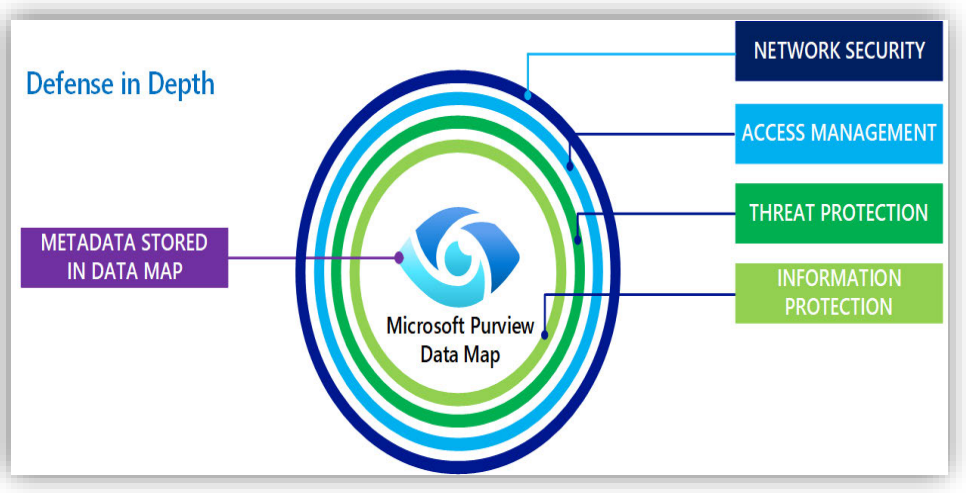


Figure 17: Purview Security Layers(F17)

4.1 Networking

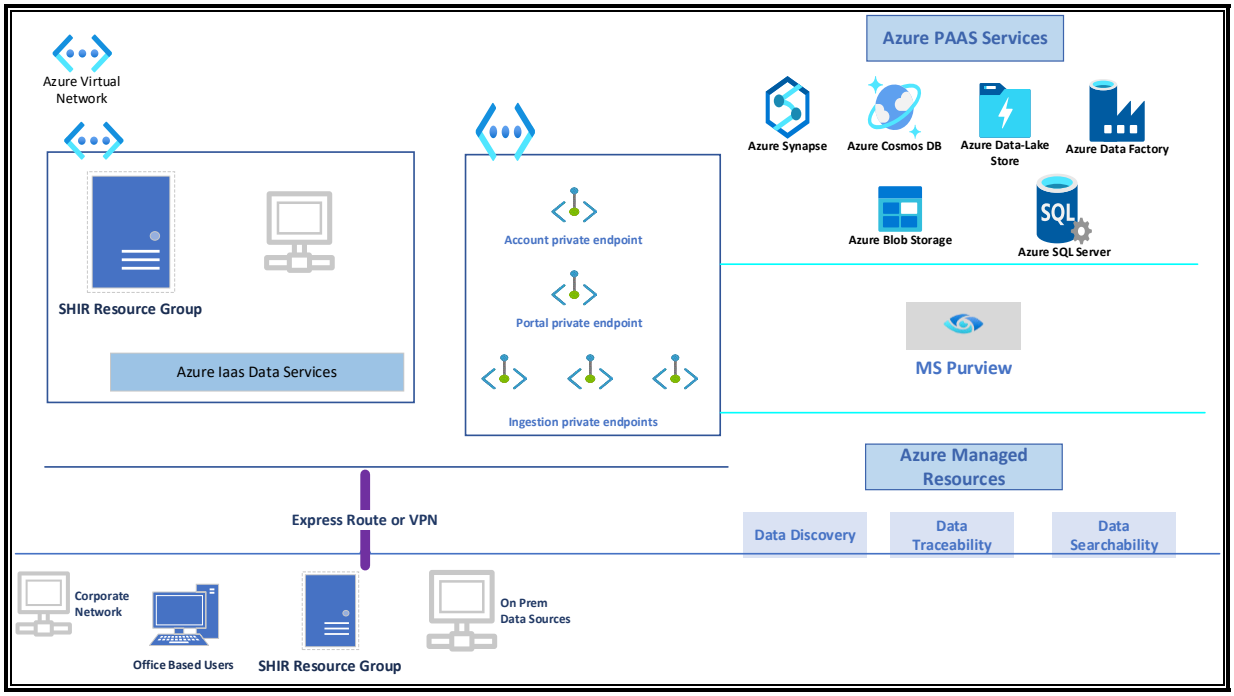


Figure 18: MS Purview Network Security (F18)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1099 of 1295

MS Purview, being a Platform as a Service (PAAS), the following network security capabilities will be enabled for MS Purview accounts:

- End-to-end network isolation using Private Link Service.
- Microsoft Purview Firewall to disable Public access.
- Deployment of Network Security Group (NSG) rules for subnets where Azure data sources private endpoints, Microsoft Purview private endpoints and self-hosted runtime VMs are deployed.
- Implement Microsoft Purview with private endpoints managed by a Network Virtual Appliance, such as Azure Firewall for network inspection and network filtering.
- As MS Purview will be used from inside our private network for STARS, we will use Azure Private Link Service with MS Purview accounts for partial or end-to-end isolation to connect to MS Purview governance portal, access MS Purview endpoints and to scan data sources.
- The Microsoft Purview account private endpoint will be used to add another layer of security, so only client calls that are originated from within the virtual network are allowed to access the Microsoft Purview account. This private endpoint will also be a prerequisite for the portal private endpoint.
- The Microsoft Purview portal private endpoint will be required to enable connectivity to Microsoft Purview governance portal using a private network.
- Microsoft Purview can scan data sources in Azure or an on-premises environment by using ingestion private endpoints.

For further in-depth details on MS Purview Networking – please refer to the **'DEWR - TSD - Azure Service Design – Purview'** document.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1100 of 1295

5 Security

5.1 Identity Governance and Access Management (Authentication and Authorization)

To gain access to STARS Microsoft Purview, users will be authenticated and authorized.

- Authentication is the process of proving the user is who they claim to be.
- Authorization refers to controlling access inside Microsoft Purview assigned on collections.

We will use Azure Active Directory to provide authentication and authorization mechanisms for Microsoft Purview inside Collections.

Microsoft Purview roles will be assigned to the following security principals from the Azure Active Directory tenant that is associated with Azure subscription where Microsoft Purview instance for STARS will be hosted:

- Users and guest users (if they're already added into your Azure AD tenant)
- Security groups
- Managed Identities
- Service Principals

Microsoft Purview fine-grained roles will be assigned to a flexible Collections hierarchy inside the Microsoft Purview instance.

For further in-depth details on MS Purview Security – please refer to the **'DEWR - TSD - Azure Service Design – Purview'** document.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1101 of 1295

6 Requirements

6.1 Functional

FR No	Requirement Name	Description	Acceptance Criteria	Traceability
FR038	Data lineage	The VDS system must provide capability for data lineage so that consumers and System Operator know how data has changed in its lifecycle from source to consumption.	Given submitted training data When users query the changes in data between source and consumption Then an audit log of changes can be viewed	MS Purview will have the ability to perform lineage – from target to source – performing reverse trace capability through API connectors. Please see section 3.5.
FR039	Metadata	The VDS system must enable metadata management. DEWR is currently using Aristotle and Azure Purview for metadata management. The compatibility of these systems needs to be ensured.	Given the ability to maintain data about data (i.e. metadata) When data is edited and captured Then metadata will be compatible with DEWR's systems - Aristotle and Azure Purview	All Purview metadata will be stored in its own repository and metadata is Upserted on the fly as it changes through live connectors. Please see section 3.2

6.2 Non-Functional

The Non-Functional requirements are more applicable to a Data Management Framework and will be addressed in a separate document after the Data Management Framework document(s) are available.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1102 of 1295

7 Appendix

7.1 Acronyms

Acronym/ Abbreviation	Definition
API	Application Programming Interfaces
AVETMISS	Australian Vocational Education and Training Management Information Statistical Standard
DEWR	Department of Employment and Workplace Relations
DXC	STARS Solution Provider and all round rockstars
FFS	Fee For Service
FR	Functional Requirement
MDM	Master Data Management
NCVER	National Centre for Vocational and Education Research
NFR	Non-Functional Requirement
Error! Reference source not found.	Organisational Change Management
RTO	Registered Training Organisation
SMS	Student Management System
STA	State Training Authority
STARS	Student Training & Activity Reporting System
TGA	Training.gov.au
TO	Training Organisation
VDS	VET Data Streamlining
VET	Vocational Education & Training
MS Purview	Microsoft Data Governance Tool
SRMP	Security Risk Management Plan
SSP	System Security Plan
IRP	Incident Response Plan
SOP	Standard Operating Procedures
AD	Active Directory
ARM	Azure Resource Manager
AWS	Amazon Web Services
CA	Certificate Authority
DNS	Domain Name System
DR	Disaster Recovery
FQDN	Fully Qualified Domain Name
HCE	Hybrid Cloud Environment
SHIR	Self-Hosted Integration Runtime
SOP	Standard Operating Procedure
SQL	Structured Query Language
TSD	Technology and Services Division
VM	Virtual Machine
VNET	Virtual Network

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1103 of 1295

7.2 Databases and Repositories

Database	Azure
Provider DB	Azure Cosmos + Azure Blob Storage for Uploaded files
Validation DB	Azure Cosmos
Submission DB	Azure Cosmos
Event DB	Azure Cosmos
Release DB	Azure Cosmos
Master DB	SQL Server
Refernce Data	SQL Server
R & A	Sql Server
Notification	Azure Cosmos
Batch Submission Service	Azure Blob Storage for Uploaded files

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1104 of 1295

Australian Government
Department of Employment and Workplace Relations

Detail Design Document

Data Quality Framework

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1105 of 1295

1 Document Management

1.1 Document Control

Version	Change Description	Date
0.1	DRAFT	05 May 2023
0.2		
1.0		

NOTE: This version of this document was valid at the time it was printed. Please regularly check for any updated versions. This is a living document and will be updated as required.

1.2 Document Author

Role	Name
s 47F(1)	Data Architect

1.3 Document Reviewers

Role	Name

1.4 Document Approval

Role	Name	Approval Date
Director, STARS Delivery	s 47F(1)	

1.5 Related Documents

Reference	Document
A	STARS Solution Requirements Document – D.0.3.docx
B	STARS Solution Design – D.0.3.docx
C	Detailed Design Document – Rules Service D.0.1.docx
D	Detailed Design Document - Master Data Service D.0.3.docx
E	Detailed Design Document – Submission Service D.0.3.docx
F	Detailed Design Document – Validation Service D.0.2.docx
G	Detailed Design Document – Notification Service V2.0.docx
H	Detailed Design Document – Reference Data Service D.0.3.docx

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1106 of 1295

Reference	Document
I	Quality Data Framework Diagrams - V0.1
J	Detailed Design Document – Release Service D.0.1.docx
K	Detailed Design Document – Provider Service D.0.4.docx
L	Detailed Design Document – R&A Provisioning Service D.0.3.docx
M	STARS Detailed Design - Azure Infrastructure and Networking - v1.0
N	Detailed Design Document - IDAM D.0.4

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1107 of 1295

Contents

1 Document Management 2
1.1 Document Control 2
1.2 Document Author 2
1.3 Document Reviewers 2
1.4 Document Approval 2
1.5 Related Documents 2
Contents 4
2 Introduction 6
2.1 Purpose 6
2.2 Intended Audience 6
3 Scope and Exclusions 7
3.1 In Scope 7
3.2 Out of Scope 7
4 Design Decisions 8
5 Data Dependencies 8
6 Functional Requirements 9
7 Feature Overview 10
7.1 STARS Data Quality Framework 10
7.1 Current State 13
7.2 Target State 14
7.2.1 Data Quality Gateways 14
7.2.2 Data Quality Data Flow 16
8 Data Quality Reports 18
8.1 Bronze Data Layer – Data Quality Reports 18
8.1.1 Error/Warning Summary Report 18
8.1.2 Error/Warning Detail Report 19
8.2 Silver Data Layer – Data Quality Reports 20
9 Data Profiling 21
9.1.1 Column Quality 21
9.1.2 Column Distribution 22
9.1.3 Column Profile 22
10 Azure Synapse Link for Cosmos DB for Reporting and Data Profiling 23

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1108 of 1295

11 Aggregated Data Checks 23
11.1 Aggregated Data Checks and Rules 25
11.1.1 Shift in training activity 25
11.1.2 Shift in program completions 25
11.1.3 Shift within an element 25
11.1.4 Large shifts in demographics 25
11.2 Aggregated Data Output Edit Action 25
12 Dependency on Other Services 25
12.1 Provider Service 26
12.2 Submission Service 26
12.3 Validation Service 26
12.4 Notification Service 26
12.5 Rules Service 26
12.6 Reference Data Service 26
12.7 Provider Management Portal 26
12.8 Release Service 26
12.9 IDAM 26
13 Appendix 27
13.1 <Appendix 1> 27
14 Acronyms 28

Figure 1: STARS Data Quality Framework (I1) 11
Figure 2: Current Data Quality Stages (I2) 13
Figure 3: Current AVS Data Quality Checks and Reports (I3) 14
Figure 4: Target State Data Quality Stages (I4) 15
Figure 5: Data Quality Data Flow (I5) 17
Figure 6: Current Collection error/warning Summary (I6) 19
Figure 7: Current Collection error/warning Detailed (I7) 20
Figure 8: Data Profiling (I8) 21
Figure 9: Azure Synapse Link for Azure Cosmos DB 23
Figure 10: Aggregated Data Check (I9) 24

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1109 of 1295

2 Introduction

This document covers the design of the STARS Data Quality Framework.

Note: This is a living document that will be updated as designs for various other STARS services are developed and evolve. Accordingly, this document will evolve to reflect and document Data Quality Framework components as the STARS solution build progresses.

STARS Data Quality Framework will provide a systematic approach for assessing the quality of collected data as it moves through the data layer stages in STARS.

The Data Quality Framework design will include

- the data quality stage gates and the data flow through the various data quality layers
- the data quality reports and dashboards
- the data profiling of submitted training activity data
- the aggregated data checks on the master data

2.1 Purpose

The purpose of this detailed design document is to provide a technical description of the Data Quality Framework in STARS. The document is intended for technical stakeholders, including architects, software engineers, administrators, and operations teams, who are responsible for designing, developing, deploying, and operating the STARS solution in Azure.

The document also serves as a reference for ongoing management and evolution of the STARS solution and to ensure that the STARS solution is deployed and operated in accordance with the best practices and guidance provided by the Microsoft Cloud Adoption Framework and the Azure Well-Architected Framework.

The document is intended to be a living document, updated as the STARS solution evolves and changes over time. It provides a clear and concise understanding of the Azure infrastructure and networking components and their configurations, helping to ensure the STARS solution is deployed, operated, and managed in a consistent and efficient manner.

2.2 Intended Audience

This detailed design document provides technical information and guidance to help technical stakeholders understand the Data Quality Framework Service components of the STARS solution. The document is expected to be used by technical stakeholders who have a good understanding of cloud computing concepts and are experienced in the Azure platform.

This document is not intended for non-technical readers, nor for readers who are not experienced in the design and configuration of Azure-based solutions; informational and educational resources regarding Azure are readily available at https://Azure.microsoft.com

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1110 of 1295

The primary audience for this document includes:

- Program technical staff: This includes architects, software engineers, administrators, and operations teams within the program responsible for the delivery of STARS solution.
- External technical stakeholders: This includes technical staff from other federal, state, and territory governments agencies, including VDS embedded team members.
- ICT Technical Staff: This includes a variety of technical roles involved during the VDS program and post-go live, including network engineers, systems administrators, database administrators, security engineers, and DevOps engineers.

3 Scope and Exclusions

3.1 In Scope

The following are in-scope of this detail design solution:

- This Data Quality Framework design will include the functional aspects of data quality including data quality check gates, and data quality data flow
- Data Quality Framework will be applied to data submitted from Training Providers
- This design will cover the data quality reports including validation and error trend reports
- This design will cover the data profiling to easily assess data quality trends
- This design will cover the Aggregated data checks business rules and the process

3.2 Out of Scope

The following are out of scope:

- This Data Quality Framework design will be supported by various STARS services like Provider Service, Submission Service, Validation Service, Rule Service, STA Integration Service etc. as outlined in Section 1.5 of this document. These STARS services are out of scope for this document and will be covered in the specific services detail design solution.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1111 of 1295

4 Design Decisions

At the time of writing this document, the requirements for below design are not available and will be provided later as part of NCVER's Work Package 02 (WP02).

- the data quality reports and dashboards
- the aggregated data checks and process on the master data

The above design items have been captured in DevOps as action items detailed below.

Table 1: Design Decisions/Action Items

Decision #	Design Decision	Status
DD01	Data Quality Aggregated Data Check schedule/trigger frequency (monthly/quarterly/yearly) will be confirmed as part of NCVER WP02 package. Action Item 3589	Open
DD02	5-year historical data in Master database or R&A database and Aggregated Data check business rules will be confirmed as part of NCVER WP02 package. Action Item 3153	Open
DD03	Data Quality report and dashboard requirement on Bronze Data Layer will be confirmed as part of NCVER WP02 package. Action item 3130, 3131, 3132, 3134, 3135	Open
DD04	Data Quality report and dashboard requirement on Silver Data Layer will be confirmed as part of NCVER WP02 package. Action item 3142, 3143, 3145	Open
DD05	Details on this dashboard and metrics required for tracking RTOs from NCVER, this will be confirmed as part of NCVER WP02 package. Action item 3152	Open

5 Data Dependencies

To implement Data Quality checks in STARS, the build and test is dependent on having appropriate test data (5 years data) in STARS environment (Open item 3153).

And for Aggregated data checks development in Production, the implementation depends on the availability of historical data in STARS and data migration plan.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1112 of 1295

6 Functional Requirements

FR No	Requirement Name	Description	Traceability
FR075	Data Profiling	The VDS system must have the ability to perform data profiling quickly, and to easily assess data quality trends on new datasets or new elements in existing datasets, for which there is no historical data to support trend analysis or the creation of data quality business rules. Data Profiles should be reportable and accessible to custodians of those datasets.	Section 9
FR046	Reporting on data submission and validation outcomes	The VDS system must have the ability to generate validation reporting that can be extracted. Standard reports on data validation will be readily available outlining the outcomes of attempted data submissions from training organisations, including success rates, reasons for validation failures, and related trends.	Section 8
FR109	Error and Warning Report for STA	As a data custodian, I want to be able to view the list of unresolved errors and warnings so that I am aware of the data quality and can follow up from there.	Section 8
FR111	Error and Warning Report for Training Org	As a TO user, I want to be able to view the list of unresolved errors and warnings related to my organisation so that I am aware of the data quality and can follow up from there.	Section 8
FR016a	Output Edits	The VDS System must perform output edits on the aggregated data when data is updated/refreshed in the Master Database which includes checking data correctness against aggregated historical VET data and/or usage of statistical analysis to detect data inconsistencies.	Section 11.1
FR016b	Output Edits Action	The VDS system must perform output edits on the aggregated data in the Master Database when the system detects data inconsistency a) Identify the inconsistency b) Notify the Provider (fee for service) and the STA/Funder (government subsidised) of the	Section 11.2

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1113 of 1295

FR No	Requirement Name	Description	Traceability
		inconsistency for resolution through an agreed process	
FR017	Store output edited data	The VDS System must store the output edited data in the data warehouse (analytics databases).	Section 11

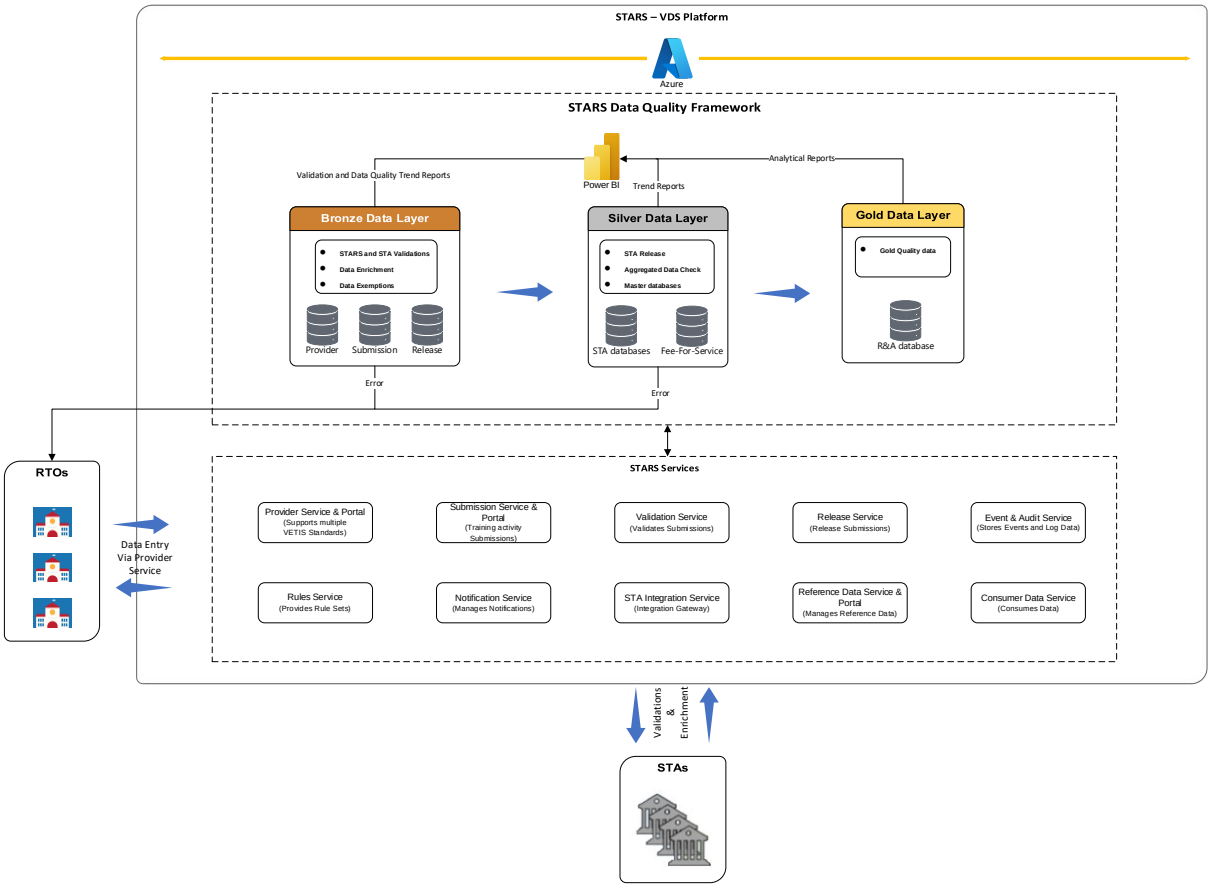
7 Feature Overview

Focus of the STARS Data Quality Framework design is to build in data quality control mechanisms, data quality rules, and data quality management workflows with the support of STARS services to manage and improve data quality.

7.1 STARS Data Quality Framework

The STARS Data Quality Framework Design will reflect the following broad categorisations for data quality based on the Medallion Architecture.

- Bronze data layer
- Silver data layer
- Gold data layer



Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1114 of 1295

Figure 1: STARS Data Quality Framework (I1)

Bronze: the raw VET data received via Provider Services API in STARS from Training Providers and STAs undergo data quality checks and validations and are stored in non-relational data storage in the Submission database.

The Provider service covers 'pre-validation', or validation in preparation for submission and will be implemented in Azure Cosmos DB.

The Submission Database will be implemented in Azure Cosmos DB. The submission DB is a document database that holds all records submitted via the provider service. The submission service facilitates the validation of records in the submission database using the validation database. It will store VET-IS Activity packets alongside submission information including validation status.

The Submission database stores

- Submitted records

For more details on the submission services, refer Submission Service detail design document.

The Release database stores

- Data enrichment changes
- Data enriched records
- STA release information

STAs might review their data if it requires data enrichment before releasing data in the STAs Master database.

For more details on the release services, refer Release Service detail design document.

Azure Synapse Link will be used for near real-time insights on bronze data layer to build data quality reports and dashboards, this will have no impact on operational workloads.

Silver: derived from VET bronze level data that has undergone validation using STARS and STA validation rules. Silver data will be suitable for STA operational needs (e.g., to support RTO payment cycles) and STA reporting.

The sliver layer master databases stores

- Validated data enriched STA specific records (STAs master database)
- Fee for service records

Silver level data will be stored in STA specific Azure SQL master databases. Data quality reports and dashboards will be built in Power BI as required for reporting.

Fee-For service records will be stored in its own Azure SQL database.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1115 of 1295

For more details on the Master databases, refer Master Data Service detail design.

Gold: derived from VET silver level data that has further undergone data quality processes like aggregated data checks and the release process. Gold data is suitable for national and STA reporting needs and will be stored in read-optimised reporting data stores (R&A database) to facilitate reporting.

The R&A Database will be implemented in Azure SQL Database.

The R&A database stores

- All released records that have passed Aggregated data checks

The above Data Quality Model (medallion architecture) in STARS will be supported by many STARS services as below.

Provider Service: The Provider Service stores, maintains activity-related data, and will support VET Information standards. When it receives a request to submit activity, it extracts relevant activity data and formats it into an activity submission request (i.e., JSON payload) for transmission to the Submission Service.

Submission Service: The Submission Service receives, records, and validates all submissions along with the full history of submission events. It is the source of truth for all submissions (valid or not). Valid submissions are passed to the Mastering service automatically.

Release Service: The release service enriches data received from the Submission service and stores information pertaining to data release and data enrichment in the Release database.

Validation Service: The Validation Service validates submissions based on applicable data quality and business rules obtained from the Rules Service. It records all validation requests and results along with the full history of validation events.

Rule Service: The Rules Service stores and then provides rules & rule sets for the main services within STARS along with the full history of rules management events. Rules are managed via the Rules Portal.

Notification Service: The Notifications Service manages and distributes Notifications raised by any of the other services. Notifications are usually raised as events on the STARS Event Grid.

STA Integration Service: The integration gateway exchanges messages between STARS and projects (maps) VIS-schema data to STA-specific schema data.

Reference Data Service: The Reference Service manages and provides Reference Data for many business rules, the Rules Engines, and workflows throughout STARS. The Reference database will also house the threshold check value that will be used in DQ Aggregated Data Check.

Data Consumption Service: Data Consumption Services are provided to facilitate the exchange of VET data between STARS and downstream VET data consumers and systems via ETL Pipelines services and R&A database.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1116 of 1295

7.1 Current State

Data quality is an important element of NCVER's role as the data custodian of the national VET administrative collections. NCVER's data quality framework provides a systematic approach for assessing the quality of collected data as it moves through stage gates of quality.

NCVER uses AVETMISS Validation Software (AVS) for data file validation and submission system.

There are five stages of data quality process in NCVERS Data Quality framework as shown in the diagram below.

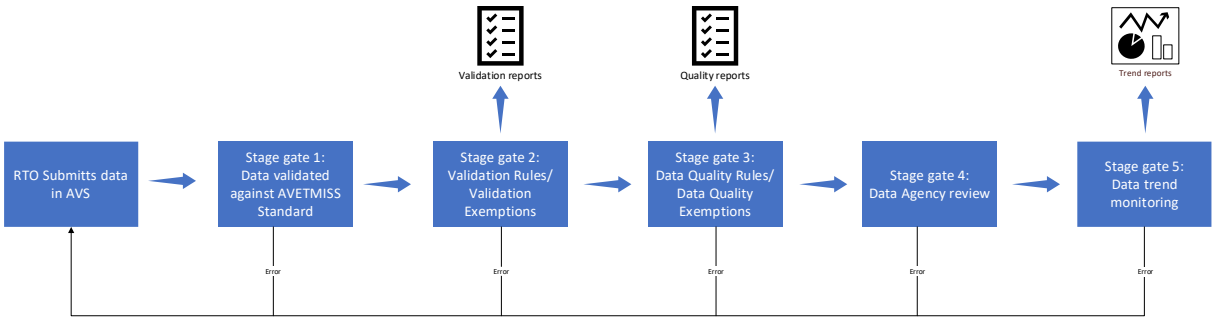


Figure 2: Current Data Quality Stages (I2)

Stage gate 1: NCVER validates the submitted data against the AVETMISS standard.

Stage gate 2: AVS generates errors against data records that do not comply with the business rules in the AVETMIS Standard. These errors are then reviewed and fixed by data providers before they submit their final data to NCVER.

Stage gate 3: AVS performs data quality checks and produces reports that profile data for an agency prior to submission.

Stage gate 4: Data agencies review their data using their own internal controls to assess the accuracy of their data before submitting data to NCVER

Stage gate 5: NCVER uses the AVS data quality reports to review data trends.

Table 1 (taken from NCVERs Data Quality Framework) shows the number of data validations, data quality checks, reports, and data trend analysis applied at each stage gate, which varies depending on the collection, collection period and type of submitter.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1117 of 1295

Table 1: The number of AVS data validations and data quality checks (errors and warnings), AVS validation and data quality reports and data trends analysis for the national VET administrative collections

	National VET administrative collection			
	National VET Provider collection		National VET in Schools collection	National Apprentice and Trainee collection
Type of data submitters	STAs	Direct RTO submissions to NCVER	BoS	STAs
Number of data submitters via this pathway	14-16 STAs	2-3k RTOs	8 BoS	8 STAs

Stage gates		**Number of quality controls performed**			
Gate 1	AVETMIS Standard	N/A	N/A	N/A	N/A
Gate 2	AVS validation errors[a]	200		160	94
	AVS validation warnings	34		27	5
	AVS validation reports	5	5	4	4
Gate 3	AVS data quality checks	25-87[b]	×	29	21
	AVS data quality reports	18	×	18	3
Gate 4	Data agency review	Q	Q	Q	Q
Gate 5	Data trend monitoring	73	×	74	13

Key

Direct RTO submissions to NCVER - Registered Training Organisations (RTOs) who submitted directly to NCVER

Data agency - For this purpose of this paper, Data agency is State Training Authorities (STAs), Boards of Study (BoS) or Registered Training Organisations (RTOs)

N/A - Not applicable as a numeric measure

× - Not accessible

Q - Represents data agency validations and quality checks, the number of which varies and is unknown to NCVER

a) The number of AVS validation errors excludes the number of AVS basic validations performed (e.g. short records, illegal characters, blank records).

b) There are 25 distinct AVS data quality checks for Q1 to Q3 and 26 for Q4 distinct AVS data quality checks. When a STA submission is a combined aggregate submission (i.e., all their training activity for a jurisdiction is in one submission) all the AVS data quality checks performed on the Training activity (NAT00120) file and Program Completed (NAT00130) file are analysed by NCVER reporting type to represent the various sectors reported. When this occurs, the number of data quality checks is reported by NCVER reporting type within a data submission.

Figure 3: Current AVS Data Quality Checks and Reports (I3)

7.2 Target State

This section covers the Data Quality gateways and the data quality data flow in STARS.

7.2.1 Data Quality Gateways

There are two stages of data quality check process in the STARS Data Quality framework as shown in the diagram below.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1118 of 1295

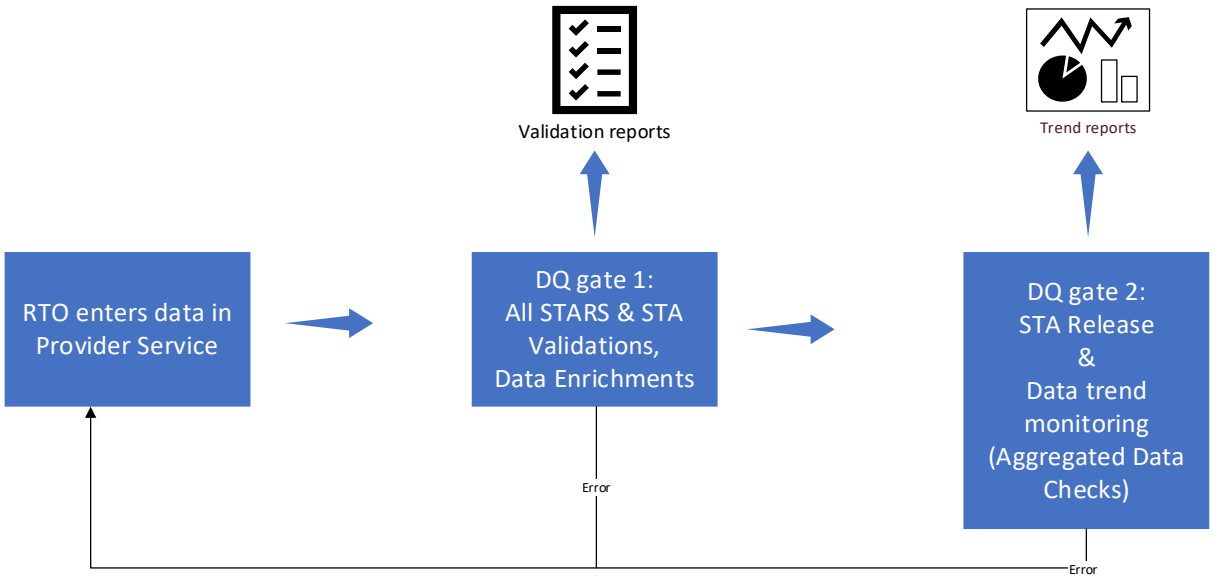


Figure 4: Target State Data Quality Stages (I4)

DQ gate 1:

The Provider Service acts as the first DQ gateway between Training Providers and STARS. Provider Services in conjunction with Validation Services & Rules Services validates the entered data against the STARS rules. The Provider service covers 'pre-validation', or validation in preparation for submission.

If there is any error with the entered data, the Provider service will return an error and the error will have to be corrected before the data is entered again.

If there is no error in the entered data, the data is submitted to the Submission service for validations.

The submission DB holds all records submitted via the provider service. The submission service facilitates the validation of records in the submission database using the validation database via Validation Service and the Rules service.

The data validation/exemptions will include e.g.

- valid values (e.g., checks that PO, CMB, RSD etc are the values reported for Postal delivery type code)
- reference data which is incorrect at source may require an exemption.

The above validation rules and exemptions will be covered in detail in the Rules Service detailed design document.

For more details on the submission services, refer Submission Service detail design document.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1119 of 1295

On successful validation by the submission service the Record is checked by the release service to determine if it requires data enrichment by the STA. If data enrichment is not required by STAs, then the record moves to the STA specific master database. If data enrichment is required for, the record goes through DQ gate 2 stage.

Data Quality validation trend reports will be generated in Power BI. Data Quality reports are covered in section 8 of this design document.

DQ gate 2:

There are two checks applied at this stage

- STA release, in case data enrichment is required by STA
- Aggregated data check

STA Release: If the record does require enrichment

- it notifies the STA and acquires the Data Enrichment information from the STA via the release service API and stores this information in the release database.
- it acquires the original record from the Submission DB and applies the patch to create a data enriched version of the record and again stores that in the release database.
- This data enriched record in the release database is used to perform validation on the enriched version of the record using the validation service.
- On successful validation the data enrichment is accepted, and the record is moved on to the STA specific master database.

For more details on the release services and its data enrichment process, refer Release Service detail design document.

In parallel to the STA release, STARS will also perform aggregated data checks on the validated data to review data trends. This is to control the data transmission between Master and the R&A database to detect any errors or flaws in the data before it is submitted into R&A database. Data that passes the aggregated data checks and released by STAs are copied to the R&A database for reporting.

The aggregated data checks include using the frequency counts for the current quarter and comparing the results with the previous corresponding quarter, looking for outliers in data trends and then examining variances (e.g., +/- 5% shift in program completions).

Aggregated Data Checks are covered in section 10 of this design document.

7.2.2 Data Quality Data Flow

The below figure shows the data quality data flow steps in STARS Data Quality Framework.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1120 of 1295

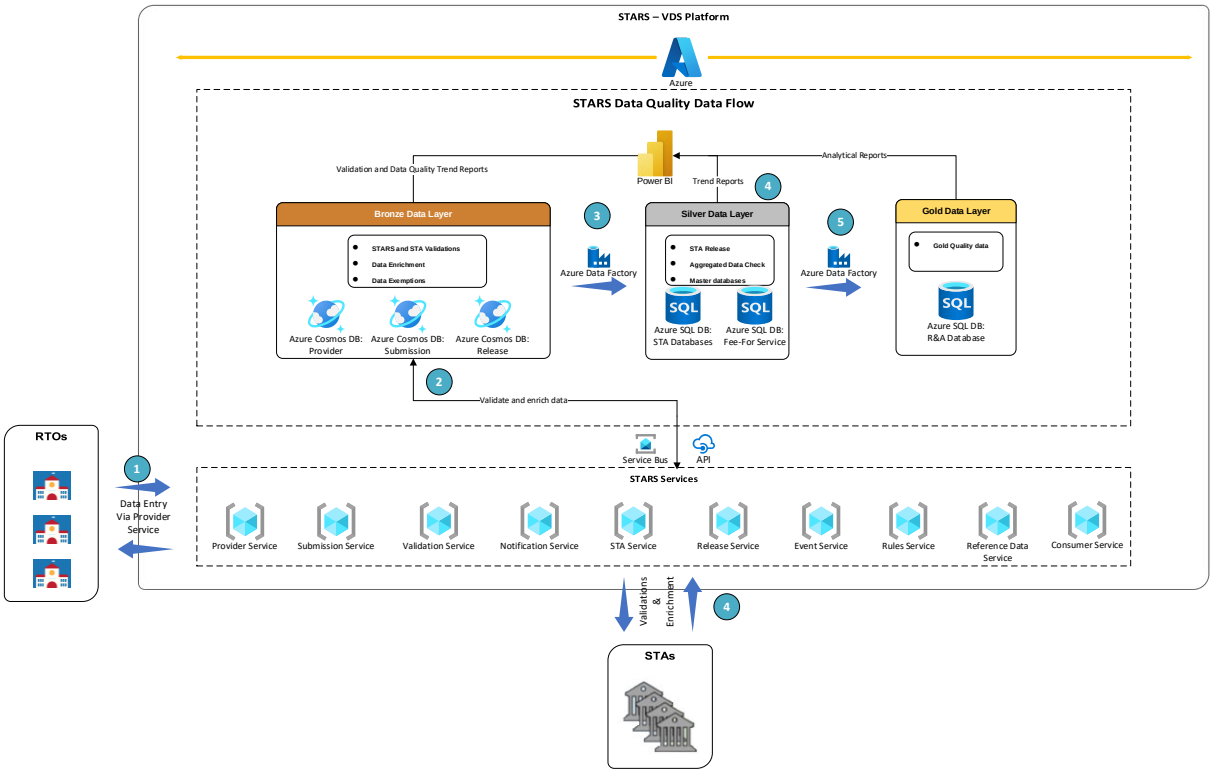


Figure 5: Data Quality Data Flow (I5)

Table 2:Data Quality Data Flow Steps

Step No	Description	Technologies
1	Training Providers submit training activity data via Provider Service	APIM, Azure Cosmos DB
2	Provider Service and Submission service in conjunction with Validation Services & Rules Services validates the entered data against the STARS and STA validation rules. Any errors are notified to STAs and Training Providers via Notification services.	APIM, Azure Cosmos DB, Drools, Service Bus
3	Training activity data that has passed validation rules and enriched are transferred to STA specific master databases and fee-for service data to Fee-For Service database	Azure Data Factory, Azure SQL Database
4	Enriched data is released in STA specific master databases and aggregated data checks applied on master databases.	Azure SQL Database, APIM
5	Data that passes the aggregated data checks and released by STAs are then transferred to the R&A database for reporting.	Azure SQL Database, Azure Data Factory

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1121 of 1295

8 Data Quality Reports

Data Quality Reporting detailed requirements have not yet been provided. DQ reports and dashboards will likely be developed during the life of the project as requirements are developed from NCVER WP02.

This section will cover the data duality reports required in STARS at the Bronze and Silver data layer. Data Quality reports in STARS will provide valuable insights into data quality issues and enable STA and NCVER to take proactive steps to address them.

8.1 Bronze Data Layer – Data Quality Reports

At the time of writing this document only South Australia jurisdiction has provided data quality report requirements as below. Data quality reports by other states, regulators and NCVER are yet to be provided.

8.1.1 Error/Warning Summary Report

Table 3: Report name: Error/Warning Summary

Report Attribute	Description
Severity Type	Error/Warning/Info
Category	National Validation Error/STA Validation Error/National Warnings/STA Payment Warnings/Info only etc
Count	Count of Type (Error/Warning)
RTO	Registered Training Organiser name
Ruled	Rule Id from Rules database
Rule Description	Rule description from Rules database
Date Time	Date and Time of the submitted records
State	Sate name

Filter: Category, RTO, Date

Drill Thru: RTO, Category

8.1.1.1 Current Sample Error /Warning Summary Report

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1122 of 1295

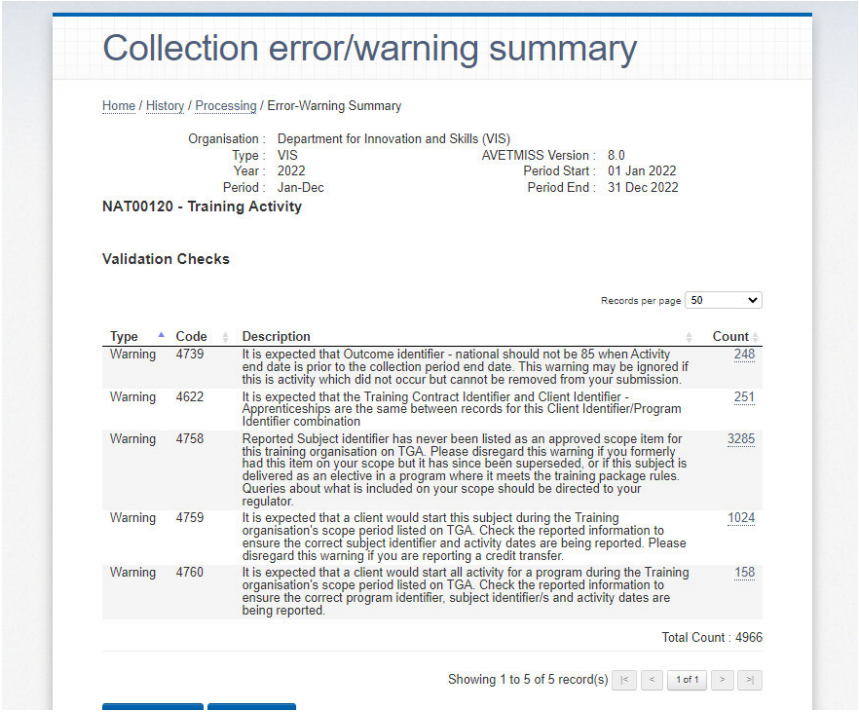


Figure 6: Current Collection error/warning Summary (I6)

8.1.1.2 Data Mapping of Report attributes

Data Mapping cannot be completed at the time of writing this document as the Collection packets are not yet confirmed. This section will be completed at a later stage.

