Base Api Service Environment (BASE)

**A Base For Rapidly Delivering Accreditable API Functionality to meet Business Objectives**

Solution Architecture Description

This document is a core Technical Domain SAD, intended to be read in conjunction with provided Extension SADs which describe Business Domain Specific Functionality as extensions.

Contents

[2 Document 1](#_Toc506120157)

[2.1 Purpose 1](#_Toc506120158)

[2.2 Intended Audience 1](#_Toc506120159)

[2.3 Reviewed By 2](#_Toc506120160)

[2.4 Versions 2](#_Toc506120161)

[2.5 Document Structure, Context, and Lifespan 2](#_Toc506120162)

[2.5.1 Views 2](#_Toc506120163)

[2.5.2 Structure 2](#_Toc506120164)

[2.5.3 Frameworks and References 3](#_Toc506120165)

[2.5.4 Living Document 3](#_Toc506120166)

[2.5.5 API Examples and ScreenGrabs 3](#_Toc506120167)

[2.6 Terms 3](#_Toc506120168)

[2.7 Citations and Bibliography 4](#_Toc506120169)

[2.8 References 5](#_Toc506120170)

[3 Business View 6](#_Toc506120171)

[3.1 Problem Definition 6](#_Toc506120172)

[3.2 Vision Statement 6](#_Toc506120173)

[3.3 Background 6](#_Toc506120174)

[3.3.1 Options Considered and Evaluated 7](#_Toc506120175)

[3.3.2 Recommendation 8](#_Toc506120176)

[3.3.3 Approval by Project Governance 8](#_Toc506120177)

[3.3.4 Further Modules and Extensions 9](#_Toc506120178)

[3.4 Purpose Alignment 10](#_Toc506120179)

[3.5 Principle Alignment 10](#_Toc506120180)

[3.6 Strategic Alignment 10](#_Toc506120181)

[3.7 Tactical Alignment 11](#_Toc506120182)

[3.8 Business Benefits 11](#_Toc506120183)

[3.9 Stakeholder Groups 11](#_Toc506120184)

[3.10 Scope 12](#_Toc506120185)

[3.10.1 In Scope 12](#_Toc506120186)

[3.10.2 Out of scope 13](#_Toc506120187)

[3.11 Constraints 13](#_Toc506120188)

[3.11.1 Principles 13](#_Toc506120189)

[3.11.2 Schedule 13](#_Toc506120190)

[3.11.3 Design Constraints 13](#_Toc506120191)

[3.12 Risks 13](#_Toc506120192)

[3.13 Controls 14](#_Toc506120193)

[3.14 Assumptions 14](#_Toc506120194)

[3.15 High Level Business Domain Requirements 15](#_Toc506120195)

[3.16 Governance 15](#_Toc506120196)

[4 Service Context View 16](#_Toc506120197)

[4.1 Synopsis 16](#_Toc506120198)

[4.2 System Context Summary 16](#_Toc506120199)

[4.3 Actor Context 19](#_Toc506120200)

[4.4 Integration – Organisation Business Service Systems 19](#_Toc506120201)

[4.5 Integration – Infrastructure Systems 19](#_Toc506120202)

[4.6 Integration - Service Reuse 21](#_Toc506120203)

[4.7 Design Scope Exclusions 21](#_Toc506120204)

[4.8 Unknowns 22](#_Toc506120205)

[4.9 Design and Operations Decision Register 22](#_Toc506120206)

[4.10 Technologies 23](#_Toc506120207)

[4.11 Protocols 23](#_Toc506120208)

[4.12 By Design 24](#_Toc506120209)

[5 Qualities View 25](#_Toc506120210)

[5.1 Abstract 25](#_Toc506120211)

[5.2 Synopsis 25](#_Toc506120212)

[5.3 Quality Objectives 25](#_Toc506120213)

[5.4 Security Qualities 26](#_Toc506120214)

[5.5 Functionality Qualities 27](#_Toc506120215)

[5.6 Usability Qualities 28](#_Toc506120216)

[5.7 Compatibility Qualities 28](#_Toc506120217)

[5.8 Reliability Qualities 29](#_Toc506120218)

[5.9 Performance Qualities 30](#_Toc506120219)

[5.10 Maintainability Qualities 31](#_Toc506120220)

[5.11 Portability Qualities 31](#_Toc506120221)

[5.12 Supportability Qualities 32](#_Toc506120222)

[6 Security Qualities View 33](#_Toc506120223)

[6.1 Synopsis 33](#_Toc506120224)

[6.2 ISO-27001:2013 33](#_Toc506120225)

[6.3 Execution Environment 33](#_Toc506120226)

[6.4 Relational Database Security 33](#_Toc506120227)

[6.4.1 Access Control 33](#_Toc506120228)

[6.4.2 In Transit 34](#_Toc506120229)

[6.4.3 At Rest 34](#_Toc506120230)

[6.5 Access Control Decision Flow 34](#_Toc506120231)

[6.6 Operation Access Control Model 35](#_Toc506120232)

[6.6.1 Core ACM 35](#_Toc506120233)

[6.6.2 Extension Module ACM 36](#_Toc506120234)

[6.7 Resource Data Classification 37](#_Toc506120235)

[6.8 Testing Data 37](#_Toc506120236)

[7 Functionality Qualities View 38](#_Toc506120237)

[7.1 Synopsis 38](#_Toc506120238)

[7.2 System Capabilities 38](#_Toc506120239)

[7.3 System Functionality 38](#_Toc506120240)

[7.3.1 Out of Scope Functionality 39](#_Toc506120241)

[7.4 Role Functionality 40](#_Toc506120242)

[7.5 Security Specialist Functionality 40](#_Toc506120243)

[7.5.1 Session and Operation Query Functionality 40](#_Toc506120244)

[7.5.2 Examples 41](#_Toc506120245)

[7.6 System Support Specialist Functionality 43](#_Toc506120246)

[7.6.1 System Tenant Management Functionality 43](#_Toc506120247)

[7.6.2 System Principal Management Functionality 46](#_Toc506120248)

[7.6.3 Diagnostics Functionality 48](#_Toc506120249)

[7.6.4 Example Use Cases 49](#_Toc506120250)

[7.7 Business Support Specialist Functionality 50](#_Toc506120251)

[7.7.1 Example Query Responses 50](#_Toc506120252)

[7.7.2 Example Use Cases 51](#_Toc506120253)

[7.8 Notification Functionality 52](#_Toc506120254)

[7.8.1 Examples 53](#_Toc506120255)

[7.8.2 Use Cases 53](#_Toc506120256)

[7.9 Media Functionality 55](#_Toc506120257)

[7.9.1 Example 55](#_Toc506120258)

[8 Information View 57](#_Toc506120259)

[8.1 Sumary 57](#_Toc506120260)

[8.2 Data Classification 57](#_Toc506120261)

[8.3 Data Ownership and Stewardship 58](#_Toc506120262)

[8.4 Data Sources 58](#_Toc506120263)

[8.5 Data Schemas and Entities 58](#_Toc506120264)

[8.6 Data Security 59](#_Toc506120265)

[8.7 Testing Constraints 59](#_Toc506120266)

[8.8 Data Provisioning 60](#_Toc506120267)

[8.9 Data Access Auditing 60](#_Toc506120268)

[8.10 Data Retention 61](#_Toc506120269)

[8.11 Database Access 61](#_Toc506120270)

[8.12 Database Users and Logins 62](#_Toc506120271)

[8.13 Database Security 62](#_Toc506120272)

[8.14 Operation Log Policies 62](#_Toc506120273)

[8.15 Data Backup Policies 62](#_Toc506120274)

[8.16 Data Archiving 62](#_Toc506120275)

[8.17 Data Integration 63](#_Toc506120276)

[8.18 Data Exportation 63](#_Toc506120277)

[8.19 Data Quality 63](#_Toc506120278)

[8.19.1 Validation 63](#_Toc506120279)

[8.20 Benefit of Information Collection 63](#_Toc506120280)

[9 Integration View 65](#_Toc506120281)

[9.1 Synopsis 65](#_Toc506120282)

[9.2 Azure Active Directory Service Integration 65](#_Toc506120283)

[9.3 Azure Key Vault Services Integration 66](#_Toc506120284)