8.1.2 Error/Warning Detail Report

Drill through to detailed error messages, report attributes are subject to change as per the collection packet received in STARS. These attributes are currently reported in AVS.

**Please note that the attributes mentioned below are from current AVS system and will change as per the collection packets received in STARS.*

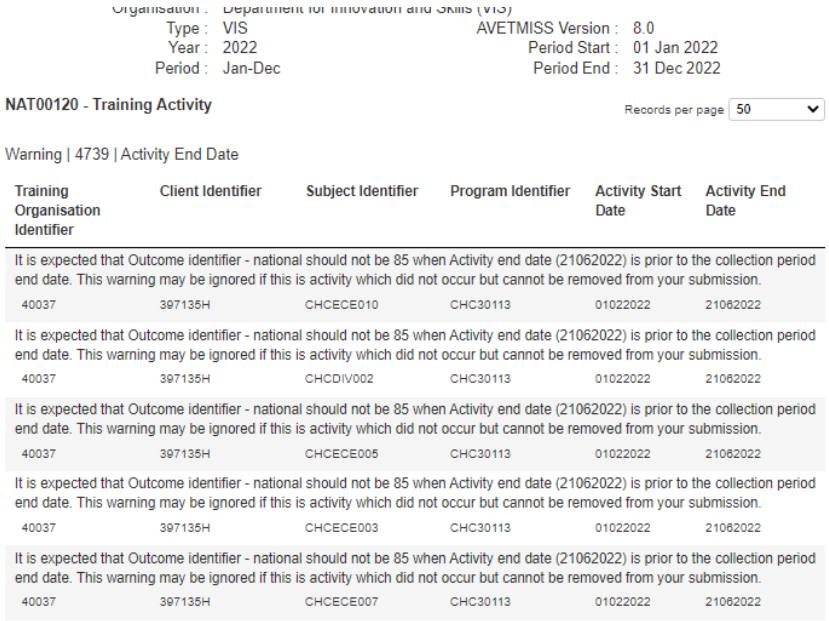
Table 4: Error/Warning Detail Report

Report Attribute	Description
File Id	File Id
File Name	File Name
Severity Type	Error/Warning/Info from Validation database
Ruled	Rule Id from Rules database
Rule Description	Rule description from Rules database

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1123 of 1295

RTO Identifier	Training Organisation Identifier
Client Identifier	Client Identifier
Subject Identifier	Subject Identifier
Program Identifier	Program Identifier
Activity Start Date	Activity Start Date
Activity End Date	Activity End Date

8.1.2.1 Current Sample Error /Warning Detail Report



Organisation : Department for Innovation and Skills (VIS)
Type : VIS
Year : 2022
Period : Jan-Dec
AVETMISS Version : 8.0
Period Start : 01 Jan 2022
Period End : 31 Dec 2022

NAT00120 - Training Activity

Records per page 50

Warning | 4739 | Activity End Date

Training Organisation Identifier	Client Identifier	Subject Identifier	Program Identifier	Activity Start Date	Activity End Date
It is expected that Outcome identifier - national should not be 85 when Activity end date (21062022) is prior to the collection period end date. This warning may be ignored if this is activity which did not occur but cannot be removed from your submission.					
40037	397135H	CHCECE010	CHC30113	01022022	21062022
It is expected that Outcome identifier - national should not be 85 when Activity end date (21062022) is prior to the collection period end date. This warning may be ignored if this is activity which did not occur but cannot be removed from your submission.					
40037	397135H	CHCDIV002	CHC30113	01022022	21062022
It is expected that Outcome identifier - national should not be 85 when Activity end date (21062022) is prior to the collection period end date. This warning may be ignored if this is activity which did not occur but cannot be removed from your submission.					
40037	397135H	CHCECE005	CHC30113	01022022	21062022
It is expected that Outcome identifier - national should not be 85 when Activity end date (21062022) is prior to the collection period end date. This warning may be ignored if this is activity which did not occur but cannot be removed from your submission.					
40037	397135H	CHCECE003	CHC30113	01022022	21062022
It is expected that Outcome identifier - national should not be 85 when Activity end date (21062022) is prior to the collection period end date. This warning may be ignored if this is activity which did not occur but cannot be removed from your submission.					
40037	397135H	CHCECE007	CHC30113	01022022	21062022

Figure 7: Current Collection error/warning Detailed (I7)

8.1.2.2 Data Mapping of Report attributes

Data Mapping cannot be completed at the time of writing this document as the Collection packets are not yet confirmed. This section will be completed at a later stage.

8.2 Silver Data Layer – Data Quality Reports

Data Quality Reporting detailed requirements have not yet been provided. DQ reports and dashboards will likely be developed during the life of the project as requirements are developed from NCVER WP02.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1124 of 1295

9 Data Profiling

Data Profiling report requirements have not yet been provided. This will be developed during the life of the project as requirements are developed from NCVER WP02.

This section covers the approach and the out of box data profiling available in Power BI.

Power BI desktop tool will be used to do data profiling of submitted data by the Training Providers in Submission database. This profiling tool will be accessible to the data custodians of this dataset, this will be defined in the security matrix in IDAM.

Data custodians will use Power BI desktop tool to perform data profiling to analyse data and identify patterns, relationships, and inconsistencies in the submitted data from Training Providers.

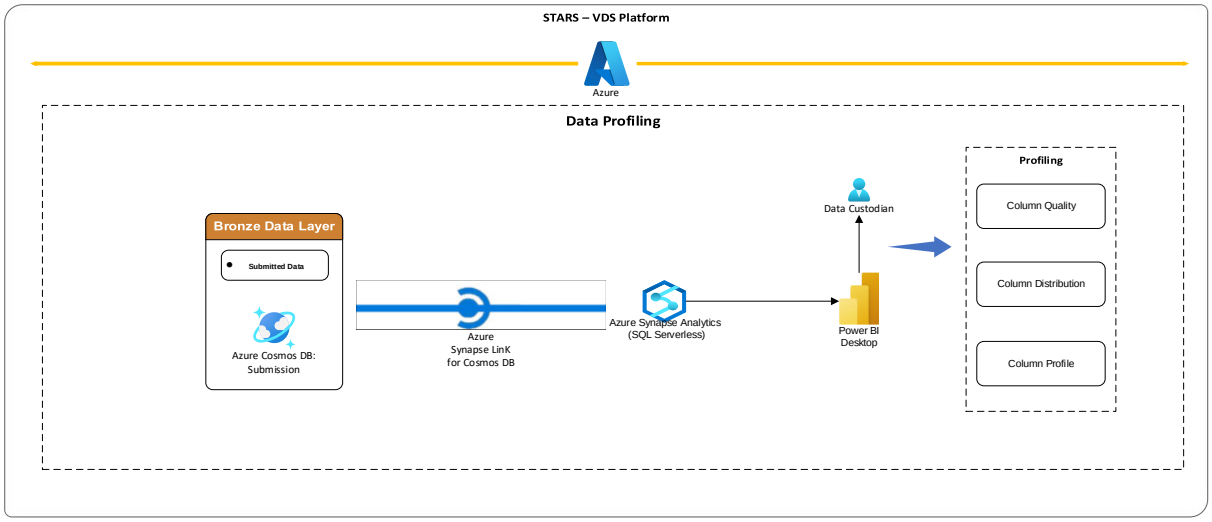


Figure 8: Data Profiling (I8)

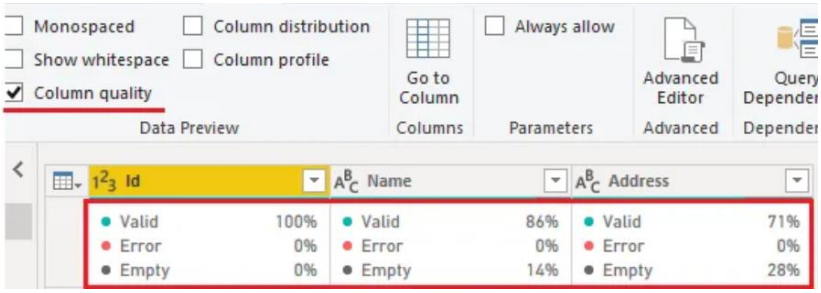
Below are the key features of data profiling in Power BI:

- Column quality
- Column distribution
- Column profile

9.1.1 Column Quality

Column quality will show valid, Error and Empty percentage of data values associated with the Selected table.

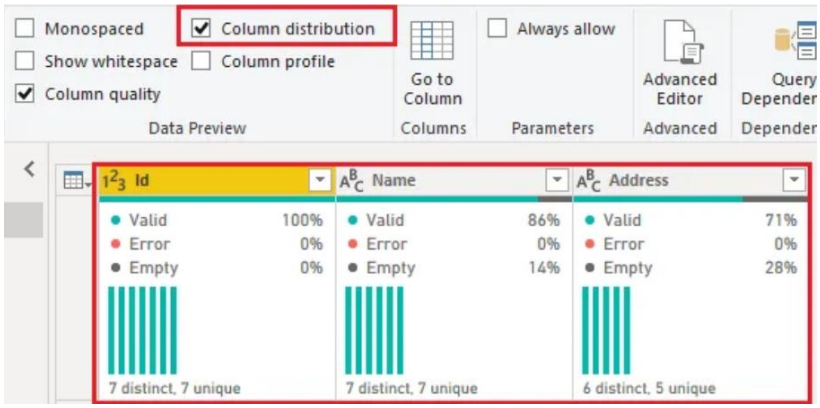
Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1125 of 1295



various transformations can be applied if required, like Remove Errors, Remove Empty, Replace Errors, keep duplicates, Remove Duplicates, etc.

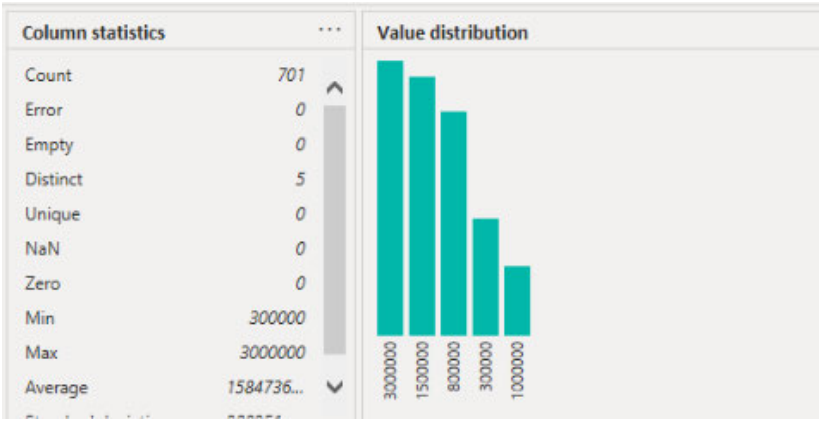
9.1.2 Column Distribution

Column distribution will show the inline value distribution histogram. For example, distinct records, unique records.



9.1.3 Column Profile

Column profile will show Column statistics which includes distribution measures like Count, error, empty, distinct, Unique, Empty string, Min, Max.



Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1126 of 1295

10 Azure Synapse Link for Cosmos DB for Reporting and Data Profiling

Power BI will connect to the T-SQL views, in Synapse serverless SQL pools, over Synapse link for Azure Cosmos DB. With Azure Synapse link, Power BI will analyse the Cosmos DB data in near real-time by using either Direct Query or Import mode.

Following are the steps to run analytical queries with the Azure Synapse Link for Azure Cosmos DB, these steps will be configured/deployed in Azure portal.

- Enable Azure Synapse Link for Azure Cosmos DB accounts
- Enable Azure Synapse Link for containers that will be enabled for analytics with Synapse link
- Connect your Azure Cosmos DB database to an Azure Synapse workspace
- Use Azure Synapse serverless SQL pool to analyse and visualize data in Power BI

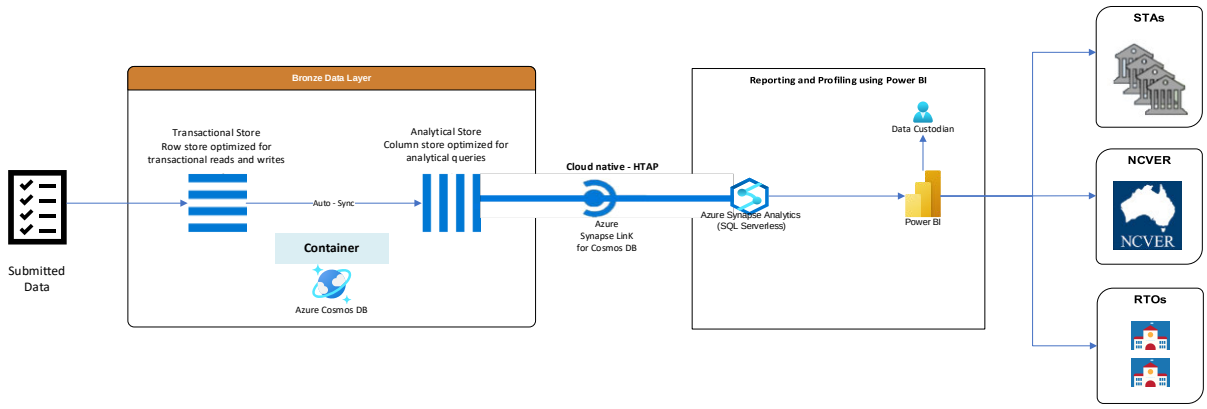


Figure 9: Azure Synapse Link for Azure Cosmos DB

As shown in the above diagram, the approach will be used for creating reports and dashboards and for data profiling. Reports created in Power BI desktop will then be published into Power BI service for collaboration with States, NCVER and regulators.

Separate workspace will be created for States, NCVER and Regulators as appropriate.

11 Aggregated Data Checks

Aggregated data quality check detailed requirements have not yet been provided. The business rules will likely be developed during the life of the project as requirements are developed from NCVER WP02.

STARS will perform the aggregated data checks when data is updated/refreshed in the Master Database which will include checking data correctness against aggregated historical VET data and/or usage of statistical analysis to detect data inconsistencies. If the data quality checks are deemed 'fatal', this will stop the flow of data downstream to Reporting & Analytics database.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1127 of 1295

Azure Data Factory will be used as an orchestration tool to implement Aggregated data checks on the master database, following is the high-level flow of this data quality check.

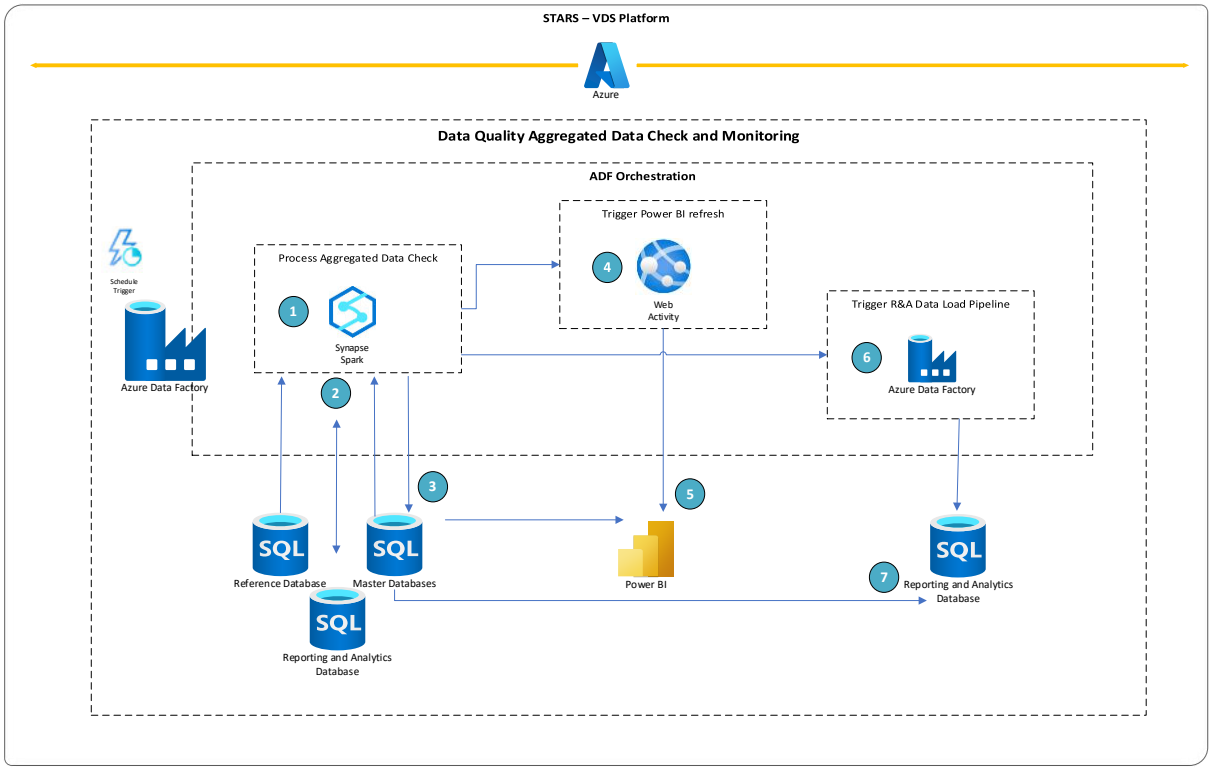


Figure 10: Aggregated Data Check (I9)

Table 5: Aggregated Data Check Steps

Step No	Description	Technologies
1 & 2	Azure data factory will run the Azure spark notebook that will extract data from Reference, Master and R&A database required for Aggregated data check	Azure Synapse Spark, Azure SQL Database, Azure Data Factory
3	Azure spark will process the aggregated data checks on the master data and publish the results back into the master database for further processing by setting the flag 'IsGoldready' to true for the records that have passed aggregated data checks.	Azure Synapse Spark, Azure SQL Database, Azure Data Factory
4	After successful completion of steps 1 to 3, Azure Data factory will trigger Power BI dataset refresh via Web activity task (this is an optional step, will be triggered if there is a use case)	Azure Data factory
5	Power BI imports the required data from Master database into its dataset for producing data quality reports to review data trends	Azure SQL Database, Power BI
6	After the aggregated data checks are performed on the master data, the data will be moved to R&A database either by a schedule refresh or a manual trigger based on a business process that is to be confirmed (Action item 3381). As per the business process, Azure Data factory will trigger ADF pipeline to load gold ready and released data from Master to R&A database for further reporting	Azure SQL Database, Azure Data Factory

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1128 of 1295

Step No	Description	Technologies
7	ADF pipeline extracts the gold and release ready data from the Master database and transforms the data into dimension and fact tables in R&A database	Azure SQL Database, Azure Data Factory

11.1 Aggregated Data Checks and Rules

This Aggregated data check will include using the frequency counts for the current quarter (monthly/quarter/yearly tbc) and comparing the results with the previous corresponding quarter, looking for outliers in data trends and then investigating anomalies like

- +/- 5% shift in training activity (student or enrolments)
- +/- 5% shift in program completions
- +/- 5% shift within an element
- 15% unknown for demographics

The following section will cover the business rules for performing these aggregated checks on the Master database.

11.1.1 Shift in training activity

Business rules to be added after NCVERs WP02.

11.1.2 Shift in program completions

Business rules to be added after NCVERs WP02.

11.1.3 Shift within an element

Business rules to be added after NCVERs WP02.

11.1.4 Large shifts in demographics

Business rules to be added after NCVERs WP02.

11.2 Aggregated Data Output Edit Action

Aggregated data check detailed requirements have not yet been provided. The business rules will likely be developed during the life of the project as requirements are developed from NCVER WP02.

12 Dependency on Other Services

Following are STARS services upon which Data Quality Framework has dependencies.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1129 of 1295

12.1 Provider Service

The Provider Service acts as the gateway between Training Providers and STARS. It provides the entry point for submissions from SMSs using the STARS public APIs.

12.2 Submission Service

The Submission Service API will be used for Submitting training activity and retrieving details of Training Activity that has been submitted.

12.3 Validation Service

The Validation Service API will be used for handling validation requests.

12.4 Notification Service

The Notification Service API will be used when sending notifications.

12.5 Rules Service

The Rules Service API will be used as part of the validation request process, specifically, to run Business Rules to determine the funding relationships for submitted records.

12.6 Reference Data Service

The Reference Data Service API will be used retrieve configurable static content, and VET-IS classification data for pre-population of forms and for Training Provider Data management.

12.7 Provider Management Portal

The Provider Management Portal will depend on this service to provide API endpoints for Training Provider management.

12.8 Release Service

The release service enriches data received from the Submission service and stores information pertaining to data release and data enrichment in the Release database.

12.9 IDAM

IDAM captures the detailed design requirements for Identity and Access Management for various stake holders, systems, and services in STARS.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1130 of 1295

13 Appendix

13.1 <Appendix 1>

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1131 of 1295

14 Acronyms

Acronym	Definition
ADF	Azure Data Factory
ADLS	Azure Data Lake Storage
API	Application Programming Interface
BI	Business Intelligence
CMB	Community Mail Bag
DEWR	Department of Employment and workplace Relations
DQ	Data Quality
ETL	Extract, Transform & Load
FR	Functional Requirement
IRAP	Information Security Registered Assessors Program
ISM	Australian Government Information Security Manual
MDM	Master Data Management
NFR	Non-functional Requirement
PO	Post Office Box
RMB	Roadside Mailbox/bag
STARS	Student Training & Activity Reporting System
VET	Vocational Education Training
VDS	VET Data Streamlining
IVETS	Integrated Vocational Education and Training System
DB	Database
DDL	Data Definition Language
R&A	Reporting and Analytics
RTO	Registered Training Organisation
SQL	Structured Query Language
SIT	System Integration testing

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1132 of 1295

STA	State Training Authority
SMS	Student Management
STARS	Student Training & Activity Reporting System
TP	Training Provider
UAT	User Acceptance Testing
VDS	VET Data Streamlining
VET	Vocational Education and Training
VIS	VET Information Standard
VM	Virtual Machine
VNet	Virtual Network

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1133 of 1295

Document 19

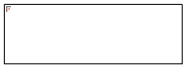
Australian Government
Department of Employment and Workplace Relations

High-Level Design

Student and Training Activity Reporting System (STARS)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1134 of 1295

ISBN
xxx-x-xxx-xxxxx-x [PRINT]
xxx-x-xxx-xxxxx-x [PDF]
xxx-x-xxx-xxxxx-x [DOCX]



With the exception of the Commonwealth Coat of Arms, the Department's logo, any material protected by a trade mark and where otherwise noted all material presented in this document is provided under a Creative Commons Attribution 4.0 International (https://creativecommons.org/licenses/by/4.0/) licence.

The details of the relevant licence conditions are available on the Creative Commons website (accessible using the links provided) as is the full legal code for the CC BY 4.0 International (https://creativecommons.org/licenses/by/4.0/legalcode).

The document must be attributed as the (Student and Activity Reporting System (STARS) High-Level Design).

Document control

Document authors

Version	Name	Description	Organisation	Date
v0.1	s 47F(1) and s 47F(1)	Initial draft	DEWR	27/08/2024
v0.2	s 47F(1)	Create v0.2	DEWR	28/08/2024
v0.3	s 47F(1)	Create v0.3 for provision to NCVER	DEWR	02/09/2024
v0.4	s 47F(1)	Removed Azure Synapse Analytics due to functionality overlap with Azure DataBricks. Updated references to registered training orgs (RTOs) to the more general 'training organisations' so as not to unintentionally exclude non-registered training organisations. Updated links to current requirements. Added clarifications on data validation/quality rules storage and management. Removed references to 'data ingestion' from 'Data Operations' section, so not to confuse with initial data ingestion	DEWR	09/09/2024

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1136 of 1295

Version	Name	Description	Organisation	Date
		from training organisations. Noted in 'Reporting, analytics and data consumption' section that provisionally submitted data will not be available to all consumers.		
v0.5	s 47F(1)	Added further environment details in 'Development and hosting approach' section. Re-write of 'Data flows' section.	DEWR	16/09/2024
v0.6	s 47F(1)	Updated environment details to align with what has been physically provisioned in the oobe Azure environment.	DEWR	21/10/2024
v0.7	s 47F(1)	Incorporated feedback from s 47F(1) and s 47F(1) De-coupled external IdP decision from current HLD.	DEWR	24/10/2024
v1.0	s 47F(1)	Bumped to v1.0, following DAF endorsement.	DEWR	07/11/2024
v1.1	s 47F(1)	Documented solution alignment with the DEWR Digital Business Strategy.	DEWR	07/11/2024
v1.2	s 47F(1)	Added updated business capability map.	DEWR	20/01/2025

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1137 of 1295

Version	Name	Description	Organisation	Date
		Updated document after feedback from states/territories/NCVER. Added Microsoft Purview as a solution component.		
v1.3	s 47F(1)	Azure Kubernetes Service replacing Azure App Service as hosting service for web application solution components. Updated Data flows section. Added Identity and access management section and updated solution overview to include Okta. Updated 'migrated' to 'replicated' in Table 7. Updated Integration section to provide a clearer overview. Updated Environments section.		

Document reviewers

Version	Name	Role	Organisation	Date
v0.2	s 47F(1)	Tech Writer review.	DEWR	28/08/2024

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1138 of 1295

Document approvers

Version	Name	Description	Organisation	Date
v0.7	Design and Architecture Forum (DAF)	HLD endorsed by the DAF. Bumped to v1.0.	DEWR	07/11/2024

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1139 of 1295

Contents

Document control **3**
Document authors 3
Document reviewers 5
Document approvers 6
1 About this document **10**
2 Related documents **10**
3 Stakeholder consultation **11**
4 Introduction **12**
4.1 Background 12
4.2 Dependencies 13
4.3 Assumptions 13
4.4 Constraints 13
4.5 Risks and mitigations 14
4.6 Scope 14
4.6.1 In scope 14
4.6.2 Out of scope 15
4.7 Stakeholders 15
5 Business context **20**
5.1 Objectives 20
5.2 Strategic alignment 20
5.2.1 National Skills Reform alignment 20
5.2.2 DEWR Digital Business Strategy alignment 21
5.3 Users 22
5.4 Requirements 22
5.5 Business process 22
6 Information security **23**
6.1 Information assets 23
6.2 System classification 24
6.3 STARS business capability map 24
7 Solution context **24**
7.1 Overview 24
7.2 STARS architecture and key components 25

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1140 of 1295

7.3 Key design decisions 27
7.4 Architecture patterns 30
7.4.1 Adoption of cloud architecture 30
7.4.2 Modular component deployment 30
7.4.3 Reuse existing DSD architecture patterns 30
7.4.4 Microsoft DataOps architecture 30
7.5 Architecture principles alignment 31
7.6 Components 36
7.6.1 Application 36
7.6.2 Data 41
7.6.3 Integration 49
7.6.4 Infrastructure 52
7.6.5 Network 57
7.6.6 Security 58
8 Operations 70
8.1 System class and availability 70
8.2 Backup and recovery 70
8.3 Logging and monitoring 71
8.4 Application/technology decommissioning requirements 72
8.5 Support 72
9 Development and hosting approach 73
9.1 Environments 73
9.2 Technical Debt register 74
9.3 Proof of Concept outcomes 74

Appendices

Appendix A | Acronyms and Abbreviations 74

Figures

Figure 1 | DSD Project Delivery Approach 10
Figure 2 | STARS business capability map 24
Figure 3 | STARS solution overview 25
Figure 4 | STARS Data Collection 36
Figure 5 | STARS System Operational Support 38
Figure 6 | Data Operations 40

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1141 of 1295

Figure 7 | Reporting and Analytics 41
Figure 8 | Data Flow Pattern 1a - direct submission pathway 45
Figure 9 | Data Flow Pattern 1b - direct submission pathway 46
Figure 10 Data Flow Pattern 2a indirect submission pathway data flow 47
Figure 11 Data Flow Pattern 2b indirect submission pathway data flow 48
Figure 15 | STARS integration with NCVER for importing reference data 50
Figure 16 | STARS integration with NCVER for importing classification data 51
Figure 17 | STARS integration with NCVER for importing AVETMISS data 51
Figure 18 | STARS integration with NCVER for exporting VET IS data 52
Figure 19 | Solution networking for STARS 58
Figure 20 | Training organisation primary user onboarding 60
Figure 21 | DataHub primary user invites subordinate user 60
Figure 22 | DataHub user authenticates to STARS Provider Portal 61
Figure 23 | DataHub training org user registers M2M credentials for use with STARS web API 62
Figure 24 | M2M authentication to STARS web API, using Workload Identity Federation 63
Figure 25 | Partner organisation user authentication - SAML example 65
Figure 26 | STA/STARS OAuth certificate-based authentication 67
Figure 27 | Azure Monitor 72

Tables

Table 1 | Related documents 11
Table 2 | HLD stakeholder consultation 11
Table 3 | STARS solution assumptions 13
Table 4 | STARS solution constraints 14
Table 5 | STARS solution risks and mitigations 14
Table 6 | STARS solution stakeholders 15
Table 7 | STARS solution information assets 23
Table 8 | STARS solution – key components 25
Table 9 | STARS key design decisions 27
Table 10 | STARS architecture principles alignment 31
Table 11 | STARS components 37
Table 12 | STARS Operational Support 38
Table 13 | Solution components utilised by STARS 40
Table 14 | Power BI component 41
Table 15 | Data stores crucial to the STARS solution 41
Table 16 | STARS reporting, analytics and data consumption 43
Table 17 | Data Flow Pattern 1a - direct submission pathway data flow steps 45
Table 18 | Data Flow Pattern 1b - direct submission pathway data flow steps 46
Table 19 | Data Flow Pattern 2a - indirect submission pathway data flow steps 47
Table 20 | Data Flow Pattern 2b - indirect submission pathway data flow steps 48
Table 21 | STARS infrastructure resources 52
Table 22 | Partner integration authentication methods 65
Table 23 | Access control and authorisation 67
Table 24 | Additional identity security measures 67

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1142 of 1295

Table 25 | System class and availability 70
Table 26 | Azure service redundancy settings 70
Table 27 | Azure SQL database backup and retention settings 71
Table 28 - Environments 73

1 About this document

This document articulates the STARS high-level design (HLD), consisting of the system architecture, data design, applications, and interfaces.

This is a key deliverable for the 'Pre-Delivery' phase, as outlined in the Digital Solutions Division (DSD) Project Delivery Approach. A range of stakeholders have been involved in the consultation, review, and approval process of the HLD.

The HLD must be approved as a key deliverable of the 'Pre-Delivery' phase, prior to this document being utilised in the 'Delivery' phase. The Portfolio Strategy and Assurance Team (PSAT) as well as the VDS Project's Solution Management Team (SMT) are key stakeholders in the consultation and review process. The HLD will be presented to the Design Architecture Forum (DAF) and Architecture Review Board (ARB) for approval. Figure 1 shows the DSD project delivery approach.

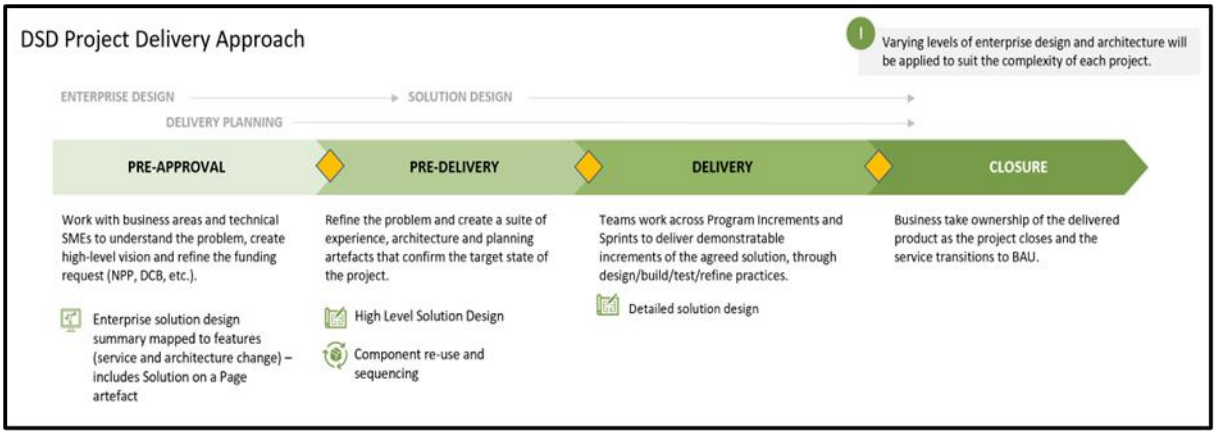


Figure 1 | DSD Project Delivery Approach

2 Related documents

Table 1 lists related documents.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1143 of 1295

Table 1 | Related documents

Title	Version, date	Author	Location
Requirements	September 2024	Project team	Requirements (GovTEAMS)[1]
VDS – Perimeta for Azure - Solution Design	1.2	oobe Pty Ltd	VDS - Perimeta for Azure - Solution Design[2]
VDS DSD Cloud Hosting Options Analysis REBASELINED	August 2024	s 47F(1)	VDS DSD Cloud Hosting Options Analysis REBASELINED[3]

3 Stakeholder consultation

Table 2 lists the stakeholders that were involved in the consultation, review, and approval process of this HLD.

Table 2 | HLD stakeholder consultation

Stakeholder name	Team / area representation
s 47F(1)	Lead Architect for Skills and Training Value Stream, Portfolio Strategy & Assurance (PSA) Team
s 47F(1) ,s 47F(1)	IT Platform Design and Architecture Team
s 47F(1) s 47F(1)	VDS Project Solution Management Team
s 47F(1) s 47F(1) s 47F(1)	DSD Cyber Security Team
s 47F(1) s 47F(1)	TSD Cyber Assurance Team
s 47F(1)	DSD Digital Delivery Capability
s 47F(1) ,s 47F(1)	Data Analytics Platform and Warehousing
s 47F(1) ,s 47F(1)	TSD Architecture Team
s 47F(1)	TSD Business Management Team

[1] You will need to use your GovTEAMS account to access this location.
[2] Internal DEWR document, link will not work outside of the DEWR network.
[3] Internal DEWR document, link not accessible outside of the DEWR network.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1144 of 1295

Stakeholder name	Team / area representation
s 47F(1) ,s 47F(1)	National Centre for Vocational Education Research (NCVER)
s 47F(1)	ACT Government
s 47F(1)	NSW Government
s 47F(1)	QLD Government
s 47F(1)	SA Government
s 47F(1)	TAS Government
s 47F(1)	VIC Government
s 47F(1)	WA Government

4 Introduction

4.1 Background

The current Vocational Education and Training (VET) ecosystem is complex, with approximately four million VET students within Australia and offshore. These students are supported by more than 4000 VET providers, delivering a variety of courses including engineering, hospitality and community services.

Timely and accurate data is critical for deciding the future of vocational education within Australia. The current process used to collect and report data involves significant lag, with delays of up to 21 months from the time of training to when reporting is available. This delays decision-making, resulting in a VET environment that is not in step with the needs of VET providers and students – both current and prospective.

The VET Data Streamlining (VDS) project seeks to uplift the VET sector in Australia through the implementation of a comprehensive, reliable, and near real-time data management repository. This repository will serve as a single source of truth. It will provide VET regulators, governments, VET stakeholders and data consumers with the comprehensive information they require to make timely and proactive decisions regarding the education and training requirements of Australians. The VDS Project will transform current VET data collection, validation, storage and reporting processes by implementing a streamlined data submission pathway to a central repository. The new solution will improve the overall transparency of the VET ecosystem, enhance decision-making, support policy development and improve consumer engagement and trust.

The solution being proposed and developed will be called the Student and Training Activity Reporting System (STARS).

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1145 of 1295

4.2 Dependencies

The dependencies for the STARS solution include:

- Approval of the STARS requirements elaboration outputs, developed through the Requirements Elaboration activity that informs the scope of the STARS Solution end-state.
- STARS high-level integration requirements with external organisations – such as registered training organisations (RTOs) and non-registered training organisations, the National Centre for Vocational Education Research (NCVER) and State Training Authorities (STAs) – need to be clearly defined, to inform the STARS Solution end-state.
- Department of Employment and Workplace Relations (DEWR) requires a detailed understanding of the NCVER Value Stream. This will inform the integration design between STARS and NCVER.
- The data model underpinning the solution needs to be defined to inform the data collection web Application Programming Interface (API) design and data interchange formats for STARS.
- A final decision on the external Identity and Access Management (IDAM) approach is required.

4.3 Assumptions

Table 3 lists assumptions that are relevant to the STARS solution.

Table 3 | STARS solution assumptions

#	Description
A1	The scope of STARS will be as detailed in the user requirements approved by the program. **NOTE: assumption proven.**
A2	The hosting environment for the solution is Microsoft Azure. **NOTE: assumption proven.**
A3	The project will begin build in the existing Dedicated VDS Hosting Environment, managed by oobe Pty Ltd. **NOTE: assumption proven.**
A4	OAuth 2.0, OpenID Connect and Security Assertion Markup Language (SAML) 2.0 are the preferred standards for implementing federated authentication. **NOTE: assumption proven.**

4.4 Constraints

Table 4 lists constraints for the STARS solution.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1146 of 1295

Table 4 | STARS solution constraints

#	Constraint	Potential Impacts	Mitigations
C1	DEWR will develop the STARS solution and handover to NCVER for operation.	Some established DEWR patterns for the in-house solution build may not be appropriate for the STARS solution.	The STARS solution will be built with portability in mind.
C2	VET IS.	The VET sector has a developed standard for data exchange, that the STARS solution will need to account for.	A clear view of the data standard and how it is intended to be collected.

4.5 Risks and mitigations

Table 5 lists the key risks for the STARS solution.

Table 5 | STARS solution risks and mitigations

#	Risk	Mitigation
R1	The project operates within a complex stakeholder environment, with various views and understandings.	Stakeholder engagement that occurs early and continues throughout the life of the project.
R2	The project is expected to begin build in the existing dedicated VDS hosting environment provisioned by oobe Pty Ltd. Whether this is the end-state hosting arrangement for the solution is yet to be determined.	An enduring decision on hosting arrangements for the solution as soon as possible. **NOTE: Risk resolved. Project to be built in the VDS hosting environment provisioned by oobe Pty Ltd.**
R3	VDS partners may not meet project timelines for submission of information or the review and approval of content.	The project may need to proceed with development of capability at risk.

4.6 Scope

4.6.1 In scope

The following is in scope for this HLD:

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1147 of 1295

- Solution scope includes the known/funded end-state for the VDS program, including, but not limited to, an initial MVP release of the STARS solution[4].

4.6.2 Out of scope

The following is out of scope for this HLD:

- Modifications to external systems, such as Student Management Systems (SMS) supported by commercial vendors, and STA IT systems
- Provision of the cloud infrastructure to support the STARS solution will be undertaken by DEWR as a separate activity, to support the STARS Solution.
- Activity relating to the VET in Schools data collection.

4.7 Stakeholders

Table 6 lists STARS solution stakeholders.

Table 6 | STARS solution stakeholders

Role	Key responsibilities	Identified resource
Senior Supplier	The senior supplier represents the interests of all those involved in the design, development, facilitation, procurement, and implementation of the project's products. The Senior Supplier's minimum responsibilities include: • reviews and signs off concept briefs and business cases • assigns Lead Delivery Manager (LDM) and other key roles • shares insights, strategic requirements, and suggestions to shape and inform successful delivery practices and outcomes (where required) • reviews and signs off key changes and artefacts developed by the LDM and delivery team at large (where required) • approves Status Report submissions provided by LDMs	s 47F(1)

[4] User stories stored on GovTEAMS.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1148 of 1295

Role	Key responsibilities	Identified resource
	• represents the project at the monthly DSD Delivery Committee – escalating risks, issues, dependencies and changes (where required) • escalates notable changes and outcomes to Executive (where required).	
Lead Delivery Manager	The LDM's minimum responsibilities include: • reviews key pre-delivery artefacts and requirements • provides support in the development of concept briefs and business cases, including costings and technical intel (where required) • identifies key stakeholders and delivery team resources • develops: Delivery Plan, Detailed Financials, Stakeholder Engagement Plan, Risks, Actions, Issues and Dependencies (RAID) log, Status Report template • submits status reports on time to the Senior Supplier for review and endorsement • submits status reports on time to the DSD IT PMO for inclusion in the DSD Delivery Committee, once endorsed by the Senior Supplier • submits status and escalates notable outcomes to the Senior Responsible Officer (SRO) for inclusion in Project Board agendas • escalates notable outcomes to the Senior Supplier, SRO, and boards and forums where required • consolidates and stores all design artefacts and documentation • oversees and assures that all agreed PI Planning ceremonies are undertaken in alignment to departmental standards	s 47F(1)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1149 of 1295

Role	Key responsibilities	Identified resource
	• responsible for accurately sizing and monitoring effort required to deliver objectives.	
Product Owner	The Product owner's minimum responsibilities include: • helps inform key discovery design artefacts, that is, Features and acceptance criteria, Product Roadmap, Detailed Design and Requirements, Pattern on a page, Proof of Concept • maintains Features, Stories and tasks (DevOps) • helps the Team with defining stories, sequencing activities, as well as drafting the Team's PI Objectives • responsible for conveying visions and goals from upper management to the team • has content authority to make decisions at the User Story level • schedules working sessions to refine the Backlog • responsible for splitting broad user stories, adding or reviewing estimates for tasks and ordering backlog tasks to deliver high-priority outcomes first • remains engaged with the development team throughout the sprint • answers questions on how things are supposed to function and look, as well as making any trade-offs between ICT and business when necessary • accepts user stories as 'done' by ensuring stories and tasks have been QA tested and meet agreed expectations • facilitates feedback from stakeholders across delivery (where required).	s 47F(1)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1150 of 1295

Role	Key responsibilities	Identified resource
Scrum Master	The Scrum Master's minimum responsibilities include: • facilitates preparation for PI Planning • helps develop Plan-on-a-Page, PI Objectives, Team Boards, Dependency Maps • helps the team estimate their capacity for Iterations and finalises Team PI Objectives, dependencies, and ambiguities during Team Breakout sessions • helps define techniques to improve product management and backlog refinement • responsible for preparing the sprint backlog prior to each meeting • helps the delivery team understand the need for clear and concise product backlog items and management • schedules and facilitates sprint ceremonies and meetings • responsible for preparing the sprint backlog prior to each meeting • finds techniques to maximise productivity • communicates any blocking issues requiring input from the Product Owner.	Multiple roles within the project delivery team
Delivery Team	The Delivery Team's minimum responsibilities include: • provides support for the development of concept briefs and business cases, including costings and technical intel (where required) • participates in PI Planning breakout sessions by creating and refining user stories and acceptance criteria as well as adjusting the working plan with the Product Owner	STARS delivery team

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1151 of 1295

Role	Key responsibilities	Identified resource
	• helps identify risks and dependencies, and supports the team in drafting and finalising PI Objectives • supports the Product Owner to refine the backlog, by providing insights and estimations regarding stories, work items, risks, issues and estimations co-creates the working plan for the Sprint and sprint backlog, including tasks and acceptance criteria • provides quality assurance by adhering to a definition of 'Done' • adapts and iterates the plan each day towards the Sprint Goal • holds team members accountable to delivering agreed outcomes.	
Enterprise Architecture (Technology and Services Division [TSD] and DSD)	Enterprise Architecture (TSD and DSD) responsibilities include: • define and maintain the enterprise architecture (EA) for DEWR, aligning ICT strategy and planning with the DSD Digital Business Strategy • develop and maintain the EA governance structure, and ensure that IT investments align with strategic objectives • work closely with stakeholders across DEWR to ensure that the IT architecture supports DEWR's overall objectives • provide guidance and direction in the solution design and implementation phases of projects.	TSD Architecture Team DSD IT Platform and Architecture Team
National Centre for Vocational Education Research (NCVER)	Project partners.	Various

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1152 of 1295

Role	Key responsibilities	Identified resource
State/Territory Training Authorities and relevant statutory bodies	Project partners.	Various
VET sector regulators	Project partners.	Various

5 Business context

5.1 Objectives

STARS objectives include:

- STARS will be an extensible platform which will lead to the replacement of applicable legacy systems.
- The design of STARS will facilitate the transfer of the platform, administration, management, support and operation of the system to NCVER.
- Applicable data and processing in relation to the collections and products will be moved to STARS.
- VET Provider data is in scope for delivery in STARS. The design will be considerate of the need to include other collections based on timings to be defined in the roadmap.
- STARS will hold and manage data through to final TVA URF maturity.

5.2 Strategic alignment

5.2.1 National Skills Reform alignment

The overarching program objectives, in alignment with the National Skills Reform, include:

- Improved Service Delivery. Data analytics to drive evidence-based decision-making and continuous government and business operational refinement, enabling effective and efficient service delivery.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1153 of 1295

- Job Creation and Economic Growth. Quality data enhances opportunities for innovative data use and analytics to drive new products, services, business models and industries, spurring job growth and economic activity.
- Agility and Resilience. Enabling government and businesses to quickly identify, understand and respond to emerging issues of national importance (such as COVID-19) though quality data and strong data use and sharing, governance and management.
- Better VET Sector and Student Outcomes. Enhancing opportunities to reduce VET sector regulatory and administrative burden, through improved data quality, data use, governance and frameworks (such as reusing student information to streamline the business and reporting process).

5.2.2 DEWR Digital Business Strategy alignment

The DEWR Digital Business Strategy outlines several objectives targeted at individuals, providers, businesses, industry groups/regulators, and government staff. From these cohorts, the nature of the STARS solution relates directly to providers (training organisations), industry groups/regulators and government staff.

The list below outlines how the STARS solution aligns with the goals of the Digital Business Strategy for these relevant cohorts.

- Providers/industry groups/regulators:
    - Direct integration – 'I can connect my software directly with DEWR services'. STARS will provide improved means by which training organisations will submit data to government systems, modernising submission of student and training activity data via web service and web portal pathways.
    - Real time data – 'I can access data and insights in real time'. An objective of STARS is to improve timeliness of data submission from training organisations via web service and web portal pathways, a 'progressive submission' model, where real time data submission is enabled and encouraged.
    - Automation – 'I can get automatic and instant data validation'. The STARS solution includes a significant amount of automated/instant data validation, implementing the VET IS data standard for data submitted by training organisations.
    - Efficient – 'I only have to provide the government my business data once'. STARS upholds the scenario where training organisations will provide their data once to a government entity, be it state training authorities (in the case of 'indirect states'), that then pass data onto STARS, or direct submission to STARS (in the case of 'direct states').
- Government (and partner) staff:

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1154 of 1295

- Frictionless – 'I can do my job in user friendly systems, without having to do error prone, manual workarounds'. STARS will provide user friendly and accessible interfaces for government and partner staff to interact with the solution. This will be validated by way of a focus on service design, user consultation and testing.
- Data driven decisions – 'I can easily access and use data to make informed decisions and provide advice'. STARS will provide improved and more timely access to data, better supporting informed decisions and advice.
- Linked data across government – 'I have improved visibility of broader trends through linked data across government and industry'. STARS is envisioned to be the central collection point for student and training activity data, supporting broader trend analysis across the VET sector.

5.3 Users

The following users have been identified for STARS:

- Registered training organisations, using VET IS compliant student management systems (SMS), or manual reporting processes (e.g. in the case of small training organisations that may not have compliant student management software)
- Non-registered training organisations that may be contractually obligated to transfer data to STARS
- DEWR
- NCVER, the National VET Data Custodian
- STAs
- Relevant statutory bodies
- National VET sector regulators, including the Australian Skills Quality Authority (ASQA), the Victorian Registration and Qualifications Authority (VRQA) and the Training Accreditation Council (TAC).

5.4 Requirements

Requirements can be found on the VDS GovTEAMS site at VET Data Streamlining Requirements.

5.5 Business process

Service design blueprint snapshots can be found on the VDS GovTEAMS site.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1155 of 1295

6 Information security

6.1 Information assets

Table 7 lists STARS solution information assets.

Table 7 | STARS solution information assets

Name	Classification	Description
VET provider collection	OFFICIAL: Sensitive	The main body of data collected by the system.
STA business rules	OFFICIAL	Business validation rules based on STA business requirements.
Reference data from various sources	OFFICIAL	Reference data (pre-existing lists of values) relevant to the solution.
Classification data (VET IS and AVETMISS 8)	OFFICIAL	Information on the data standards that assist to guide the solution's data model. The old outgoing AVETMISS standard and the new incoming VET IS standard.
Emails or other notifications generated by the solution	OFFICIAL	Processes/Workflows supported by the solution will likely require sending of notifications at various times.
Reports and data visualisations generated by the solution	OFFICIAL: Sensitive	Reports and data visualisations based on data collected by the solution.
User account data	OFFICIAL: Sensitive	Data relating to identity and access control (IAC) for the solution. For example, usernames, passwords, and details used to support multi-factor authentication (MFA), such as email address or phone numbers.
Role-based access control (RBAC) data	OFFICIAL	Role/Group names and the mapping of associated user accounts.
Machine credentials	OFFICIAL: Sensitive	Service accounts or managed identities used for service/machine authentication purposes.
Information stored on any web portal(s) that make up part of the solution	OFFICIAL	Instructional, reference and other information that may be present on any web portal(s).

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1156 of 1295

Name	Classification	Description
Digital certificates, API client keys and other material used for authentication purposes	OFFICIAL: Sensitive	Data stored to enable encryption features and other security functionality within the solution.
Historical data replicated from NCVER	OFFICIAL: Sensitive	Historical data replicated from NCVER required to be stored in STARS.

6.2 System classification

The overall security classification for the STARS solution is **OFFICIAL: Sensitive**.

6.3 STARS business capability map

Figure 2 shows the STARS business capability map.

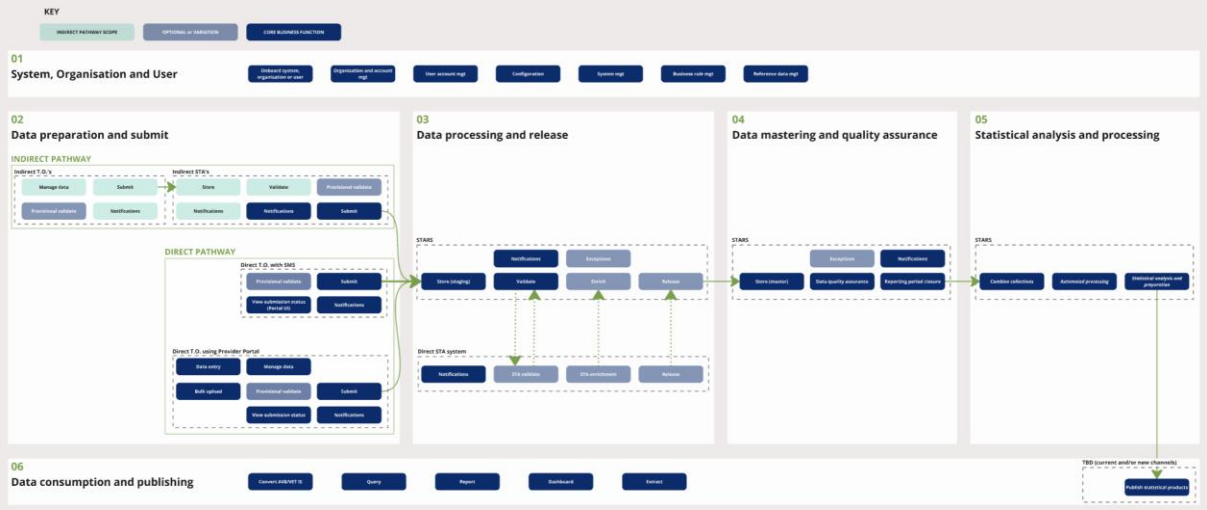


Figure 2 | STARS business capability map

7 Solution context

7.1 Overview

This section includes an overview of:

- the data sources, the STARS solution, and data consumers
- integration between training organisations, STAs, STARS, NCVER, and other consumers
- components involved in data collection, processing, enrichment, analytics, and reporting.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1157 of 1295

The solution overview highlights the sources and consumers of the STARS system. The primary information sources are training organisations, who are responsible for reporting training activity. State training authorities (STAs) act as both data sources and data consumers:

- the former in the case of 'indirect states' (WA, VIC, TAS, QLD) where training organisations submit training activity to the STA, and the STA passes that data on to STARS
- the latter, in the case of 'direct states' where training organisations submit training activity data directly to STARS.

VET regulatory bodies are data consumers and data sources. The latter in the case of a training provider closure and when student qualifications are cancelled.

The STARS solution is an evolving architecture based on requirements, user stories, and agreements from workshops with stakeholders. This architecture will be refined further as the project progresses. Figure 3 shows the STARS solution overview.

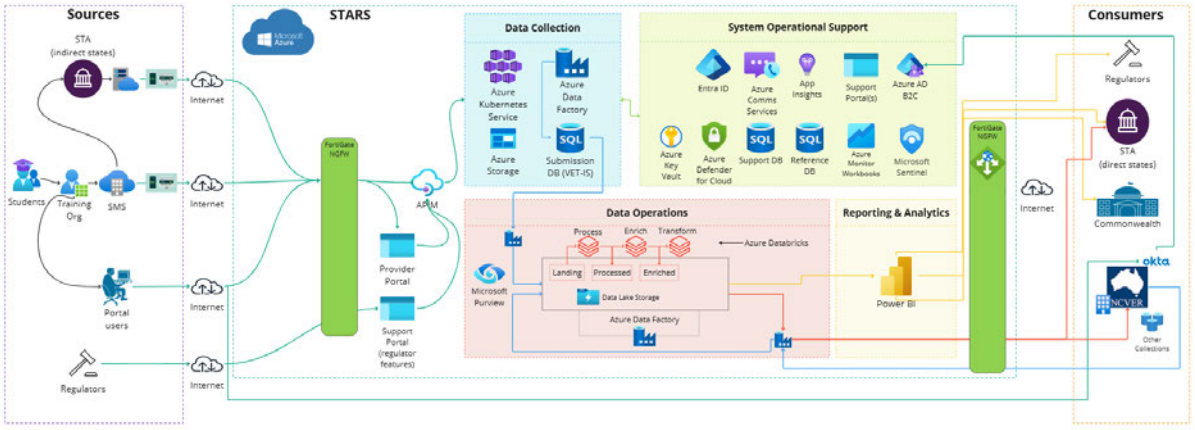


Figure 3 | STARS solution overview

7.2 STARS architecture and key components

The STARS solution leverages an architecture comprised of the key components listed in Table 8. These components are intended to ensure the system's efficiency, security, and scalability.

Table 8 | STARS solution – key components

Key component	Description
FortiGate NGFW	FortiGate Next Generation Firewall. The central aggregation point for external traffic, provides load balancing and enhanced security features.
Provider Portal	Web-based user interface for external users.
Support Portal	Web-based user interfaces for internal and partner users (administration/operational functions). This component will include functionality for regulators to manage submitting data for example, in the case of training organisation closure.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1158 of 1295

Key component	Description
APIM	Azure API Management. Handles web API calls, secures API access, and provides detailed analytics on API usage.
Azure Kubernetes Service	Hosts web API applications, supporting scalability, highly secured containers and load testing.
Submission DB (VET IS), Support DB, Reference DB	Azure SQL Database provides secure, high-performance storage for all STARS data.
Entra ID	Implements advanced Identity and Access Management (IAM) for the solution.
Azure Storage	Provides scalable, secure storage solutions for various data types within the STARS system.
Data Lake Storage	Azure Data Lake offers a hierarchical namespace for organising big data and supporting analytics.
Azure Data Factory	Orchestrates extract, transform, load (ETL) processes for data integration and transformation within STARS.
Microsoft Purview	Provides unified data governance features to help manage data across the STARS solution.
Azure Monitor Workbooks	Customisable dashboards for monitoring and analysing system performance.
Microsoft Sentinel	A security information and event management (SIEM) and security orchestration, automation, and response (SOAR) solution.
App Insights	Azure App Insights provides real-time monitoring and diagnostics for application performance and health.
Azure Databricks	Facilitates advanced data processing and analytics for big data within the STARS system.
Azure Comms Services	Azure Communication Services provides email notifications as required by the solution.
Azure Defender for Cloud	Delivers advanced threat protection and security monitoring for STARS solution components.
Azure Key Vault	A secure secrets store, providing management for secrets, keys, and digital certificates.
PowerBI	PowerBI provides advanced analytics, real-time data processing, and visualisation tools.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1159 of 1295

Key component	Description
Azure Active Directory Business to Consumer (AAD B2C)	Supports federated authentication for external users of the solution. Provides a means by which external identity providers can be integrated into the solution.
Okta	Provides authentication for training organisation users already on-boarded to the NCVER DataHub.

This architecture offers several key advantages:

- Streamlined Data Collection. The portals and APIs enable training organisations and STAs to integrate with the solution, facilitating efficient and streamlined data collection.
- Infrastructure as Code (IaC). STARS is built using IaC principles where possible, allowing for efficient deployment and management. This approach enhances scalability, simplifies maintenance processes, and streamlines the deployment, configuration, and ongoing upkeep of the STARS platform.
- Scalability. The solution is designed to scale easily, accommodating evolving business needs and user demands.
- Standalone. STARS operates independently of any existing DEWR infrastructure, minimising potential integration complexities. All components are newly developed and independent of existing DEWR systems, ensuring a smooth and efficient transition to NCVER upon go-live. The distinct separation between DEWR and NCVER environments ensures a viable ownership transition and minimises potential future operational challenges.

7.3 Key design decisions

Table 9 lists key design decisions for STARS.

Table 9 | STARS key design decisions

Decision	Impact
Extensibility, flexibility and practical implementation	Particularly as it relates to the continuing evolution of the VET IS, the solution will keep extensibility and flexibility as key objectives, while balancing these objectives with ensuring practical implementation.
Stateless Design	For optimal performance, STARS should utilise stateless services by default. This simplifies scaling and management.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1160 of 1295

Decision	Impact
	Session affinity may be considered for specific needs but requires justification and approval.
Offload resource intensive tasks	To optimise system performance and responsiveness for real-time user interactions, STARS should leverage a tiered architecture. Resource intensive data processing tasks are strategically offloaded to a dedicated batch processing layer. This approach utilises separate resources, ensuring efficient handling of complex computations without impacting the performance of client-facing APIs.
Solution should be self-healing across all layers	STARS should leverage a robust self-healing architecture to ensure high availability. External service integrations should employ a retry mechanism to gracefully handle transient failures, such as network connectivity issues or overloaded external services. In the event of persistent failures, a circuit breaker pattern should be automatically invoked to prevent STARS from continuously attempting to connect to a failing service, thereby mitigating potential cascading effects. Additionally, automated failover can be implemented to seamlessly route traffic to a healthy instance in case of an underlying resource outage.
Solution should be redundant and highly available across two regions (region-pair)	STARS should ensure continuous operation through regional failover. The multi-region architecture leverages automated URL rewrites to seamlessly redirect traffic to the secondary region (Australia Southeast) in case of a primary region (Australia East) outage.
Microsoft Azure Cloud platform	STARS will leverage a cloud-based deployment model, aligning with DEWR's enterprise strategy for infrastructure management. This approach ensures adherence to established security policies and standards, while simultaneously delivering a seamless user experience for diverse user groups.
Dedicated VDS hosting environment	STARS will be hosted in a standalone Azure environment, ensuring a seamless transition to management by NCVER.
Use of accepted industry-standard authentication and authorisation approaches	STARS will make use of industry-standard authentication approaches, such as OAuth 2.0 (client certificate-based client credentials flow for confidential clients), OpenID Connect and SAML 2.0.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1161 of 1295

Decision	Impact
Use of Representational State Transfer (REST) web APIs	STARS fosters integration with external systems by leveraging REST web service APIs for data exchange. This approach enables efficient data interchange and facilitates the development of innovative applications and services that complement the core functionalities of STARS.
Use of Azure API Management as external integration platform between training organisation SMS systems, STA systems (indirect jurisdictions) and STARS	Azure API Management simplifies web service integration, offering centralised control, security, scalability, monitoring and faster development.
Use of Azure Data Factory	This cloud-based data integration service will be used to extract, transform, and load data from STARS databases into the STARS data warehouse. It will also be used to support data integration with some partner organisations.
Use of Microsoft Purview	Provides data governance and lineage for the solution. In particular, Purview is expected to assist with management of sensitive personal data and data classification.
Use of Azure Databricks	Orchestration steps making up ETL processes may be implemented via Azure Databricks. The Databricks Unity Catalog can assist with managing Databricks workspaces but should not be considered a replacement for overarching identity and access management (provided by Entra ID) or data governance (provided by Microsoft Purview).
Use of Power BI	Power BI will provide data visualisation capabilities across staging/datastore layers and a self-service model for data visualisations. It will also publish reports for consumption via web browsers and mobile devices.
Use of Azure SQL Database	Provides a fully managed relational database service for high-performance transactional data storage, querying and management within the STARS solution.
Use of Azure Kubernetes Service	Provides a robust and scalable platform for hosting RESTful web APIs, enabling highly secure containers, load testing, integration with other STARS components, as well as external systems.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1162 of 1295

7.4 Architecture patterns

7.4.1 Adoption of cloud architecture

Adopting cloud architecture involves migrating to cloud platforms like Microsoft Azure to leverage their advanced infrastructure and services. This process includes reconfiguring existing applications or developing new ones to utilise cloud resources effectively. By leveraging Azure's capabilities, organisations can enhance scalability, performance, and security, ensuring a more resilient and agile IT environment.

7.4.2 Modular component deployment

The system's design emphasises modularity, allowing for the independent development and deployment of components. This approach ensures enhanced maintainability and flexibility. Solution components will be divided into smaller, independent services that can be developed, deployed, and scaled independently. This pattern supports modular design, allowing for flexible development and easier maintenance.

7.4.3 Reuse existing DSD architecture patterns

To optimise implementation and maintain consistency, STARS will reuse existing DSD architecture patterns by creating separate instances rather than directly integrating with platforms. This separation will facilitate the smooth handover of the application to NCVER.

This approach involves the following architecture patterns:

- API Gateway Pattern. Use Azure API Management (APIM) for managing web APIs – including the gateway functions and developer portal integration, as required – to streamline API access and consumption.
- IDAM Pattern. Implement a dedicated trust platform for user authentication and access control, ensuring a secure and consistent approach to managing user identities and permissions.

7.4.4 Microsoft DataOps architecture

STARS will harness the power of modern Microsoft DataOps architecture to deliver capabilities like those of NCVER's current value stream. The architecture will integrate several key components, orchestrated by Azure Data Factory, to provide data flow and processing. Components of the Architecture include:

- Azure Data Factory (ADF). Central to the architecture, ADF will manage the orchestration of data workflows, pulling data from the Submission DB and NCVER collections into a centralised landing storage. Azure Databricks will handle advanced data processing and analytics.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1163 of 1295

- Data Lake Storage. A scalable and secure storage solution that will serve as the foundational layer for storing raw data. This storage solution ensures flexibility and robustness in managing large volumes of data.

This architecture will process and enrich data throughout its lifecycle, aiming to produce a dataset of high maturity. Once processed, the data will be stored in a data warehouse and delivered to different consumers through Power BI, as well as leveraging site-to-site connections to NCVER. This architecture is designed to optimise data management and analytical capabilities, ensuring efficient and reliable data delivery.

7.5 Architecture principles alignment

Table 10 lists the general architecture principles used in every solution undertaken by DEWR, and alignment to these principles in the context of the STARS solution.

Table 10 | STARS architecture principles alignment

Architectural principles	Description	Alignment
Customer focused	We let customer feedback and insights drive delivery, ensuring a customer focused outcome. • In the context of the STARS solution, 'customers' would be defined as those user groups listed in the Providers/industry groups/regulators: ○ Direct integration – 'I can connect my software directly with DEWR services'. STARS will provide improved means by which training organisations will submit data to government systems, modernising submission of student and training activity data	Simplified user onboarding process. Intuitive portal interfaces. Straightforward and well documented machine-to-machine (M2M) integration. Streamlined business process, allowing progressive submission, reducing burden on training organisations.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1164 of 1295

Architectural principles	Description	Alignment
	via web service and web portal pathways. ○ Real time data – 'I can access data and insights in real time'. An objective of STARS is to improve timeliness of data submission from training organisations via web service and web portal pathways, a 'progressive submission' model, where real time data submission is enabled and encouraged. ○ Automation – 'I can get automatic and instant data validation'. The STARS solution includes a significant amount of automated/instant data validation, implementing the VET IS data standard for data submitted by training organisations. ○ Efficient – 'I only have to provide the government my business data once'. STARS upholds the scenario where training organisations will provide their data once to a	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1165 of 1295

Architectural principles	Description	Alignment
	government entity, be it state training authorities (in the case of 'indirect states'), that then pass data onto STARS, or direct submission to STARS (in the case of 'direct states'). • Government (and partner) staff: ○ Frictionless – 'I can do my job in user friendly systems, without having to do error prone, manual workarounds'. STARS will provide user friendly and accessible interfaces for government and partner staff to interact with the solution. This will be validated by way of a focus on service design, user consultation and testing. ○ Data driven decisions – 'I can easily access and use data to make informed decisions and provide advice'. STARS will provide improved and more timely access to data, better supporting	

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1166 of 1295

Architectural principles	Description	Alignment
	informed decisions and advice. ○ Linked data across government – 'I have improved visibility of broader trends through linked data across government and industry'. STARS is envisioned to be the central collection point for student and training activity data, supporting broader trend analysis across the VET sector. Users section.	
Speed to value	We deliver value more quickly, by reducing size and risk of changes and releasing more frequently.	Improving the quality and timeliness of training activity reporting. This will provide a high-speed central 'clearing house' for VET data and to provide an extensible, future-ready platform for future reporting and operational requirements of stakeholders involved with VET sector data.
Data-driven	We treat data as an asset to deliver modern and personalised experiences.	The integration of Power BI and the planned data warehouse with advanced analytics tools demonstrates the commitment to data-driven insights.
Aligned, governed and modern	Leveraging modern mindsets to assure alignment to DEWR delivery strategies, through practical governance.	The adoption of Azure cloud services, a dedicated trust platform, and open standards like RESTful APIs showcases

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1167 of 1295

Architectural principles	Description	Alignment
		the emphasis on modernisation and innovation.
Platforms and reuse	We promote platform, pattern-based architecture and Whole-of-government (WoG) services to drive reuse and consistency.	The utilisation of a dedicated trust platform and exiting cloud service patterns highlights the focus on reuse and efficiency.
Open standards	We leverage open and trusted standards across delivery.	Use of OAuth, OpenID Connect and SAML open standards showcases the intention to integrate with the VET sector in a transparent manner, supporting openness and interoperability.
Accessible	We deliver accessible services regardless of location, device, or capability.	The core services of STARS are exposed as standard web APIs to be consumed by any software capable of integration with REST web services. In addition to web APIs, STARS will provide accessibility tested web portals, that meet WCAG 2.2 requirements.
Secure by design	We deliver services that have robust security embedded from inception.	The dedicated VDS hosting platform is secure by design; it will control and safeguard the STARS solution via role-based access control, multi-factor authentication and other complimenting approaches/strategies, taking guidance from the Australian Government's Information Security Manual. Native security tools and in the cloud infrastructure – such as Azure Defender for Cloud Apps and FortiGate appliances – also assist to provide security assurance.
Leverage cloud tools	We achieve speed, innovation, and business agility by choosing cloud tools.	STARS is built on the Microsoft Azure cloud platform, leveraging the appropriate features available within the context of the platform.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1168 of 1295

Architectural principles	Description	Alignment
Reliable engineering	We deliver digital services that are available, scalable, and performant.	STARS is built on a reliable, dedicated VDS hosting platform that will support appropriate scalability and performance.

7.6 Components

7.6.1 Application

This section provides a high-level logical grouping of the STARS application layer components, categorised by their function:

1. Data Collection
2. System Operational Support
3. Data Operations
4. Reporting and Analytics.

7.6.1.1 Data collection

The STARS system is designed to facilitate data integration and collection from training organisations and STAs. To achieve this, the system leverages REST web APIs, enabling efficient data submission and retrieval. Additionally, a dedicated portal is supplied for training organisations to help streamline their reporting processes. Figure 4 shows the STARS data collection.

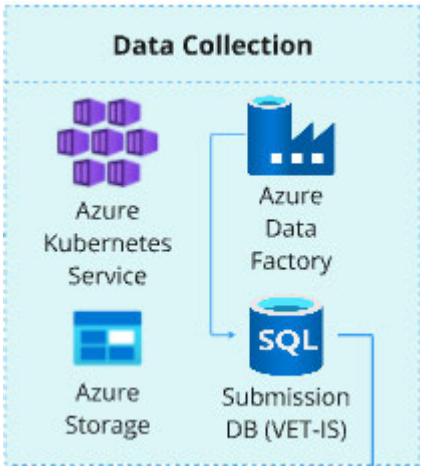


Figure 4 | STARS Data Collection

Each web API will incorporate robust data validation mechanisms to ensure that submitted data meets predefined standards and formats. This helps to maintain data integrity and accuracy. All APIs are secured using industry-standard authentication and authorisation protocols. This is expected to be a

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1169 of 1295

certificate-based OAuth 2.0 client credentials flow, to ensure only authorised entities can access and submit data. Table 11 lists STARS APIs.

In cases where, for example, asynchronous processing is required, STARS APIs will support the ability for client systems to determine the outcome of such processing, allowing for process automation on the client side.

Table 11 | STARS components

Component	Purpose	Functionality	Azure services/resources
Provider API	Integrates with training organisation SMS to validate, submit, and read provider data.	Enables data validation, submission, and retrieval for training organisations.	Application Firewall, APIM, Azure Kubernetes Service, Submission DB (VET IS)
Direct States API	Used by STAs in states where training organisations submit data directly to STARS.	Interacts with 'direct state' STAs to support STA specific validations, and access to VET data.	Application Firewall, APIM, Azure Kubernetes Service, Submission DB (VET IS)
Indirect States API	Used by STAs in states where training organisations submit to STAs, and STAs submit data to STARS.	Supports submission of, and access to, VET data.	Application Firewall, APIM, Azure Kubernetes Service, Submission DB (VET IS)
Provider Portal	Offers a browser-based user interface for training organisations to interact with STARS.	Allows manual data entry, report generation, and monitoring of submission statuses.	Application Firewall, APIM, Azure Kubernetes Service, Submission DB (VET IS)
Quality Check Job	Triggered by a specific volume of training data, this process analyses data in aggregate to identify potential inconsistencies across providers and states.	Perform aggregate checks, quality checks, apply business validations across all data submitted by training organisations and STAs.	Azure Data Factory

7.6.1.2 System Operational Support

To enable the STARS solution, several components are required. These components serve as the backbone for operations. Figure 5 shows STARS System Operational Support.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1170 of 1295

Figure 5 | STARS System Operational Support

Table 12 provides further detail on STARS System Operational Support.

Table 12 | STARS Operational Support

Component	Usage	Azure services/resources
Support Portal(s)	The STARS backend support portal is an essential component designed to streamline and manage various administrative and operational tasks associated with the system. Portal users would include STARS administrators, i.e. DEWR and/or NCVER staff. This portal serves various functions, such as: • managing Classification and Reference Data • managing Submissions • Managing validation/data quality rules • handling exceptions.	APIM, Azure Kubernetes Service, Azure SQL Database
Entra ID	Microsoft Entra ID is used to support user authentication and authorisation. This includes training organisation users, STA users, SMS clients, SMS vendors, backend internal operation support staff and NCVER Staff.	Entra ID
Azure Comms Service	Email notifications are expected to be delivered on key events such as successful data submission, quality check errors, scheduled reports and user account activities.	Azure Communication Services
Support DB	Contains supplementary data required for operational support, data validation/data quality rules and system	Azure SQL Database

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1171 of 1295

Component	Usage	Azure services/resources
	configurations, including user and organisation management.	
Reference DB	Contains reference data used for validation and enrichment, such as lookup tables and code mappings.	Azure SQL Database
Azure Monitor Workbooks	Azure Monitor Workbooks, for logging of events within the STARS environment.	Azure Monitor Workbooks
Microsoft Sentinel	Provides SIEM (Security Information and Event Management) and Security Orchestration, Automation and Response (SOAR) for the solution.	Microsoft Sentinel
App Insights	For operational monitoring of solution components. It will also provide real-time insights into the performance of the STARS application components.	Azure Application Insights
Azure Defender for Cloud Apps	Provides cloud security posture management (CSPM) for Azure workloads and resources.	Azure Defender for Cloud Apps
Azure AD B2C	Supports federated authentication/authorisation for the solution.	Azure AD B2C
Azure Key Vault	Provides secure storage for secrets, keys, and digital certificates.	Azure Key Vault

7.6.1.3 Data Operations

The Azure Data Operations architecture for the STARS system is designed to align with the NCVER value stream process, ensuring a seamless flow from data landing to processing and enrichment. Data Operations are shown at Figure 6.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1172 of 1295

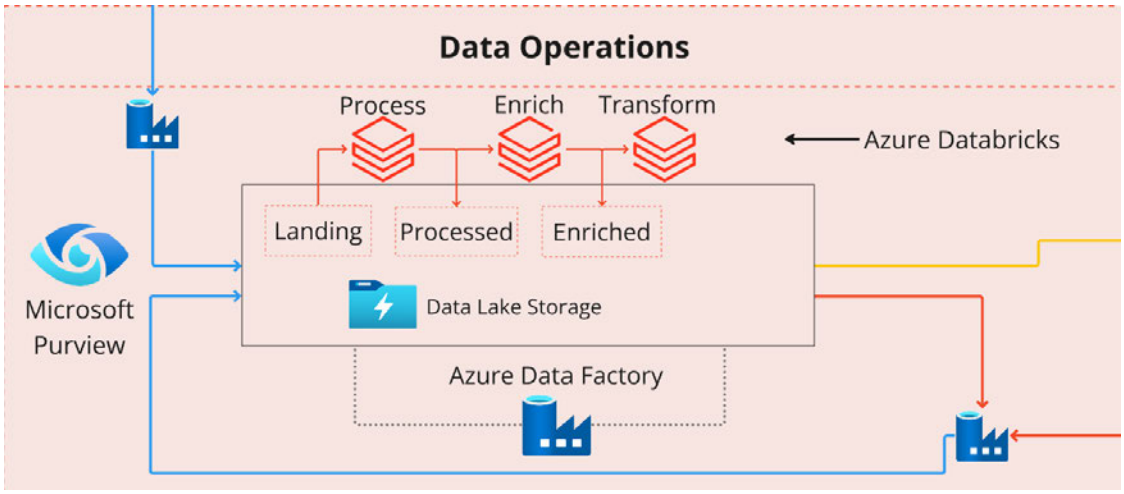


Figure 6 | Data Operations

The Data Operations architecture is a framework designed to manage the entire lifecycle of data. It begins with storing raw data in a centralised repository, i.e. Azure Data Lake Storage. Azure Data Factory then orchestrates the movement and transformation of this data. Advanced processing and enrichment can be performed using Azure Databricks, which can support complex transformations and analytics. Finally, the processed data is stored in a data warehouse, optimised for querying and reporting. Microsoft Purview is included to provide unified data governance functionality. This architecture ensures efficient data handling and high-quality insights for decision-making.

The solution components utilised by STARS at each processing stage are shown in Table 13.

Table 13 | Solution components utilised by STARS

Process stage	Description	Azure services/resources
Raw Data Storage	Raw data is stored, providing a centralised repository for unprocessed data.	Azure Data Lake Storage
Data Processing	Orchestrate data movement and transformation, preparing it for further processing.	Azure Data Factory, Azure Data Bricks
Data Enrichment	Application of advanced transformations and analytics to enhance the data.	Azure Databricks
Data Storage	Store processed and enriched data, optimised for querying and reporting.	Azure Data Lake Storage
Data Governance	Appropriate treatment of sensitive data and application of data classifications.	Microsoft Purview

7.6.1.4 Reporting and analytics

Reporting and Analytics provides data visualisations and reports for consumption by stakeholders, making use of the outputs from the Data Operations solution component group. Figure 7 shows Reporting and Analytics.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1173 of 1295

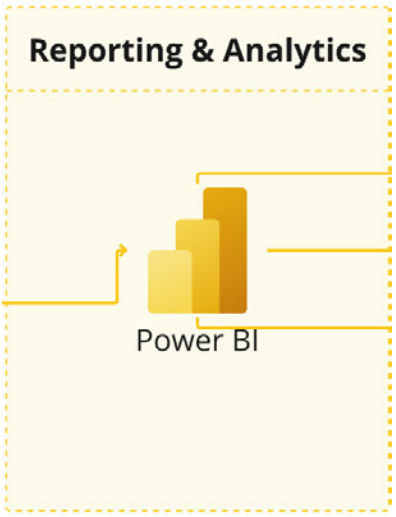


Figure 7 | Reporting and Analytics

Table 14 shows the Power BI component and usage.

Table 14 | Power BI component

Component	Usage	Azure services/resources
Power BI	Provides data visualisations and reports. The primary audience is expected to be training organisations, state training authorities, state regulators and any other relevant government partners. NCVER has existing data reporting and visualisation capabilities that will still receive STARS data.	Power BI

7.6.2 Data

The STARS solution employs a data management framework utilising various Azure data stores, to ensure efficient data operations and comprehensive reporting. Table 15 provides an overview of the data stores crucial to the STARS solution, highlighting how each store is used and managed to achieve optimal performance and reliability.

Table 15 | Data stores crucial to the STARS solution

Data store	Description	Azure service	Usage in application domain
Landing Database	Stores VET IS data submitted by training organisations and indirect jurisdictions for initial processing, validation, and quality checks.		Data Collection

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1174 of 1295

Data store	Description	Azure service	Usage in application domain
Submission Database	Stores VET IS data submitted by training organisations and indirect jurisdictions for processing, validation, and quality checks beyond provisional validation.	Azure SQL Database	Data Collection
Reference Database	Maintains reference data used for validation and enrichment of data, such as lookup tables and code mappings.	Azure SQL Database	Support Operations
Support Database	Contains supplementary data required for operational support and system configurations.	Azure SQL Database	Support Operations
Storage Account	General-purpose storage account for any uploaded files. Also stores any imported data which must be processed before being stored in other data stores. Defender for Storage will be used to enable on-demand malware scanning of uploaded files.	Azure Storage Account	Data Collection
Data Lake Storage	Centralised repository for raw data from various sources such as submission database and other imported collections from NCVER. Store data during the transformation phases – landing, processed and enriched.	Azure Data Lake Storage	Data Operations

7.6.2.1 Reporting, analytics and data consumption

Reporting, analytics, and data consumption are essential features of STARS. These features enable training organisations, STAs, the Commonwealth, Regulators and NCVER to extract meaningful insights and to drive informed decision-making. The STARS system utilises Azure services for comprehensive reporting, analytics, and data consumption, scoped appropriately to different audiences (e.g. some audiences will not have access to provisionally validated data). STARS will support detailed reporting across various areas such as training submissions and data validation outcomes.

STAs, Commonwealth agencies, and regulators can use Power BI as the main tool for accessing interactive dashboards and performing data queries, and such partner organisations can benefit from the system's integration capabilities. This set-up ensures that all stakeholders can efficiently manage and analyse data. Table 16 shows the reporting, analytics and data consumption features of STARS.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1175 of 1295

Industry regulators have a requirement to have access to data as early as possible and in as much granularity as possible, to support data tracking and analysis processes. STARS solution will give regulators access to granular data as early as is feasible, accounting for any technical constraints, and in accordance with the enduring governance and access policy.

Table 16 | STARS reporting, analytics and data consumption

Functionality	Description	Azure services/resources
Data Consumption by partner organisations	The validated submissions received by STARS in the VET IS standard can be translated to the AVETMISS 8 standard and sent to partner organisations.	Azure Data Factory, SQL Server Database,
Reporting	Generate reports on submissions from training organisations, including: • summaries and detailed analytics • data validation • quality assurance reports • reporting exceptions • insights. Allow data export, querying, and dashboard creation for end users to access and analyse data.	Power BI

7.6.2.2 Data flows

Data validation involves several different validation types:

- National validation rules – confirming that the data meets the national standard, is complete and in the right format – these rules will reside within the STARS Solution
- State/Territory specific validation rules – confirming that data meets the specific requirements of a given state or territory training authority – in the case of each state or territory interacting with STARS, state/territory rules may reside within the STARS Solution, state/territory training authority systems, or some combination of the two
- Aggregate data validation – also referred to as data quality validation, this involves validation across a volume of submitted data, for example to check that all students do not have the same address or gender, or that the percentage of indigenous students is not above an unrealistic threshold

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1176 of 1295

Also relevant is the concept of provisional validation, where a training organisation can send their data purely for the purposes of validation, before final submission of their data. This provisional validation allows training organisations to gather confidence that their data is fit for final submission.

There can be rare cases where exceptions are made to specific validation rules for specific submissions. The STARS Solution will allow for overriding validation rules to support such cases.

The STARS Solution is expected to support the below data flow patterns:

- Pattern 1 – direct submission – relevant for 'direct' states/territories (NSW, SA, NT, ACT), where training organisations submit data directly to STARS. There are two sub-patterns within the direct submission pathway:
    - Pattern 1a – where the state/territory training authority maintains an in-house set of data validation rules and the STARS Solution communicates with state/territory training authority systems to run these rules against data stored in STARS, then receives a response
    - Pattern 1b – where the state/territory training authority has provided their data validation rules and the STARS Solution can run these internally, alleviating the need to communicate with the state/territory training authority for data validation purposes
- Pattern 2 – indirect submission – relevant for 'indirect' states (WA, VIC, TAS, QLD), where training organisations will submit some or all their data to state training authorities and these data will subsequently be passed onto STARS after validation by state/territory training authorities

7.6.2.2.1 Data Flow Pattern 1a - direct submission pathway

The state/territory training authority maintains an in-house set of data validation rules. The STARS Solution communicates with state/territory training authority systems to run these rules against data stored in STARS, then receives a response.

Figure 8 shows the Pattern 1a direct submission pathway data flow.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1177 of 1295

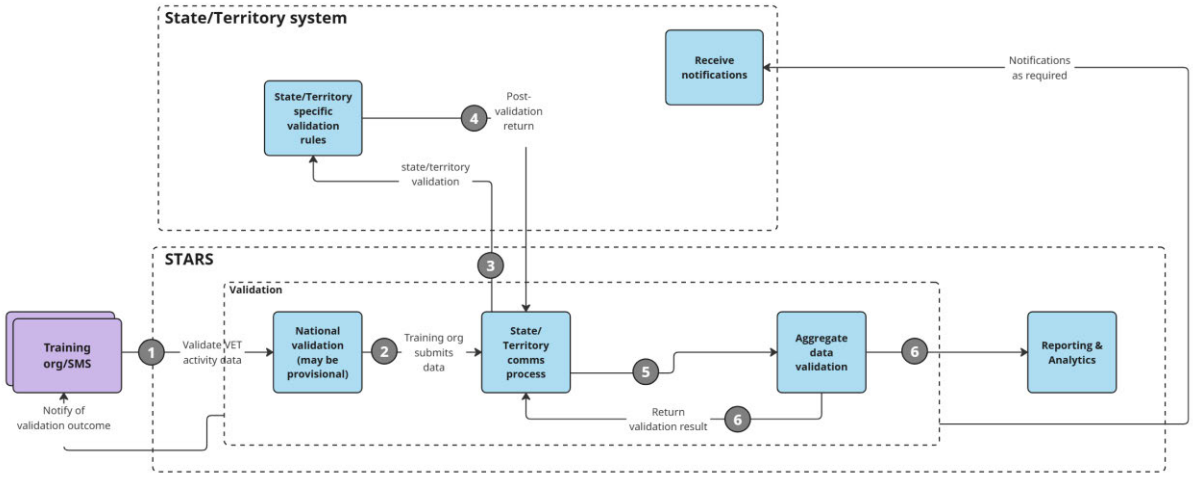


Figure 8 | Data Flow Pattern 1a - direct submission pathway

Table 17 outlines the Pattern 1a data flow steps.