[9.4 Azure SQL Services Integration 66](#_Toc506120285)

[9.5 Azure Storage Account Services Integration 67](#_Toc506120286)

[9.5.1 Private Blob Storage 68](#_Toc506120287)

[9.5.2 Publicly Addressable Blob Storage 68](#_Toc506120288)

[9.6 Azure Document Database Service 69](#_Toc506120289)

[9.7 Azure Redis Cache 69](#_Toc506120290)

[9.8 AntiMalware Service Integration 70](#_Toc506120291)

[9.9 Authentication Service Integration 71](#_Toc506120292)

[9.9.1 Open Id Connect (OIDC) Authentication Protocol 71](#_Toc506120293)

[9.9.2 Modern Client side generated Views 72](#_Toc506120294)

[9.9.3 Legacy Service side generated MVC Applications 72](#_Toc506120295)

[9.9.4 Machine to Machine Authentication 72](#_Toc506120296)

[9.10 External Data Sources 73](#_Toc506120297)

[9.10.1 Identity Attribute Services 73](#_Toc506120298)

[9.10.2 On Prem Operational System Access 74](#_Toc506120299)

[9.10.3 Synopsis 74](#_Toc506120300)

[9.10.4 Option 1: Reliance on VPN 75](#_Toc506120301)

[9.10.5 Option 2: Use of Data Factory 77](#_Toc506120302)

[9.10.6 Option 3: Custom Service Agent 78](#_Toc506120303)

[9.10.7 On-Premise System Recommendation 79](#_Toc506120304)

[9.11 Client Service Integration 79](#_Toc506120305)

[10 Delivery View 81](#_Toc506120306)

[10.1 Synopsis 81](#_Toc506120307)

[10.2 Sequence 81](#_Toc506120308)

[10.3 Application Lifecycle Management (ALM) Suite 81](#_Toc506120309)

[10.4 Organisation’s ALM Account 82](#_Toc506120310)

[10.5 ALM Project Space 82](#_Toc506120311)

[10.6 ALM Version Control Service 83](#_Toc506120312)

[10.6.1 Multiple Version Control Repositories 83](#_Toc506120313)

[10.7 First Build 84](#_Toc506120314)

[10.7.1 Trunk Branch Protection 84](#_Toc506120315)

[10.8 ALM Build Management Service 85](#_Toc506120316)

[10.9 ALM Deployment Management Service 85](#_Toc506120317)

[10.9.1 ARM Templates 85](#_Toc506120318)

[10.9.2 ARM Template Parameters, variables and outputs 86](#_Toc506120319)

[10.9.3 ARM Template Invocation Sequence 86](#_Toc506120320)

[10.9.4 ARM Design Constraints 87](#_Toc506120321)

[11 Development View 88](#_Toc506120322)

[Synopsis 88](#_Toc506120323)

[11.1 Constraints 88](#_Toc506120324)

[11.1.1 Development Principles 88](#_Toc506120325)

[11.2 Prerequisites 88](#_Toc506120326)

[11.2.1 Use of a Distributed Version Control 88](#_Toc506120327)

[11.2.2 Continuous Delivery Service 89](#_Toc506120328)

[11.2.3 Verified Code Branch Integration 89](#_Toc506120329)

[11.2.4 Domain Driven Design Component Approach 90](#_Toc506120330)

[11.2.5 Test Driven Development Process 90](#_Toc506120331)

[11.3 Accessibiliy 91](#_Toc506120332)

[11.3.1 Acccessibility Obligations 91](#_Toc506120333)

[11.3.2 Usability Obligations 92](#_Toc506120334)

[11.4 Risks 92](#_Toc506120335)

[11.5 System Components 92](#_Toc506120336)

[11.6 System Components 93](#_Toc506120337)

[11.7 Service Facade ODATA API Controllers 94](#_Toc506120338)

[11.8 Core Infrastructure Services 95](#_Toc506120339)

[11.8.1 Service Design 96](#_Toc506120340)

[12 Testing View 97](#_Toc506120341)

[12.1 Synopsis 97](#_Toc506120342)

[12.2 ‘Delivery Contracts’ versus ‘Tests’ 97](#_Toc506120343)

[12.2.1 Driver 97](#_Toc506120344)

[12.3 Delivery Contract Sequence 98](#_Toc506120345)

[12.4 Delivery Contract Definitions 98](#_Toc506120346)

[12.5 Delivery Contract Development 98](#_Toc506120347)

[12.6 Delivery Contract Development Scope 99](#_Toc506120348)

[12.7 Delivery Contract Execution 100](#_Toc506120349)

[12.8 Security Concerns 100](#_Toc506120350)

[12.9 Privacy Concerns 101](#_Toc506120351)

[13 Operational View 102](#_Toc506120352)

[13.1 Synopsis 102](#_Toc506120353)

[13.2 Operating System and Execution Environment Patching 102](#_Toc506120354)

[13.3 Domain Name System (DNS) Records 102](#_Toc506120355)

[13.4 System Configuration 102](#_Toc506120356)

[13.5 System Diagnostics Tracing 103](#_Toc506120357)

[13.6 System Monitoring 103](#_Toc506120358)

[13.7 System Alerting 103](#_Toc506120359)

[13.8 Operation Auditing 104](#_Toc506120360)

[13.9 Data Archiving 104](#_Toc506120361)

[13.10 Support Reports 104](#_Toc506120362)

[13.11 High Availability 104](#_Toc506120363)

[13.12 High Availability – Sustained and Burst Load Handling 104](#_Toc506120364)

[13.13 Failover – Addressing Device Failure 105](#_Toc506120365)

[13.14 FailOver – Active Geo Replication 105](#_Toc506120366)

[13.15 Fail Over - Testing 106](#_Toc506120367)

[13.16 Backup 106](#_Toc506120368)

[13.17 Restore 106](#_Toc506120369)

[13.18 Backup Restoration Testing 107](#_Toc506120370)

[13.19 Disaster Recovery 107](#_Toc506120371)

[13.20 Business Continuity Plan 107](#_Toc506120372)

[13.21 Disengagement Plan 107](#_Toc506120373)

[13.22 Terms and Conditions 107](#_Toc506120374)

[13.23 Operational Manual 107](#_Toc506120375)

[13.24 Operational Support 108](#_Toc506120376)

[13.25 Operational Management 108](#_Toc506120377)

[13.26 Actions 108](#_Toc506120378)

[14 Appendices 109](#_Toc506120379)

[14.1 Glossary 109](#_Toc506120380)

[14.2 Summary of OIDC 114](#_Toc506120381)

[14.2.1 Description 114](#_Toc506120382)

[14.2.2 Terms 114](#_Toc506120383)

[14.2.3 OIDC Integration 115](#_Toc506120384)

[14.2.4 OIDC Brokered Integration 116](#_Toc506120385)

[14.2.5 Flows 117](#_Toc506120386)

[14.2.6 Authorisation Code Grant Flow 117](#_Toc506120387)

# Document

This section of the document describes the document’s purpose, audience, and location within the Organisation’s common document repositories.

## Purpose

The purpose of this [Solution Architecture Description (SAD) Document](#_Glossary) is to communicate the solution architecture to key stakeholders, in order to verify it meets their objectives, the proposed architecture is correct, complete and fit for purpose.

The SAD uses industry standards ([ISO/IEC/IEEE 42010:2011](#_BASE_Glossary)), conventions, notation and terms to:

* Summarize the project business drivers, goals, objectives, scope, and high level requirements,
* Summarize the context in which the system operates, the information it manages and controls access to,
* Describe how the stated business objectives can be met by automated services,
* Record important decisions and their rationale, including alternatives considered and rejected,
* Outline outstanding issues requirement management and resolution.

## Intended Audience

This document is intended for review and reference by different stakeholders, for different purposes:

* **Project Delivery Managers:** to understand the relationship and responsibilities of key stakeholders and their various objectives, where understanding assists the planning, co-ordination of design, development, delivery, maintenance and support stakeholders in order to deliver Value, while adhering to agreed Constraints.
* **Business Stakeholders:** to ensure their business objectives are understood, and an overview of how they will be addressed by automation
* **Enterprise Architects:** reviewing enterprise business and infrastructure service integration
* **Solution Architects:** of planned and integrated systems
* **Security Architects:** reviewing technical controls
* **Infrastructure Architects:** reviewing deployment
* **Review Boards:** to review Regulations, Standards, Practises and Organisation and Project [Principles](#_Principles) are met,
* **Designers and Developers:** implementing the solution
* **Operational and Support Staff:** to gain an overview of the system enough to ensure their deployment, maintenance and support requirements are met
* **Testers:** to gain an overview of the system to participate in developing assurances that expected qualities are met.