Table 17 | Data Flow Pattern 1a - direct submission pathway data flow steps

Step #	Description
1	Training organisations may send data for provisional validation prior to final submission. This allows training organisations to obtain a degree of assurance that their data is fit for submission. This provisional validation step can occur multiple times before final submission. In a scenario where a training organisation might choose to submit their data without a provisional validation step beforehand, a validation process will still occur prior to submission. In business terminology, this would not be considered 'provisional validation'. But from a technical implementation point of view, the validations run in such a scenario before the submission was accepted, would be the same as those run during a 'provisional validation' step.
2	Training organisations submit data to STARS.
3	STARS sends data to state/territory training authority systems.
4	Validation result is returned to STARS from the state/territory training authority systems. Data is approved for release to other stakeholders.
5	Aggregate data validation (AKA data quality checks) occurs in STARS.
6	Validated result is returned. Data flows onwards within STARS for reporting and analytics purposes. Notifications can be sent to stakeholders across the various data flow steps, for example indicating validation results to training organisations, or process notifications to state/territory training authorities as required.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1178 of 1295

7.6.2.2.2 Data Flow Pattern 1b - direct submission pathway

The state/territory training authority has provided their data validation rules, and the STARS Solution can run these internally, alleviating the need to communicate with state/territory training authority systems for data validation purposes.

Figure 9 shows the Pattern 1b direct submission pathway data flow.

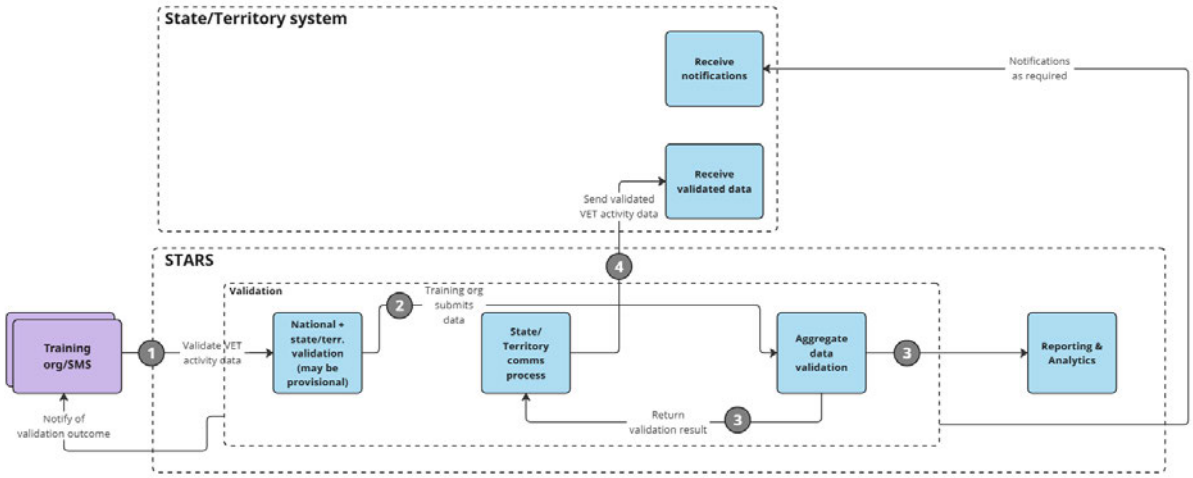


Figure 9 | Data Flow Pattern 1b - direct submission pathway

Table 18 outlines the Pattern 1b data flow steps.

Table 18 | Data Flow Pattern 1b - direct submission pathway data flow steps

Step #	Description
1	Training organisations may send data for provisional validation prior to final submission. Allowing training organisations to obtain a degree of assurance that their data is fit for submission. This provisional validation step can occur multiple times before final submission. In a scenario where a training organisation might choose to submit their data without a provisional validation step beforehand, a validation process will still occur prior to submission. In business terminology, this would not be considered 'provisional validation'. But from a technical implementation point of view, the validations run in such a scenario before the submission was accepted, would be the same as those run during a 'provisional validation' step.
2	Training organisations submit final data to STARS. Aggregate data validation (AKA data quality checks) occurs in STARS.
3	Validated result is returned. Data flows onwards within STARS for reporting and analytics purposes.
4	Final VET activity data is sent to the state/territory training authority.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1179 of 1295

Step #	Description
	Notifications can be sent to stakeholders across the various data flow steps, for example indicating validation results to training organisations, or process notifications to state/territory training authorities as required.

7.6.2.2.3 Data Flow Pattern 2a - indirect submission pathway

Figure 10 shows the Pattern 2a indirect submission pathway data flow.

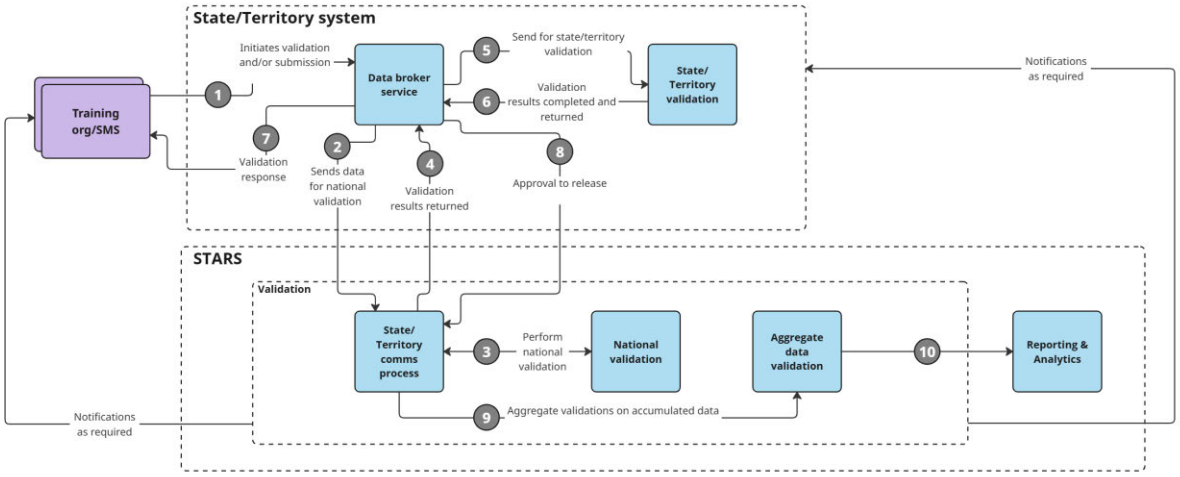


Figure 10 Data Flow Pattern 2a indirect submission pathway data flow

Table 19 outlines the Pattern 2a data flow steps.

Table 19 | Data Flow Pattern 2a - indirect submission pathway data flow steps

Step #	Description
1	Training organisations may send data for provisional validation prior to final submission. Allowing training organisations to obtain a degree of assurance that their data is fit for submission. This provisional validation step can occur multiple times before final submission. National validation rules are stored in STARS, which provides a validation service for 'indirect' jurisdictions to make use of.
2	Data sent to STARS for national validations.
3	National validations performed by STARS.
4	National validation result returned by STARS.
5	State/Territory validation performed within state/territory system.
6	State/Territory validation completed within state/territory system.
7	Validation response back to training organisation.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1180 of 1295

Step #	Description
8	Approval to release.
9	Aggregate data validation (AKA data quality checks) occurs in STARS.
10	Data flows onwards within STARS for reporting and analytics purposes.

7.6.2.2.4 Data Flow Pattern 2b - indirect submission pathway

Figure 11 shows the Pattern 2b indirect submission pathway data flow.

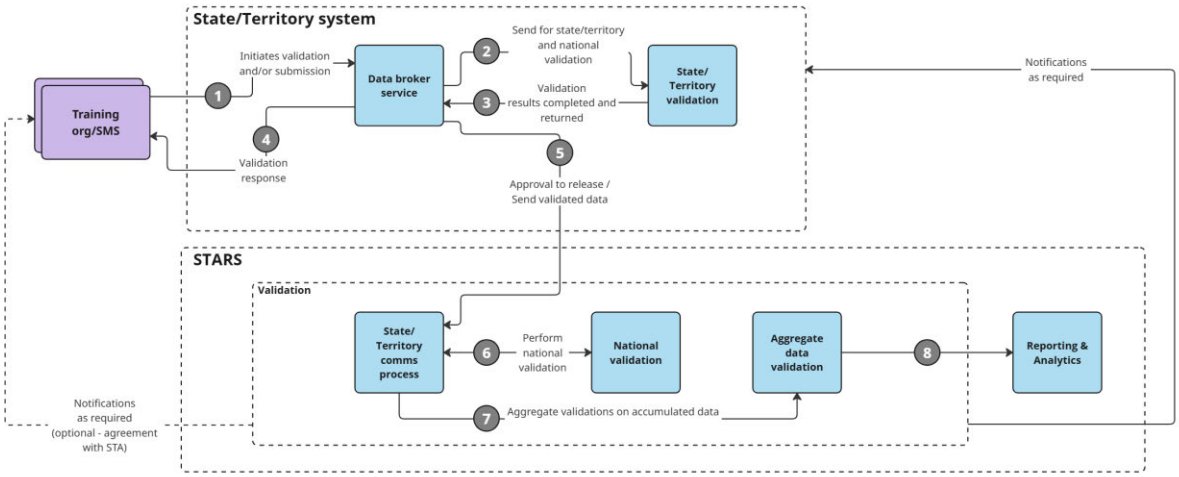


Figure 11 Data Flow Pattern 2b indirect submission pathway data flow

Table 20 outlines the Pattern 2b data flow steps.

Table 20 | Data Flow Pattern 2b - indirect submission pathway data flow steps

Step #	Description
1	Training organisations may send data for provisional validation prior to final submission. Allowing training organisations to obtain a degree of assurance that their data is fit for submission. This provisional validation step can occur multiple times before final submission. National validation rules are implemented by the indirect jurisdiction.
2	STA validates data.
3	STA completes validation.
4	Validation result returned by STA.
5	Data sent to STARS.
6	Validation performed by STARS. National validations will still be run by STARS, for assurance reasons.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1181 of 1295

Step #	Description
7	Aggregate data validation (AKA data quality checks) occurs in STARS.
8	Data flows onwards within STARS for reporting and analytics purposes.

7.6.2.3 Data retention

The existing DEWR/NCVER approach to data retention will be utilised by STARS. This will be determined based on non-functional requirements.

7.6.2.4 Data audience

The intent is for STARS to share data with the following stakeholders:

- training organisations, registered and non-registered, using student management systems or manual reporting processes
- the Australian Government DEWR
- NCVER – the national VET data custodian
- STAs
- VET sector regulators (ASQA, VRQA, WA TAC).

7.6.3 Integration

7.6.3.1 Training organisation/SMS integration with STARS – M2M

Training organisation/SMS machine-to-machine integration will follow IAM Pattern B – Training organisation machine-to-machine onboarding, authentication and authorisation.

The request flow includes the use of APIM to provide additional security, rate limiting, and API management capabilities. APIM processes requests before they reach REST APIs within the STARS solution.

7.6.3.2 STA integration with STARS – M2M

STA machine-to-machine integration will follow IAM Pattern D – Partner organisation machine-to-machine onboarding, authentication and authorisation. Both web service and data sharing integration approaches are expected for STAs.

Web service integration scenarios include those where a call/response is required. Examples would be the distributed nature of data validations inherent in Data Flow Pattern 1a - direct submission pathway and Data Flow Pattern 2b - indirect submission pathway, where some data validations are carried out by STARS and some carried out by STA systems. The request flow includes the use of APIM to provide additional security, rate limiting, and API management capabilities. APIM processes requests before they reach REST APIs within the STARS solution.

Scenarios where bulk data needs to be transferred between STARS and STA systems would be better served via a data integration/sharing approach (e.g. via Azure Data Factory).

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1182 of 1295

7.6.3.3 STARS integrations for importing reference data

Figure 12 shows STARS integrations for importing reference data.

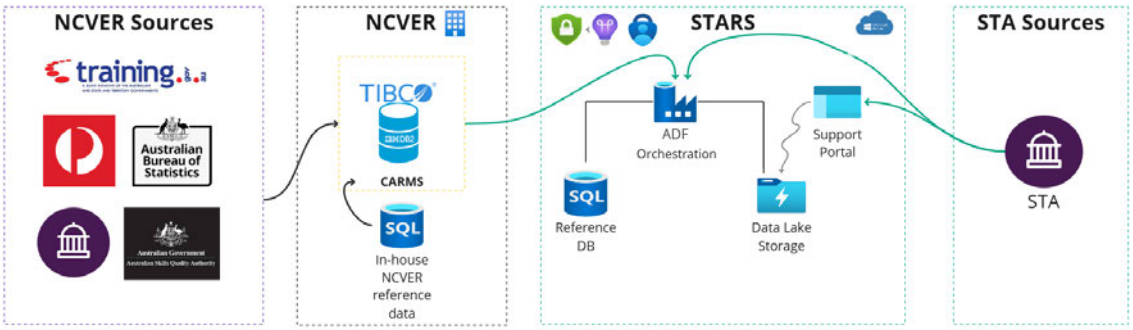


Figure 12 | STARS integration with NCVER for importing reference data

The STARS solution will import reference data from NCVER's Classification and Reference Management System (CARMS). CARMS is responsible for consolidating reference data from various sources, curating this information, and versioning it. The STARS solution will integrate with NCVER's CARMS system to receive reference data. Some reference data originate from external sources (e.g. Australia Post, Australian Bureau of Statistics) and is processed and stored by NCVER, other data originate from NCVER. Data retrieval from CARMS is expected to be managed via Azure Data Factory and stored in a reference database within the STARS system for use by STARS components. Should there be a need for non-relational storage during this process, Data Lake Storage can be used to address this need.

Any STA-specific reference data required by STARS will be imported into STARS and retrieved via a similar approach. Should any scenario call for a semi-manual approach, the STARS Support Portal can include a user interface to allow for upload of reference data in a flat file format.

7.6.3.4 STARS integration with NCVER for importing classification data

VET IS classification data is designed and managed within NCVER's Erwin system. The integration process mirrors that of reference data integration. A connection is established between the STARS Azure environment and NCVER infrastructure. This connection facilitates the import of classification data from CARMS into STARS.

Figure 13 shows the STARS integration with NCVER for importing classification data.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1183 of 1295

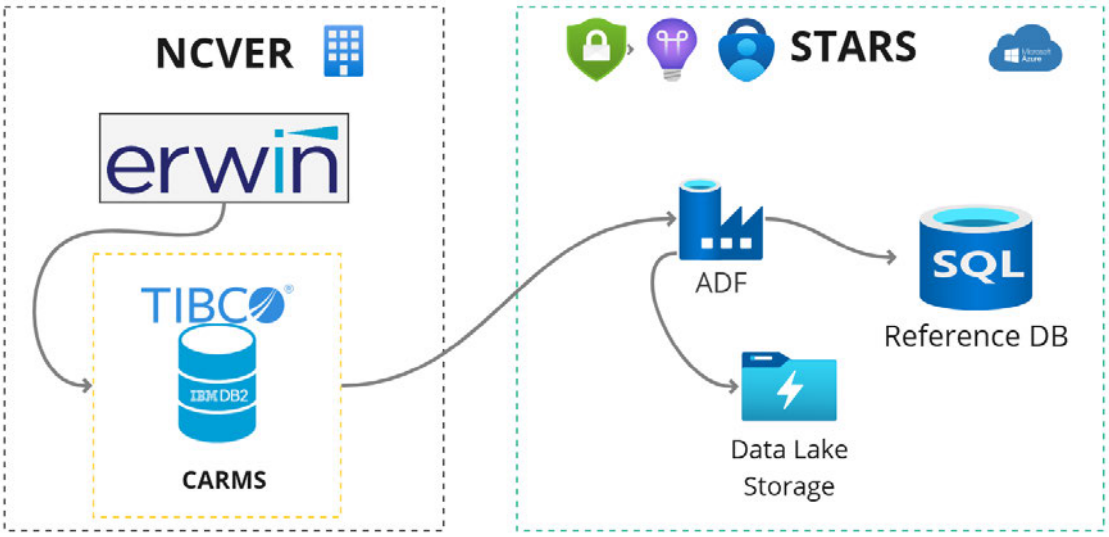


Figure 13 | STARS integration with NCVER for importing classification data

7.6.3.5 STARS integration with NCVER for importing AVETMISS 8 data

Figure 14 shows STARS integration with NCVER for importing AVETMISS data.

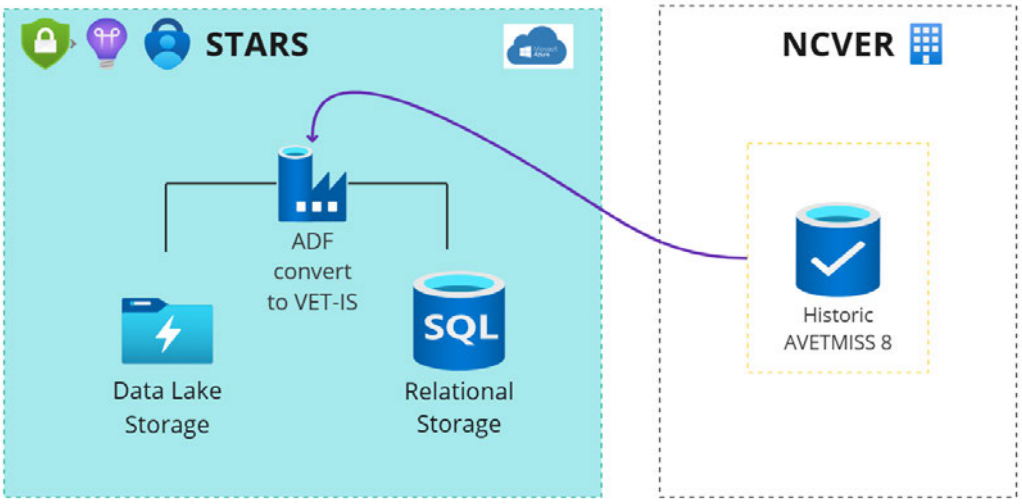


Figure 14 | STARS integration with NCVER for importing AVETMISS data

STARS will import historic AVETMISS 8 data from NCVER's Central Data Repository (CDR) system. This data import will be performed on a scheduled basis as required. Once imported, the AVETMISS 8 data will be stored in Data Lake and/or relational storage as appropriate.

Azure Data Factory will convert and process AVETMISS 8 data into the VET IS format. After processing, the transformed data will be stored by STARS. This approach supports integration of historic AVETMISS 8 data into STARS, aligning it with the VET IS standard for consistent use.

7.6.3.6STARS integration with NCVER for exporting VET IS data

Figure 15 shows STARS integration with NCVER for exporting VET IS data.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1184 of 1295

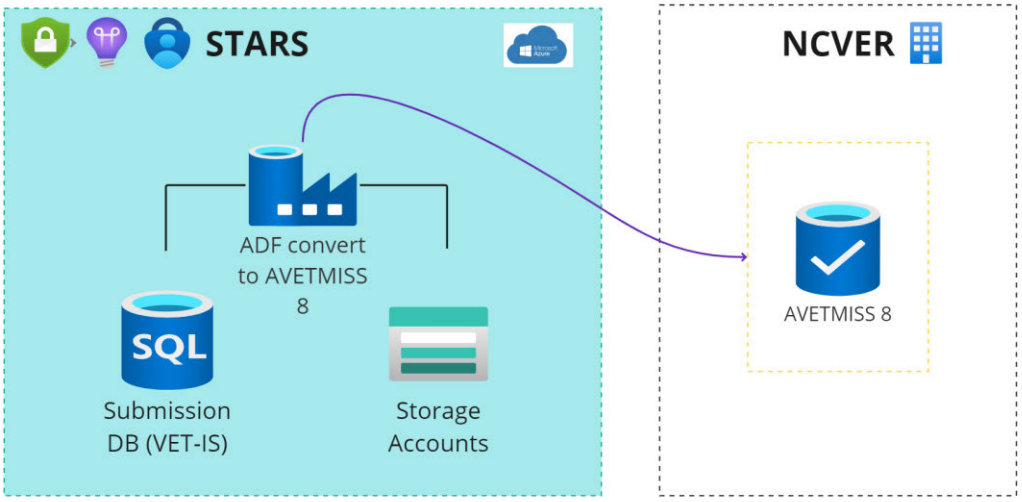


Figure 15 | STARS integration with NCVER for exporting VET IS data

STARS will export data to NCVER's on-premises systems as required, utilising an established connection and Azure Data Factory to orchestrate the export process. VET IS formatted data is stored in the Submission DB (having already undergone both basic and aggregate quality checks) and will be prepared for export to NCVER.

Azure Data Factory will manage the conversion of the VET IS data into the AVETMISS 8 format during this process, performing the necessary ETL tasks. This ensures that the data meets NCVER's requirements and is delivered in the correct format for further processing and analysis.

7.6.4 Infrastructure

Table 21 provides an overview of the various infrastructure resources of the solution.

Table 21 | STARS infrastructure resources

Infrastructure resource	Status	Location	Network zone	Quantity
Entra ID	New	Australian data residency	Dedicated VDS hosting environment	Premium P2 tier: • 1 x Prod
Azure AD B2C	New	Australian data residency	Dedicated VDS hosting environment	• 1 x Dev/Test/E-Fix Premium P2 tier: • 1 x Staging/Pre-Prod/Load Testing/Prod

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1185 of 1295

Infrastructure resource	Status	Location	Network zone	Quantity
Azure Kubernetes Service	New	Azure – Australia East/Southeast	Dedicated VDS hosting environment	• 1 x Dev • 1 x Test • 1 x Staging • 1 x Load Testing • 1 x Pre-Prod • 1 x Hot-Fix • 1 x Prod
Azure SQL Database – Transactional	New	Azure – Australia East/Southeast	Dedicated VDS hosting environment	• 1 x Dev • 1 x Test • 1 x Staging • 1 x Load Testing • 1 x Pre-Prod • 1 x Hot-Fix • 1 x Prod
Azure SQL Database – Reporting	New	Azure – Australia East/Southeast	Dedicated VDS hosting environment	• 1 x Dev • 1 x Test • 1 x Staging • 1 x Load Testing • 1 x Pre-Prod • 1 x Hot-Fix • 1 x Prod
Azure Application Insights	New	Azure – Australia East/Southeast	Dedicated VDS hosting environment	• 1 x Dev • 1 x Test

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1186 of 1295

Infrastructure resource	Status	Location	Network zone	Quantity
				• 1 x Staging • 1 x Load Testing • 1 x Pre-Prod • 1 x Hot-Fix • 1 x Prod
Azure API Management	New	Azure – Australia East/Southeast	Dedicated VDS hosting environment	Developer tier: • 1 x Dev/Test/Hot-Fix/Pre-prod Premium tier: • 1 x Staging/Load Testing/Prod
Azure Communication Services (notifications)	New	Australian data residency	Dedicated VDS hosting environment	• 1 x Dev • 1 x Test • 1 x Staging • 1 x Load Testing • 1 x Pre-Prod • 1 x Hot-Fix • 1 x Prod
Azure Data Factory	New	Australian data residency	Dedicated VDS hosting environment	• 1 x Dev • 1 x Test • 1 x Staging • 1 x Load Testing • 1 x Pre-Prod

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1187 of 1295

Infrastructure resource	Status	Location	Network zone	Quantity
				• 1 x Hot-Fix • 1 x Prod
Microsoft Purview	New	Azure – Australia East	Dedicated VDS hosting environment	• 1 x Prod
Azure Databricks	New	Australian data residency	Dedicated VDS hosting environment	• 1 x Dev • 1 x Test • 1 x Staging • 1 x Load Testing • 1 x Pre-Prod • 1 x Hot-Fix • 1 x Prod
Azure Data Lake Storage	New	Azure – Australia East/Southeast	Dedicated VDS hosting environment	• 1 x Dev • 1 x Test • 1 x Staging • 1 x Load Testing • 1 x Pre-Prod • 1 x Hot-Fix • 1 x Prod
Azure Blob Storage	New	Azure – Australia East/Southeast	Dedicated VDS hosting environment	• 1 x Dev • 1 x Test • 1 x Staging • 1 x Load Testing • 1 x Pre-Prod

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1188 of 1295

Infrastructure resource	Status	Location	Network zone	Quantity
				• 1 x Hot-Fix • 1 x Prod
Power BI	New	Australian data residency	Dedicated VDS hosting environment	• 1 x Dev • 1 x Test • 1 x Staging • 1 x Load Testing • 1 x Pre-Prod • 1 x Hot-Fix • 1 x Prod
Azure Defender for Cloud Apps	New	Non-regional	Dedicated VDS hosting environment	• 1 x Dev • 1 x Test • 1 x Staging • 1 x Load Testing • 1 x Pre-Prod • 1 x Hot-Fix • 1 x Prod
Azure Monitor	New	Azure – Australia East/Southeast	Dedicated VDS hosting environment	• 1 x Dev • 1 x Test • 1 x Staging • 1 x Load Testing • 1 x Pre-Prod • 1 x Hot-Fix • 1 x Prod

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1189 of 1295

Infrastructure resource	Status	Location	Network zone	Quantity
Microsoft Sentinel	New	Non-regional	Dedicated VDS hosting environment	• 1 x Dev • 1 x Test • 1 x Staging • 1 x Load Testing • 1 x Pre-Prod • 1 x Hot-Fix • 1 x Prod
Azure Key Vault	New	Azure – Australia East/Southeast	Dedicated VDS hosting environment	• 1 x Dev • 1 x Test • 1 x Staging • 1 x Load Testing • 1 x Pre-Prod • 1 x Hot-Fix • 1 x Prod

7.6.5 Network

Solution networking will follow the existing pattern for the dedicated VDS hosting environment. This is shown at Figure 16.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1190 of 1295

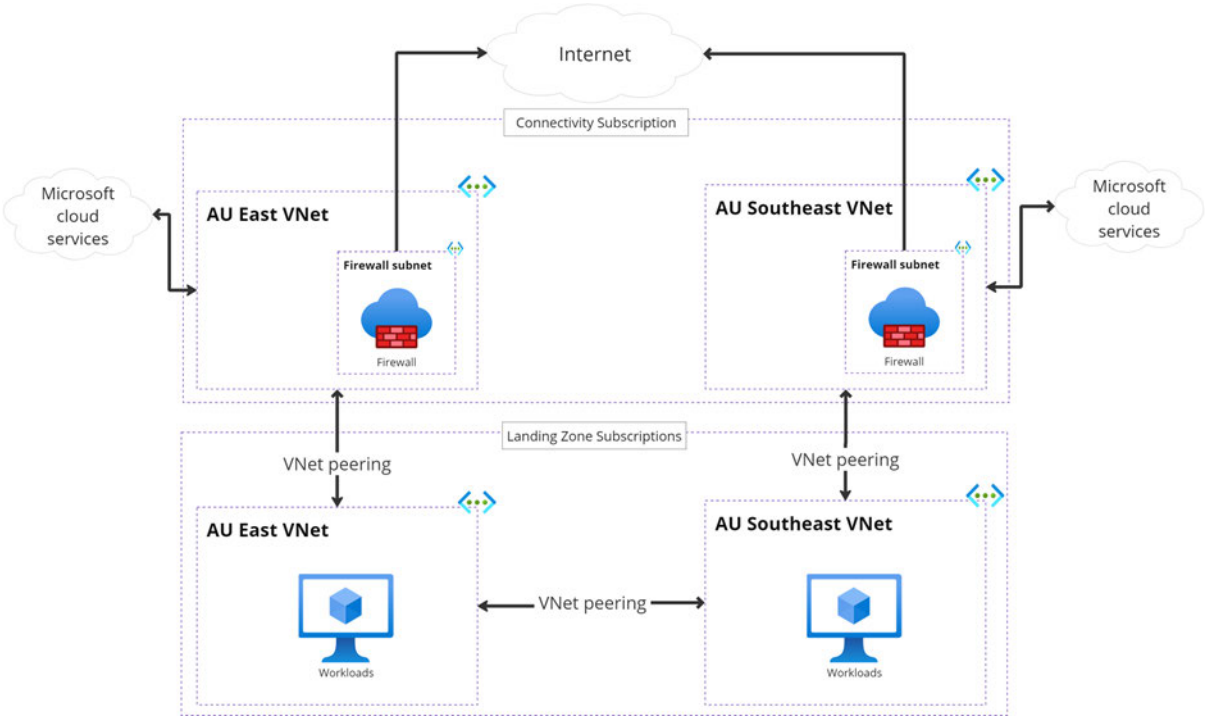


Figure 16 | Solution networking for STARS

Further information on the networking configuration of the oobe provisioned 'Perimeta' dedicated VDS hosting environment can be found in the VDS - Perimeta for Azure - Solution Design[5].

7.6.6 Security

The STARS solution will utilise various strategies to secure the platform. These strategies include fortifying network security and safeguarding deployments of solution components and associated artefacts (such as application code and the underlying databases).

NCVER has advised that they have an *NCVER Cyber Security Program and Principles* document that specifies their approach and alignment to the Australian Government's Information Security Manual (ISM). This solution will utilise the ISM to guide the accreditation processes.

The deployment strategy will strictly adhere to the principles of zero trust network access, with a heavy reliance on private connectivity to ensure secure communications between services and to limit exposure to public networks. Key security components include:

- Azure API Management, a management platform for APIs within the STARS solution. As a PaaS, API Management supports the complete API lifecycle. API Management can integrate with Entra ID, Azure AD B2C (OAuth 2.0, OpenID Connect), as well as Defender for APIs, for full lifecycle protection, detection, and response coverage for APIs.

[5] Internal DEWR document, link will not work outside of the DEWR network.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1191 of 1295

- Microsoft Sentinel, which provides SIEM and SOAR for the solution. Features provided by Sentinel includes visualisation of log data, anomaly detection and alerting, investigation of security incidents, proactive threat hunting and automated responses to security events.
- Azure AD B2C, an enterprise-grade consumer directory and broker service. Will be employed in the solution where necessary to enable OAuth 2.0 and OpenID Connect authentication/authorisation.
- Public Access Restrictions – a default 'deny all' stance should be adopted towards public access. Azure resources should be configured to reject all unsolicited inbound traffic from the internet. Exceptions to this rule should be controlled and granted solely on an as-needed basis, ensuring minimal attack vectors for potential threats.
- Firewall Policies – Firewall policies should be configured with granularity, that by default deny all inbound and outbound network flows. Any permission granted is based on explicit need, in accordance with the least privilege principle.

7.6.6.1 Identity and access management

Identity and access management (IAM) will follow differing patterns, tailored to relevant use cases. These are outlined below.

7.6.6.1.1 Training organisations

7.6.6.1.1.1 IAM Pattern A – Training organisation user onboarding, authentication and authorisation

There are approximately 4,000 training organisations active within Australia, with likely several user accounts required per training organisation, this user cohort should be expected to be in the tens of thousands. With this size user cohort, user on-boarding should be largely self-service to minimise the need for direct user support. Once a training organisation has been registered and an initial administration user created, a training organisation should be able to manage their own user accounts as they relate to the STARS solution.

A pre-existing capability assisting the STARS solution is the NCVER DataHub web portal. This web portal employs a combination of the Squiz Matrix web content management system and the Okta identity and access management platform to provide user on-boarding and authentication/authorisation. Okta acts as a federated identity provider[6] for Squiz, allowing users registered in Okta to login to Squiz, via an Okta login page.

Thousands of training organisation user accounts are currently registered in the NCVER DataHub. The DataHub user onboarding, self-service organisation management and existing user base will be leveraged by the STARS solution. In this way, the STARS solution will not need to reinvent the user onboarding wheel for training organisations. Currently registered DataHub users will be able to re-use their existing DataHub user accounts to authenticate to STARS, providing a smooth user experience

[6] https://en.wikipedia.org/wiki/Identity_provider

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1192 of 1295

and improved security by alleviating the need for training organisation users to manage a new and additional user account to interact with STARS.

Figure 17 shows the user onboarding process for a new training organisation that has not yet registered on the NCVER DataHub[7].

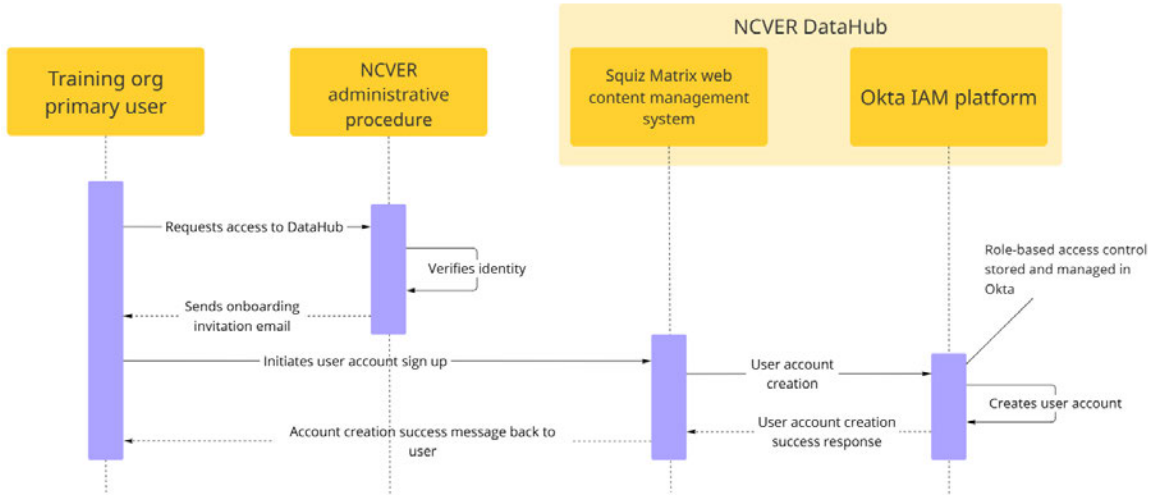


Figure 17 | Training organisation primary user onboarding

Once a training organisation primary user (e.g. a training organisation CEO) has been onboarded to the DataHub, this primary user is able to invite their own subordinate users to register on the DataHub and assign roles to these subordinate users, within the context of their own training organisation.

Figure 18 shows a primary user logging into the NCVER DataHub to invite a subordinate user from their training organisation to register for a DataHub user account. Role-based access control can be assigned by the primary user to their subordinate users.

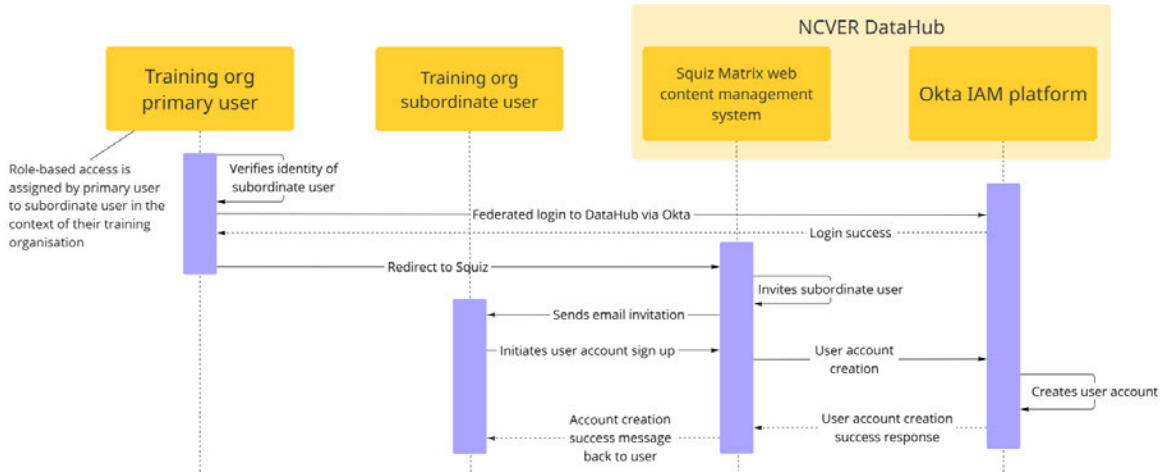


Figure 18 | DataHub primary user invites subordinate user

[7] The NCVER DataHub includes other technology platforms beyond Squiz and Okta. But other aspects of the DataHub are not directly relevant to the STARS solution.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1193 of 1295

In a similar way to how NCVER's Okta instance acts as an identity provider for Squiz, Okta acts as an identity provider for STARS. A federated trust is established between the NCVER Okta instance and the STARS solution. This allows for OpenID Connect[8] user authentication/authorisation to be extended to STARS. Okta will enforce multi-factor authentication during the login process.

Figure 19 shows a DataHub user authenticating to the STARS Provider Portal.

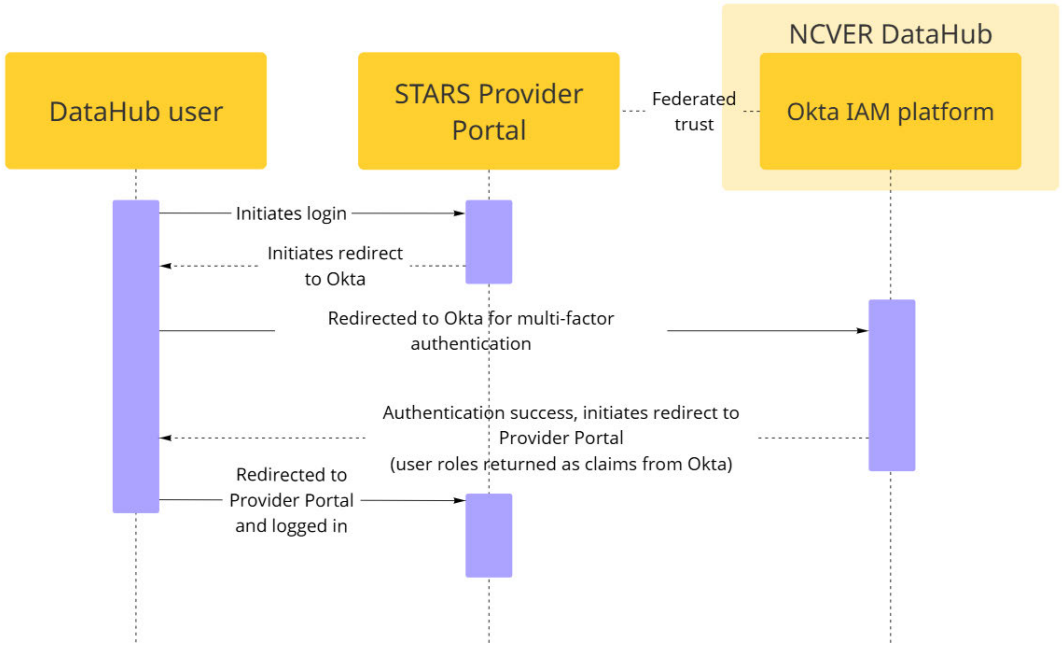


Figure 19 | DataHub user authenticates to STARS Provider Portal

7.6.6.1.1.2 IAM Pattern B – Training organisation machine-to-machine onboarding, authentication and authorisation

Training organisations make use of Student Management Systems (SMS) to interact with STARS web APIs to send VET activity data to STARS and meet their reporting obligations. Student Management Systems will need to use machine-to-machine (M2M) credentials to authenticate to STARS web APIs. Due to the size of the user cohort involved, the process for registering machine-to-machine authentication credentials should be self-service, to minimise the need for direct support.

The Squiz web content management system, as part of the NCVER DataHub, will be enhanced to allow for an appropriately authorised training organisation user to register a set of M2M credentials for use by their training organisation's SMS when interacting with STARS.

Figure 20 shows a DataHub user registering M2M credentials via a self-service process, enabled by the NCVER DataHub. A prerequisite for this process is that a public/private key pair has been created by

[8] https://en.wikipedia.org/wiki/OpenID#OpenID_Connect_(OIDC)

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1194 of 1295

the training organisation (or their chosen supporting vendor). The public key is provided via direct upload to the DataHub, or a 'jwks_uri' address[9] is provided.

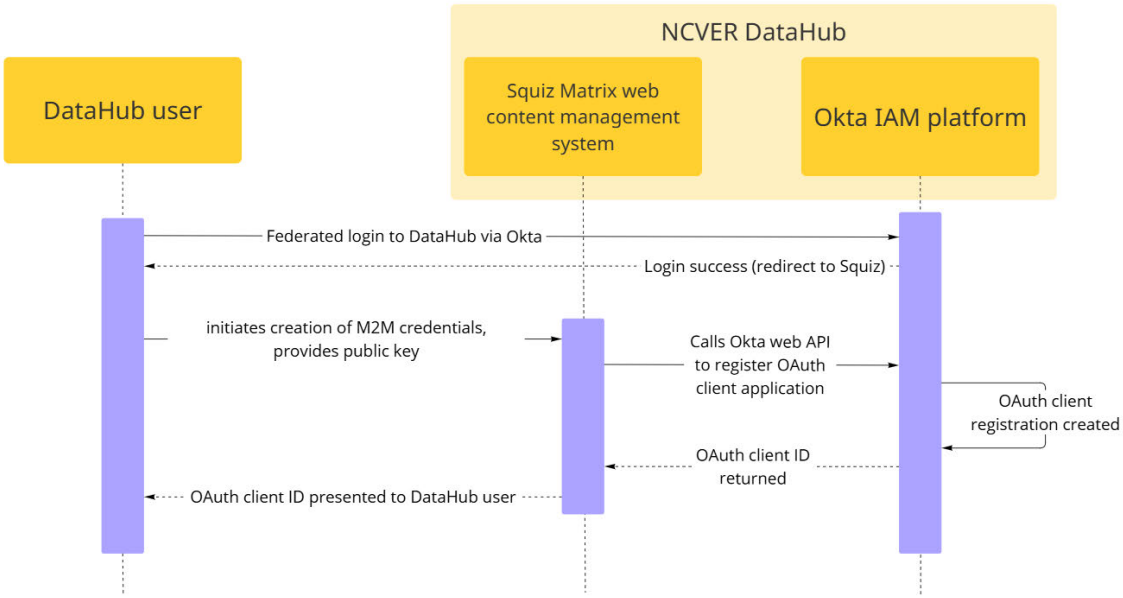


Figure 20 | DataHub training org user registers M2M credentials for use with STARS web API

Once registered, the training organisation makes use of their M2M credentials in conjunction with their student management system (SMS), allowing their SMS to authenticate with STARS web APIs on behalf of the training organisation. An Entra ID feature known 'workload identity federation'[10] provides for federated authentication in the context of M2M connections. The NCVER Okta instance and the Entra ID instance in the STARS Azure tenant work together to enable authentication to STARS web APIs using M2M credentials registered in the NCVER Okta instance.

Figure 21 shows an authentication flow involving a student management system, the NCVER Okta instance and the STARS Entra ID instance, making use of Workload Identity Federation.

[9] RFC 7591 (https://www.rfc-editor.org/rfc/rfc7591.html#section-2) notes that a 'jwks_uri' is preferable as it allows for easier key rotation.
[10] https://learn.microsoft.com/en-us/entra/workload-id/workload-identity-federation

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1195 of 1295

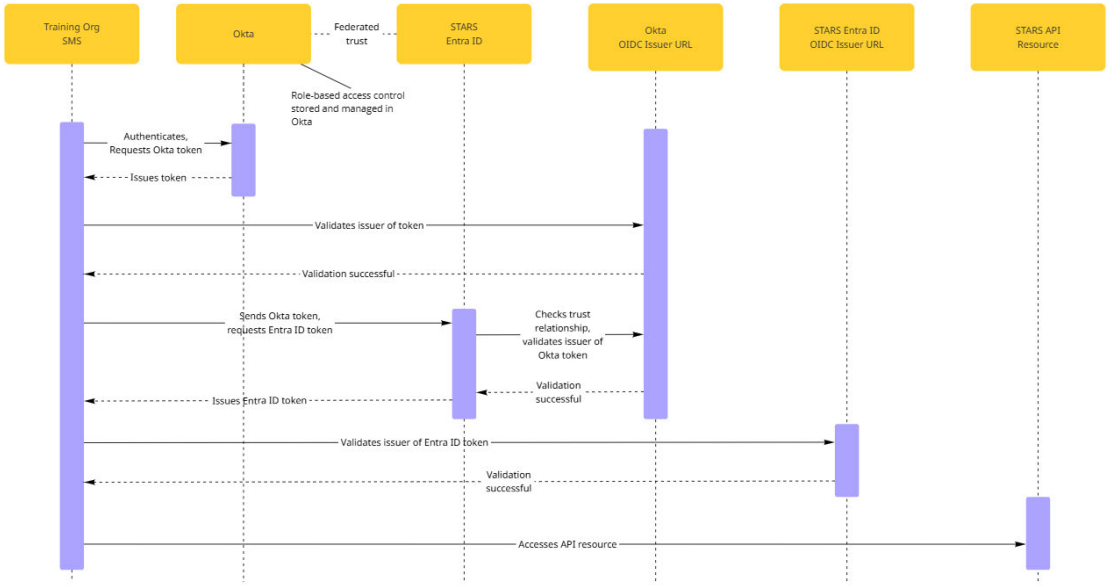


Figure 21 | M2M authentication to STARS web API, using Workload Identity Federation

This approach supports mutual authentication. Tokens issued by Okta are validated by student management systems and the STARS Entra ID instance. Tokens issued by Entra ID are validated by student management systems.

The expectation is that authorisation for training organisations will be managed within the NCVER Okta instance. A constraint (likely by design) of Workload Identity Federation is that claims (e.g. OAuth scopes) contained in the Okta token cannot be directly retrieved and copied to the subsequently issued STARS token. If differing levels of authorisation are required for training organisations interacting with STARS via M2M channels, authorisations stored in Okta can be retrieved by STARS via a separate call to the Okta web API.

7.6.6.1.2 Partner organisations

7.6.6.1.2.1 IAM Pattern C - Partner organisation user onboarding, authentication and authorisation

Partner organisations include NCVER, state (and territory) training authorities (STAs), VET sector regulators, and relevant federal government agencies. Partner organisations can act as data submitters, data consumers, or a combination of the two. The size of the partner organisation user cohort will be significantly smaller than the training organisation user cohort. Partner organisations have pre-existing enterprise-grade identity and access management platforms that can be federated directly with the STARS Entra ID instance. This will allow partner users to employ their existing organisational user accounts to access relevant portions of the STARS solution, providing seamless single sign-on and alleviating the need for partner organisations to create and manage new and separate user accounts to interact with STARS, resulting in better security and a better user experience.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1196 of 1295

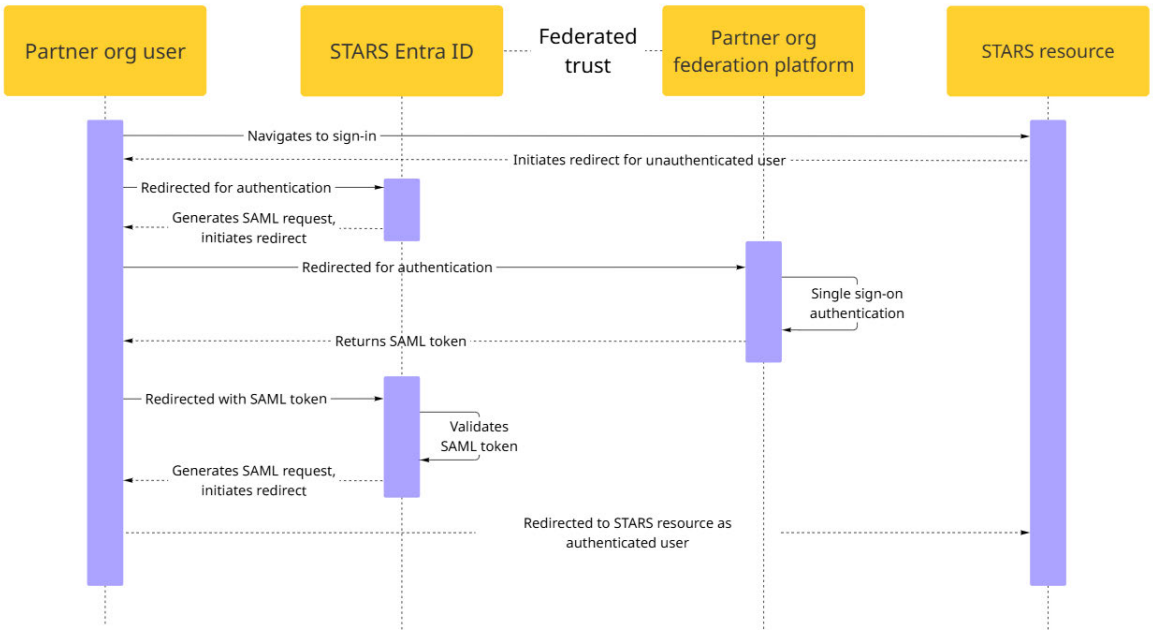
Identity federation between STARS and partner organisations will be achieved via capabilities that, in the context of Entra ID, are grouped under the moniker of 'B2B collaboration'. Entra ID supports various approaches under 'B2B collaboration'[11], two that are expected to assist in the context of STARS are:

- SAML/WS-Fed IdP federation[12] – supported by a wide range of identity and access management products from many different vendors.
- Microsoft Entra accounts – for partner organisations that make use of Entra ID as all or part of their identity and access management solution, 'guesting' user accounts from the partner organisation's Azure tenant into the STARS Azure tenant is a straightforward way to configure federation.

Once partner organisations are federated with the STARS Azure tenant, partner user accounts can be given appropriate authorisation to access STARS resources such as the STARS support portal, reports, data visualisations, etc.

For authorisation purposes, Entra ID 'dynamic membership groups'[13] can be used to automatically assign guest users to relevant groups within the STARS tenant, for example based on the email domain, or other identifying attribute of the partner user account.

Figure 22 shows a partner organisation user accessing a STARS resource, via SAML single sign-on, enabled by their own organisation's federation platform. The Prerequisite being that federated trust is configured between the partner organisation's federation platform and the STARS Entra ID instance.



[11] https://learn.microsoft.com/en-us/entra/external-id/identity-providers
[12] https://learn.microsoft.com/en-us/entra/external-id/direct-federation-overview
[13] https://learn.microsoft.com/en-us/entra/identity/users/groups-dynamic-membership

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1197 of 1295

Figure 22 | Partner organisation user authentication - SAML example

7.6.6.1.2.2 IAM Pattern D – Partner organisation machine-to-machine onboarding, authentication and authorisation

Machine-to-machine authentication in the context of STARS and partner organisations needs to support various use cases, including:

- Pathway 1a – direct submission – where STAs of direct jurisdictions run state/territory validations from within their own systems.
- Pathway 1b – direct submission– where STARS runs state/territory validations on behalf of the direct jurisdiction STA.
- Pathway 2a – indirect submission – where STAs of indirect jurisdictions submit collected VET activity data to STARS. In this model, STARS provides a validation service that is called on by the STA to run national validations.
- Pathway 2b – indirect submission – similar to pathway 2a, with the difference being that in this model, the indirect STA implements national validations within their own system.
- Web API integration with NCVER – web API connectivity between STARS and NCVER will be required to enable integration between STARS and the NCVER Okta instance.
- Data integration with NCVER – VET activity data must be transferred to NCVER and data received from NCVER, such as reference/classification data.
- Data integration with state/territory VET sector regulators

For use cases such as Pathway 1a and Pathway 2a, where data must be transferred between STARS and partner organisations for validation purposes, implying a request/response model, a web API approach that makes use of OAuth-based authentication will be used.

For use cases that involve data integration/movement, but do not require a request/response model, an approach leveraging Azure Data Factory will be used, with various authentication methods available to meet partner capabilities, including mTLS.

Expected authentication methods for partner integration use cases are outlined in Table 22.

Table 22 | Partner integration authentication methods

Partner integration	Authentication method
Pattern 1a – direct submission pathway Web API - STARS carries out national validations, sends data to STA for state/territory specific validations, receives response.	OAuth-based, mutual signature verification, over private channel if possible (e.g. VPN or Azure Private Link).

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1198 of 1295

Partner integration	Authentication method
Pattern 1b – direct submission pathway STARS carries out both national and state/territory specific validations, sends validated data to STA.	Leverage Azure Data Factory, various authentication methods, over private channel if possible (e.g. VPN or Azure Private Link).
Pattern 2a – indirect submission pathway Web API – STA carries out state/territory specific validations, sends data to STARS for national validations, receives response.	OAuth-based, mutual signature verification, over private channel if possible (e.g. VPN or Azure Private Link).
Pattern 2b – indirect submission pathway STA carries out both national and state/territory specific validations, sends validated data to STARS.	Leverage Azure Data Factory, various authentication methods, over private channel if possible (e.g. VPN or Azure Private Link).
Web API integration with NCVER To facilitate role-based access control for training organisation users, web API integration with the NCVER Okta instance is required.	OAuth-based, mutual signature verification, over private channel if possible (e.g. VPN or Azure Private Link).
Data integration with NCVER VET activity data transferred to NCVER, reference/classification data received from NCVER.	Leverage Azure Data Factory, various authentication methods, over private channel if possible (e.g. VPN or Azure Private Link).
Data integration with state/territory VET sector regulators VET activity data transferred to state/territory VET sector regulators.	Leverage Azure Data Factory, various authentication methods, over private channel if possible (e.g. VPN or Azure Private Link).

Figure 23 shows an OAuth certificate-based mutual authentication approach between an STA system and STARS. In this example, signature verification is carried out by retrieving public keys from OIDC issuer URLs, as these better support certificate rotation. However, public keys can be directly exchanged if necessary.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1199 of 1295

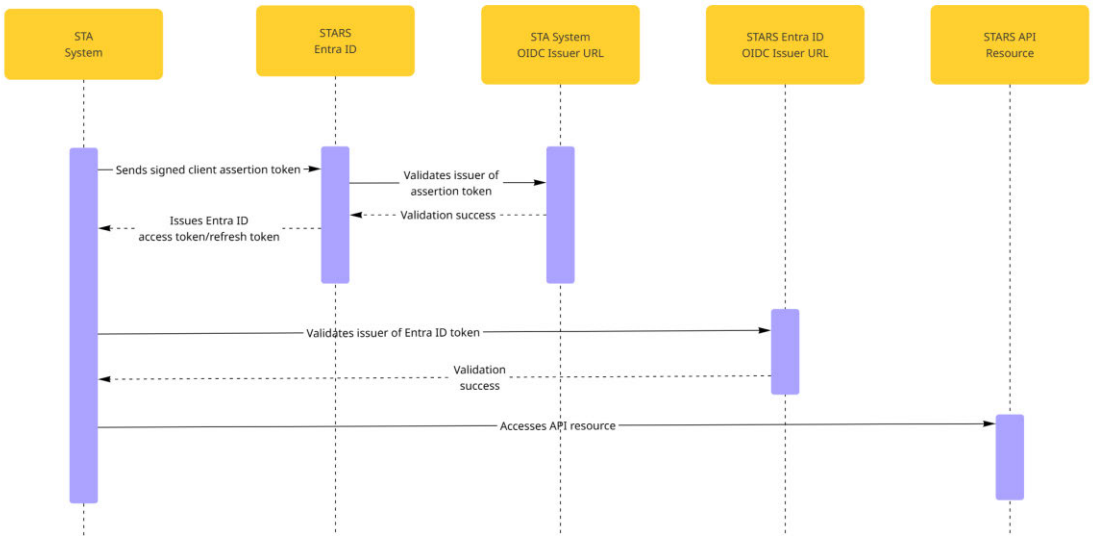


Figure 23 | STA/STARS OAuth certificate-based authentication

Data integration authentication is not diagrammed here and will depend on partner organisation capabilities and IT environments.

7.6.6.1.3 Internal authentication patterns

Covering internal use cases relating to development and administration within the STARS Azure tenant, internal authentication will leverage native Entra ID functionality.

7.6.6.1.4 Access control and authorisation

Table 23 outlines access control and authorisation for the STARS solution.

Table 23 | Access control and authorisation

Access control	Description
Azure AD B2C	Provides a federated authentication capability for external users. Working in conjunction with external identity providers as necessary. Other solution components such as APIM, will act as relying parties on Azure AD B2C.
Entra ID	The Azure tenant deployed for STARS will have an Entra ID instance that provides IDAM functionality for various solution components.

7.6.6.1.5 Additional identity security measures

Table 24 outlines additional identity security measures for the STARS solution.

Table 24 | Additional identity security measures

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1200 of 1295

Access control	Description
MFA	MFA will be integrated into the solution to add an extra layer of security during login and to meet ISM requirements, mitigating the risk of unauthorised access.
Secure Communication Protocols	All communication between users and STARS applications, as well as between different system components, will be encrypted using secure protocols such as Transport Layer Security (TLS).
Regular Security Audits	Periodic security audits will be conducted to identify and address potential vulnerabilities and to ensure ongoing compliance with security best practices.
Anti-malware	Microsoft Defender for Cloud Apps assists in mitigating malware from execution within the STARS solution.
FortiGate	Boundary FortiGate firewalls with granular Access Control Lists (ACLs) for both ingress and egress traffic.
User Role Restriction	Azure RBAC restriction utilising the IDAM capabilities within the solution.
Key management	Key management with Azure Key Vault.
Azure Policy	Tenancy configuration compliance with Azure Policy Cloud and Compliance Management. These cover prevention of public access by default, enforcing use of private subnets, use of HTTPS, defining permitted services and regions.
Data segregation	Data segregation, including prevention of traffic between Azure security boundaries.

7.6.6.2 Code security

Code security is paramount in the STARS solution, ensuring the protection of sensitive data and the overall integrity of the system. The project adheres to industry best practices and established security standards to mitigate risks and prevent vulnerabilities within application code.

7.6.6.2.1 Secure coding practices

Secure coding practices that will be employed as part of the STARS solution include:

- OWASP Guidelines: The project follows the Open Worldwide Application Security Project (OWASP) guidelines, which provide a comprehensive framework for developing secure web applications. These guidelines address common vulnerabilities such as injection flaws, broken authentication, and sensitive data exposure.
- OWASP ASVS Level 2: The project will adhere to the OWASP Application Security Verification Standard (ASVS) Level 2, an extensive set of security requirements for web application development that is referenced in the Australian Government ISM (control ISM-0971). This

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1201 of 1295

standard covers various aspects of security, including authentication, authorisation, session management, input validation, and error handling.

7.6.6.2.2 Key security measures

The key security measures that will be employed as part of the STARS solution include:

- Secrets Management. Sensitive information, such as API keys and database credentials, are securely stored in Azure Key Vault. This ensures that secrets are protected from unauthorised access and accidental exposure.
- Managed Service Identities. The project leverages Managed Service Identities (MSIs) for secure authentication between Azure services, eliminating the need for managing and embedding credentials within application code.
- Secure Cookies. Sensitive cookies are to be stored using the 'HttpOnly' and 'secure' flags, mitigating the risk of cross-site scripting (XSS) and man-in-the-middle attacks.
- TLS. All communication between users and the STARS applications, as well as between different system components, is encrypted using the TLS protocol to ensure data confidentiality and integrity.

7.6.6.2.3 Defence against common vulnerabilities

The following measures will be implemented to provide defence against common vulnerabilities:

- SQL Injection. The solution should use parameterised queries and object-relational mapping (ORM) tools (e.g. Microsoft's Entity Framework)[14] to prevent SQL injection vulnerabilities. This ensures that user-supplied input is properly validated and sanitised before being used in database queries.
- XSS. Content encoding and output escaping techniques should be employed to protect against XSS attacks. This will help to prevent malicious scripts from being injected into web pages and executed in user browsers.
- Cross-Site Request Forgery (CSRF). Anti-CSRF tokens are implemented to mitigate CSRF attacks, ensuring that only authorised users can submit requests to STARS.

[14] ORM should be used where appropriate, taking query performance into consideration.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1202 of 1295

8 Operations

8.1 System class and availability

System class and availability will be implemented as per the DEWR system availability framework. This is outlined in Table 25.

Table 25 | System class and availability

Category	Class 1	Class 2	Class 3	Class 4
Availability Profile	Continuous	High	Fault Tolerant	Low
Business Profile	Critical 24X7 Support Priority 1	High 24X7 Support Priority 2	Medium 12X5 Support Priority 3	Low 8X5 Support Priority 4
Architecture	Active – Active	Active – Standby	Active – Failover	Single Site
Availability Outage/Month	99.99% uptime per month 260 sec down	99.90% Uptime per month 45min down	98.50% per month Uptime 11hr down	95% Uptime per month 36hr down
Recoverability	3hrs RTO <5mins RPO	12hrs RTO <2hrs RPO	72 hrs RTO <8 hrs RPO	3 weeks RTO Best Efforts RPO

Based on currently known requirements, system availability is Class 2.

8.2 Backup and recovery

To support the system in achieving the required Recovery Time Objective (RTO) and Recovery Point Objective (RPO), application components will be backed by supported service redundancy settings for the production systems in Microsoft Azure. These settings are shown in Table 26.

Table 26 | Azure service redundancy settings

System component	Targeted Azure services	Redundancy setting (production)
STARS Data Collection, Integration Services and Master Data Management	Azure Kubernetes Service	Azure SQL: Zone-redundant

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1203 of 1295

System component	Targeted Azure services	Redundancy setting (production)
	Azure API Management Azure SQL Database	Azure Kubernetes Services Geo-redundant deployment Azure API Management Premium V2 for zone redundancy Azure SQL: Zone-redundant
STARS Support Portal	Azure Kubernetes Service Azure SQL Database	Azure Kubernetes Services Geo-redundant deployment Azure Kubernetes Services Geo-redundant deployment Azure SQL: Zone-redundant

Table 27 shows the backup and retention settings for Azure SQL databases that will be configured.

Table 27 | Azure SQL database backup and retention settings

Settings	Production (Prod) settings
Full Backups (automatic)	Every week for 6 weeks
Differential Backups (automatic)	Every 12 Hours
Transactional Log Backups	Every 5-10 minutes
LTR Weekly	Retained for 12 Weeks
LTR Monthly	Retained for 12 Months
LTR Annual	Retained for 7 Years
Geo-Replication and Auto-Failover	Zone-redundant storage (ZRS) or Geo-redundant storage (GRS), depending on system criticality definition

In the event of a full system rebuild being required, administrators will have the capability to re-run the latest production deployment pipelines in Azure DevOps.

8.3 Logging and monitoring

Logging and monitoring in STARS will be supported via:

- standard Azure Monitor-based functionality, with the addition of Microsoft Sentinel for SIEM capabilities

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1204 of 1295

- each Azure service will be configured to log events at an appropriate level
- applications will be configured to use Application Insights and to log all system and audit events, using structured logging.

Figure 24 shows the Azure monitoring that will be utilised for STARS.

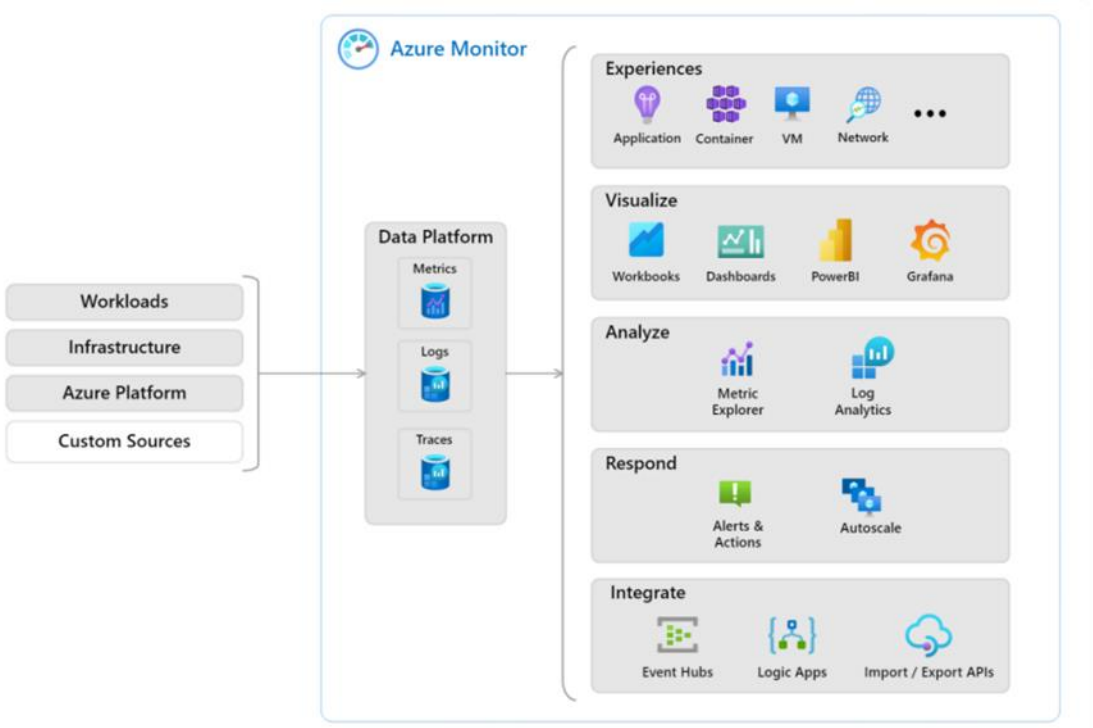


Figure 24 | Azure Monitor

8.4 Application/technology decommissioning requirements

STARS is to be deployed in a green field hosting environment. Decommissioning is out of scope.

8.5 Support

The support arrangements for STARS includes:

-
- Digital Experience and Services within DSD is responsible for the configuration, build, deployment, and resolution of any issues and defects related to this application (prior to handover to NCVER). NCVER and DEWR's role in the support of the sector during transition will be captured and agreed through the Business Operating Model work led by NCVER and agreed to be DEWR.
- For standard operations support (such as access control and licensing), existing departmental support protocols will be followed (prior to handover to NCVER).

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1205 of 1295

9 Development and hosting approach

9.1 Environments

Table 28 shows the environments that will be used for the solution.

Table 28 - Environments

Environment	Azure subscription	Description	Purpose	End-to-end connectivity
Development	landingzone-dev	Main development environment, for infrastructure and application development experimentation.	Smoke testing Development Unit testing Functional testing	Required
Feature	Landingzone-ftr	Feature environment. Hosts 'ephemeral' application instances. These instances are based on feature branches from source control, deployed via infrastructure as code (IaC), to allow for isolated testing of code branches. In this context, 'ephemeral' means that infrastructure resources to support a given feature are created and torn down on demand, via IaC automation.	Smoke testing Unit testing Functional testing	Required
Test	landingzone-tst	Main test environment. Release testing occurs here before promotion to higher environments.	Smoke testing Integration testing End-to-end testing	Required

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1206 of 1295

Environment	Azure subscription	Description	Purpose	End-to-end connectivity
			Acceptance Testing	
Early Access/Sandpit	landingzone-ear	Early Access/Sandpit environment. This environment is for external developers to verify their integrations with STARS.	Smoke testing External integration testing	Required
Pre-Production	landingzone-pre	Pre-production environment. Also used for hotfix purposes.	Smoke testing Performance testing Failover testing Penetration Testing	Required
Production	landingzone-pro	The production environment for the solution.	Hosts the production release of the solution	Required

See also the VDS - Perimeta for Azure - Solution Design[15].

9.2 Technical Debt register

To be added if required.

9.3 Proof of Concept outcomes

To be added if required.

Appendix A | Acronyms and Abbreviations

Term	Expansion
AD	Active Directory
AI	Artificial Intelligence

[15] Internal DEWR document, link will not work outside of the DEWR network.

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1207 of 1295

Term	Expansion
API	Application Programming Interface
APIM	Azure API Management
ARB	Architecture Review Board
ASQA	Australian Skills Quality Authority
ASVS	Application Security Verification Standard
B2C	Business to Consumer
CARMS	Classification and Reference Management System
CDR	Central Data Repository
CIAM	Customer Identity and Access Management
CSPM	Cloud Security Posture Management
CSRF	Cross-Site Request Forgery
DAF	Design Architecture Forum
DEWR	Department of Employment and Workplace Relations
DSD	Digital Solutions Division
EA	Enterprise Architecture
ETL	Extract, Transform, Load
FFS	Fee for Service
GRS	Geo-redundant Storage
HLD	High-Level Design
IaC	Infrastructure as Code
IDAM	Identify and Access Management
IdP	Identity Provider
IRAP	Infosec Registered Assessors Program
ISM	Information Security Manual
LDM	Lead Delivery Manager
M2M	Machine-to-Machine

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1208 of 1295

Term	Expansion
MFA	Multi-Factor Authentication
MSI	Managed Service Identities
NCVER	National Centre for Vocational Education Research
ORM	Object-relational Mapping
OWASP	Open Worldwide Application Security Project
PaaS	Platform as a Service
PSAT	Portfolio Strategy and Assurance Team
RAID	Risks, Issues, Actions and Dependencies
RBAC	Role-Based Access Control
RPO	Recovery Point Objective
RTO	Recovery Time Objective
SAML	Security Assertion Markup Language
SIEM	Security Information and Event Management
SLDM	Solution Logical Data Model
SMS	Student Management System
SMT	Solution Management Team
SOAR	Security, Orchestration, Automation and Response
SRO	Senior Responsible Officer
STA	State Training Authority
TAC	Training Accreditation Council
TLS	Transport Layer Security
TSD	Technology and Services Division
TVA	Total VET Activity
VDS	VET Data Streamlining
VET	Vocational Education and Training

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1209 of 1295

Term	Expansion
VRQA	Victorian Registration and Qualifications Authority
WoG	Whole of Government
XSS	Cross-Site Scripting
ZRS	Zone-redundant Storage

Department of Employment and Workplace Relations - Documents Released under FOI - LEX 1611 1210 of 1295

Decisions list

Data Element Formats

Unicode

Decisions

- It was agreed to maintain the current (non-unicode) support for MVP, with the idea certain fields may need to support unicode post MVP (e.g. email).
- For MVP, 'ASCII characters with the extended character set' is to be used to determine allowable characters in the API. NCVER acknowledged this may result in some characters from other languages not being supported in certain fields (e.g. email).

Student Address Format

Decisions

- Student Addresses will be collected in a fully structured format.

Date Format

Decision

- Dates will be collected in the API according to https://swagger.io/docs/specification/v3_0/data-models/data-types/ which does not necessarily align with how dates are collected through AVS.

Student Activity Date

Decision

- This data element will remain a date (not datetime) as there is no legitimate scenario in which the same Student Status will be collected for the same enrolment with the same Student Activity Date attached.

Spaces

Decision

- NCVER happy to take direction from DEWR but at this point only spaces (code 32) will be accepted as supported 'spaces'.

Data Element Classification Values

ENUMS / Classification Codes

Decisions

- Enums will be used with business friendly and meaningful values (to make the APIs more readable and self-describing).
- Enum nullability and/or default values will be a technical implementation decision for DEWR to make. Values such as 'not stated' do not require a VET IS change.
- Changes to classification values and any potential mappings from classification values to enums will be guided by DEWR.
- NCVER confirmed if enums in the API are used, they should be the same value as the Classification Value in the Business Glossary (i.e. no mapping between the enum and the classification value).
- AVETMISS codes will be removed from classification values (short names).

Missingness

Decisions

- DEWR will accommodate 'missing' classification values (i.e. not propose any additional classification values be added to the VET IS) as most probably speak to data quality.
- NCVER acknowledge there may be gaps in business rules which may require additional business rules being added and/or changes to existing business rules if no further classification values to the VET IS are added.

Indicator data elements

Decision

- Indicator data elements will only ever have two values by design.

Training Relationship Type

Decisions

- Training Relationship Type is an MVP requirement (excluding apprenticeship and traineeship classification values).

USI Exemption Reason

Decisions

- There will be classification values other than 'INDIV' but these classification values will be hidden and should not be exposed through enums.

Training Organisation Type

Decision

- This information will be collected from TGA reference data (no requirement from this data element for an organisation top level resource).

Program or Subject Identifier / Name

Decision

- The two data elements of Program or Subject Identifier (DEPR0011) and Program or Subject Name (DEPR0009) will be renamed for use in the API Open Specification.

Stakeholder Role Type Code

Decisions

- 'Delivery Provider' is for Victoria Only
- 'Employer' is meant to be removed
- 'Provider of funding' is meant to be removed

Program Type Code

Decision

- NCVER are happy for DEWR to proceed with the assumption two extra classification values of 'Jurisdictionally recognised Specialisation' and 'Nationally accredited specialisation' will be added to Program Type Code. These extra classification values would render the Specialisation Name and Specialisation Number data elements obsolete as this information would be collected under Program equivalents (Program or Subject Identifier = Specialisation Number and Program or Subject Name = Specialisation Name). This is a VET IS change.

Data Elements with both national and state based classification values

Decisions

- NCVER confirmed there are no conflicts for direct state based classification values. i.e. it will either be the state based classification value or the national classification value submitted, not both.
- It was acknowledged there may be conflicts for indirect submissions.

Organisation Type Code

Decision

- 'School' will be a national classification value under this data element and NCVER is happy for DEWR to proceed assuming STARS is only collecting schools under this data element in the context of third parties for MVP.

Data Element Locations

Third Party Providers

Decision

- Third Party Provider information will be collected under Program and Subject Enrolment, assuming we can uniquely identify third parties.
- Third Party Identifier aligns to Stakeholder Identifier Type Reference Number (DESH0032) when Stakeholder Identifier Type Code (DESH0030) is 'Training organisation identifier'
- Third Party ABN aligns to ABN (DESH0001)
- Third Party Name aligns to Stakeholder Name Type Value (DESH0034) despite the only value of Stakeholder Name Type Code (DESH0033) being 'Training Organisation'

Nominal Hours

Decisions

- Nominal Hours will be placed under Offering and is to be included for MVP.

International Student Type

Decision

- International Student Type will be placed under both Program Enrolment and Subject Enrolment.

Training Venue Type and Predominant Training Venue Type

Decisions

- Move Training Venue Type and Predominant Training Venue Type from the Delivery Location top level resource to the Subject Enrolment top level resource. This constitutes a change to the VET IS.

Vocational Education Indicator

Decision

- Vocational Education Indicator will be moved to Offering (under both program and subject offering). This is a change to the VET IS, noting certain states may treat this data element differently.

Data elements in the API contract

USI Data Attribute

Decisions

- The USI data attribute will be split into two attributes (the USI number and USI exemption) to facilitate structural validation. This is a change to the VET IS.

Date of Birth Accuracy Code

Decision

- Date of Birth Accuracy Code will appear as a new attribute in the API specification but does not constitute a change to the VET IS. This is an artefact of the API design and TOs are not being asked to submit any extra information. The SMS vendor will be responsible for supplying this information.

USI Verification

Decisions

- The USI Verification Date and USI Verification Status WA specific attributes will not be promoted to national attributes.
- The system will support USI verification technically as it is a requirement from regulators (with other parties also having a strong interest in it).

Non Reportable Data Elements

Decisions

- Noting political reasons, Reportable and Non-Reportable Data Elements will not change in the VET IS. This decision acknowledges further discussions around validation implementation need to occur with a wider group.
- Two examples discussed include:
    - Program Category Code and Program Class Code are not required to be reported for not nationally recognised programs. All elements required to be reported for not nationally recognised programs are already reportable data elements (e.g. ANZSCO, Level of Education, Field of Education).
    - Location Usage Type Code will remain not reportable as it is assumed this is a back-end system attribute which will be implemented by SMS vendors and does not need to be collected from RTOs.
- Confirmation the following data elements will remain non reportable:
    - Subject Type Code – program type code has more detailed values.
    - Program Scheduled Start Date – this is to be inferred from the first Subject scheduled Start Date in a Program.
    - Government Type Code – initially included in the model for future use but ultimately deemed unnecessary for the VET Provider collection.
- Non reportable data elements which can always be inferred do not need to be included in the API contract. All other non reportable data elements need to be included in the API contract.

Responsibility Lifecycle State Expiry Date

Decision

- This data element is not required for transport and is intended to show the end of a particular status (calculated field one day before beginning of next status).

Previous Qualifications Achieved

Decisions

- NCVER confirmed the intent of this data element is to allow only one value for each level to be reported (i.e. only one bachelor can be reported, only one diploma can be reported).
- NCVER confirmed we do not need to collect the previous qualification name, the level is enough.
- NCVER confirmed the classification value of “No” will remain for backwards compatibility to AVETMISS (the definition in the Business Glossary will change).

Training Organisation Type

Decision

- This information will be collected from TGA reference data (no requirement from this data element for an organisation top level resource).

Offering

Decision

- As at June 2025 TWG, NCVER have reviewed and are happy for this design to go up for proposal to governance, noting it isn't quite complete.

System Behaviour for MVP

Responsibility Financial Detail

Decision

- Confirming Responsibility Financial Detail Class, Category and Type codes will be included in MVP.
- For MVP, all that needs to be implemented is; if any information is received in either Financial Detail Class, Category or Type code, the information should fail validations.
- The mapping between Responsibility Financial Detail Class, Category and Type codes is only needed for stage 2 (once we start receiving funding information).
- Concessions and Waivers will not be included in MVP. They may not even be necessarily for STARS to accept post MVP.

State Specific Data Elements

Decisions

- Financial detail type code will be used to determine which state based validation rules should be applied on data submitted. In CARMS, the financial detail type code will map to a state or the commonwealth and a state, in both cases the state determines the validation rules to be applied. Prior to receiving the financial detail type code, no state specific validation rules should be run.
- For routing (who can see the data), financial detail type code is not the only attribute which determines this.

Multiple USIs for the same Student

Decisions

- The system needs to be able to accept multiple USIs for the same student and accept USI updates for the Student top level resource. The system should not lose the ability to see historical versions of Student USIs linked to enrolments.
- NCVER confirmed multiple USIs for the same student submitted at once should fail business validation rules.

Training Relationship Type

Decisions

- For state specific data elements, we can expect a 1 to many relationship between entities which are linked through training relationship type.
- The training relationship type always needs an associated identifier to specify entities which are being linked through training relationship type.
- Transferred Into represents a deliberate choice from the student and links program enrolments from the same training organisation.
- Superseded By represents a classification change and can link subject or program enrolments.

Program/Subject Enrolment Lifecycles

Decisions

- DEWR will proceed with the assumption the same student status can occur more than once for a given enrolment.

Subject Offering to Program Offering relationship

Decision

- A subject offering must be able to be related to multiple program offerings.

Delivery Location Nullability

Decisions

- Postcode is nullable.
- SuburborTownName is not nullable. Noting for an overseas delivery location this is free text and cannot be validated.

Partitioning by Organisation

Decision

- NCVER confirmed there is a requirement to partition the data under Student and Delivery Location top level resources by organisation.
    - Organisations define Delivery Locations when they setup an instance of Offering.
    - Organisations define Students as a stakeholder relationship between an Organisation and Student.

Non-structural validation messages

Decisions

- NCVER is responsible for creating non-structural validation messages. These messages will be using business glossary terminology.

Indigenous Status

Decision

- For MVP, NCVER is okay with the API only collecting a single value (no child records for indigenous status). Note this could be expanded in future as NCVER have had questions from students about why their indigenous status to their home country is not included.

Student API Detailed Design

1. Summary

This document provides the technical design for the Student increment APIs.

It provides information on

- The data entities for Student interactions – this includes fields, entity relationships, and information to perform a technical validation on the entities to determine if the message is an acceptable format
- The API endpoints, request objects and response objects for Asynchronous interactions
- Invocation of the rules engine
- Persistence of the response to the staging datastore (Cosmos DB)
- Handling of errors and return of appropriate error codes and responses

2. Data Entities

This section defines the structure of the entities in the request/response objects.

The Min & Max length should reflect that required by the master database. Required indicates whether the field is required for the API. Mandatory fields are normally defined by business rules.

3. Valid Characters

The default character set is UTF-8. Only characters in the range 0x20 to 0x7e inclusive are permitted in payloads.

The double quote character " and the backslash \ character must be escaped when used within strings according to JSON standards to ensure that a payload is valid. A payload which contains invalid JSON will be rejected.

4. Request Payload Field value Truncation

In accordance with AVETMISS and State standards, behavioural validations truncate field values when received values are longer than the lengths indicated in this document. This is done prior to saving payloads to the Cosmos database.

Truncation operates by:

- Trimming trailing spaces from the received string value of a field
- Truncating if the character length of the value is more than the documented length and Can Be Truncated? Is set to Y in the user story behaviour rules table
- Trimming again to remove any trailing spaces

5. Initial Payload Validations

Validation Type	Value
NULL	Returns 400 BadRequest if payload is null
Invalid Payload	Returns 400 BadRequest if payload cannot be deserialized

Verify Identifier (URL/Payload)	Returns 400 BadRequest if supplied identifiers in URL don't match with identifiers supplied in body payload.
Not Found	Returns 404 NotFound – If requested entity was not found.

6. Response Codes

200 – OK

A 200 OK response will be returned for the GET requests if validation is complete and the entity has been mastered. The response will include a payload with the requested entity retrieved from the database, along with any validation details. See the respective endpoint responses for detailed information.

202 – Accepted

A 202 accepted response will be returned for the following requests:

1. POST the payload and after the payload passes the initial validation
2. DELETE, if it's a valid request
3. GET the validation status, while the validation process hasn't been completed

The response will include a payload with the following format.

```
{
    "validationId": "24924aae-b968-4086-a9bd-d7a7b9bd55e0",
    "validationTimestamp": "2020-08-13T02:34:53.251379Z",
    "status": "Validating"
}
```

400 – Bad Request

```
Response body
{
  "errors": [
    {
      "id": "0d014999-c408-48fc-8044-95c4264781a3",
      "detail": "Request payload must not be NULL",
      "code": "400"
    }
  ]
}
```

404 – Not Found

```
Response body
{
  "errors": [
    {
      "id": "71780af2-2a4b-4efb-b4e2-149d519232ac",
      "detail": "Requested entity was not found",
```

```
      "code": "404"
    }
  ]
}
```

413 – Request Entity Too Large

A 413 Request Entity Too Large response will be returned if any single request entity is too large to be published to the Azure Service Bus queue.

```
Response Body:
{
  "errors": [
    {
      "id": "0709427b-ae2d-48ad-b74c-49695c770392",
      "detail": "One or more records in the payload exceeded the acceptable maximum size",
      "code": "413"
    }
  ]
}
```

422 – Unprocessable Entity (Client error)

A 422 Unprocessable Entity response will be returned if payload is serialized but there are validation error(s).

```
},
        "validationResults": [
          {
            "ruleSet": "Basic Validations",
            "errorCount": 2,
            "warningCount": 0,
            "infoCount": 0
          },
          {
            "ruleSet": "StudentAVETMISSFieldValidations",
            "errorCount": 3,
            "warningCount": 1,
            "infoCount": 0
          }
        ],
        "errors": [
          {
            "code": "Required",
            "ruleSet": "Basic Validations",
            "source": "Student.ClientId",
            "message": "ClientId must not be blank, please correct and re-
submit."
          },
          {
            "code": "Required",
            "ruleSet": "Basic Validations",
```

```
            "source": "Student.FamilyName",
            "message": "FamilyName must not be blank, please correct and re-
submit."
          },
```

405 – Method Not Allowed

If the “latest” parameter is not true on the Get latest single student validation endpoint, return a 405 (Method Not Allowed) response.

```
{
    "errors": [
      {
        "id": "6930a9bc-6dba-4996-8c32-acf3b2db02cc",
        "detail": "An invalid URL has been specified",
        "code": "405"
      }
    ]
  }
```

500 - Internal Server Error

```
{
    "errors": [
      {
        "id": "3e5fa897-51f3-4639-8ecb-b8958470b6ce",
        "detail": "Unexpected Error",
        "code": "500"
      }
    ]
  }
```

The response will also contain additional header values:

- Location - A URL the client should poll for a response status
- Retry-After – A timespan that indicates when the client should check back for the status. This is currently set to 60 seconds.