## Reviewed By

Feedback incorporated via reviews by the following:

* Sandy Britain, Enterprise Architect, BE&S ICT Strategy, Planning & Architecture.
* Mike O’Connor, Enterprise Architect, Security, BE&S ICT Strategy, Planning & Architecture
* Martin Rothbaum, Enterprise Architect, Security, BE&S ICT Strategy, Planning & Architecture
* Mehrdad Fatemi, Architect, BE&S ICT Strategy, Planning & Architecture
* Jeff Grove, Security Specialist, BE&S ICT Assurance,
* Nitima Pednekar, Security Specialist, BE&S ICT Assurance
* Grayson Mitchell, Solution Architect, BE&S ICT SPA
* Andy Dockerty, Architect, BE&S ICT Strategy, Planning & Architecture
* Quintes van Aswegen, Architect, BE&S ICT Strategy, Planning & Architecture
* Stuart McGrigor, Architect, BE&S ICT Strategy, Planning & Architecture
* Dries Venter, Cloud Systems Engineer | BE&S Operations & Infrastructure Services Management
* Diane Simpson, Infrastructure Architect, BE&S ICT Strategy, Planning & Architecture

## Versions

* **0.1:** 15/11/2017: Initial Draft
* **0.2:** 10/01/2018: Submitted for Initial Feedback
* **0.3:** 15/01/2018: Incorporate Feedback
* **0.8:** 27/01/2018: Incorporate Feedback. Added Security View
* **0.9:** 29/01/2018: Submittal to TRW
* **1.0:** 12/02/2018: Submitted to ESDA

## Document Structure, Context, and Lifespan

### Views

The document follows the industry accepted convention ([ISO/IEC/IEEE 42010:2011](#_BASE_Glossary)) of describing complex models as a series of *Views* intended for different audiences. Archimate and Unified Modeling Language (UML) notation is used where appropriate.

### Structure

The structure of the document errs towards the use of section headers as opposed to free flowing descriptive text, for faster location of pertinent information, via the document’s Table of Contents (TOC).

### Frameworks and References

Where appropriate, this document references independently developed sector and organisation reference artefacts.

### Living Document

This document is periodically updated to reflect appropriately governance endorsed changes as they emerge. Structure and content may change to reflect feedback on content focus and clarity.

### API Examples and ScreenGrabs

The system is still under current development and ongoing design. Any cut/pasted API calls are to be taken as indicative only, to facilitate understanding and discussion purposes.

## Terms

Solutions benefit from problem definitions and designs described using a restrained vocabulary of correct terms.

The [Glossary](#_Glossary) within the [Appendices](#_Appendices) provides an extended list of domain specific terms and acronyms used within this document.

The following are required to understand this document.

Note:  
Having been queried several times during review, special attention to the terms Principal and Principle.   
There is a difference.

* **View:** an industry term for a Section of a SAD used to describe a complex model from a specific audience’s Viewpoint (refer to [ISO/IEC/IEEE 42010:2011](#_Glossary)).
* **Quality: a positive or negative trait of a service.**
* **ISO-25010:**  the current authoritative standards-based list of positive qualities which stakeholders consider valuable when assessing a sevice. ISO-25010 supersedes the earlier ISO-9126 standard, and earlier-still FURPS acronym, allowing one to leverage a lot of experienced independent thought on the matter of valuable qualities, and allows for a common agreed terminology which removes ambiguity as to what is actually being asked for, or being delivered. The top level expected Qualities are: Security, Functionality, Usability, Compatibility, Reliability, Performance, Maintainability, Portability, Supportabilty.
* **Service:** a manual or automated service to render assistance to others.
* **Business Service:** a *service*, that may in turn rely on or more automated services, commonly referred to a solution.
* **Principle:** (distinct from Principal) a fundamental truth or proposition that serves as the foundation for a chain of reasoning.
* **API:** Application Programming Interface. A misleading term, as modern APIs do not provide *programmability* in any shape or form, and only provide Browse, Read, Add, Edit, Delete (BREAD) access to permitted service resources by appropriate users.
* **REST:** Representational State Transfer protocol.
* **Person:** a Natural Person (Individual) or Legal Person (Organisation).
* **PersOrg:** a common industry equivalent for the term Person.
* **Organisation:** a group of one or more Persons.
* **Subject:** any entity that requests access to an object[[1]](#footnote-1).
* **Principal:** (distinct from Principle) A uniquely identifiable Person, Service, or System Subject that accesses a System[[2]](#footnote-2).
* **User:** A subset of Principal.
* **Actor:** a commonly used term for Principal.
* **Attribute:** a Key/Value pair describing summarizing a property of an entity.
* **Claim:** an Attribute distributed by a trusted authoritative source (often an Identity Provider (IDP)), often cryptographically signed.
* **Credential:** a secured and undisclosed Claim by an Identity Provider to verify the Identity is who they say they are, before issuing an Identity Token (eg: Password).
* **System Permission:** a logical system permission associated to an Identity Token.
* **System Role:** a logical grouping of System Permissions, associated to an Identity Token.
* **Identity:** a unique set of claims representing one of many identities belonging to a Person.
* **Identity Token:** a short-lived digital token issued by an Identity [Token] Provider (IDP) transmitted with messages to uniquely identify the Subject sending the message.
* **CRUD:** a commonly used industry term for data asset operations (Create, Read, Update, Delete) that is logically inappropriate for meeting common target objectives. Refer to CRUS.
* **CRUS:** an acronym for data asset operations (Create, Read, Update, State) that addresses Logical deletes as State changes (providing a means to Undo, Merge and optionally Archive if and as required).

## Citations and Bibliography

* Fielding, R. (n.d.). *Representational state transfer*. Retrieved from Wikipedia.org: https://en.wikipedia.org/wiki/Representational\_state\_transfer
* ICT, N. (2016, October). *API Standard and Guidelines*. Retrieved from ICT.govt.nz: https://www.ict.govt.nz/guidance-and-resources/standards-compliance/api-standard-and-guidelines/
* IETF. (2008). *The Transport Layer Security (TLS) Protocol, Version 1.2*. Retrieved from IETF.ORG: https://tools.ietf.org/html/rfc5246
* ISO. (2016). *ISO/IEC 20802-1:@016*. Retrieved from ISO.ORG: https://www.iso.org/standard/69208.html
* ISO. (2016). *ISO/IEC 20802-2:2016*. Retrieved from ISO.ORG: https://www.iso.org/standard/69209.html
* ISO. (2017). *ISO/IEC 21788:2017*. Retrieved from ISO.org: https://www.iso.org/standard/71616.html
* Microsoft. (2017, March 8). *Microsoft says Google's Cloud Reliability claims versus Azure and Amazon Web Services does not Compute*. Retrieved from GeekWire: https://www.geekwire.com/2017/microsoft-says-googles-cloud-reliability-claim-vs-azure-amazon-web-services-not-compute/
* Microsoft. (2017). *Service Level Agreeements*. Retrieved from microsoft.com: https://azure.microsoft.com/en-us/support/legal/sla/
* MOE. (2018). *Organisation Principal Catalog*. Retrieved from Confluence: http://confluence.moe.govt.nz/display/BSA/Principle+Catalog
* Project Journal. (2016). *66% of IT Projects Fail*. Retrieved from Project Journal: https://projectjournal.co.uk/2016/02/20/66-of-it-projects-fail/
* Sigal, S. (2018). *Core Service Principles*. Retrieved from Leanpub: https://leanpub.com/principles/
* W3C. (2006, September 29). *Extensible Markup Language (XML) 1.1 (Second Edition)*. Retrieved from W3C.org: https://www.w3.org/TR/xml11/
* Wikipedia. (2018). *Semantic URL*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Semantic\_URL

## References

* [Integrations Architecture Options Paper](https://docs.moe.govt.nz/navigator/bookmark.jsp?desktop=MinistryDocuments&repositoryId=FNMinistryDocuments&repositoryType=p8&docid=DMOE%2C%7B1C5A444A-9678-4E41-96C4-A8365566408A%7D%2C%7B53477922-DB61-C106-868B-614F24B00000%7D&mimeType=application%2Fvnd.openxmlformats-officedocument.wordprocessingml.document&template_name=DMOE&version=current&vsId=%7BCD39E2B1-4126-C2B7-ACB8-61072B500006%7D).