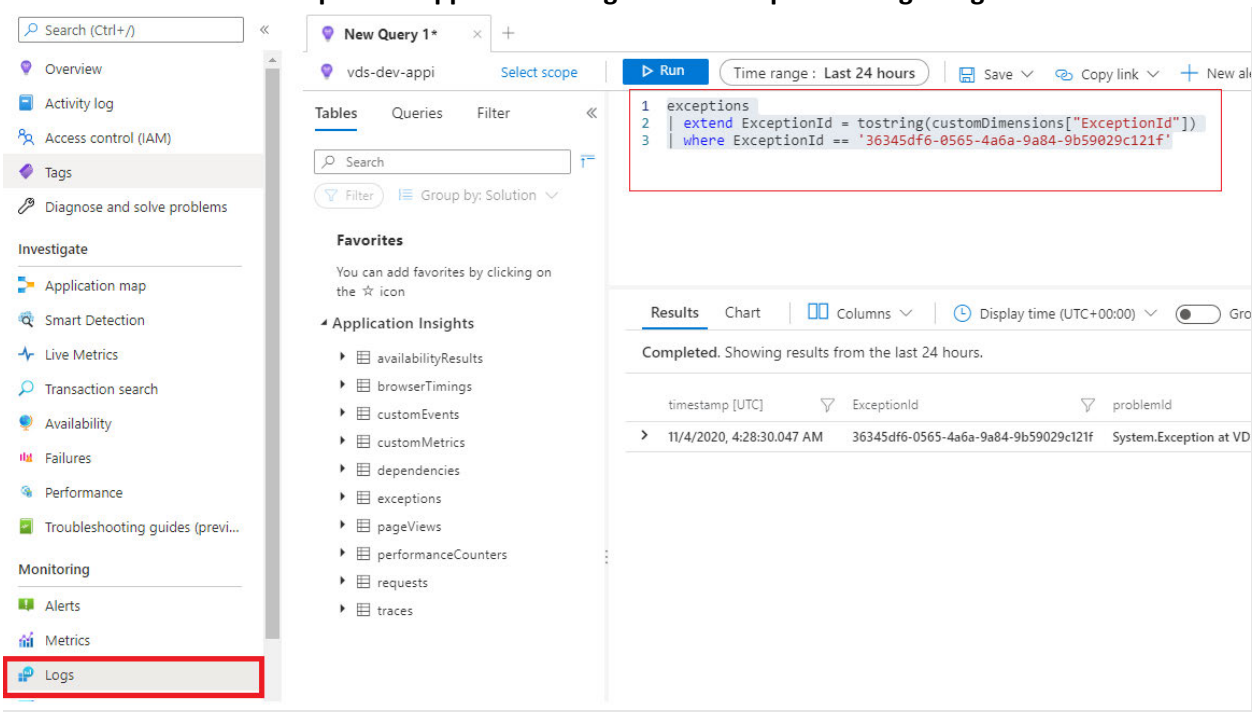
```
content-length: 1762
content-type: application/json; charset=utf-8
date: Fri, 14 Aug 2020 01:51:54 GMT
location: http://vds-dev-
sf.australiacentral.cloudapp.azure.com:8484/api/v1/students/validations/4ab4bef0-
2279-4b1c-b805-4b4df65d2987
retry-after: 60
server: Kestrel
```

7. Error Response Fields

Element	Content

validationId	A unique value which identifies a request/response combination
validationTimestamp	A timestamp of when the request was processed
validationResults	An array which lists the different rulesets which were applied to the request. The elements within each item provide counts of the errors, warnings and info messages which occurred under the ruleset
errors	An array which lists every error which resulted from validating a request. The elements within each item identify the specific error code, the name of the rule and a human-readable message. Source is the name of the field which generated the errors. This is in JSON notation. This element may be an empty array if there are no errors
warnings	An array which lists every warning which resulted from validating a request. The elements within each item identify the specific warning code, the name of the rule and a human-readable message. Source is the name of the field in the payload in JSON notation which generated the warning. This is in JSON notation. This element may be an empty array if there are no warnings
infos	An array which lists every information message which resulted from validating a request. The elements within each item identify the specific information code, the name of the rule and a human-readable message. Source is the name of the field in the payload which generated the information message. This is in JSON notation. This element may be an empty array if there are no information messages
<entityname>	The original request payload (**Student**)

How to search for an exception in application insights? All exceptions are getting tracked now.



Query to search for an exception

```
exceptions

| extend ExceptionId = tostring(customDimensions["ExceptionId"])

| where ExceptionId == '36345df6-0565-4a6a-9a84-9b59029c121f'
```

8. Architecture overview

The different elements of the architecture in this increment are:

1. **Message Router** – its role is:
    1. Receive a message from the Service Bus queue
    2. Determine which mastering microservice the message needs to be routed to
    3. Send the message to the relevant microservice and wait for the response
    4. Complete the message (Router is now free to take the next message from the queue)
2. **Mastering Microservice** – its role is:
    1. Receive a message from the Message Router
    2. Validate the entity in the message (Behavioral validations)
    3. Call inRule for BRMS validation
    4. Master the entity (if CanBeMastered = true from inRule)

The new services required for Student are:

- **VDS.Student.Microservice** – This service will include two ServiceFabric projects and two WebApi projects (one for Student endpoints and one for Student Mastering endpoints)
- **VDS.MessageRouter.Service** – This is a global service (required for all increments), not a microservice. It will be extended to support the messages types in later increments (e.g. unit-enrolment etc.).

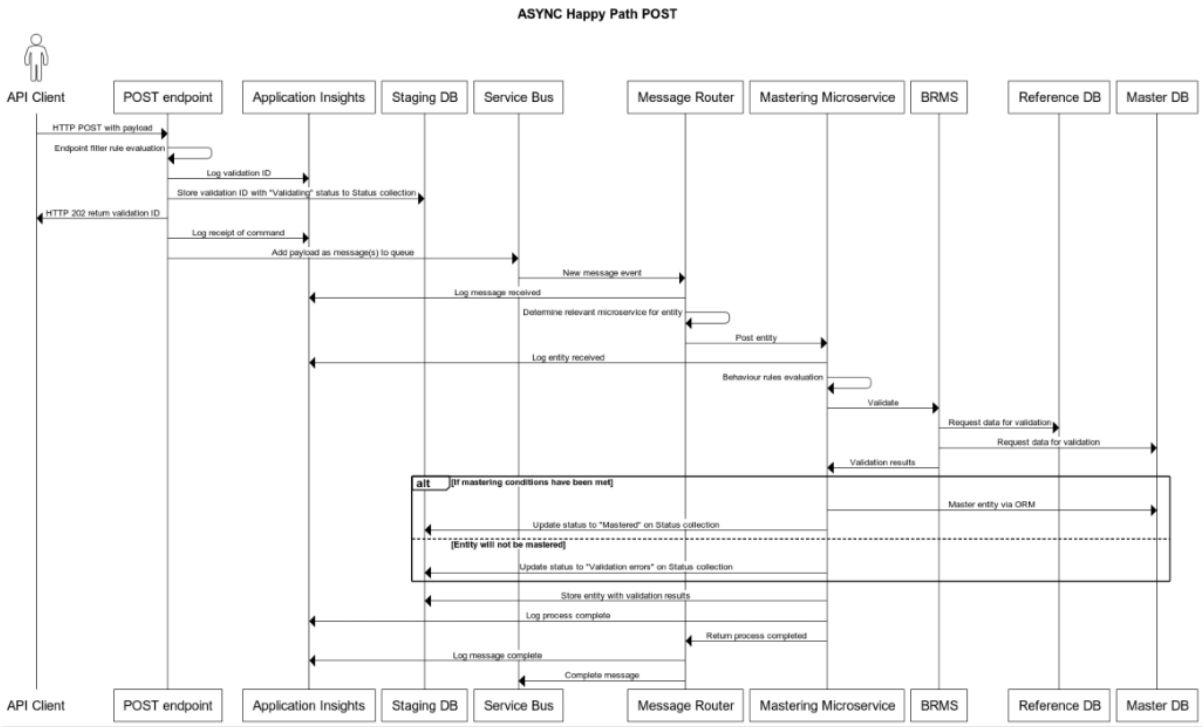


Figure 1 Sequence diagram that displays the process path of a post request from the API endpoint until mastering in the SQL Server database.

9. Get request by validation ID

A new Cosmos DB document collection – Validation Status – has been created to support payloads with collections.

Without the use of the transaction scope feature, each entity in a collection that gets to the queue will go to the mastering evaluation and may be mastered. This can happen when not all of the entities in a payload get on the queue. To ameliorate this problem, we will follow the rules below.

- Validation Status is set to **Validating** initially;
- If any entities fail to be published to the queue due to a large message size, we set the Validation Status for the entire request/payload to **Message Size Exceeded**;
- If any entities fail to be published to the queue for any other reason, we set the Validation Status for the entire request/payload to **Critical Failure**;
- If all entities are mastered, we change the Validation Status to **Completed**;
- If any of the entities cannot be mastered because there is a business rules validation error, the status changes to **CompletedNotMastered**;
- In case of any mastering failure, the status changes to **CompletedNotMastered**

The sequence diagram below shows how a get request will respond depending on the current status of the payload's Validation Status document.

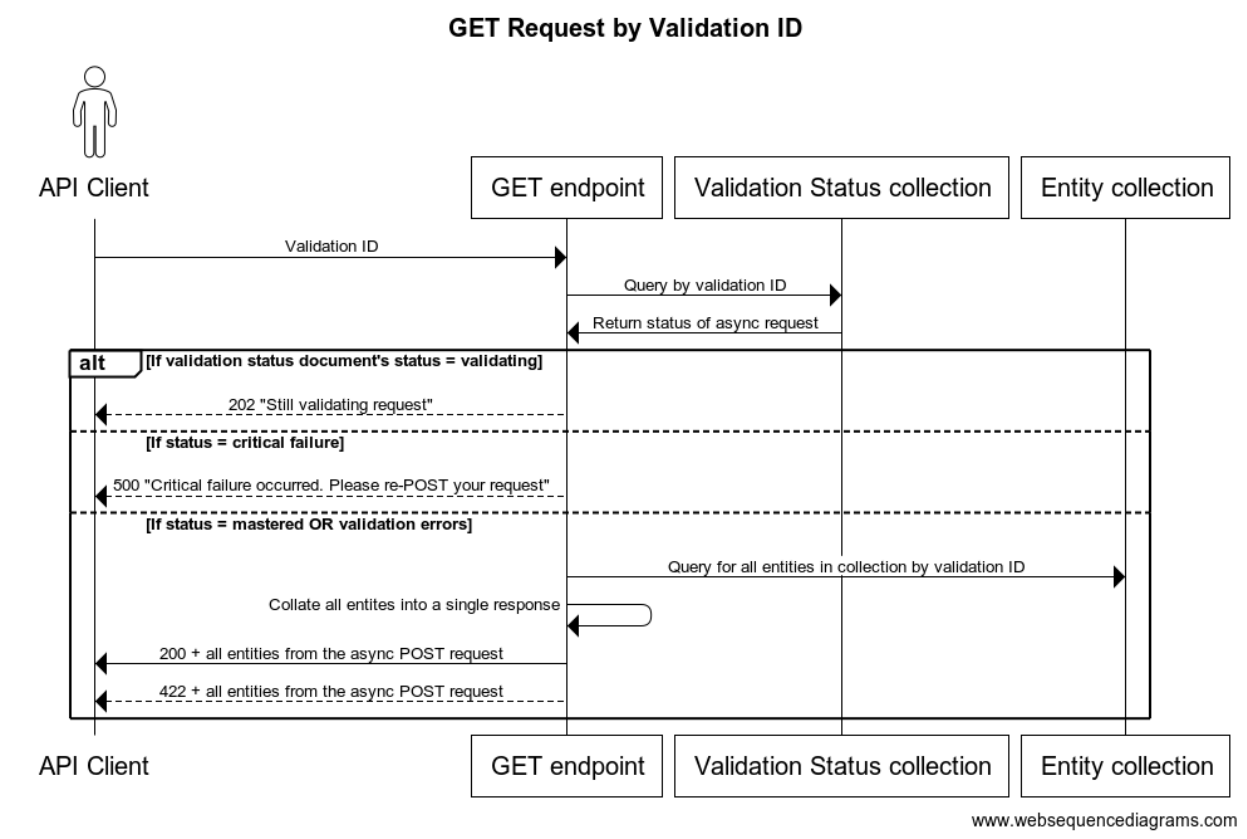


Figure 2 Get Request by Validation ID

The Status field in the Validation Status document will have the following possible values:

- **Validating** – Asynchronous process still underway. Returns 202 Accepted while in this state.
- **Critical Failure** – Post request failed. Client will need to re-POST the request. Returns 500 Unexpected Error.
- **Message Size Exceeded** - Post request failed; one or more entities exceed the acceptable size limit. Client will need to truncate and re-POST the request. Returns 413 Request Entity Too Large.
- **Completed** – Process complete. All entities have been mastered. Returns 200 OK, along with the mastered entity.
- **CompletedNotMastered** – Process complete. One or more entities have failed to be mastered. Returns 422 Unprocessable entity or 200 OK

In case you receive status CompletedNotMastered and exceptionId has a GUID value, that means there was a mastering failure and exception has been logged in Telemetry. The exceptionId can be used to check the exact error in Application Insights.

200

Response body


```
        "errorCount": 0,
        "warningCount": 0,
        "infoCount": 0
      },
      {
        "ruleSet": "ContactAddressFieldValidations",
        "errorCount": 0,
        "warningCount": 0,
        "infoCount": 0
      }
    ],
    "errors": [],
    "warnings": [],
    "infos": [
      {
        "code": "ST0041",
        "ruleSet": "StudentAVETMISSFieldValidations",
        "source": "ClientId",
        "message": "From the data supplied, this Student at your Organis
      }
    ],
    "exceptionId": "bfb4ece0-b576-4a81-ad33-54f277256e9f",
    "validationId": "a0e1e4b8-a13d-4074-a99e-cf9faf031cf9",
    "validationTimestamp": "2021-01-14T23:27:29.1058608Z",
    "status": "CompletedNotMastered"
  }
```


10. Get request by entity ID

The GET endpoints that request an entity by its ID will have a different process. The process is described in the sequence diagram below.

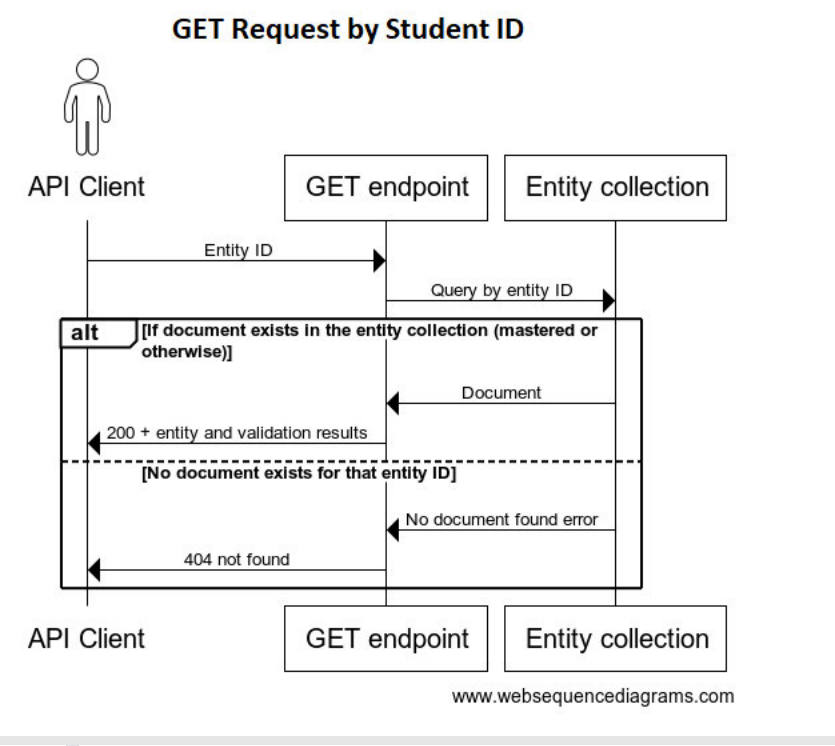


Figure 3 GET Request by Student ID

11. Delete endpoint

The Master DB implements soft-delete. i.e., entities have an IsDeletedFlag on the database. We need to implement an async delete endpoint that will permit the SMS to change this field from false to true. We will implement this by an upsert process from the level of Service Bus onwards. At the API endpoint level, we will provide a HTTP delete command that expects the client ID of the entity.

Endpoint looks like:

v1/students/{clientId}

The method that implements delete command creates a new instance of the entity (rather than converting the payload into the entity) and sets the provider ID (received via the header), client ID and the IsDeletedFlag. For example, an entity, as JSON, will look similar to the below.

```
{
    "ProviderId": "Provider1",
    "ClientId": "Client1",
    "IsDeletedFlag": true
}
```

To uniquely identify a student, we need to combine the client ID with the provider ID. The provider ID is used as the session ID on Service Bus. The session ID could be used to find the record in the master DB.

For clarity, however, we will set the ProviderId in the entity before queuing the message onto the Service Bus.

All other fields will be null (and ignored by the mastering service if they're not). Mastering will check the IsDeleteFlag and update that field in the DB if it's set to true. Cosmos DB: persist entity into a document in the usual way.

12. Student mastering microservice

The role of the mastering microservice is:

1. Receive a message from the Message Router
2. Deserialise a message into an entity
3. Validate the entity

This process involves:

- Executing .NET based behavior rules that
- Trim any leading and trailing whitespace
- Truncate fields based on maximum length rules
- Check if values exist for required fields
- Check the length of fields
- Check that a non-truncatable field is within its length limits

4. Calling the BRMS to validate a set of rulesets as specified by national and state bodies

This process involves:

- Checking the field CanBeMastered
- If true, perform a .NET based ORM process to transform the entity into the relational model of the entity in the SQL Server database

5. Finally, executing a BRMS ruleset that determines if an entity can be mastered (field name = CanBeMastered)
6. Master the entity
7. Save the entity to the staging database
8. Set the status in ValidationStatus

13. Student (Single): Endpoints and HTTP return codes

Http Verb or end point	Header Parameter	Http status codes
POST (Create / Validate) e.g. Validate Single Student POST /v1/students/{clientId}/validate	Training Organisation Identifier	202 - Accepted 400 - Cannot desterilize the payload Or Identifiers are not matched 403 - Forbidden (to be implemented once security design completed)
GET (Get Validation by Validation ID) e.g. Get Single/Batch Student Validation GET v1/students/validations/{validationId}	Training Organisation Identifier	200 - OK (if the validation has occurred) 202 – Accepted (if the validation hasn't occurred yet) 400 - If entity identifier(s) was not supplied or not able to parse/bind 404 - Not Found – if the validation request doesn't exist 403 - Forbidden (to be implemented once security design completed)

		413 – RequestEntity TooLarge – If Service Bus is unable to publish due to large message size 422 - Unprocessable entity – If there is any validation error
GET (Get Validation by Student ID) e.g. Get latest Single Student Validation GET v1/students/{clientId}/validations?latest=true	Training Organisation Identifier	200 - OK (if the validation has occurred) 405 – Not Allowed (method is not implemented for this resource) – if "false" is passed for latest flag 400 - If entity identifier(s) was not supplied or not able to parse/bind 404 - Not Found – if the validation request doesn't exist 403 - Forbidden (to be implemented once security design completed) 422 - Unprocessable entity – If there is any validation error
DELETE (Delete ID) e.g. Delete Single Student v1/students/{clientId}	Training Organisation Identifier	202 - Accepted 400 - If entity identifier(s) was not supplied or not able to parse/bind 403 - Forbidden (to be implemented once security design completed) 404 – Not found (if ID does not exist in the database - ST0088 retuned from InRule)

Training Organisation Identifier – Header parameter

Training Organisation Identifier	providerId	varchar (10)	Alphanumeric	N	N	10	1

Student request payload model

US Field Name	Payload Field Name	Length	Data Type	Nullable?	Variable Length String Can be truncated ?	Max Length	Min Length
Client Identifier	clientId	varchar(10)	Alphanumeric	N	N	10	1
Student Title	title	varchar(4)	Alphanumeric	Y	N	4	1
Student Family Name	familyName	varchar(40)	Alphanumeric	N	N	40	1
Student Given Name First	givenNameFirst	varchar(40)	Alphanumeric	Y	N	40	1
Student Given Name Other	givenNameOther	varchar(40)	Alphanumeric	Y	N	40	1
Highest School Level Completed Identifier (L)	highestSchoolLevelCompletedId	varchar(2)	Alphanumeric	N	N	2	2

Gender (L)	gender	varchar(1)	Alphanumeric	N	N	1	1
Date of Birth	dateOfBirth	Date (dd/mm/yyyy)	Date	N	N	10	10
Labour Force Status Identifier (L)	labourForceStatusId	varchar(2)	Alphanumeric	N	N	2	2
Prior Educational Achievement Identifier	priorEducationalAchievementIds	List of Prior Educational Achievement Identifiers					
At School Flag	atSchoolFlag	varchar(1)	Alphanumeric	Y	N	1	1
Unique Student Identifier	uniqueStudentId	varchar(10)	Alphanumeric	Y	N	10	10
Survey Contact Status (L)	surveyContactStatus	varchar(1)	Alphanumeric	Y	N	1	1
	residentialAddress	Residential Address					
	contactAddress	Contact Address					
Contact Telephone number [home]	contactHomePhoneNumber	varchar(22)	Numeric	Y	Y	22	1
Contact Telephone number [work]	contactWorkPhoneNumber	varchar(22)	Numeric	Y	Y	22	1
Contact Telephone number [mobile]	contactMobilePhoneNumber	varchar(22)	Numeric	Y	Y	22	1
Contact Email address	contactEmail	varchar(80)	Alphanumeric	Y	Y	80	1
Contact Email address [alternative]	contactAlternativeEmail	varchar(80)	Alphanumeric	Y	Y	80	1
Deceased	deceased	varchar(1)	Boolean	Y	N	1	1
Qual Since 17	qualificationSince17	varchar(1)	Numeric	Y	N	1	1
Highest Post School Qual	highestPostSchoolQualification	varchar(1)	Numeric	Y	N	1	1
State or Territory Currently Living In	stateCurrentlyLivingIn	varchar(2)	Alphanumeric	Y	N	2	2
Residency Status	residencyStatus	varchar(1)	Numeric	Y	N	1	1

State or Territory Currently Working In	stateCurrentlyWorkingIn	varchar(2)	Alphanumeric	Y	N	2	2
Employers Organisation Name	employerOrganisationName	varchar(500)	Alphanumeric	Y	Y	500	1
Employers Organisation Postcode	employerOrganisationPostcode	varchar(4)	Alphanumeric	Y	N	4	4
Employers Organisation Suburb	employerOrganisationSuburb	varchar(50)	Alphanumeric	Y	N	50	1
Employers Organisation Street Number	employerOrganisationStreetNumber	varchar(15)	Alphanumeric	Y	Y	15	1
Employers Organisation Street Name	employerOrganisationStreetName	varchar(70)	Alphanumeric	Y	Y	70	1

Prior Educational Achievement Identifiers

US Field Name	Payload Field Name	Length	Data Type	Nullable?	Variable Length String Can be truncated?	Max Length	Min Length
Prior Educational Achievement Identifier	achievementId	varchar(3)	Numeric	N	N	3	3

Residential Address

US Field Name	Payload Field Name	Length	Data Type	Nullable?	Variable Length String Can be truncated?	Max Length	Min Length
Residential Country Identifier (L)	countryId	varchar(4)	Alphanumeric	N	N	4	4
Residential Postcode	postcode	varchar(4)	Alphanumeric	N	N	4	4
Residential State Identifier (L)	stateId	varchar(4)	Alphanumeric	N	N	2	2
Residential Address Suburb Locality or Town	suburb	varchar(50)	Alphanumeric	Y	Y	50	1
Residential Address Building / Property Name	buildingName	varchar(50)	Alphanumeric	Y	Y	50	1
Residential Address Flat / Unit Details	unitNumber	varchar(30)	Alphanumeric	Y	Y	30	1

Residential Address Street Number	streetNumber	varchar(15)	Alphanumeric	Y	Y	15	1
Residential Address Street Name	streetName	varchar(70)	Alphanumeric	Y	Y	70	1

Contact Address

US Field Name	Payload Field Name	Length	Data Type	Nullable?	Variable Length String Can be truncated?	Max Length	Min Length
Contact Address Postal Delivery Box	postalDeliveryBox	varchar(22)	Alphanumeric	Y	Y	22	1
Contact Postcode	postcode	varchar(4)	Alphanumeric	N	N	4	4
Contact State Identifier (L)	stateId	varchar(4)	Alphanumeric	N	N	2	2
Contact Address Suburb Locality or Town	suburb	varchar(50)	Alphanumeric	N	Y	50	1
Contact Address Building / Property Name	buildingName	varchar(50)	Alphanumeric	Y	Y	50	1
Contact Address Flat / Unit Details	unitNumber	varchar(30)	Alphanumeric	Y	Y	30	1
Contact Address Street Number	streetNumber	varchar(15)	Alphanumeric	Y	Y	15	1
Contact Address Street Name	streetName	varchar(70)	Alphanumeric	Y	Y	70	1

Nullability = Y/N
Variable Length String (Can be truncated) = Y/N
Max = Maximum length of value
Min = Minimum length of value
(L) = Lookup

Cells with a green background are extra optional fields only required by NDE.

Allowed Values

US Field Name	Payload Field Name	Values Allowed/Not Allowed

Highest School Level Completed Identifier (L)	highestSchoolLevelCompletedId	ıavioral rules will support this value (Numeric check ignored for value): '@@' Not Specified
Date of Birth	dateOfBirth	Behavioral rules will support this value (Date check ignored for this value): • a valid month and year with day '@' filled (e.g. @@/12/1984), or • a valid year with day and month '@' filled (e.g. @@/@@/1984), or • completely '@' filled where the client does not specify their birth date (e.g. @@/@@/@@@@).
Labour Force Status Identifier (L)	labourForceStatusId	ıavioral rules will support this value (Numeric check ignored for value): '@@' - Not Specified
At School Flag	atSchoolFlag	Behavioral rules will support '@'

Request Payload – Student POST Single


```
{
  "Student": {
      "ProviderId": "91451",
      "ClientId": "91451-USI",
      "Title": "QB",
      "FamilyName": "Karlikova",
      "GivenNameFirst": "Edwardo",
      "GivenNameOther": "D",
      "HighestSchoolLevelCompletedId": "12",
      "Gender": "M",
      "DateOfBirth": "26/05/1968",
      "LabourForceStatusId": "@@",
      "PriorEducationalAchievementIds": [{
          "AchievementId": "008"
      }, {
          "AchievementId": "410"
      }],
      "AtSchoolFlag": "N",
      "UniqueStudentId": "ZU33XZDZXU",
      "SurveyContactStatus": "A",
      "ResidentialAddress": {
          "CountryId": "1101",
          "UnitNumber": "60",
          "BuildingName": "Schrute Farms",
          "StreetNumber": "60",
```

```
            "StreetName": "Margaret St",
            "Suburb": "Sydney",
            "StateId": "01",
            "Postcode": "2000"
        },
        "ContactAddress": {
            "PostalDeliveryBox": "Contact Postal Box",
            "UnitNumber": "10",
            "BuildingName": "Sears Center",
            "StreetNumber": "10",
            "StreetName": "George St",
            "Suburb": "Sydney",
            "StateId": "01",
            "Postcode": "2000"
        },
        "ContactHomePhoneNumber": "0294391234",
        "ContactWorkPhoneNumber": "0294397910",
        "ContactMobilePhoneNumber": "0470123456",
        "ContactEmail": "abc@Rxp.com",
        "ContactAlternativeEmail": "abc@Rxp.com",
        "Deceased": "0",
        "QualificationSince17": "2",
        "HighestPostSchoolQualification": "5",
        "StateCurrentlyLivingIn": "01",
        "ResidencyStatus": "1",
        "StateCurrentlyWorkingIn": "01",
        "EmployerOrganisationName": "Sample Employer Name",
        "EmployerOrganisationPostcode": "2155",
        "EmployerOrganisationSuburb": "Kellyville",
        "EmployerOrganisationStreetNumber": "123",
        "EmployerOrganisationStreetName": "Sample Street"
    }
}
```


Response Payload – Student POST Single


```
{
    "validationId": "67dd1fbc-5073-41f3-a638-ea902bf411c9",
    "validationTimestamp": "2020-08-24T05:49:03.3669986Z",
    "status": "Validating"
}
```


Response Student GET by Validation ID – Validating


```
{
```

```
    "validationId": "67dd1fbc-5073-41f3-a638-ea902bf411c9",
    "validationTimestamp": "2020-08-24T05:49:03.3669986Z",
    "status": "Validating"
}
```


Response Student GET by Validation ID – Pending Delete


```
{
    "validationId": "26124a87-fa05-479f-b106-c394fc7dcfd1",
    "validationTimestamp": "2020-08-24T05:54:20.261139Z",
    "status": "Pending Delete"
}
```


Response Student GET by Validation ID – Completed


```
{
    "students": [
      {
        "student": {
          "providerId": "91451",
          "clientId": "91451-USI",
          "title": "QB",
          "familyName": "Karlikova",
          "givenNameFirst": "Edwardo",
          "givenNameOther": "D",
          "highestSchoolLevelCompletedId": "12",
          "gender": "M",
          "dateOfBirth": "26/05/1968",
          "labourForceStatusId": "@@",
          "priorEducationalAchievementIds": [
            {
              "achievementId": "008"
            },
            {
              "achievementId": "410"
            }
          ],
          "atSchoolFlag": "N",
          "uniqueStudentId": "ZU33XZDZXU",
          "surveyContactStatus": "A",
          "residentialAddress": {
            "countryId": "1101",
            "unitNumber": "60",
            "buildingName": "Schrute Farms",
            "streetNumber": "60",
            "streetName": "Margaret St",
            "suburb": "Sydney",
```

```
      "stateId": "01",
      "postcode": "2000"
    },
    "contactAddress": {
      "postalDeliveryBox": "Contact Postal Box",
      "unitNumber": "10",
      "buildingName": "Sears Center",
      "streetNumber": "10",
      "streetName": "George St",
      "suburb": "Sydney",
      "stateId": "01",
      "postcode": "2000"
    },
    "contactHomePhoneNumber": "0294391234",
    "contactWorkPhoneNumber": "0294397910",
    "contactMobilePhoneNumber": "0470123456",
    "contactEmail": "abc@Rxp.com",
    "contactAlternativeEmail": "abc@Rxp.com",
    "deceased": "0",
    "qualificationSince17": "2",
    "highestPostSchoolQualification": "5",
    "stateCurrentlyLivingIn": "01",
    "residencyStatus": "1",
    "stateCurrentlyWorkingIn": "01",
    "employerOrganisationName": "Sample Employer Name",
    "employerOrganisationPostcode": "2155",
    "employerOrganisationSuburb": "Kellyville",
    "employerOrganisationStreetNumber": "123",
    "employerOrganisationStreetName": "Sample Street"
  },
  "validationResults": [
    {
      "ruleSet": "Basic Validations",
      "errorCount": 0,
      "warningCount": 0,
      "infoCount": 0
    },
    {
      "ruleSet": "USI Validation",
      "errorCount": 0,
      "warningCount": 0,
      "infoCount": 0
    },
    {
      "ruleSet": "StudentAVETMISSFieldValidations",
```

```
            "errorCount": 0,
            "warningCount": 0,
            "infoCount": 1
          },
          {
            "ruleSet": "ResidentialAddressFieldValidations",
            "errorCount": 0,
            "warningCount": 0,
            "infoCount": 0
          },
          {
            "ruleSet": "ContactAddressFieldValidations",
            "errorCount": 0,
            "warningCount": 0,
            "infoCount": 0
          },
          {
            "ruleSet": "StudentNDEFieldValidations",
            "errorCount": 0,
            "warningCount": 0,
            "infoCount": 0
          }
        ],
        "errors": [],
        "warnings": [],
        "infos": [
          {
            "code": "ST0041",
            "ruleSet": "StudentAVETMISSFieldValidations",
            "source": "ClientId",
            "message": "From the data supplied, this Student at your Organisation
 already exists within the ePoC. Data supplied will be treated as an update to th
at record"
          }
        ],
        "exceptionId": null,
        "validationId": "feb841d7-7a63-4313-9950-dfb1fd68df6a",
        "validationTimestamp": "2021-02-16T05:57:03.0161591Z",
        "status": "Mastered"
      }
    ]
  }
```

Response Student GET by Validation ID – Critical Failure


```
{
      "errors": [
        {
          "id": "6ee91a5c-0d14-4a03-aa66-5eb974062d39",
          "detail": "Unexpected Error",
          "code": "500"
        }
      ]
}
```


Response Student GET by Client ID


```
{
    "student": {
      "providerId": "91451",
      "clientId": "91451-USI",
      "title": "QB",
      "familyName": "Karlikova",
      "givenNameFirst": "Edwardo",
      "givenNameOther": "D",
      "highestSchoolLevelCompletedId": "12",
      "gender": "M",
      "dateOfBirth": "26/05/1968",
      "labourForceStatusId": "@@",
      "priorEducationalAchievementIds": [
        {
          "achievementId": "008"
        },
        {
          "achievementId": "410"
        }
      ],
      "atSchoolFlag": "N",
      "uniqueStudentId": "ZU33XZDZXU",
      "surveyContactStatus": "A",
      "residentialAddress": {
        "countryId": "1101",
        "unitNumber": "60",
        "buildingName": "Schrute Farms",
        "streetNumber": "60",
        "streetName": "Margaret St",
        "suburb": "Sydney",
        "stateId": "01",
        "postcode": "2000"
      },
```

```
    "contactAddress": {
      "postalDeliveryBox": "Contact Postal Box",
      "unitNumber": "10",
      "buildingName": "Sears Center",
      "streetNumber": "10",
      "streetName": "George St",
      "suburb": "Sydney",
      "stateId": "01",
      "postcode": "2000"
    },
    "contactHomePhoneNumber": "0294391234",
    "contactWorkPhoneNumber": "0294397910",
    "contactMobilePhoneNumber": "0470123456",
    "contactEmail": "abc@Rxp.com",
    "contactAlternativeEmail": "abc@Rxp.com",
    "deceased": "0",
    "qualificationSince17": "2",
    "highestPostSchoolQualification": "5",
    "stateCurrentlyLivingIn": "01",
    "residencyStatus": "1",
    "stateCurrentlyWorkingIn": "01",
    "employerOrganisationName": "Sample Employer Name",
    "employerOrganisationPostcode": "2155",
    "employerOrganisationSuburb": "Kellyville",
    "employerOrganisationStreetNumber": "123",
    "employerOrganisationStreetName": "Sample Street"
  },
  "validationResults": [
    {
      "ruleSet": "Basic Validations",
      "errorCount": 0,
      "warningCount": 0,
      "infoCount": 0
    },
    {
      "ruleSet": "USI Validation",
      "errorCount": 0,
      "warningCount": 0,
      "infoCount": 0
    },
    {
      "ruleSet": "StudentAVETMISSFieldValidations",
      "errorCount": 0,
      "warningCount": 0,
      "infoCount": 1
```

```
      },
      {
        "ruleSet": "ResidentialAddressFieldValidations",
        "errorCount": 0,
        "warningCount": 0,
        "infoCount": 0
      },
      {
        "ruleSet": "ContactAddressFieldValidations",
        "errorCount": 0,
        "warningCount": 0,
        "infoCount": 0
      },
      {
        "ruleSet": "StudentNDEFieldValidations",
        "errorCount": 0,
        "warningCount": 0,
        "infoCount": 0
      }
    ],
    "errors": [],
    "warnings": [],
    "infos": [
      {
        "code": "ST0041",
        "ruleSet": "StudentAVETMISSFieldValidations",
        "source": "ClientId",
        "message": "From the data supplied, this Student at your Organisation alr
eady exists within the ePoC. Data supplied will be treated as an update to that r
ecord"
      }
    ],
    "exceptionId": null,
    "validationId": "feb841d7-7a63-4313-9950-dfb1fd68df6a",
    "validationTimestamp": "2021-02-16T05:57:03.0161591Z",
    "status": "Mastered"
  }
```


Response Student DELETE


```
{
    "validationId": "26124a87-fa05-479f-b106-c394fc7dcfd1",
    "validationTimestamp": "2020-08-24T05:54:20.261139Z",
    "status": "Validating"
}
```

14. Student SEE Data: Endpoints and HTTP return codes

Http Verb or end point	Header Parameter	Http status codes
POST (Create / Validate) e.g. Validate Student SEE Assessment POST /v1/students/{clientId}/seedata/validate	Training Organisation Identifier	202 - Accepted 400 - Cannot desterilize the payload Or Identifiers are not matched 403 - Forbidden (to be implemented once security design completed)
GET (Get Validation by Validation ID) e.g. Get Student SEE Assessment Validation GET v1/students/seedata/validations/{validationId}	Training Organisation Identifier	200 - OK (if the validation has occurred) 202 – Accepted (if the validation hasn't occurred yet) 400 - If entity identifier(s) was not supplied or not able to parse/bind 404 - Not Found – if the validation request doesn't exist 403 - Forbidden (to be implemented once security design completed) 413 – RequestEntity TooLarge – If Service Bus is unable to publish due to large message size 422 - Unprocessable entity – If there is any validation error
GET (Get Validation by Client Id) e.g. Get latest Student SEE Assessment Validation GET v1/students/{clientId}/seedata/validations?latest=true	Training Organisation Identifier	200 - OK (if the validation has occurred) 405 – Not Allowed (method is not implemented for this resource) – if "false" is passed for latest flag 400 - If entity identifier(s) was not supplied or not able to parse/bind 404 - Not Found – if the validation request doesn't exist 403 - Forbidden (to be implemented once security design completed) 422 - Unprocessable entity – If there is any validation error
DELETE (Delete ID) e.g. Delete SEE Assessment v1/students/{clientId}/seedata/{assessmentId}	Training Organisation Identifier	202 - Accepted 400 - If entity identifier(s) was not supplied or not able to parse/bind 403 - Forbidden (to be implemented once security design completed) 404 – Not found (if ID does not exist in the database - ST0088 retuned from InRule)

Training Organisation Identifier – Header parameter

Training Organisation Identifier	providerId	varchar (10)	Alphanumeric	N	N	10	1

Student SEE Data request payload model

US Field Name	Payload Field Name	Length	Data Type	Nullable?	Variable Length String Can be truncated?	Max Length	Min Length
Unique Student Identifier	uniqueStudentId	Varchar(10)	Alphanumeric	N	N	10	10
Client Identifier	clientId	Varchar(10)	Alphanumeric	N	N	10	1
Assessment Identifier	assessmentId	Varchar(10)	Alphanumeric	N	N	10	1
Assessment Date	assessmentDate	Date (dd/mm/yyyy)	Date	N	N	10	10
Assessment Type (L)	assessmentType	Varchar(3)	Alphanumeric	N	N	3	1
Referral Date	referralDate	Date (dd/mm/yyyy)	Date	N	N	10	10
Site ID (L)	siteId	Varchar(6)	Alphanumeric	N	N	6	6
Training Recommendation	trainingRecommendation	Varchar(1)	Boolean	C	N	1	1
	trainingAssessment	Training Assessment Data (upto 11 ratings)					

Assessment data

US Field Name	Payload Field Name	Length	Data Type	Nullable?	Variable Length String Can be truncated?	Max Length	Min Length
ACSF Skill Type (L)	acsfSkillType	varchar(100)	Alphanumeric	N	N	100	1
ACSF Level (L)	acsfLevel	varchar(20)	Alphanumeric	N	N	20	1

Nullability = Y/N/C
Variable Length String (Can be truncated) = Y/N
Max = Maximum length of value
Min = Minimum length of value

(L) = Lookup

Request Payload – Student SEE Data POST


```
{
    "StudentSEEData": {
        "ProviderId": "91451",
        "UniqueStudentId": "ZU33XZDZXU",
        "ClientId": "91451-USI",
        "AssessmentId": "C02-A01",
        "AssessmentDate": "02/01/2021",
        "AssessmentType": "PTA",
        "ReferralDate": "01/01/2021",
        "SiteId": "L70014",
        "TrainingAssessment": [{
                "ACSFSkillType": "Learning.01",
                "ACSFLevel": "PreLevelB"
            },
            {
                "ACSFSkillType": "Learning.02",
                "ACSFLevel": "PreLevel"
            },
            {
                "ACSFSkillType": "Reading.03",
                "ACSFLevel": "PreLevelA"
            },
            {
                "ACSFSkillType": "Reading.04",
                "ACSFLevel": "Level2"
            },
            {
                "ACSFSkillType": "Writing.05",
                "ACSFLevel": "Level4"
            },
            {
                "ACSFSkillType": "Writing.06",
                "ACSFLevel": "PreLevelB"
            },
            {
                "ACSFSkillType": "Oral Communication.07",
                "ACSFLevel": "Level1"
            },
            {
                "ACSFSkillType": "Oral Communication.08",
                "ACSFLevel": "Level3"
```

```
            },
            {
                "ACSFSkillType": "Numeracy.09",
                "ACSFLevel": "PreLevelA"
            },
            {
                "ACSFSkillType": "Numeracy.10",
                "ACSFLevel": "PreLevel"
            },
            {
                "ACSFSkillType": "Numeracy.11",
                "ACSFLevel": "Level4"
            }
        ],
        "TrainingRecommendation": "Y"
    }
}
```

Response Payload – Student SEE Data POST

```
{
    "validationId": "67dd1fbc-5073-41f3-a638-ea902bf411c9",
    "validationTimestamp": "2020-12-24T05:49:03.3669986Z",
    "status": "Validating"
}
```

Response Student SEE Data GET by Validation ID – Validating

```
{
    "validationId": "67dd1fbc-5073-41f3-a638-ea902bf411c9",
    "validationTimestamp": "2020-12-24T05:49:03.3669986Z",
    "status": "Validating"
}
```

Response Student SEE Data GET by Validation ID – Pending Delete

```
{
    "validationId": "26124a87-fa05-479f-b106-c394fc7dcfd1",
    "validationTimestamp": "2020-12-24T05:54:20.261139Z",
    "status": "Pending Delete"
}
```

Response Student SEE Data GET by Validation ID – Completed

```
{
    "studentSEEDataList": {
    "StudentSEEData": {
```

```
        "ProviderId": "91451",
        "UniqueStudentId": "ZU33XZDZXU",
        "ClientId": "91451-USI",
        "AssessmentId": "C02-A01",
        "AssessmentDate": "02/01/2021",
        "AssessmentType": "PTA",
        "ReferralDate": "01/01/2021",
        "SiteId": "L70014",
        "TrainingAssessment": [{
                "ACSFSkillType": "Learning.01",
                "ACSFLevel": "PreLevelB"
            },
            {
                "ACSFSkillType": "Learning.02",
                "ACSFLevel": "PreLevel"
            },
            {
                "ACSFSkillType": "Reading.03",
                "ACSFLevel": "PreLevelA"
            },
            {
                "ACSFSkillType": "Reading.04",
                "ACSFLevel": "Level2"
            },
            {
                "ACSFSkillType": "Writing.05",
                "ACSFLevel": "Level4"
            },
            {
                "ACSFSkillType": "Writing.06",
                "ACSFLevel": "PreLevelB"
            },
            {
                "ACSFSkillType": "Oral Communication.07",
                "ACSFLevel": "Level1"
            },
            {
                "ACSFSkillType": "Oral Communication.08",
                "ACSFLevel": "Level3"
            },
            {
                "ACSFSkillType": "Numeracy.09",
                "ACSFLevel": "PreLevelA"
            },
            {
```

```
                "ACSFSkillType": "Numeracy.10",
                "ACSFLevel": "PreLevel"
            },
            {
                "ACSFSkillType": "Numeracy.11",
                "ACSFLevel": "Level4"
            }
        ],
        "TrainingRecommendation": "Y"
    },
        "validationResults": [{
                "ruleSet": "Basic Validations",
                "errorCount": 0,
                "warningCount": 0,
                "infoCount": 0
            },
            {
                "ruleSet": "StudentAVETMISSFieldValidations",
                "errorCount": 0,
                "warningCount": 0,
                "infoCount": 0
            },
            {
                "ruleSet": "StudentNSWFieldValidations",
                "errorCount": 0,
                "warningCount": 0,
                "infoCount": 0
            }
        ],
        "errors": [],
        "warnings": [],
        "infos": [],
        "exceptionId": null,
        "validationId": "1e3cc5d9-75a8-46dd-b4e6-f56e92157054",
        "validationTimestamp": "2020-12-25T02:05:58",
        "status": "Mastered"
    }]
}
```


Response Student SEE Data GET by Validation ID – Critical Failure


```
{
      "errors": [
        {
          "id": "6ee91a5c-0d14-4a03-aa66-5eb974062d39",
```

```
          "detail": "Unexpected Error",
          "code": "500"
        }
      ]
}
```


Response Student SEE Data GET by Client ID


```
{
    "StudentSEEData": {
        "ProviderId": "91451",
        "UniqueStudentId": "ZU33XZDZXU",
        "ClientId": "91451-USI",
        "AssessmentId": "C02-A01",
        "AssessmentDate": "02/01/2021",
        "AssessmentType": "PTA",
        "ReferralDate": "01/01/2021",
        "SiteId": "L70014",
        "TrainingAssessment": [{
                "ACSFSkillType": "Learning.01",
                "ACSFLevel": "PreLevelB"
            },
            {
                "ACSFSkillType": "Learning.02",
                "ACSFLevel": "PreLevel"
            },
            {
                "ACSFSkillType": "Reading.03",
                "ACSFLevel": "PreLevelA"
            },
            {
                "ACSFSkillType": "Reading.04",
                "ACSFLevel": "Level2"
            },
            {
                "ACSFSkillType": "Writing.05",
                "ACSFLevel": "Level4"
            },
            {
                "ACSFSkillType": "Writing.06",
                "ACSFLevel": "PreLevelB"
            },
            {
                "ACSFSkillType": "Oral Communication.07",
                "ACSFLevel": "Level1"
            },
```

```
            {
                "ACSFSkillType": "Oral Communication.08",
                "ACSFLevel": "Level3"
            },
            {
                "ACSFSkillType": "Numeracy.09",
                "ACSFLevel": "PreLevelA"
            },
            {
                "ACSFSkillType": "Numeracy.10",
                "ACSFLevel": "PreLevel"
            },
            {
                "ACSFSkillType": "Numeracy.11",
                "ACSFLevel": "Level4"
            }
        ],
        "TrainingRecommendation": "Y"
    },
    "validationResults": [{
            "ruleSet": "Basic Validations",
            "errorCount": 0,
            "warningCount": 0,
            "infoCount": 0
        },
        {
            "ruleSet": "StudentAVETMISSFieldValidations",
            "errorCount": 0,
            "warningCount": 0,
            "infoCount": 0
        },
        {
            "ruleSet": "StudentNSWFieldValidations",
            "errorCount": 0,
            "warningCount": 0,
            "infoCount": 0
        }
    ],
    "errors": [],
    "warnings": [],
    "infos": [],
    "exceptionId": null,
    "validationId": "1e3cc5d9-75a8-46dd-b4e6-f56e92157054",
    "validationTimestamp": "2020-12-25T02:05:58",
    "status": "Mastered"
```

```
}
```

Response Student DELETE

```
{
    "validationId": "26124a87-fa05-479f-b106-c394fc7dcfd1",
    "validationTimestamp": "2020-12-24T05:54:20.261139Z",
    "status": "Validating"
}
```

15. Student (Batch): Endpoints and HTTP return codes

Http Verb or end point	Header Parameter	Http status codes
POST (Create / Validate) - Batch e.g. Validate Student List POST /v1/students/validate	Training Organisation Identifier	202 - Accepted 400 - Cannot desterilize the payload Or Identifiers are not matched 403 - Forbidden (to be implemented once security design completed)

Training Organisation Identifier – Header parameter

Training Organisation Identifier	providerId	varchar (10)	Alphanumeric	N	N	10	1

Request Payload – Student POST Batch

```
{
  "studentList": [{
    "student": {
      "clientId": "CLIENT123",
        "title": "Mr.",
        "familyName": "Doe",
        "givenNameFirst": "John",
        "givenNameOther": "Johnny",
        "highestSchoolLevelCompletedId": "12",
        "gender": "M",
        "dateOfBirth": "18/08/1980",
        "labourForceStatusId": "01",
        "priorEducationalAchievementIds": [
          {
            "achievementId": ""
          }
        ],
        "atSchoolFlag": "N",
        "uniqueStudentId": "USI123-456",
        "surveyContactStatus": "A",
        "residentialAddress": {
          "countryId": "1101",
```

```
          "unitNumber": "60",
          "buildingName": "Building 123",
          "streetNumber": "60",
          "streetName": "Miller St",
          "suburb": "North Sydney",
          "stateId":"01",
          "postcode": "2060"

        },
        "contactAddress": {
          "postalDeliveryBox": "Contact Postal Box",
          "unitNumber": "10",
          "buildingName": "Building 456",
          "streetNumber": "10",
          "streetName": "George St",
          "suburb": "Sydney",
          "stateId": "01",
          "postcode": "2000"
        },
        "contactHomePhoneNumber": "02 9439 1234",
        "contactWorkPhoneNumber": "02 9439 7910",
        "contactMobilePhoneNumber": "02 1234 5678",
        "contactEmail": "Y",
        "contactAlternativeEmail": "Y",
        "deceased": "0",
        "qualificationSince17": "2",
        "highestPostSchoolQualification": "5",
        "stateCurrentlyLivingIn": "01",
        "residencyStatus": "1",
        "stateCurrentlyWorkingIn": "01",
        "employerOrganisationName": "Sample Employer Name",
        "employerOrganisationPostcode": "2155",
        "employerOrganisationSuburb": "Kellyville",
        "employerOrganisationStreetNumber": "123",
        "employerOrganisationStreetName": "Sample Street"
    }
  }]
}
```


Response – Student POST Batch


```
{
    "validationId": "67dd1fbc-5073-41f3-a638-ea902bf411c9",
    "validationTimestamp": "2020-08-24T05:49:03.3669986Z",
    "status": "Validating"
}
```

16. Student SEE Data (Batch): Endpoints and HTTP return codes

Http Verb or end point	Header Parameter	Http status codes
POST (Create / Validate) - Batch e.g. Validate Student SEE Data List POST /v1/students/seedata/validate	Training Organisation Identifier	202 - Accepted 400 - Cannot desterilize the payload Or Identifiers are not matched 403 - Forbidden (to be implemented once security design completed)

Training Organisation Identifier – Header parameter

Training Organisation Identifier	providerId	varchar (10)	Alphanumeric	N	N	10	1

Request Payload – Student SEE Data Batch POST


```
{
    "studentSEEDataList": [{
        "StudentSEEData": {
          "ProviderId": "91451",
          "UniqueStudentId": "ZU33XZDZXU",
          "ClientId": "91451-USI",
          "AssessmentId": "C02-A01",
          "AssessmentDate": "02/01/2021",
          "AssessmentType": "PTA",
          "ReferralDate": "01/01/2021",
          "SiteId": "L70016",
          "TrainingAssessment": [
            {
              "ACSFSkillType": "Learning.01",
              "ACSFLevel": "PreLevelB"
            },
            {
              "ACSFSkillType": "Learning.02",
              "ACSFLevel": "PreLevel"
            },
            {
              "ACSFSkillType": "Reading.03",
              "ACSFLevel": "PreLevelA"
            },
            {
              "ACSFSkillType": "Reading.04",
              "ACSFLevel": "Level2"
            },
            {
              "ACSFSkillType": "Writing.05",
              "ACSFLevel": "Level4"
```

```
            },
            {
              "ACSFSkillType": "Writing.06",
              "ACSFLevel": "PreLevelB"
            },
            {
              "ACSFSkillType": "Oral Communication.07",
              "ACSFLevel": "Level1"
            },
            {
              "ACSFSkillType": "Oral Communication.08",
              "ACSFLevel": "Level3"
            },
            {
              "ACSFSkillType": "Numeracy.09",
              "ACSFLevel": "PreLevelA"
            },
            {
              "ACSFSkillType": "Numeracy.10",
              "ACSFLevel": "PreLevel"
            },
            {
              "ACSFSkillType": "Numeracy.11",
              "ACSFLevel": "Level4"
            }
          ],
          "TrainingRecommendation": "Y"
        }
      }
    ]
}
```

Response – Student SEE Data Batch POST

```
{
    "validationId": "e05f6140-2d34-43d9-8c6d-e809feee406d",
    "validationTimestamp": "2021-01-28T05:21:15.4503111Z",
    "status": "Validating"
}
```

17. Student secure end points – Authentication & Authorisation

In staging and production environment, all the traffic to the API end points will flow through the APIM and each request will get authenticated and authorised before it actually hits the end point.

If you are looking for what and how the security is designed and implemented for API end points then please refer this document.

https://nsitechnology.sharepoint.com/:w:/r/sites/DepartmentofEducation-VDSePoC/Shared%20Documents/General/Technical/02%20Solution%20Architecture/VET%20ePoC%20-%20Identity%20Access%20Management%20Design.docx?d=w02d289c25ff542979d2a220290065f98&csf=1&web=1&e=Fo77Sk

This section will focus more on the how to consume secure end points.
Base Student URIs for secure end points:

DEV

```
https://vds-dev-apimgt.azure-api.net/Student/api/v1/students/{clientId}
```

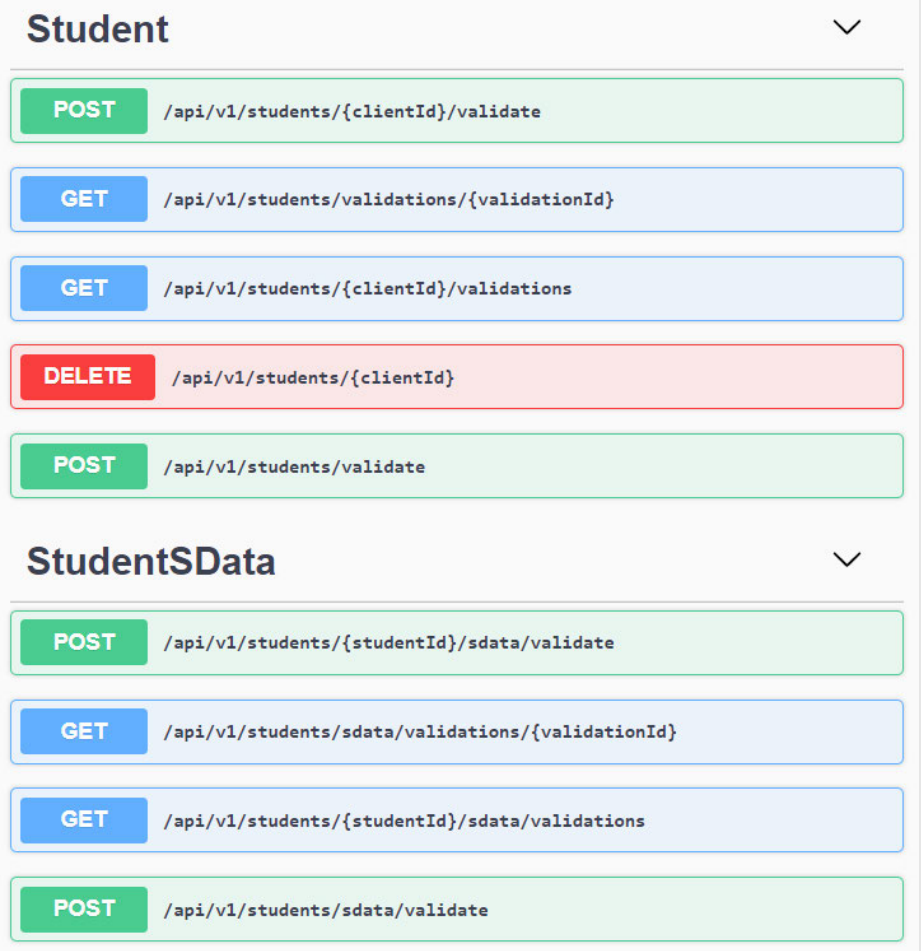
TEST

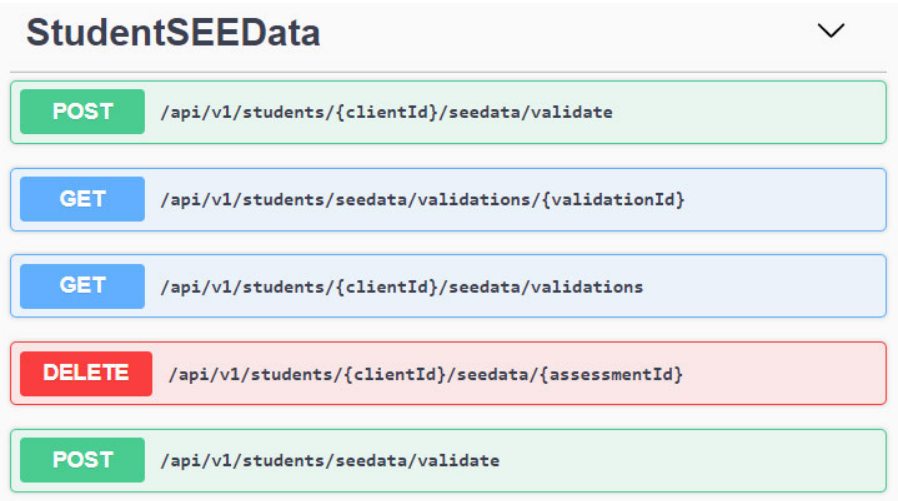
```
https://vds-test-apimgt.azure-api.net/Student/api/v1/students/{clientId}
```

UAT

```
https://vds-uat-apimgt.azure-api.net/Student/api/v1/students/{clientId}
```

External Public secure end points are:





- **How to access these end points:**
  Complete end point URI example to validate a program:

https://vds-dev-apimgt.azure-api.net/CourseAdmin/api/v1/programs/instances/Ins1/validate

This end point expects authorisation header (jwt token) and providerId in header of the request.

How to get a JWT token for authorisation header?

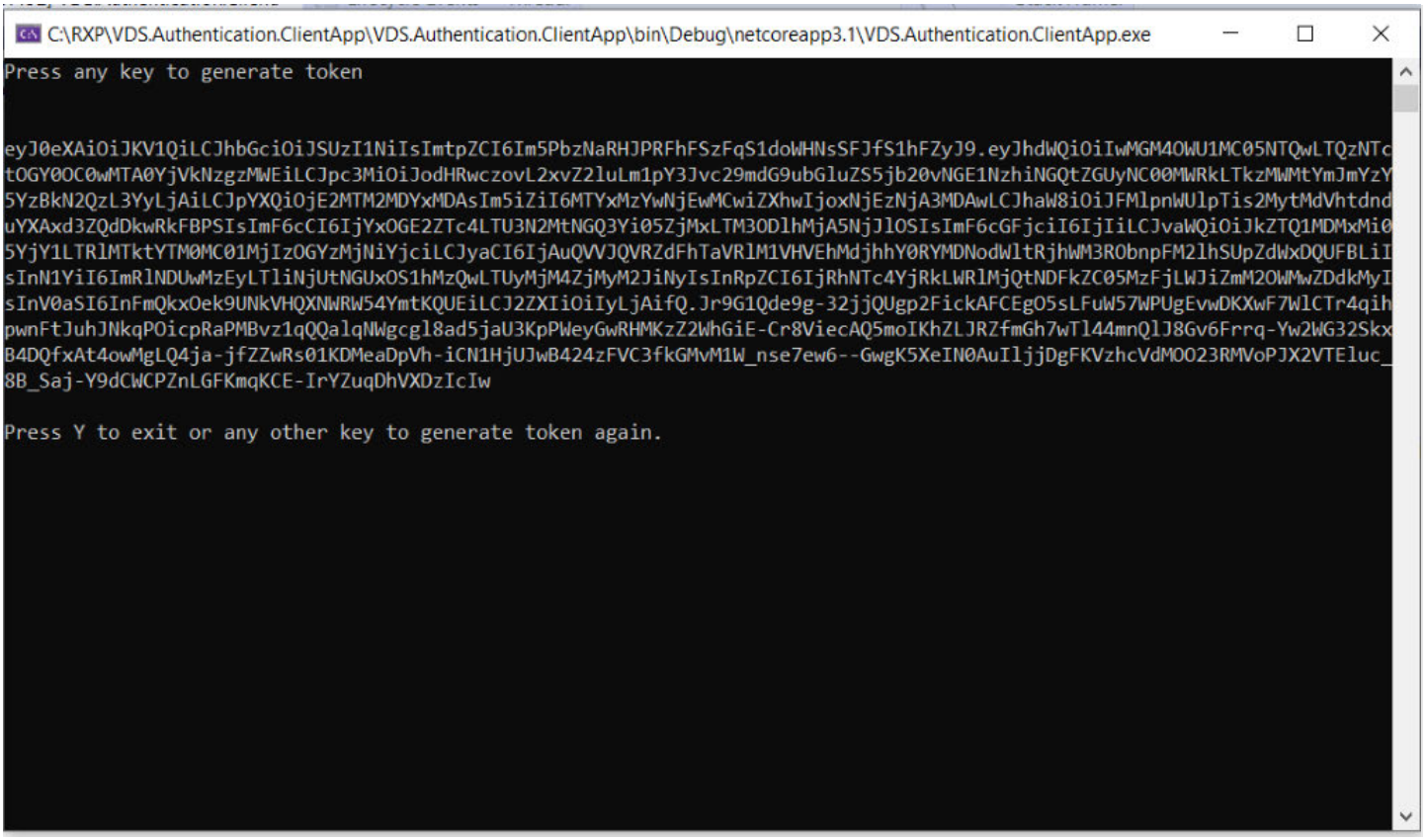
Please use the TokenGenerator console app to create a token for a particular environment like DEV, TEST and UAT. This application can be found at this location and is available for download:

DEV/TEST/UAT

https://nsitechnology.sharepoint.com/:f:/r/sites/DepartmentofEducation-VDSePoC/Shared%20Documents/General/Technical/03%20Detailed%20Design/Security?csf=1&web=1&e=kSUKsW

Run these apps to get the token for a SMS provider, currently this application will generate a token for provider 110069.

Run exe - **VDS.Authentication.ClientApp.exe** from the folder and press any key to generate token



Copy the token and paste that to Authorization header

☑	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6Im5PbzNa...
☑	providerId	110069

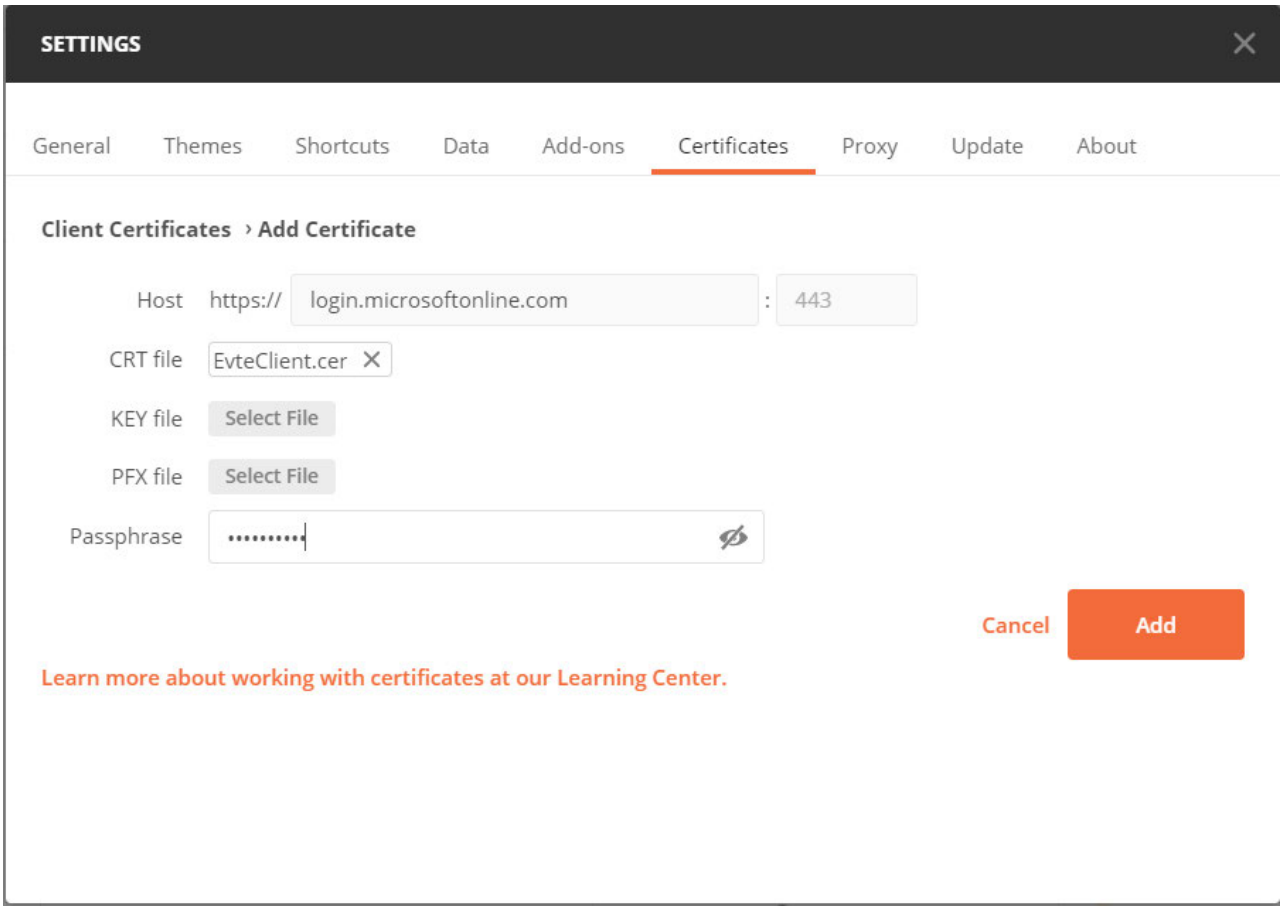
If not a valid token has been passed or token has been expired then authentication failed and status code 401 Unauthorised will be returned.

```
{
  "statusCode": 401,
  "message": "Unauthorized"
}
```

In case the authorisation after the authentication failed then the same response will be returned.

- **Alternate way to create a Token using the postman**

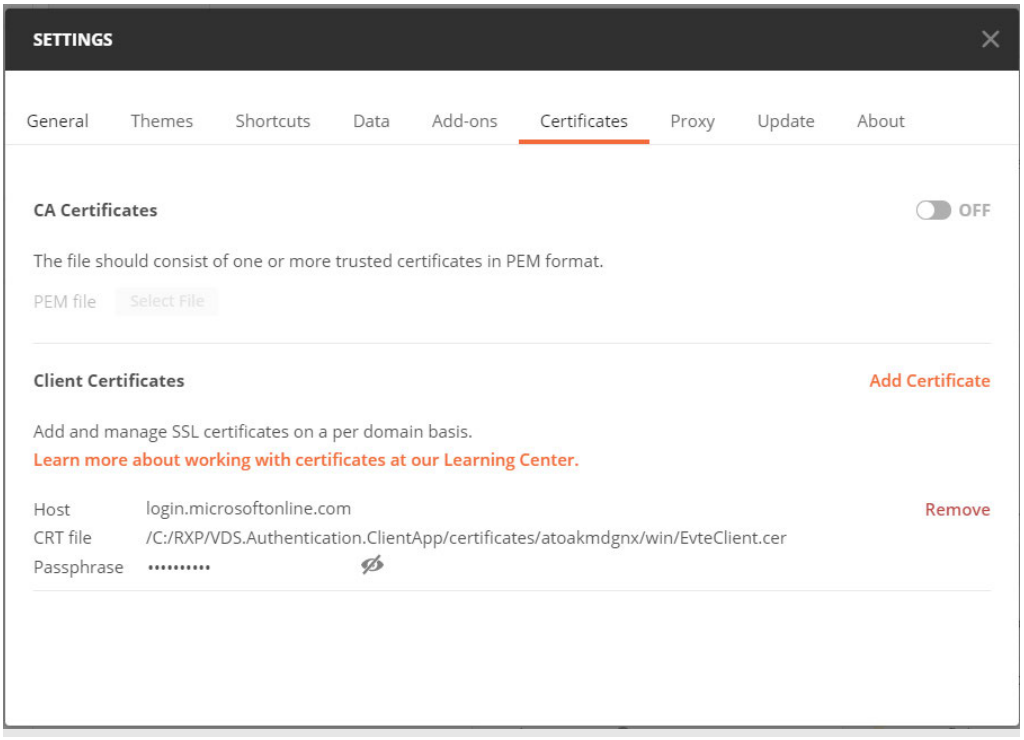
  Add a client certificate in the settings section:



- Download certificate from this location:

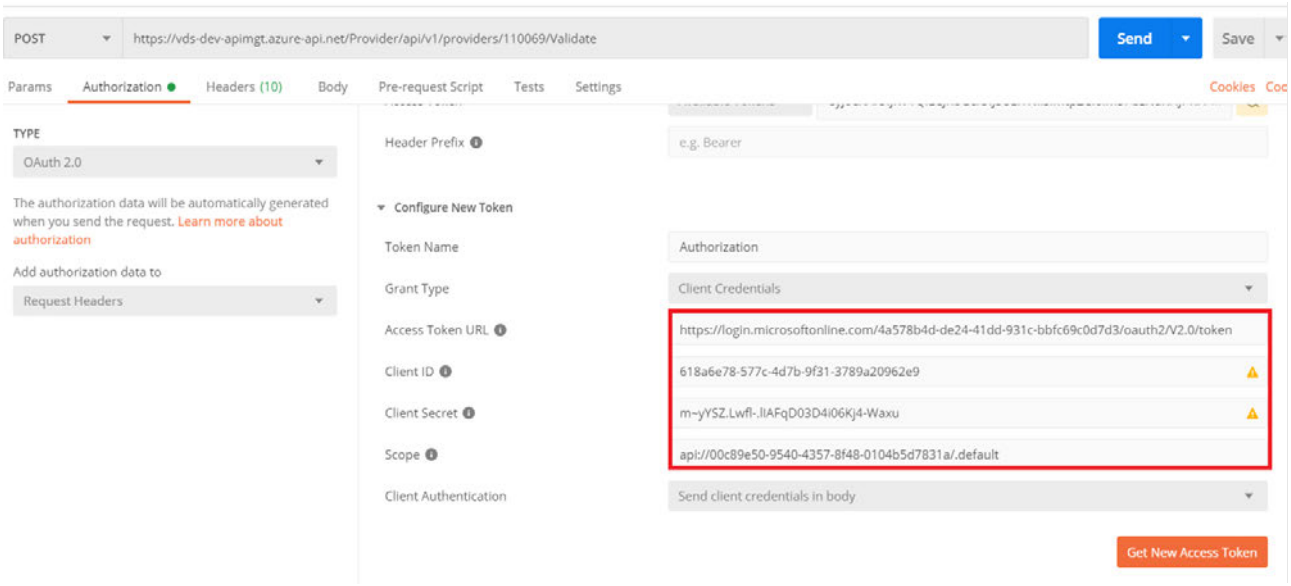
  https://nsitechnology.sharepoint.com/:u:/r/sites/DepartmentofEducation-VDSePoC/Shared%20Documents/General/Technical/03%20Detailed%20Design/Security/EvteClient.cer?csf=1&web=1&e=Zahfsb

- passphrase is Password1!



Certificate is added for login.microsoftonline.com

- **How to generate a new token using postman?**



DEV

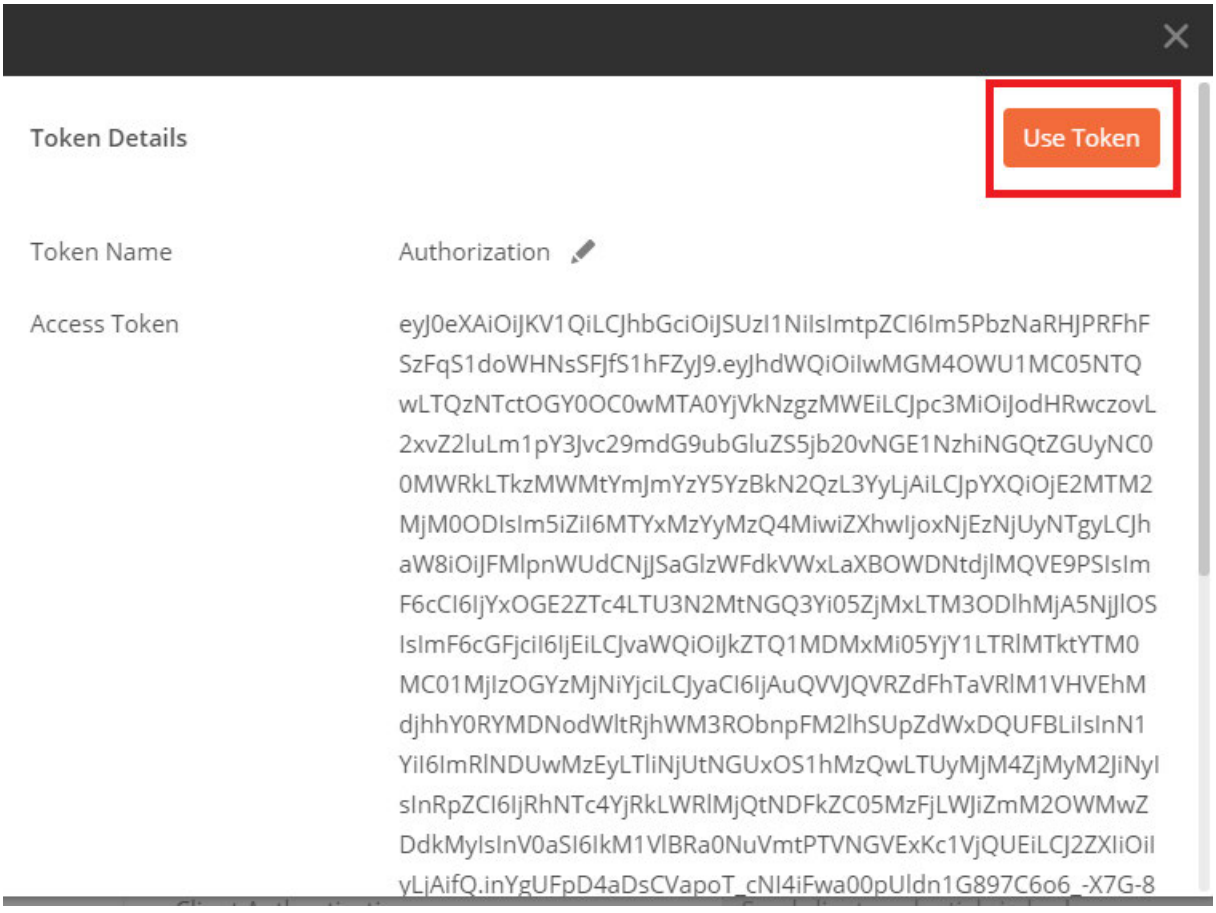
- Access Token URL: https://login.microsoftonline.com/4a578b4d-de24-41dd-931c-bbfc69c0d7d3/oauth2/V2.0/token
- ClientId: 618a6e78-577c-4d7b-9f31-3789a20962e9
- Client Secret: m~yYSZ.Lwfl-.IIAFqD03D4i06Kj4-Waxu
- Scope: api://00c89e50-9540-4357-8f48-0104b5d7831a/.default

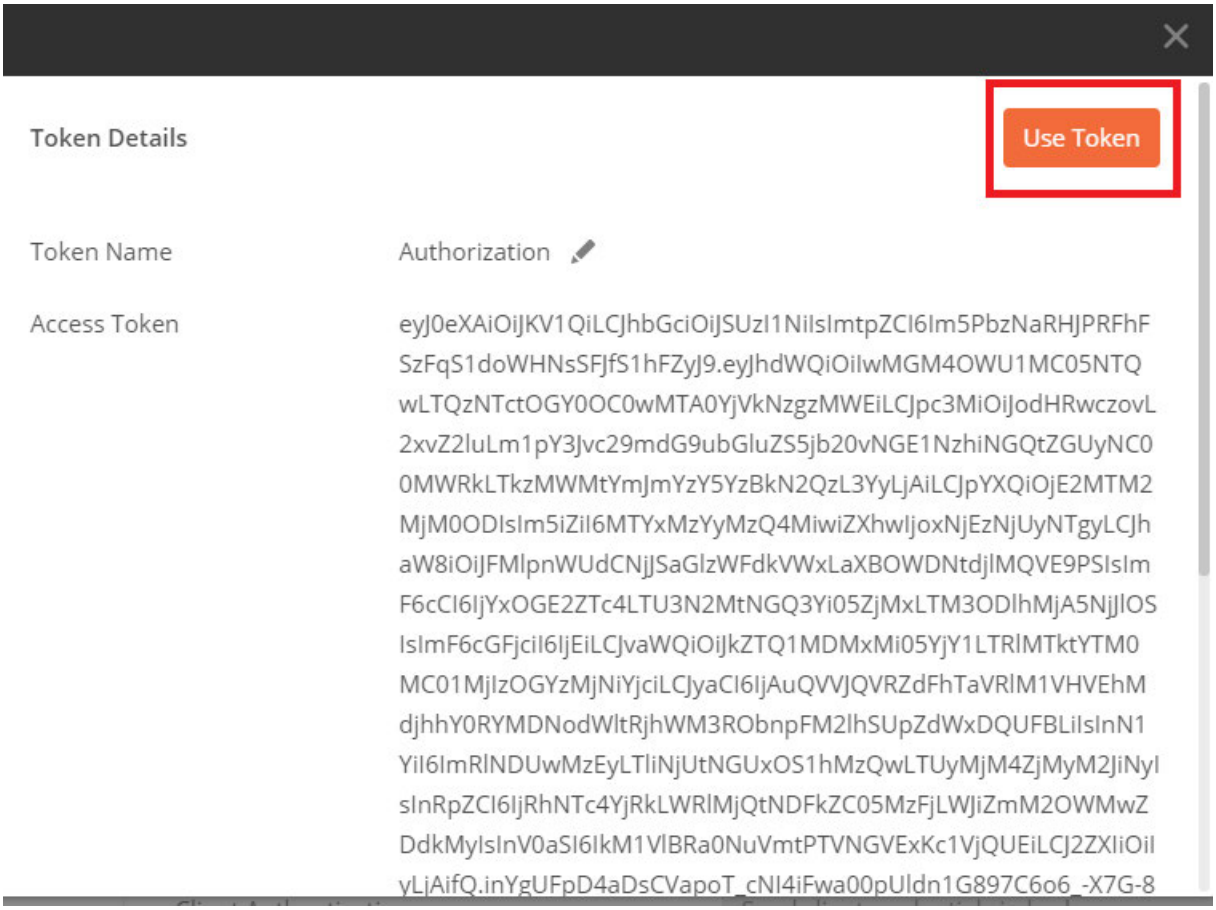
TEST

- Access Token URL: https://login.microsoftonline.com/fac941be-74a3-43b1-bbf4-96f9aa71a625/oauth2/V2.0/token
- ClientId: `30ef00a2-9ea9-420d-ba6c-69f846f62d74`
- Client Secret: `_fGn_~4W4zOs844YyLLj2esAap~i.NgbJN`
- Scope: `api://6059cf21-c8b9-4ca6-a1d7-de013a2f136a/.default`

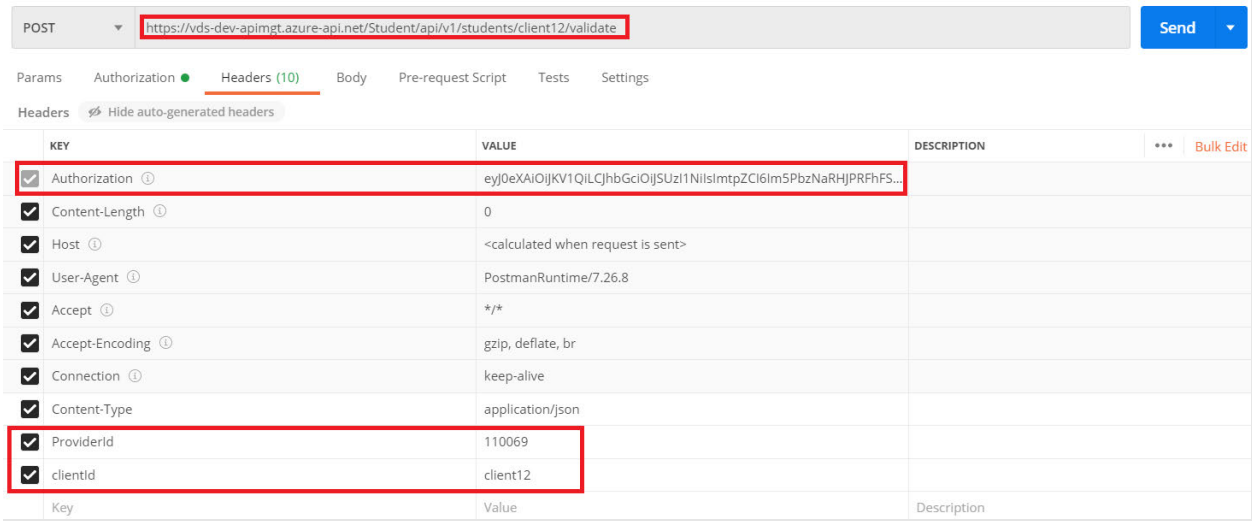
UAT

- Access Token URL: https://login.microsoftonline.com/b0c2e0cc-ff38-4dd2-b8c4-21723a430f20/oauth2/V2.0/token
- ClientId: `b19079cc-2d77-4ee9-9e06-4f1f44f35ee3`
- Client Secret: `aujiAi_3Yf6Y1P5LZ6CO._7t87~S4.u-j6`
- Scope: `api://b8eda86a-0dba-4db6-8567-356986229b8b/.default`





- **Send a new request using the Token**



Make sure the information is same as above screen shot and click the send request button.

18. USI Validation

Student API validates the USI with the USI verification service.
USI Verification service runs in two different modes:

1. Mock
2. Live

Student Validate end points validate the USI with the verification service and add the result error or warning to the validation result like InRule validation or behavior validation.

Mock mode USI input and corresponding error/warning in validate response:

USI - null

- Expected = Actual - No USI warnings - Mastered USI null
- Note: USI Client does not call USI Service Proxy

USI - “1234567890",

- Expected - Even 10 digit Mastered with USI value
- Note: Mock returns true

```
{
    "ruleSet": "USI Validation",
    "errorCount": 0,
    "warningCount": 0,
    "infoCount": 0
  }
```

USI - “1234567891",

- Expected - Validated Warning
- Note: Mock returns false

```
{
    "ruleSet": "USI Validation",
    "errorCount": 0,
    "warningCount": 1,
    "infoCount": 0
  }
```

```
"warnings": [
  {
    "code": "ST3077.4",
    "ruleSet": "USI Validation",
    "source": "UniqueStudentId",
```

```
    "message": "Your Unique Student Identifier supplied is
invalid. Please correct and re-submit."
  }
]
```

USI - “ZZZA8X3856",

- Expected - USI Match, Mastered with USI value, no USI warnings
- Note: Mock returns true

```
{
  "ruleSet": "USI Validation",
  "errorCount": 0,
  "warningCount": 0,
  "infoCount": 0
}
```

Below Exception messages will now include ExceptionId:{GUID} which we can search in the log to trace the issue.

USI - “Error_500_",

- Expected - USI Service Error Handled - Mastered with USI null, Exception Message
- Note: Mock returns 500 status

```
{
  "ruleSet": "USI Validation",
  "errorCount": 0,
  "warningCount": 1,
  "infoCount": 0
}
```

```
"warnings": [
  {
    "code": "ST3077.500",
    "ruleSet": "USI Validation",
    "source": "UniqueStudentId",
    "message": "Exception occurred when trying to validate the Unique Stud
ent Identifier. USI can not be validated at this time. Please re-
send the request. ExceptionId: {UniqueGUID}"
  }
]
```

USI - “Error_400_",

- Expected - USI Service Error Handled - Mastered with USI null, Exception Message
- Note: Mock returns 400 status


```
{
  "ruleSet": "USI Validation",
  "errorCount": 0,
  "warningCount": 1,
  "infoCount": 0
}
```

```
"warnings": [
  {
    "code": "ST3077.400",
    "ruleSet": "USI Validation",
    "source": "UniqueStudentId",
    "message": "Exception occurred when trying to validate the Unique Stud
ent Identifier. USI can not be validated at this time. Please re-
send the request. ExceptionId: {UniqueGUID} "
  }
]
```


USI - “Error_503_",

- Expected - USI Service Error Handled - Mastered with USI null, Exception Message
- Note: Mock returns 503 status


```
{
  "ruleSet": "USI Validation",
  "errorCount": 0,
  "warningCount": 1,
  "infoCount": 0
}
```

```
"warnings": [
  {
    "code": "ST3077.503",
    "ruleSet": "USI Validation",
    "source": "UniqueStudentId",
    "message": "Exception occurred when trying to validate the Unique Stud
ent Identifier. USI can not be validated at this time. Please re-
send the request. ExceptionId: {UniqueGUID} "
  }
]
```


USI - “Error_401_",

- Expected - USI Service Error Handled - Mastered with USI null, Exception Message
- Note: Mock returns 401 status

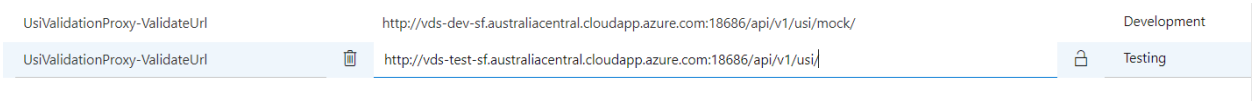
```
{
  "ruleSet": "USI Validation",
  "errorCount": 0,
  "warningCount": 1,
  "infoCount": 0
}
```

```
"warnings": [
  {
    "code": "ST3077.401",
    "ruleSet": "USI Validation",
    "source": "UniqueStudentId",
    "message": "Exception occurred when trying to validate the Unique Stud
ent Identifier. USI can not be validated at this time. Please re-
send the request. ExceptionId: {UniqueGUID} "
  }
]
```

How to switch USI service linked to Mock or Live mode?