# Business View

This section of the document summarizes key business drivers, goals and new value delivered by delivering the Solution.

## Problem Definition

Business stakeholders wish to provide interoporable automated services – specifically APIs -- to meet market requested needs, yet the cost of doing so far exceeds the value derived.

A key reason for this is that the organisation currently builds custom systems from the ground up, often duplicating effort and expense, security mistakes (and their necessary remediation and/or risk controls), leading to delayed deliveries, cost overruns and reputation impact. Simply put, our IT projects have a failure rate greater than the IT industry’s failure rate of over 66% (Project Journal, 2016).

## Vision Statement

Mature organisations improve their ability to successfully meet business expectations by leveraging previously made investments and knowledge gains in custom digital system development.

Mature organisations do this by developing and maintaining a core “starting point” appropriate to current norms and the organisation’s specific objectives.

A base “starting point” appropriate for our sector would be a secure, cloud enabled, modular base framework that encapsulate both industry-wide and sector-specific identity and integration patterns.

Such a base would benefit business specific projects by removing the long lead times required to repeat the very same development, testing, and accreditation efforts and costs of previous projects.

A means of delivering this mature, cost effective and business responsive baseline “starting point” is described within this document.

This document is the foundation for a series of smaller “micro-SAD” documents describing specific APIs Modules commissioned by various business groups to be hosted on the common ITC service provided Base.

## Background

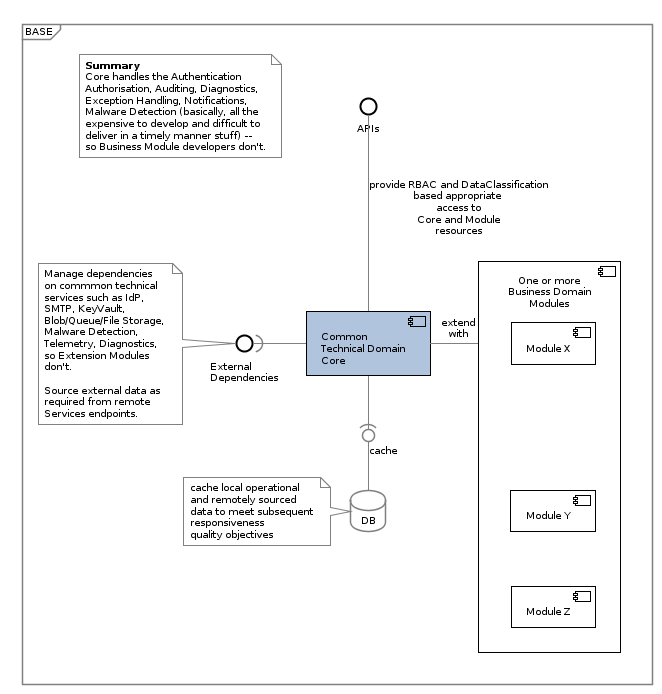
The need for this Solution was again made when a business unit approached the Organisation’s ITC unit for a means to expose an API of UNCLASSIFIED data to its sector and business users.

The discussion that ensued can be summarized as having covered the following points:

* API stands for Application Programming Interface, and therefore
* could not be delivered without an Application being available, on which to develop an Interface, and
* an Organisation managed Application would not obtain the Certification and Accreditation required to go live without meeting – in addition to the original Business Functionality – a minimum accreditable list of ISO-25010 Quality Requirements (Security, Usability, Supportability, Maintainability, etc.) required to not negatively impact other business units.
* The explanation of why these other objectives had to be included were given as:
  + The system’s functionality – whatever its Data Classification – must be Secured against tamperability, Accessibility and/or Economic Denial of Service attacks,
  + that its development meet supportable norms in order to meet unanticipated and changing business needs to make the initial investment worthwhile,
  + Be able to monitor for unexpected behaviour in order to better respond to end user issues.
  + All to get better value for the investments made, while protecting the organisation’s reputation.

The agreement reached was to develop the Solution in a modular manner, in two separate Deliverables:

* **Core**: system functionality that would be common to a vast majority of custom Systems the organisation would be interested in developing, which would handle all the non-business specific tasks, such as request routing, User management, Access control management, auditing, notifications, media management, and diagnostics.
* **Module:** a discrete Business specific Module, containing only the business specific code (in this specific first case, refer to the [Education Count Schools Directory API BASE Extension Module SAD](https://docs.moe.govt.nz/navigator/bookmark.jsp?desktop=MinistryDocuments&repositoryId=FNMinistryDocuments&repositoryType=p8&docid=DMOE%2C%7B1C5A444A-9678-4E41-96C4-A8365566408A%7D%2C%7BABB26F26-BFB8-CB5A-87E2-611C58B00000%7D&mimeType=application%2Fvnd.openxmlformats-officedocument.wordprocessingml.document&template_name=DMOE&version=current&vsId=%7B34321101-6897-C73C-AD48-60D814400000%7D)).



### Options Considered and Evaluated

Options considered prior to selecting this approach to delivering a solution included:

* **Not Deliver:** essentially express to business users that custom solutions could not cost effectively be developed, tested, accredited and deployed by ITC Services to meet their business needs.
* **Leverage Current Patterns:** use Application Support Services skillsets and legacy organisation managed infrastructure. It was agreed with the project sponsors that Small Business Systems as a group is appropriately skilled to meet support development needs but there was a risk that as a group were maybe not ready to begin new app development following current development patterns and technologies.
* **Solely leverage newly available Cloud Services:** use Azure’s Data Factory Services to combine both ETL operations and expose an API. This would effectively:
  + be misusing Data Factory functionality (it can be made to expose an API, even if it was not its primary intended purpose),
  + be repeating in the cloud design patterns that have been designated as an integration pattern to be avoided in the future due to its recognized negative impact on the Organisation’s ability to remain loosely coupled, modular and adaptable,
  + required scarce specialised cloud based ETL skillsets,
  + produce a brittle and expensive to maintain system, the outcome of which might have met the original business unit’s initial requirements for an UNCLASSIFIED data API endpoint, but would have resulted in a solution that who’s apis could not be versioned to protect the organisation’s reputation,
  + would include a perpetual un-optimized service cost liability, while being only a single use solution that would not have generated any reusable business result for its investment.

Data Factory’s primary functionality is a cloud based ETL service between systems, and less of an API Versioning, Security Role Management and secure data exposure system.   
Data Factory – and SQL Server SSIS before it – remains simply a powerful ETL and not a full-fledged Development Platform, even if it can be stretched to meet an initial set of low hanging business objectives, but not reach much further easily or cost-effectively.

* **Leverage a mix of Custom Development and available Cloud Services:** develop a core system to handle Users, Principals, Sessions, Operation Auditing and API Versioning to a Government Organisation satisfactory level, while leveraging cost effective Azure Cloud services in general, as well as leveraging Azure Data Factory for the communication between legacy and other organisation data sources that do not have an API.
  + **Expected benefits:**
    - Solves centrally common integration requirements
    - Code base can be used by other agencies, within and external to the Sector.
    - Can base can be cloud, as well as reused for on-prem (see [Integration View](#_External_Data_Sources)’s documentation of on-prem integration patterns) systems.
    - Code base can be used for both machine 2 machine API connections, as well as systems that need User Interfaces.

### Recommendation

The recommendation made was that both the long term investment and short term investment was towards the last option “Leveraging a mix of Custom Development and Azure Cloud Services”.

### Approval by Project Governance

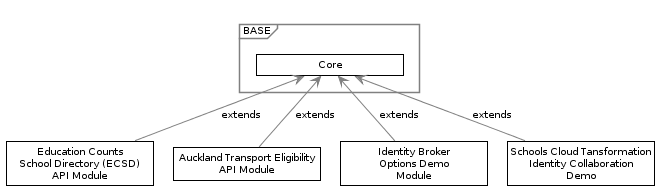
Approval was obtained to follow the recommended approach, while prioritising the development and management effort to meeting the initial business needs – exposing UNCLASSIFIED School directory data – while meeting the minimum viable ISO-25010 Quality Requirements of other groups (Security, Maintainability, etc.).   
  
Specifically agreed was to deliver the Core framework with sufficient functionality to deliver data via a Versioned API over a non-tamperable, confidential communication, using a code base that was maintainable and delivered using an automated build pipeline.  
  
Specifically agreed was the understanding that subsequent effort would be required to complete BASE to the level as designed.

### Further Modules and Extensions

In the short time since the original agreement was made to proceed in a modular way, more demand for the reuse of Core has emerged and MOUs have been agreed to.

The list at present includes:

* The (original) [Education Counts Schoold Directory API Module](https://docs.moe.govt.nz/navigator/bookmark.jsp?desktop=MinistryDocuments&repositoryId=FNMinistryDocuments&repositoryType=p8&docid=DMOE%2C%7B1C5A444A-9678-4E41-96C4-A8365566408A%7D%2C%7BABB26F26-BFB8-CB5A-87E2-611C58B00000%7D&mimeType=application%2Fvnd.openxmlformats-officedocument.wordprocessingml.document&template_name=DMOE&version=current&vsId=%7B34321101-6897-C73C-AD48-60D814400000%7D)
* The [Auckland Transport Entitlement API Module](https://docs.moe.govt.nz/navigator/bookmark.jsp?desktop=MinistryDocuments&repositoryId=FNMinistryDocuments&repositoryType=p8&docid=MoEFolder%2C%7B1C5A444A-9678-4E41-96C4-A8365566408A%7D%2C%7BA3F04430-EF99-C681-85C6-61023CB00000%7D&mimeType=folder&template_name=MoEFolder)
* The Identity Broker Demo Module
* The Schools Clouds Transformation Demo Module, which includes a UX component that is also built in a Modular manner (refer to the Base UX – [Solution Architecture Description Document](https://docs.moe.govt.nz/navigator/bookmark.jsp?desktop=MinistryDocuments&repositoryId=FNMinistryDocuments&repositoryType=p8&docid=DMOE%2C%7B1C5A444A-9678-4E41-96C4-A8365566408A%7D%2C%7B52E1949A-6E08-C2B5-8530-60FDE6D00000%7D&mimeType=application%2Fvnd.openxmlformats-officedocument.wordprocessingml.document&template_name=DMOE&version=released&vsId=%7B74AFC1EE-DCE9-CC09-87BE-60FDE6D00000%7D) in FileNet).
* Potentially providing augmented FIRST School & School Roll master data to iERS.

[](http://skysigal.com/lib/plugins/plantuml/img.php?width=0&height=0&title=PlantUML%20Graph&align=&version=2011-07-16&md5=fc573642975d197583b335eb154ac530)

Note:  
More BASE Modules are expected to be commissioned to reuse Core BASE functionality.   
Interested parties will include Resource Teacher Learning Behaviour (RTLB).

Supported by the above evidence, it is clear that there is pent-up demand for a solution suchas as BASE to provide in an efficient way to provide APIs in a accessible yet secure audited manner.

## Purpose Alignment

By addressing the business problem, the solution contributes to Ministry’s purpose statement:

We shape an education system that delivers equitable and excellent outcomes  
He mea tārai e mātou te mātauranga kia rangatira ai, kia mana taurite ai ōna huanga

## Principle Alignment

The solution aligns with current ESDA approved Sector and Organisation Principles, as well as the following [Key Service Delivery Principles](#_Key_Service_Delivery)[[3]](#footnote-3):

* **Sector First Enablement:** principled, regulated, opportunity-focused services to enable Sector people and organisations operate more effectively.
* **Secure Open Direct Access:** confidential, usable, appropriate, audited access anytime from anywhere from any device to reliable, timely, correctable, versioned information using current open integration standards, through the least layers.
* **Open Cloud Services:** cloud first, services first, suite first, web API first, open standards first, open source first.
* **Compliant and Accreditable:** Privacy Act 1993, NZDIMP, NZISM, NZGWAS, NZGWUS, NZGOAL, Public Records Act 2005, Education Act 1989, ESDA.
* **Collaborative Swift Value Delivery:** automated rapid iterative delivery of validated value to sector people by collaborative cross functional teams.

In addition, the solution aligns with the [Organisation’s Detailed Service Delivery Principles](#_Detailed_Service_Delivery) which provide Objective, Service, Security, Information, Integration, Delivery and Development guidance.

## Strategic Alignment

* Deliver to the Ministry of Education Four Year Plan 2016-2020:
  + Better targeting of investment, resources, support and expertise to drive innovation and improve results
  + Deliver on our intentions to deliver to business the means to:
    - “Provide tailored services to support and to raise achievement”
    - “Invest more effectively to raise achievement”
  + By putting in place stated enabling conditions that are required to be in place first:
    - “Information technology”.
* Deliver to Government’s ICT 2017 Strategy and Action Plan:
  + Support for “Better public services – Result 9 “Better for Business is making it easier for business customers to deal with Government”
  + Support for “Better public services – Results 10 “New Zealanders can complete their transactions with government easily in a digital environment”
    - Deliver to the Integrated Work Programme to support the Government ICT Strategy:
      * Removing barriers to the adoption of cloud services
      * Development of infrastructure and capability to produce new insights to improve government services and to inform government policy and investment
      * Ensuring that ICT investments are targeted to support digital services, and that benefits are realised

## Tactical Alignment

* **Provides the means for the iEd to deliver a key deliverable:** a key deliverable of the iEd Program is the delivery of APIs over Views of instant or time-ranges of information consolidated from multiple Operational Services (eg: ERS Services), which cannot be provided by the Operational Services themselves.
* **Provides a referenceable API Architecture:** the organisation has required for 3+ years a reference architecture that combines the capabilities of meeting government-ready security while providing government mandated (NZDIMP[[4]](#footnote-4)) access to this Organisation’s data in a manner that meets security and legal constraints, including NZISM[[5]](#footnote-5), Cabinet mandated Usability, Accessibility and Archiving constraints.   
    
  Having a code base which demonstrates OASIS Standards based, ISO/IEC approved ODATA[[6]](#footnote-6), OAuth, OIDC[[7]](#footnote-7) integration capabilities in an unambiguous, operating manner would address the problem definition by having pre-assembled components and knowledge that would lower uncertainty of delivery and duplication of research costs, while ensuring our qualities expectations are met.  
    
  The Reference architecture can then be used by both Vendors and Small Businsess Services.

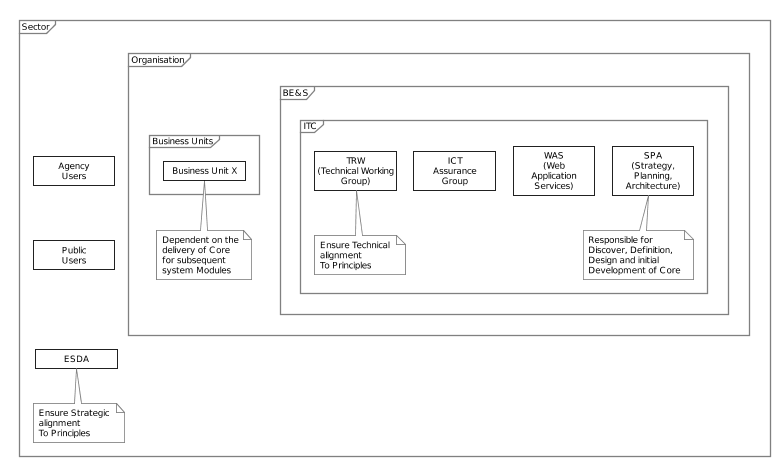
## Business Benefits

The following lists the business benefits of putting in place this solution:

* *Better value for money*
* *Better Reputation*

## Stakeholder Groups

The following are impacted in one way or another by the current BASE Service:



* **MOE:**
  + **BE&S:** Business Enablement and Support
    - **ICT**
      * **SPA:** Strategy Planning and Architecture
        + **Solution Architects Group:** Responsible for Initial Discovery, Definition, and Design of the Core Service.
        + **TRW:** Responsible for Reviewing and Accrediting on an ongoing basis the technical designs of the solution’s solutions, escalating to ESDA appropriate service design decisions.
      * **Assurance Group:** Responsible for ensuring the system is Accreditable.
      * **WAS:** Web Application Services
        + **Test Services:** Responsible for providing specialized input on the development of Functional Test Contract Definitions. Testing is applied by automation via the deployment pipeline.
        + **SBS:** Small Business Support Services: Responsible for ongoing maintenance needs of the deployed service.
      * **CAB:** Responsible for ensuring that C&A processes have been completed and permission to Operate has been granted to the deployed system.
* **Sector Users:** this sectors agencies will be able to access data in an appropriate manner and convert from existing antiquated Character Separated Value (CSV) data sources to the more flexible ODATA based REST API endpoints where they can query and retrieve custom shaped data in JSON format.
* **Other government agencies:** other agencies will be access data in an appropriate manner.
* **Public Users:** Public developers may utilize UNCLASSIFIED data made available via APIs to meet needs the ministry cannot envision.
* **ESDA:** Responsible for Accrediting the design alignment of the solutions services.
* Note:   
  it is to be expected that during the early stages of projects, not all Stakeholder roles will have been assigned.

## Scope

### In Scope

* SCOPE-0001: Develop a BASE infrastructure for subsequently deployment of cost-effective deployment of secure Modules. This includes management of Configuration, Exceptions, Users, Roles, DataClassification, Notifications.

### Out of scope

* XSCOPE-0001: Business Domain functionality. As stated earlier, this SAD documents the Core system on which other Modules, described by other SADs, are reliant on.

## Constraints

### Principles

This Business concern for which the solution is designed is part of the Organisation.

As such this solution is constrained to meet the the Service Delivery Principles Framework comprised of the following:

#### Key Service Delivery Principles

The [Key Service Delivery Principles](https://docs.moe.govt.nz/navigator/bookmark.jsp?desktop=MinistryDocuments&repositoryId=FNMinistryDocuments&repositoryType=p8&docid=DMOE%2C%7B1C5A444A-9678-4E41-96C4-A8365566408A%7D%2C%7BAE38638D-7652-CE77-875D-61774DF00000%7D&mimeType=application%2Fvnd.openxmlformats-officedocument.wordprocessingml.document&template_name=DMOE&version=current&vsId=%7BED5B0F9B-63A5-C409-87D6-61734D000000%7D)[[8]](#footnote-8) summarize the objective, information and security, platform, compliance and delivery context in which this service is delivered.

#### Detailed Service Delivery Principles

The [Detailed Service Delivery Principles (Abridged)](https://docs.moe.govt.nz/navigator/bookmark.jsp?desktop=MinistryDocuments&repositoryId=FNMinistryDocuments&repositoryType=p8&docid=DMOE%2C%7B1C5A444A-9678-4E41-96C4-A8365566408A%7D%2C%7B6BE34130-2572-C952-ADED-61775E60001A%7D&mimeType=application%2Fvnd.openxmlformats-officedocument.wordprocessingml.document&template_name=DMOE&version=current&vsId=%7B253E97BB-A944-C51D-85EA-60F189E00000%7D) provide more guidance on how to meet the above Key Service Delivery Principles. The (approximately) 60+ detailed principles are categorized per architecture, security, information, integration, design, delivery, development.

### Schedule

The following lists known dates that frame the project’s schedule.

The codebase, accompanying delivery integration and testing pipeline and necessary documentation is expected to be functional by the end of Q1/2018.

### Design Constraints

Below are listed key additional Constraints specific to this solution.

* CONS -0005: ESL in not expected to offer a public or student authentication service in the near to medium term.

## Risks

Delivery risks include the following:

* RISK -0001: Neither the Ministry’s Education Sector Logon (ESL) service alone or in combination with the Organisation’s ESTI/FED program is sufficient to authenticate the largest section of potential users. Note that this risk is not limited to this project: it is currently shared by any service to be offered by the Organisation or Sector.
* RISK-0002: The amount of time allocated to developing this system is relatively short.
* RISK-0003: The ministry does not have mature development practises that limit access to production data in part or whole, in clear text or obfuscated, in individual or aggregate form.
* RISK-0004: The ministry does not currently have a mature development practise that limits access to deployment environment secrets (system credentials, etc.). This document provides solutions to this.
* RISK -0013: Ministry cannot provide performant devices and current tools to consultant developers in order for them to work effectively on and off premise, without delay.
* RISK-0015: It is a key sector and organisation principle that we deliver to Government Standards. This includes the use of HTTPS for ODATA based queryable APIs -- but the relationship to the SIF format is still to be determined.

## Controls

* CONT-0004, addressing RISK-0004:: This Document defines an architecture that relies on Azure AD RBAC access between components, rather than rely on either disclosing secrets/credentials to developers, or rely on slower manual coordination between infrastructure and development specialist groups.
* CONT-0005, addressing RISK-0004: This document defines an automated deployment process that remains “dumb” -- without relying on secrets to successfully deploy target environments – as opposed to a Build and deployment process that requires lots of secrets being embedded in build definitions.
* CONT-0006, addressing RISK-0016: As per DES-0305, APIs should continue to be delivered to the government standard of OData – and secondary API endpoints provided for SIF compliance where required.

## Assumptions

The design of the solution is based on the information available at the time.   
  
The following lists Assumptions made in order to progress the solution’s design:

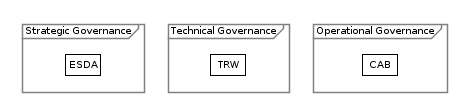
* ASS-0001: Azure Service can meet the Organisation’s C&A requirements.  
  Note:   
  Microsoft holds several current accreditations[[9]](#footnote-9):
  + ISO-27001, ISO-27018 Certifications
  + undergoes SOC1 and SOC2 reporting of the Azure platform,
  + has FedRAMP Authority to Operate (includes AzureAD),
  + has Australian Government Information Security Registered Assessors Program (IRAP)
  + Singapore Multi-Tier Cloud Security (MTCS) granted to Azure’s Datacenters, O365, Azure AD, and Azure.
* ASS-0002: Near-Shore cloud storage remains acceptable for Data classified as SENSITIVE.
* ASS-0003: The organisation will provide an OIDC compliant authentication service of some kind.

## High Level Business Domain Requirements

The following list Business Domain Requirements.

* Deliver a deployable baseline code base that addresses ISO-25010 quality objectives of its Stakeholders.

## Governance



As per the the [Stakeholders](#_Stakeholder_Groups) section above, governance is provided by multiple boards:

* **ESDA:** Responsible for Accrediting the design alignment of the solutions services.
* **TRW:** Responsible for Reviewing and Accrediting on an ongoing basis the technical designs of the solution’s solutions, escalating to ESDA appropriate service design decisions.
* **CAB:** Responsible for ensuing C&A processes have been completed and permission to Operate has been granted to the deployed system.

|  |
| --- |
|  |

# Service Context View

This section of the document describes the context within which the Solution operates, the business systems with which it interacts, extends, improves and/or replaces.

## Synopsis

The solution is a Cloud based PaaS based solution that takes advantages of the sector’s ESTI-provisioned Azure available services, to provide a highly secure, API base framework.

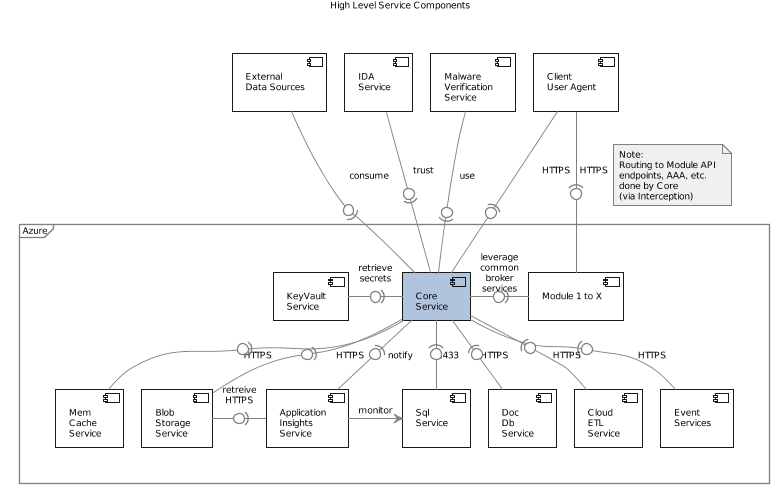
The solution’s is cost effective to run using Azure PaaS infrastructure correctly. It is highly secure, available, modular and maintainable. The solution provides a valuable framework on which to deliver immediate and future functionality to meet business demand in a timely and accreditable manner while meeting [Key Service Principles](#_Key_Service_Delivery).