By default DEV and TEST stages are connected to USI mock. USI service can be linked in Live mode to any of the environment.

Go to Student mastering CD pipeline and create this variable if doesn't exist for that stage.



Variable name is "UsiValidationProxy-ValidateUrl" and value is URI to the USI service.

USI service has two end points:

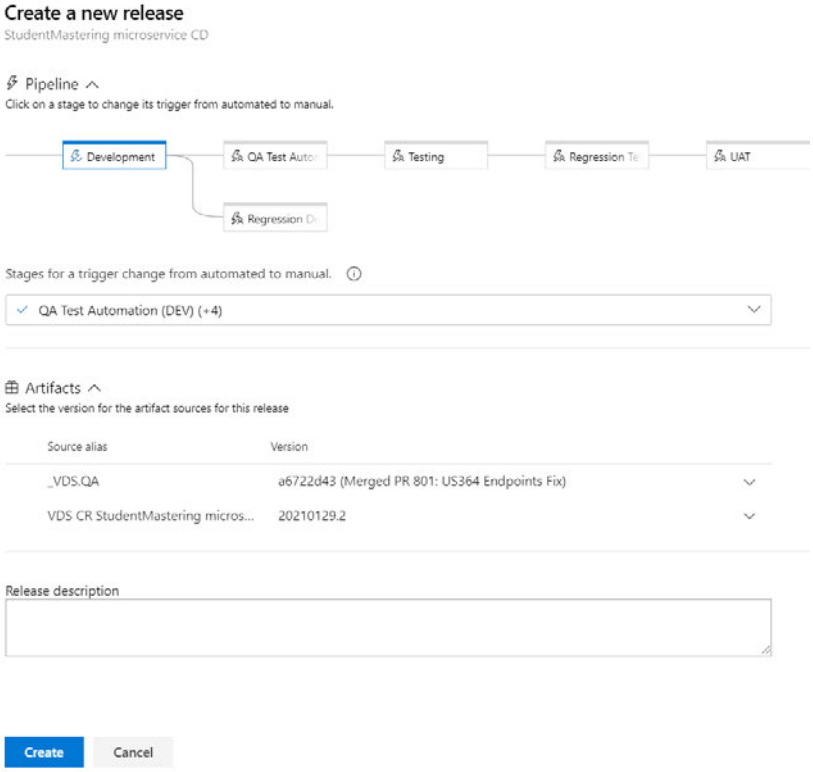
- Mock
- Live

Mock mode URI of USI service for DEV stage
http://vds-dev-sf.australiacentral.cloudapp.azure.com:18686/api/v1/usi/mock/

Live mode URI of USI service for TEST stage
http://vds-test-sf.australiacentral.cloudapp.azure.com:18686/api/v1/usi/

So if mock/ is removed from Mock mode URI then it will become the Live mode URI.

If you would like to change the mode from Mock to Live or vice versa then first change the URI value of this variable accordingly for that stage or environment and then **create a new release for that environment**.



19. Acceptance Criteria

User Story	Acceptance Criteria	Notes/Comments/Questions	Task(s) #
3067 (Student - Create / Update Student Data)	AC1	Handled via the API endpoint for Validate Single Student	3419
	AC2, AC3, AC4	Handled via behavioral rule templates	
	AC5, AC6	Handled via Error Response Fields (section 7)	
3071 (Student - Create / Update Student Prior Learning Data)	AC1	Handled via the API endpoint for Validate Single Student	3447
	AC2, AC4	Handled via inRule	
	AC3	Handled via collection for priorEducationalAchievementIds in json request payload	
	AC5, AC6, AC7	Handled via behavioral rule templates	
	AC8, AC9	Handled via Error Response Fields (section 7)	
4014 (Student - Create / Update Student NDE Specific Data)	AC1	Handled via the API endpoint for Validate Single Student	4030
	AC2, AC3, AC4	Handled via behavioral rule templates	

	AC5, AC6	Handled via Error Response Fields (section 7)	
4158 (Student - Create / Update Student SEE Specific Data)	AC1, AC2	Handled via the API endpoint for Validate Single Student and behavioral rule templates	4167
	AC3	Values supplied through the API endpoint for Student; Rule to be implemented in inRule	
	AC4	Allowed values for ACSF Skill Type to be handled via inRule	
	AC5	Allowed values for ACSF Level to be handled via inRule	
6212: Unit Enrolment - Retro - Student - SEE	AC1	New Create/Update SEE Assessment endpoint	
	AC2, AC3, AC4	Handled via behavioral rule templates	
	AC5, AC6	Handled via Error Response Fields (section 7)	
	AC7, AC8	Assessment ratings supplied through the endpoint. Error messages and business rules (US7406) handled in BRMS	
	AC9, AC10, AC11, AC12, AC13	Values supplied through the endpoint. Error messages and business rules (US7406) handled in BRMS	
7158: Unit Enrolment - Retro - Student - Delete Pre-training Assessments (SEE)	AC1, AC2	New Delete SEEAssessment endpoint	
	AC3, AC5	To be handled in BRMS	
	AC4, AC6	Handled via Error Response Fields (section 7)	
7998: Unit Enrolment - Retro - Student - SEE - Batch SEE Assessments	AC1		
	AC2	Behavior rule validations to be handled by mastering service. Rules mentioned in US 7406 to be handled via inRule	
	AC3	Handled via Error Response Fields (section 7)	
3070 (Student - Get Validation Result - Student Data)	AC1	Handled via the API endpoint (GET by Client ID)	3440
	AC2	Handled via the API endpoint (GET by Validation ID)	
	AC3	This will be handled by fetching the Provider ID from the header of the request and fetching data for that Provider (based on the Student ID or Validation ID)	
	AC4	Added a clarificationto user story for different GET response statuses	
3069 (Student - Delete Student Data - Student Data – SINGLE)	AC1	Handled via the API endpoint for Delete Single Student (DELETE by Client ID) Added a question related to the Deletion of linked Client record.	3433
	AC2	Facilitated by having Provider ID as part of the request header	

	AC3, AC4	Handled through the Validation Status collection. Error response handled via Error Response Fields (section 7)	
	AC5	Not intended for Student increment – confirmed by Matt	
3068 (Student - Batch Student Data)	AC1	Handled via the API endpoint for Validate Student List (Only Student Data, Prior Learnings Data and NDE specific data. The Sensitive Data and Disability Data would be a separate endpoint)	3426
	AC2	Handled via behavioral rule templates	
	AC3	Handled via Error Response Fields (section 7)	

20. Service Bus Publishing

The Student endpoint Microservice publishes a Message to the Service Bus, which has the following fields:

Header Fields	Description	Type / Possible Values
MessageId	A concatenation of a Guid and a UtcNow timestamp	string
SessionId	ProviderId	string
Label	Details whether the entity is to be validated or deleted, and whether the entity is a Student Data or a Sensitive Student Data	“CreateStudentData”, “DeleteStudentData”, “CreateSensitiveStudentData”, “DeleteSensitiveStudentData”

Body (ServiceBusMessage)	Description	Possible Values
PayloadType	Indicates the type of the payload and whether it is part of a sequence of entities.	“StudentData”, “StudentDataList”, “SensitiveStudentData”, “SensitiveStudentDataList”
Payload	If PayloadType is “StudentData” or “StudentDataList”, Payload will be of type StudentRequest. If PayloadType is “SensitiveStudentData” or “SensitiveStudentDataList”, Payload will be of type SensitiveStudentRequest.	StudentRequest, SensitiveStudentRequest
ValidationId	Unique ID that is returned to the client once request is accepted. This value is also stored in the respective entry in the CosmosDB AsyncValidationStatus collection.	Guid
ValidationTimestamp	Timestamp that is returned to the client once request is accepted. This value is also stored in the respective entry in the CosmosDB AsyncValidationStatus collection.	DateTime
SequenceNumber	Indicates the sequence number of the entity in its request, ranging from 1 to TotalModelsInPayload.	int

TotalModelsInPayload	Indicates the total number of entities in the request; for POST Single and DELETE, this will be int 1.	
Action	Indicates the action to be taken on the entity; if request is through POST then “Validate”, else if request is through DELETE then “Delete”.	“Validate”, “Delete”

The ServiceBusMessage has the following format:

```
{
  "serviceBusMessage": {
    "payloadType": "StudentDataList",
    "payload": {
      "student": {
        "clientId": "CLIENT123",
        "title": "Mr.",
        "familyName": "Doe",
        "givenNameFirst": "John",
        "givenNameOther": "Johnny",
        "highestSchoolLevelCompletedId": "12",
        "gender": "M",
        "dateOfBirth": "18/08/1980",
        "labourForceStatusId": "01",
        "priorEducationalAchievementIds": [
          {
            "achievementId": ""
          }
        ],
        "atSchoolFlag": "N",
        "uniqueStudentId": "USI123-456",
        "surveyContactStatus": "A",
        "residentialAddress": {
          "countryId": "1101",
          "unitNumber": "60",
          "buildingName": "Building 123",
          "streetNumber": "60",
          "streetName": "Miller St",
          "suburb": "North Sydney",
          "stateId":"01",
          "postcode": "2060"

        },
        "contactAddress": {
          "postalDeliveryBox": "Contact Postal Box",
          "unitNumber": "10",
          "buildingName": "Building 456",
```

```
          "streetNumber": "10",
          "streetName": "George St",
          "suburb": "Sydney",
          "stateId": "01",
          "postcode": "2000"
        },
        "contactHomePhoneNumber": "02 9439 1234",
        "contactWorkPhoneNumber": "02 9439 7910",
        "contactMobilePhoneNumber": "02 1234 5678",
        "contactEmail": "Y",
        "contactAlternativeEmail": "Y",
        "deceased": "0",
        "qualificationSince17": "2",
        "highestPostSchoolQualification": "5",
        "stateCurrentlyLivingIn": "01",
        "residencyStatus": "1",
        "stateCurrentlyWorkingIn": "01",
        "employerOrganisationName": "Sample Employer Name",
        "employerOrganisationPostcode": "2155",
        "employerOrganisationSuburb": "Kellyville",
        "employerOrganisationStreetNumber": "123",
        "employerOrganisationStreetName": "Sample Street"
      }
    },
    "validationId": "00000000-0000-0000-0000-000000000000",
    "validationTimestamp": "2020-08-25T06:59:56",
    "sequenceNumber": 1,
    "totalModelsInPayload": 100,
    "action": "Validate"
  }
}
```


21. Appendix

1. Limitations in Service Bus

Number of subscriptions per topic	Entity	Subsequent requests for creating additional subscriptions for the topic are rejected. As a result, if configured through the portal, an error message is shown. If called from the management API, an exception is received by the calling code.	2,000 per-topic for the Standard tier.
Number of SQL filters per topic	Entity	Subsequent requests for creation of additional filters on the topic are rejected, and an exception is received by the calling code.	2,000
Number of correlation filters per topic	Entity	Subsequent requests for creation of additional filters on the topic are rejected, and an exception is received by the calling code.	100,000
Size of SQL filters or actions	Namespace	Subsequent requests for creation of additional filters are rejected, and an exception is received by the calling code.	Maximum length of filter condition string: 1,024 (1 K). Maximum length of rule action string: 1,024 (1 K). Maximum number of expressions per rule action: 32.
Number of SharedAccessAuthorizationRule rules per namespace, queue, or topic	Entity, namespace	Subsequent requests for creation of additional rules are rejected, and an exception is received by the calling code.	Maximum number of rules per entity type: 12. Rules that are configured on a Service Bus namespace apply to all types: queues, topics.
Number of messages per transaction	Transaction	Additional incoming messages are rejected, and an exception stating "Cannot send more than 100 messages in a single transaction" is received by the calling code.	100 For both **Send()** and **SendAsync()** operations.
Number of virtual network and IP filter rules	Namespace		128

Figure 4 Service Bus quotas - https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-quotas

Student Sensitive Data

1. Summary

This document provides the technical design for the Student increment APIs.

It provides information on

- The data entities for Student interactions – this includes fields, entity relationships, and information to perform a technical validation on the entities to determine if the message is an acceptable format
- The API endpoints, request objects and response objects for Asynchronous interactions
- Invocation of the rules engine
- Persistence of the response to the staging datastore (Cosmos DB)
- Handling of errors and return of appropriate error codes and responses

2. Data Entities

This section defines the structure of the entities in the request/response objects.

The Min & Max length should reflect that required by the master database. Required indicates whether the field is required for the API. Mandatory fields are normally defined by business rules.

3. Valid Characters

The default character set is UTF-8. Only characters in the range 0x20 to 0x7e inclusive are permitted in payloads.

The double quote character " and the backslash \ character must be escaped when used within strings according to JSON standards to ensure that a payload is valid. A payload which contains invalid JSON will be rejected.

4. Request Payload Field value Truncation

In accordance with AVETMISS and State standards, behavioural validations truncate field values when received values are longer than the lengths indicated in this document. This is done prior to saving payloads to the Cosmos database.

Truncation operates by:

- Trimming trailing spaces from the received string value of a field
- Truncating if the character length of the value is more than the documented length and Can Be Truncated? Is set to Y in the user story behaviour rules table
- Trimming again to remove any trailing spaces

5. Initial Payload Validations

Validation Type	Value
NULL	Returns 400 BadRequest if payload is null
Invalid Payload	Returns 400 BadRequest if payload cannot be deserialized
Verify Identifier (URL/Payload)	Returns 400 BadRequest if supplied identifiers in URL don't match with identifiers supplied in body payload.

Not Found	Returns 404 NotFound – If requested entity was not found.

6. Response Codes

200 – OK

A 200 OK response will be returned for the GET requests if validation is complete and the entity has been mastered. The response will include a payload with the requested entity retrieved from the database, along with any validation details. See the respective endpoint responses for detailed information.

202 – Accepted

A 202 accepted response will be returned for the following requests:

1. POST the payload and after the payload passes the initial validation
2. DELETE, if it's a valid request
3. GET the validation status, while the validation process hasn't been completed

The response will include a payload with the following format.

```
{
    "validationId": "24924aae-b968-4086-a9bd-d7a7b9bd55e0",
    "validationTimestamp": "2020-08-13T02:34:53.251379Z",
    "status": "Validating"
}
```

400 – Bad Request

```
Response body
{
  "errors": [
    {
      "id": "0d014999-c408-48fc-8044-95c4264781a3",
      "detail": "Request payload must not be NULL",
      "code": "400"
    }
  ]
}
```

```
Response body
{
  "errors": [
    {
      "id": "0d014999-c408-48fc-8044-95c4264781a3",
      "detail": "One of the mandatory fields should be supplied",
      "code": "400"
    }
  ]
}
```

404 – Not Found

```
Response body
{
  "errors": [
    {
      "id": "71780af2-2a4b-4efb-b4e2-149d519232ac",
      "detail": "Requested entity was not found",
      "code": "404"
    }
  ]
}
```

413 – Request Entity Too Large

A 413 Request Entity Too Large response will be returned if any single request entity is too large to be published to the Azure Service Bus queue.

```
Response Body:
{
  "errors": [
    {
      "id": "0709427b-ae2d-48ad-b74c-49695c770392",
      "detail": "One or more records in the payload exceeded the acceptable maximum size",
      "code": "413"
    }
  ]
}
```

422 – Unprocessable Entity (Client error)

A 422 Unprocessable Entity response will be returned if payload is serialized but there are validation error(s).

```
},
        "validationResults": [
          {
            "ruleSet": "Basic Validations",
            "errorCount": 1,
            "warningCount": 0,
            "infoCount": 0
          }
        ],
        "errors": [
          {
            "code": "Required",
            "ruleSet": "Basic Validations",
            "source": "SensitiveData.ClientId",
            "message": "ClientId must not be blank, please correct and re-
submit."
          },
```

500 - Internal Server Error

```
{
    "errors": [
      {
```

```
        "id": "3e5fa897-51f3-4639-8ecb-b8958470b6ce",
        "detail": "Unexpected Error",
        "code": "500"
      }
    ]
  }
```


The response will also contain additional header values:

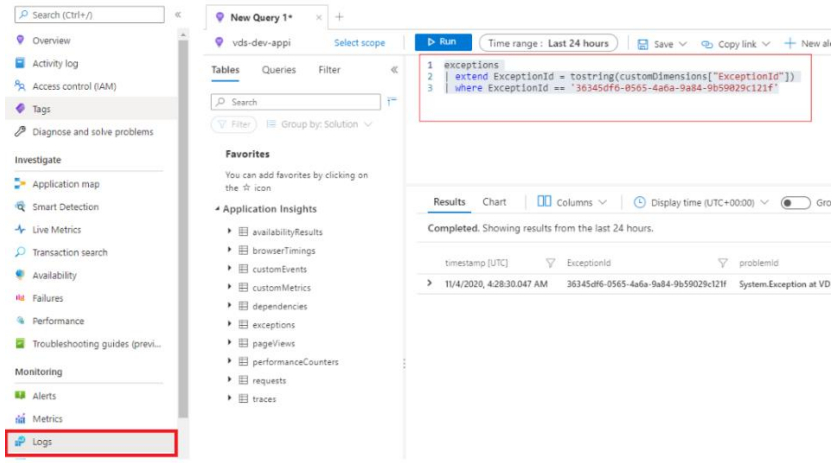
- Location - A URL the client should poll for a response status
- Retry-After – A timespan that indicates when the client should check back for the status. This is currently set to 60 seconds.


```
content-length: 1762
content-type: application/json; charset=utf-8
date: Fri, 14 Aug 2020 01:51:54 GMT
location: http://vds-dev-
sf.australiacentral.cloudapp.azure.com:8484/api/v1/sensitivedata/validations/4
ab4bef0-2279-4b1c-b805-4b4df65d2987
retry-after: 60
server: Kestrel
```


7. Error Response Fields

Element	Content
validationId	A unique value which identifies a request/response combination
validationTimestamp	A timestamp of when the request was processed
validationResults	An array which lists the different rulesets which were applied to the request. The elements within each item provide counts of the errors, warnings and info messages which occurred under the ruleset
errors	An array which lists every error which resulted from validating a request. The elements within each item identify the specific error code, the name of the rule and a human-readable message. Source is the name of the field which generated the errors. This is in JSON notation. This element may be an empty array if there are no errors
warnings	An array which lists every warning which resulted from validating a request. The elements within each item identify the specific warning code, the name of the rule and a human-readable message. Source is the name of the field in the payload in JSON notation which generated the warning. This is in JSON notation. This element may be an empty array if there are no warnings
infos	An array which lists every information message which resulted from validating a request. The elements within each item identify the specific information code, the name of the rule and a human-readable message. Source is the name of the field in the payload which generated the information message. This is in JSON notation. This element may be an empty array if there are no information messages
<entityname>	The original request payload (**SensitiveData**)

How to search for an exception in application insights? All exceptions are getting tracked now.



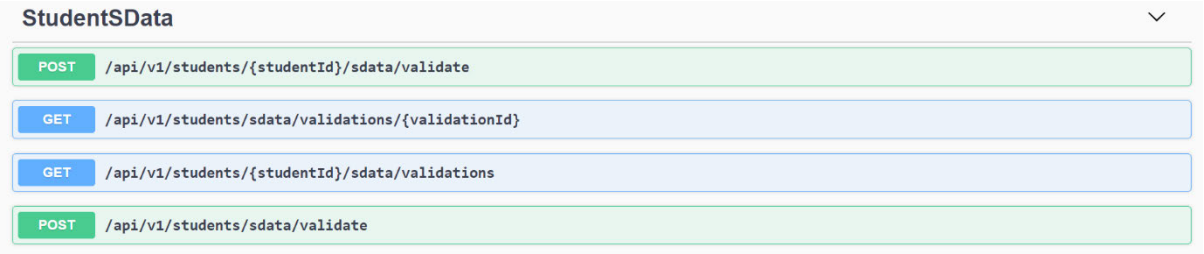
Query to search for an exception

```
exceptions

| extend ExceptionId = tostring(customDimensions["ExceptionId"])

| where ExceptionId == '36345df6-0565-4a6a-9a84-9b59029c121f'
```

8. Endpoints and HTTP return codes



Http Verb or end point	Header Parameter	Http status codes
POST (Create / Validate) e.g. Validate Single Student Sensitive Data POST /v1/students/{studentId}/sdata/validate	Training Organisation Identifier	202 - Accepted 400 - Cannot desterilize the payload Or Identifiers are not matched 403 - Forbidden (to be implemented once security design completed)
GET (Get Validation by Validation ID) e.g. Get Single/Batch Student Sensitive Data Validation GET v1/students/sdata/validations/{validationId}	Training Organisation Identifier	200 - OK (if the validation has occurred) 202 – Accepted (if the validation hasn't occurred yet) 400 - If entity identifier(s) was not supplied or not able to parse/bind 404 - Not Found – if the validation request doesn't exist

		403 - Forbidden (to be implemented once security design completed) 413 – RequestEntity TooLarge – If Service Bus is unable to publish due to large message size 422 - Unprocessable entity – If there is any validation error
GET (Get Validation by Student ID) e.g. Get latest Single Student Sensitive Data Validation GET v1/students/{studentId}/sdata/validations?latest=true	Training Organisation Identifier	200 - OK (if the validation has occurred) 405 – Not Allowed (method is not implemented for this resource) – if "false" is passed for latest flag 400 - If entity identifier(s) was not supplied or not able to parse/bind 404 - Not Found – if the validation request doesn't exist 403 - Forbidden (to be implemented once security design completed) 422 - Unprocessable entity – If there is any validation error

9. Training Organisation Identifier – Header parameter

Training Organisation Identifier	providerId	varchar (10)	Alphanumeric	N	N	10	1

10. Student Sensitive Data Request Payload Model

Field Name	Length	Data Type	Nullable?	Variable Length String Can be truncated?	Maximum Length	Minimum Length
**Disability Type Identifier (L)	varchar(2)	Numeric	N	N	2(300 characters in total)***	2
Indigenous Status Identifier (L)	varchar(1)	AlphaNumeric	N	N	1	1
Language Identifier (L)	varchar(4)	AlphaNumeric	N	N	4	4
Birth Country Identifier	Varchar(4)	AlphaNumeric	N	N	4	4
Disability Status (L)	varchar(1)	Numeric	Y	N	1	1
Disability Assess Type (L)	varchar(1)	Numeric	Y	N	1	1
Welfare Status (L)	varchar(1)	Numeric	Y	N	1	1
Welfare Type (L)	varchar(2)	Numeric	Y	N	2	2

In Social Housing Resident or Wait List	varchar(1)	Boolean	Y	N	1	1
Long Term Unemployed flag	varchar(1)	Boolean	Y	N	1	1

Nullability = Y/N
Variable Length String (Can be truncated) = Y/N
Max = Maximum length of value
Min = Minimum length of value
(L) = Lookup
**Comma separated list of disability type identifiers (Please check the sample payload below)
***Comma separated list of 300 characters maximum including commas (#5968)

Although all of the fields are optional, at least one of the fields needs to be provided.

For first time upload of Student Sensitive data, all fields are mandatory

The field values are required if supplied.

If disability type identifiers contain blanks, the blank values are removed

Request

```
{
 "name": "DisabilityTypeIds",
 "value": "11,,"
}
```

Response

```
{
 "name": "DisabilityTypeIds",
 "value": "11"
}
```

Similar implementation for Welfare Types.

If any value is passed in DisabilityType and WelfareType, existing values in database will be soft deleted and new ones inserted.

Condition of the payload:

- The payload will be in a form of an array of key-value pairs which will represent the fields in the table above
- Each of the fields are optional, meaning that the array in the payload does not have to contain all the key-value pairs
- If “Welfare Status” is 0, then no Welfare Types can be provided

Disability Flag

Disability Flag has been excluded from the API payload and won’t be stored in the database, as the value can be implied or calculated based on the Disability Type Identifier in the database for the reporting purposes.

Samples of possible payload for Single Student

Sample 1:

```
{
  "studentSData": {
    "studentId" : "Stu12312345678",
    "sData":[
      {
        "name" : "languageId",
        "value" : "0000-"
      }
    ]
  }
}
```

Sample 2:

```
{
  "studentSData": {
    "studentId" : "StuId123",
    "sData":[
      {
        "name" : "disabilityTypeIds",
        "value" : "12,23"
      },
      {
        "name" : "IndigenousStatusId",
        "value" : "2"
      },
      {
        "name" : "languageId",
        "value" : "0000"
      },
      {
        "name" : "birthCountryId",
        "value" : "0000"
      },
      {
        "name" : "disabilityStatus",
        "value" : "1"
      },
      {
        "name" : "disabilityAssessType",
        "value" : "1"
      },
      {
        "name" : "welfareStatus",
```

```
        "value" : "1"
      },
      {
        "name" : "welfareTypes",
        "value" : "01,02"
      },
      {
        "name" : "inSocialHousingResidentOrWaitList",
        "value" : "0"
      },
      {
        "name" : "longTermUnemployedFlag",
        "value" : "0"
      }
    ]
  }
}
```

Sample 3:

```
{
  "studentSData": {
    "studentId" : "StuId123",
    "sData":[
      {
        "name" : "IndigenousStatusId",
        "value" : "2"
      },
      {
        "name" : "languageId",
        "value" : "0000"
      },
      {
        "name" : "disabilityStatus",
        "value" : "0"
      },
      {
        "name" : "disabilityAssessType",
        "value" : "1"
      },
      {
        "name" : "welfareStatus",
        "value" : "0"
      },
    ]
  }
}
```

Sample 4:

```
{
  "studentSData": {
    "studentId" : "StuId123",
    "sData":[
      {
        "name" : "languageId",
        "value" : "0000"
      }
    ]
  }
}
```

Sample Payload for Multiple Students

```
{
  "studentSDataList": [
    {
      "studentSData": {
        "studentId" : "Stu12312345678",
        "sData"
          {
            "name" : "disabilityTypeIds",
            "value" : "12,23"
          },
          {
            "name" : "IndigenousStatusId",
            "value" : "2"
          },
          {
            "name" : "languageId",
            "value" : "0000"
          },
          {
            "name" : "disabilityStatus",
            "value" : "1"
          },
          {
            "name" : "disabilityAssessType",
            "value" : "1"
          },
          {
            "name" : "welfareStatus",
            "value" : "1"
          },
          {
            "name" : "welfareTypes",
            "value" : "01,02"
```

```
          },
          {
            "name" : "inSocialHousingResidentOrWaitList",
            "value" : "0"
          },
          {
            "name" : "longTermUnemployedFlag",
            "value" : "0"
          }
        ]
      }
    },
    {
      "studentSData": {
        "studentId" : "StuId123",
        "sData":[
          {
            "name" : "languageId",
            "value" : "0000"
          }
        ]
      }
    }
  ]
}
```

Response (202) for POST and GET (if still pending)

```
{
 "validationId": "24924aae-b968-4086-a9bd-d7a7b9bd55e0",
 "validationTimestamp": "2020-08-13T02:34:53.251379Z",
 "status": "Validating"
}
```

Response 200 for GET (if validation completed)

For single student

```
{
  "studentSData": {
    "providerId": "110069",
    "studentId": "StuId123",
    "sData": [
      {
        "name": "DisabilityTypeIds",
```

```
        "value": "12,23"
      },
      {
        "name": "IndigenousStatusId",
        "value": "2"
      },
      {
        "name": "LanguageId",
        "value": "0000"
      },
      {
        "name": "BirthCountryId",
        "value": "0000"
      },
      {
        "name": "DisabilityStatus",
        "value": "1"
      },
      {
        "name": "DisabilityAssessType",
        "value": "1"
      },
      {
        "name": "WelfareStatus",
        "value": "1"
      },
      {
        "name": "WelfareTypes",
        "value": "01,02"
      },
      {
        "name": "InSocialHousingResidentOrWaitList",
        "value": "0"
      },
      {
        "name": "LongTermUnemployedFlag",
        "value": "0"
      }
    ]
  },
  "validationResults": [
    {
      "ruleSet": "Basic Validations",
      "errorCount": 0,
      "warningCount": 0,
      "infoCount": 0
    }
  ],
```

```
  "errors": [],
  "warnings": [],
  "infos": [],
  "exceptionId": null,
  "validationId": "6e6f4898-dd9d-433a-82ac-84ca51f61b6e",
  "validationTimestamp": "2020-09-22T06:55:51.3739092Z",
  "status": "CompletedNotMastered"
}
```


Response 422 for GET (if validation completed with errors in payload)

For single student


```
{
"studentSData": {
    "providerId": "11006912345678",
    "studentId": "Stu12312345678",
    "sData": [
      {
        "name": "DisabilityTypeIds",
        "value": "123-,231-"
      },
      {
        "name": "IndigenousStatusId",
        "value": "2-"
      },
      {
        "name": "LanguageId",
        "value": "0000-"
      },
      {
        "name": "DisabilityStatus",
        "value": "1-"
      },
      {
        "name": "DisabilityAssessType",
        "value": "1-"
      },
      {
        "name": "WelfareStatus",
        "value": "1-"
      },
      {
        "name": "WelfareTypes",
        "value": "01-,02-"
      },
      {
        "name": "InSocialHousingResidentOrWaitList",
```

```
        "value": "0-"
      },
      {
        "name": "LongTermUnemployedFlag",
        "value": "0-"
      }
    ]
  },
  "validationResults": [
    {
      "ruleSet": "Basic Validations",
      "errorCount": 14,
      "warningCount": 0,
      "infoCount": 0
    }
  ],
  "errors": [
    {
      "code": "MaxLength",
      "ruleSet": "Basic Validations",
      "source": "StudentSData.ProviderId",
      "message": "ProviderId is too long and your data cannot be saved, please
 correct and re-submit."
    },
    {
      "code": "MaxLength",
      "ruleSet": "Basic Validations",
      "source": "StudentSData.StudentId",
      "message": "StudentId is too long and your data cannot be saved, please
correct and re-submit."
    },
    {
      "code": "MaxLength",
      "ruleSet": "Basic Validations",
      "source": "StudentSData.IndigenousStatusId",
      "message": "IndigenousStatusId is too long and the data cannot be saved,
 please correct and re-submit."
    },
    {
      "code": "MaxLength",
      "ruleSet": "Basic Validations",
      "source": "StudentSData.LanguageId",
      "message": "LanguageId is too long and the data cannot be saved, please
correct and re-submit."
    },
    {
      "code": "MaxLength",
      "ruleSet": "Basic Validations",
```

```
      "source": "StudentSData.DisabilityTypeIds[0].DisabilityTypeId",
      "message": "DisabilityTypeId is too long and the data cannot be saved, p
lease correct and re-submit."
    },
    {
      "code": "MaxLength",
      "ruleSet": "Basic Validations",
      "source": "StudentSData.DisabilityTypeIds[1].DisabilityTypeId",
      "message": "DisabilityTypeId is too long and the data cannot be saved, p
lease correct and re-submit."
    },
    {
      "code": "MaxLength",
      "ruleSet": "Basic Validations",
      "source": "StudentSData.DisabilityStatus",
      "message": "DisabilityStatus
is too long and the data cannot be saved, please correct and re-submit."
    },
    {
      "code": "MaxLength",
      "ruleSet": "Basic Validations",
      "source": "StudentSData.DisabilityAssessType",
      "message": "DisabilityAssessType
is too long and the data cannot be saved, please correct and re-submit."
    },
    {
      "code": "MaxLength",
      "ruleSet": "Basic Validations",
      "source": "StudentSData.WelfareStatus",
      "message": "WelfareStatus
is too long and the data cannot be saved, please correct and re-submit."
    },
    {
      "code": "MaxLength",
      "ruleSet": "Basic Validations",
      "source": "StudentSData.WelfareTypes[0].WelfareType",
      "message": "WelfareType
is too long and the data cannot be saved, please correct and re-submit."
    },
    {
      "code": "MaxLength",
      "ruleSet": "Basic Validations",
      "source": "StudentSData.WelfareTypes[1].WelfareType",
      "message": "WelfareType
is too long and the data cannot be saved, please correct and re-submit."
    },
    {
      "code": "MaxLength",
```

```
      "ruleSet": "Basic Validations",
      "source": "StudentSData.InSocialHousingResidentOrWaitList",
      "message": "InSocialHousingResidentOrWaitList
is too long and the data cannot be saved, please correct and re-submit."
    },
    {
      "code": "MaxLength",
      "ruleSet": "Basic Validations",
      "source": "StudentSData.LongTermUnemployedFlag",
      "message": "LongTermUnemployedFlag
is too long and the data cannot be saved, please correct and re-submit."
    }
  ],
  "warnings": [],
  "infos": [],
  "exceptionId": null,
  "validationId": "ca207c24-9da2-444d-b873-a2174109d961",
  "validationTimestamp": "2020-09-22T05:57:20.0442603Z",
  "status": "CompletedNotMastered"
}
```


For multiple student


```
{
  "studentSDataList": [
    {
      "studentSData": {
        "providerId": "11006912345678",
        "studentId": "Stu12312345678",
        "sData": [
          {
            "name": "DisabilityTypeIds",
            "value": "123-,231-"
          },
          {
            "name": "IndigenousStatusId",
            "value": "2-"
          },
          {
            "name": "LanguageId",
            "value": "0000-"
          },
          {
            "name": "DisabilityStatus",
            "value": "1-"
          },
          {
            "name": "DisabilityAssessType",
            "value": "1-"
```

```
          },
          {
            "name": "WelfareStatus",
            "value": "1-"
          },
          {
            "name": "WelfareTypes",
            "value": "01-,02-"
          },
          {
            "name": "InSocialHousingResidentOrWaitList",
            "value": "0-"
          },
          {
            "name": "LongTermUnemployedFlag",
            "value": "0-"
          }
        ]
      },
      "validationResults": [
        {
          "ruleSet": "Basic Validations",
          "errorCount": 14,
          "warningCount": 0,
          "infoCount": 0
        }
      ],
      "errors": [
        {
          "code": "MaxLength",
          "ruleSet": "Basic Validations",
          "source": "StudentSData.ProviderId",
          "message": "ProviderId is too long and your data cannot be saved, pl
ease correct and re-submit."
        },
        {
          "code": "MaxLength",
          "ruleSet": "Basic Validations",
          "source": "StudentSData.StudentId",
          "message": "StudentId is too long and your data cannot be saved, ple
ase correct and re-submit."
        },
        {
          "code": "MaxLength",
          "ruleSet": "Basic Validations",
          "source": "StudentSData.IndigenousStatusId",
          "message": "IndigenousStatusId is too long and the data cannot be sa
ved, please correct and re-submit."
```

```
        },
        {
          "code": "MaxLength",
          "ruleSet": "Basic Validations",
          "source": "StudentSData.LanguageId",
          "message": "LanguageId is too long and the data cannot be saved, ple
ase correct and re-submit."
        },
        {
          "code": "MaxLength",
          "ruleSet": "Basic Validations",
          "source": "StudentSData.DisabilityTypeIds[0].DisabilityTypeId",
          "message": "DisabilityTypeId is too long and the data cannot be save
d, please correct and re-submit."
        },
        {
          "code": "MaxLength",
          "ruleSet": "Basic Validations",
          "source": "StudentSData.DisabilityTypeIds[1].DisabilityTypeId",
          "message": "DisabilityTypeId is too long and the data cannot be save
d, please correct and re-submit."
        },
        {
          "code": "MaxLength",
          "ruleSet": "Basic Validations",
          "source": "StudentSData.DisabilityStatus",
  "message": "DisabilityStatus
is too long and the data cannot be saved, please correct and re-submit."
        },
        {
          "code": "MaxLength",
          "ruleSet": "Basic Validations",
          "source": "StudentSData.DisabilityAssessType",
          "message": "DisabilityAssessType
is too long and the data cannot be saved, please correct and re-submit."
        },
        {
          "code": "MaxLength",
          "ruleSet": "Basic Validations",
          "source": "StudentSData.WelfareStatus",
        "message": "WelfareStatus
is too long and the data cannot be saved, please correct and re-submit."
        },
        {
          "code": "MaxLength",
          "ruleSet": "Basic Validations",
          "source": "StudentSData.WelfareTypes[0].WelfareType",
```

```
          "message": "WelfareType
is too long and the data cannot be saved, please correct and re-submit."
        },
        {
          "code": "MaxLength",
          "ruleSet": "Basic Validations",
          "source": "StudentSData.WelfareTypes[1].WelfareType",
          "message": "WelfareType
is too long and the data cannot be saved, please correct and re-submit."
        },
        {
          "code": "MaxLength",
          "ruleSet": "Basic Validations",
          "source": "StudentSData.InSocialHousingResidentOrWaitList",
          "message": "InSocialHousingResidentOrWaitList
is too long and the data cannot be saved, please correct and re-submit."
        },
        {
          "code": "MaxLength",
          "ruleSet": "Basic Validations",
          "source": "StudentSData.LongTermUnemployedFlag",
          "message": "LongTermUnemployedFlag
is too long and the data cannot be saved, please correct and re-submit."
        }
      ],
      "warnings": [],
      "infos": [],
      "exceptionId": null,
      "validationId": "ca207c24-9da2-444d-b873-a2174109d961",
      "validationTimestamp": "2020-09-22T05:57:20.0442603Z",
      "status": "CompletedNotMastered"
    }
  ]
}
```


11. Sensitive data structure passed to InRule for validations


```
{
  "StudentSensitiveData": {
    "StudentId": "StuId123",
    "DisabilityTypeIds": [
      { "DisabilityTypeId": "12" },
      { "DisabilityTypeId": "23" }
    ],
    "IndigenousStatusId": "2",
```

```
    "LanguageId": "0000",
    "BirthCountryId": "0000",
    "DisabilityStatus": "1",
    "DisabilityAssessType": "1",
    "WelfareStatus": "1",
"WelfareTypes": [
      { "WelfareType": "01" },
      { "WelfareType": "02" }
    ],
    "InSocialHousingResidentOrWaitList": "0",
    "LongTermUnemployedFlag": "0",
    "ProviderId": "110069",
    "IsDeleted": false
  },
  "ValidationResults": [
    {
      "RuleSet": "Basic Validations",
      "ErrorCount": 0,
      "WarningCount": 0,
      "InfoCount": 0
    }
  ],
  "Errors": [],
  "Warnings": [],
  "Infos": null,
  "CanBeMastered": false,
  "Status": "Mastering Failure",
  "exceptionId": null,
  "ValidationId": "5e19ceaf-0dc9-44c2-a40f-87ae88d0a844",
  "ValidationTimestamp": "2020-09-22T07:39:31.250741Z"
}
```


12. Data Mapping between payload and Database fields

Field Name	Lookup		Saved into		
	Table	Field	Table	Field	Nullable
Disability Type Identifier (L)	Lookup.DisabilityType	DisabilityTypeId	Stakeholder.ProfileDisability	DisabilityTypeId	N
Indigenous Status Identifier (L)	Lookup.ATSICode	Code	Stakeholder.Profile	ATSICodeId	Y
Language Identifier (L)	Lookup.Language	LanguageCode	Stakeholder.Profile	LanguageId	Y
Birth Country Identifier	Pending	Pending	Pending	Pending	Pending
Disability Status (L)	Lookup.DisabilityStatus	DisabilityStatusId	Stakeholder.Profile	DisabilityStatus	Y
Disability Assess Type (L)	Lookup.DisabilityAssessType	DisabilityAssessTypeId	Stakeholder.Profile	DisabilityAssessType	Y
Welfare Status (L)	Lookup.WelfareStatus	WelfareStatusId	Stakeholder.Profile	WelfareStatus	Y

Welfare Type (L)	Lookup.WelfareType	WelfareTypeId	Stakeholder.ProfileWelfare	WelfareTypeId	N
In Social Housing Resident or Wait List	N/A	N/A	Stakeholder.Profile	SocialHousingResOrWait	Y
Long Term Unemployed flag	N/A	N/A	Stakeholder.Profile	LongTermUnemployedFlag	Y

13. Acceptance Criteria

User Story	Acceptance Criteria	Notes & Questions
3796	AC1	Training Organisation Identifier is expected in the request's header
	AC2	• The sensitive data creation or update will use its own API end point • The use of array or key-value pairs allow only some fields to be submitted. It doesn't require ALL fields or key-value pairs to be submitted at the same time
	AC3	Refer to section 10 about the removal of Disability Flag from the payload
	AC4	• The value for "Disability Type Identifiers" will be a comma separated list, to allow one or more values to be submitted. • ALL disability type identifiers relevant to the student must be submitted. Some sample cases: ○ If for example initially 3 disabilities are submitted, the 3 disabilities records will be created ○ If there were 3 disabilities initially and the next submission only contains 2 existing disabilities, that means 1 existing disability will be set to "expire" ○ If there were 2 disabilities initially and the next submission contains 1 new disability, that means the 2 existing disabilities will be set to "expire" and the new disability will be added ○ If there were 3 disabilities initially and the next submission contains 1 existing disability and 1 new disability, that means 2 existing disabilities will be set to "expire" and the new disability will be added • If the value of the disability type identifiers is "NONE", ALL disabilities of the student will be set to "expire" Assumptions: • An expired record is never re-activated once it has an end date, a new record will be added of the same type with a start date greater than the end date of the expired record

		• A record begin and end date should not be on the same day
	AC5	Refer to section 10 about the removal of Disability Flag from the payload
	AC6	This will be handled in the Behavioural validation
	AC7	This will be handled in the Behavioural validation
	AC8	This will be handled in the Behavioural validation Validation will only apply of the field / key exist in the key-pair array.
	AC9, AC10	Addressed by the Error Response Fields
4120	AC1	This will be done using the Create/Update Student sensitive data endpoint. Data would be supplied using name/value pairs. All fields are optional
	AC2, AC3, AC4	This will be handled in inRule
	AC5	The value for “Welfare Types” will be a comma separated list, to allow one or more values to be submitted. ALL relevant welfare types must be included. If Welfare Status is “1”, the Welfare Types must be provided. This will be handled in InRule If Welfare Status is “0”, the Welfare Types must not exist, and the WelfareTypeId will be set to “expire” If for example they initially send 3, that mean we add 3 welfare types. If the next time they send 2, we'll check all "non-expired" welfare types for the student and add if there's a new one and "expire" the records in the DB that are not in the list If welfare Types have empty item in the CSV list for example: ",99,,22," it should clean the blank entries before validation ("99,22") and should not throw error.
	AC6, AC7	This will be handled in the Behavioural validation
	AC8	This will be handled in the Behavioural validation?
	AC9, AC10	Addressed by the Error Response Fields