## System Context Summary

The Solution is:

* A Cloud-based Platform as a Service (PaaS) service designed to “API-First” Design [Principles](#_Principles) to host Business queryable ODATA RESTful APIs.
* Deployed to Microsoft Azure Infrastructure which meets ISO27001, ISO27018 Certifications and undergoes regular SOC2 reporting.
* Developed as a PaaS Service to meet PaaS before IaaS before Physical Design [Principles](#_Principles).
* Leverages high-value, low cost, standard Azure PaaS Services to meet its system functionality capability needs, including:
  + **Azure App Services:** a Platform as a Service (PaaS) offering that permits the system to be deployed to cost effective Standard infrastructure packages, that include by default dynamic horizontal elastic scaling, custom DNS end point discoverability and firewalled access capabilities.
  + **KeyVault Service:** for storage of Secrets required to access other services, including:
  + **IDA Services:** IDentity and Access management services, leveraging:
    - The OIDC protocol, to connect to:
    - The Organisation Cloud hosted Azure AD (using the MSAL libraries over V2 Endpoints, as opposed to the older ADAL/V1 endpoints), connected to its on premise Windows AD (WAD) via ADConnect.
    - Partner Organisations Azure AD (via B2B)
    - The wider Sector used Social Identities (including Google, Microsoft Account),
    - The Organisation’s Client facing Azure B2C based IDP bridge, when it becomes available and is functionally appropriate.
    - **Blob, Queue, File Storage Service:** for persisting data received from external API clients or ETL services.
    - **Data Factory Service:** for ETL communication over HTTPS to on premise legacy data servers.
  + **SQL Server Service:** the services takes advantage of Azure’s Database as a Service (DbaaS) offering for is relational Data storage needs.
    - Refer to the Assumptions section regarding there being no discernible need for a Non-Relational (NoSQL) database at this point in time, although if that changes, it would be relatively trivial to incorporate.
  + **Blob/Queue/File Storage Service:** Azure provides general Blob Storage, and two specialized variations that are both used by the Service. Blob Storage is used for permanent uploaded media as well rolling Diagnostics Trace Message storage. Queue Storage is used to queue for later processing data integration payloads.
* Deployed to an Nearshore Microsoft Azure Data Centre (Australia East).  
  Note: NearShore use of Azure Infrastructure – as opposed to OffShore – has been approved for storage of Data classified as SENSITIVE or below (ie, including IN-CONFIDENCE and UNCLASSIFIED).
* Backed up to the same Datacentre.
* Operation System, Operational Database and Backup Geographically Replicated to another Microsoft Data Centre (Australia SouthEast)
* Other Nearshore cloud hosted Services:
  + **Malware Detection Services:** a near shore accreditable malware detection service (scanii) will be used to scan incoming data prior to persistence, in order to not infect down-stream service clients.
  + **Organisation’s provided SMTP endpoint:** if and when one is available in a satisfactorily accessible and secure manner, it will be used to meet the service’s SMTP notification needs.
* The service development is initiated using C#, compilable to Common Intermediate Language (CIL, formerly MSIL),
* utilizing a .NET Framework that meets a minimum of .NET Standard 2.0 (ie, .NET 4.6.1 or later)
* Will be ported to .NET Core 2.0 at a later date (when it does not conflict with the key business expectations), in order to be also hostable on cheaper Linux b ased PaaS targets, while simply be better aligned with future Microsoft development roadmaps.  
  Note: A Development Decision was made to abstain form .NET Core 2.0 until OData – a key requirement for an API service -- was available on that platform. It is due Q1 2018.
* The components are developed to Domain Driven Design (DDD) conventions to increase maintainability.
* Database access is via Entity Framework.
* Database schema management is greatly simplified by using Entity’s CodeFirst (a DBasCode capability), enhancing maintainability.
* Authorisation, Auditability and Accountability is built into the core fabric of the system, rather than as an extension to business functionality.
* Communication between all system components is secured and confidential.

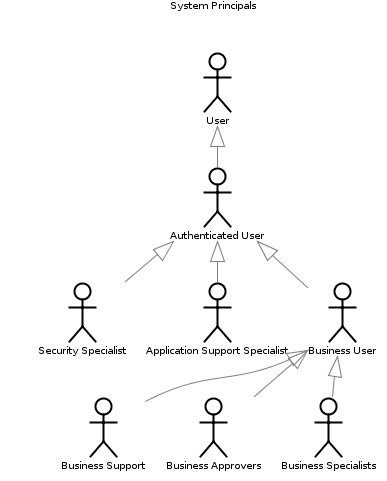
Key components in the above summary are shown in the following diagram:



## Actor Context

Core’s Functionality is used by a selection of different Actor types.

Note:   
Availability of Functionality depends on the [Access Control Model](#_Operation_Access_Control) and [Data Classification](#_Data_Classification), as described within the [Security View](#_Security_View).



## Integration – Organisation Business Service Systems

Listed below are key business service systems that will be impacted (integrates with, requires modification to and/or replaces):

* **Not-Applicable:** being a Baseline Core System Service, with no Business Functionality in and of itself, it does not integrate with any Business Services.

That said, Core can be extended with Modules that leverage common communication services provided by Core to communicate with on premise legacy. Refer to the [Integration View](#_Integration_View).

## Integration – Infrastructure Systems

Listed below are infrastructure systems the solution impacts (integration, requires modification to and/or replaces).

* **Azure Active Directory Service (AAD): the System is deployed to an PaaS environment managed by Service Accounts issued by AAD. The Service Accounts can be granted access to other Azure Resources.**
  + **Details:** Refer to this document’s [Integration View](#_Integration_View) for technical details.
* **Azure Key Vault Service:** the Azure PaaS App Service retrieves secrets (credentials and configuration settings) from Azure’s PaaS Key Vault Service.
  + **Trigger:** Execution Environment startup.
  + **Volume:** random intervals based on system usage. For example, weekly.
  + **Protocol:** serialized stream over TLS 1.2 **HTTPS, using PCI 1.2 compliant certificates.**
  + **Security: the communication channel is secured, the information is cached in the system’s memory while the system is running, and used to complete calls.**
  + **Details:** Refer to this document’s [Integration View](#_Integration_View) for technical details.
* **Azure Sql Service:** the Azure’s PaaS App Services service in turn uses Azure’s PaaS Sql service offering for relational data storage.
  + **Trigger:** Multiple interactions per user Interacts
  + **Volume:** can be optimized up to 4000 interactions per second on Standard offerings.
  + **Protocol:** SQL statements over Port 1433
  + **Security:** The services requires Azure based Integrated Security, based on Azure AD accounts (transmittal of Username/password on the Connection String is not an acceptable implementation).
  + **Details:** Refer to this document’s [Integration View](#_Integration_View) for technical details.
* **Azure Storage Service:** the Azure’s PaaS App Services service in turn uses Azure’s PaaS Storage Account service offering for non-relational data storage.
  + **Trigger:** Multiple interactions per user Interacts
  + **Volume:** can be optimized up to 4000 interactions per second on Standard offerings.
  + **Protocol:** serialized stream over TLS 1.2 **HTTPS, using PCI 1.2 compliant certificates.**
  + **Security:** access to the storage is controlled by a combination of service provided public/private access rules, and Shared Access Signature (SAS) tokens for public access. Tokens are retrieved from the Key Vault service.
  + **Details:** Refer to this document’s [Integration View](#_Integration_View) for technical details.
* **OIDC Identity [Token] Provider:** the system uses the OIDC protocol to integrate with a secure identity token provider of ESDA’s preference (see ESL and/or ESTI/FED).
  + **Trigger: Manual, as users authenticate themselves to the Service.**
  + **Protocol: OIDC, using JWT, over OAuth, over HTTPS, using PCI 1.2 compliant certificates.**
  + **Details:** Refer to this document’s [Integration View](#_Integration_View) for technical details.
* **MTA:** A secure and easily accessible Organisation managed[[10]](#footnote-10) cloud-hosted Simple Mail Transport Protocol (SMTP) Mail Transfer Agent (MTA) is expected to be provided in order for this service to integrate with, in order to meet its notification requirements.
  + **Trigger:** As functionally required.
  + **Volume:** negligible. Current design norms are to prefer handling Notifications in-system. IDA notifications (eg: Password Reset Links, Password Renewal Notifications, etc. are best handled externally, independently, in the IDA service itself).
  + **Type:** Queued individual Message.
  + **Protocol:** SMTP
  + **Security:** Retrieval from KeyVault of credentials, and traditional Machine 2 Machine xfer over a HTTPS TLS 1.2 protected secure channel.
  + **Storage:** not relevant (trivial needs, and is considered part of the remote service).
  + **Details:** Refer to this document’s [Integration View](#_Integration_View) for technical details.
* Note:   
  To be investigated is whether Azure AD integrated Security can be leveraged to avoid the storage.
* **Malware Detection Service:** the system integrates with a secure and easily accessible NearShore Malware Detection Service.
  + **Trigger:** as functionally required (when a user uploads Media of any kind, such as images, csv files, pdf, excel and word documents).
  + **Volume:** negligible. Few of our current systems rely heavily on uploaded media as FileNet is our designated EDMS, and we do not (as yet) develop many systems that allow users to upload non-documents.
  + **Protocol:** HTTPS**, using PCI 1.2 compliant certificates.**
  + **Security:** The service requires a Key + Secret, persisted in Azure KeyVault.
  + **Details:** Refer to this document’s [Integration View](#_Integration_View) for technical details.
  + **Governance:** The service relies on a Subscription fee being kept current. The trivial cost thereof is managed as an ongoing business CAPEX unit basis.

## Integration - Service Reuse

Refer to above for services integrated with, and reuse of the following systems.

* **ESTI Azure Infrastructure:** the above system uses the Azure infrastructure that the Organisation provides access to as an Education Sector Technology Infrastructure (ESTI) Project’s managed services.   
    
  Sub-services included in ESTI’s Azure service offering are:
  + Azure Key Vault
  + Azure App Services
  + Azure Storage
  + Azure Sql Services
  + Azure Redis
  + Azure Event Grid and Service Bus
  + Azure Data Factory
  + Azure App Insights
* **SMTP Service**: when an organisation wide cloud based SMTP Service is available in an accessible and secure manner, the service will be reused.
* **OIDC IDP**: upon direction from ESDA, integration will be implemented with an OIDC based IDP.

## Design Scope Exclusions

* XSCOPE-0020: Specifically excluded from scope, and design, is any database one-off, potentially manual, provisioning process.   
    
  External data needed by Core or Modules dependent on Core is to be developed as an idempotent, paged import process.
* XSCOPE-0021: Seeding Tenants or Principals directly into the system’s database, bypassing the Application logic developed to maintain quality and consistency.
* XSCOPE-0030: Specifically excluded from scope is any preparation for Manual Testing. All tests to be performed by automation, by the build pipeline. The automated tests are based on defined Delivery Contracts. Refer to [Delivery View](#_Delivery_View).

## Unknowns

Listed below are any key Unknowns that still require Answering, or are the basis of Assumptions used to proceed.

* UNKNOWN-0010: Whether contracts and HR policies are sufficiently in effect in such a manner that vendors, contractors and employees are sufficiently constrained to not directly access or use data in part or whole, in cleartext or obfuscated, in individual or aggregate form, without written consent to be obtained separately.
* UNKNOWN-0020: What Identity system will be used to authenticate and authorise non-organisation, sector and public users. See RISKS.

## Design and Operations Decision Register

Listed below are any key Design or Operational Architecturally Significant Decisions.

* DES-0101: All environments the system will be deployed to are to be scripted as to be built, configured and deployed to by automation, including Core, any dependent Modules, Database Servers, Databases, Storage Accounts, and KeyVaults.
* DES-0102: The deployment scripts will be run as part of a build deployment pipeline.
* DES-0103: Any relational database will be deployed using DBasCode (ie Microsoft Entity Framework’s CodeFirst).
* DES-0104: No Data will be migrated to the database directly (this would directly conflict with the DBasCode approach this system is designed to use). All data that is required by the system will be loaded in via an idempotent, paged, background process, passed through the Validation logic developed in the Service facades – rather than bypassing it.
* DES-0110: Testing sufficient to minimize the risk of automating the delivery of the service will be automated within the build and deployment pipeline. No manual testing will be required to ensure functionality. Test definitions may be developed by Testing Services, in order to be Automated within the continuous development and deployment pipeline.
* DES-0111: No Soak Testing will be performed before go live. Soak testing is not required until there is demonstrable proof that attention is required, at which point it will be treated as required effort, to be handled by general Agile processes, and deployed as the result of a scheduled sprint.
* DES-0112: If testing of the horizontal scaling feature is required it will be done with an artificially low threshold trigger to demonstrate dynamic scaling from 1 server to 2 works (and therefore demonstrate the threshold logic works correctly).
* DES-0201: Use of .NET Standard 2.0, but wait to port it to .NET Core until OData is available on that platform. The rationale for this decision is: a) Queryability is a key objective, b) OData is the only standards based Queryable standard, c) Microsoft had to reprioritise effort to get .NET Core 2.0 ready for go-live, and temporarily reallocated devs to other departments, d) development on OData has resumed now that .NET Core 2.0 has shipped, but it is still in Beta, e) the effort to port to >NET Core when OData is available will be trivial (expect less than a week effort).
* DES-0301: Modules can be developed to deliver APIs exposing data consolidated from multiple operational systems. Valid sources include Data Vaults. Refer to the Integrations Architecture Options paper.
* DES-0304: Queryability will be provided by ODATA, the only Standards based Queryability protocol, as per [Principles](#_Principles). Refer to the Integrations Architecture Options paper.
* DES-0305: APIs should continue to be delivered to the government standard of OData – and secondary API endpoints provided for SIF compliance where required.

## Technologies

Listed below are key technologies (Execution Environments and Frameworks) used within the current solution are in alignment with supportability constraints listed within the Enterprise Specifications.

* TECH-0001: C#, over .NET Framework, meeting .NET Standard 2.0 compliance
* IDA, using
  + TECH-0011: OIDC
  + TECH-0012: Microsoft Azure Active Directory

Important: even though the name starts with Azure, the Service is distinct from Azure Infrastructure/Compute/Services).

* + TECH-0012: Microsoft B2C (optional)
* TECH-0020: Microsoft Azure Services:
  + TECH-0021: AppServices
  + TECH-0022: KeyVault Service (for Secrets storage)
  + TECH-0023: Storage Service (for Blob, File and Queue storage needs)
  + TECH-0024: SQL Server Service (for Relational Data storage needs)
  + TECH-0025: Application Insight (for Diagnostics, Telemetry, and Alerts)
* Other Cloud Services:
  + TECH-0031: SMTP Services
  + TECH-0032: Malware Detection Service
  + Delivery Services:
    - TECH-0051: Git
    - TECH-0052: PowerShell
    - TECH-0053: Visual Studio Online Service (an Application Lifecycle Management(ALM) Service)
    - TECH-0054: Azure TinHat Service (optional)

## Protocols

Key web Protocols and Standard used are:

* TECH-0101: HTTP / HTTPS
* TECH-0102:JSON
* TECH-0103:ODATA
* TECH-0104: OpenAPI API Documentation Specification
* TECH-0201: OIDC
* TECH-0202: OAuth
* TECH-0203: JWT

## By Design

This document and any accompanying Technical Design Documentation describe a system that can be deliver the functionality described by following the proposed design -- reasonable time, effort and skillsets allocation notwithstanding.

# Qualities View

## Abstract

The solution is aimed at addressing evenly an optimal set of ISO-25010 Qualities.

Important:  
ISO-25010 Qualities include and address what is understood by the legacy terms “(Business) Functional Requirements” and “Non-Functional Requirements”.

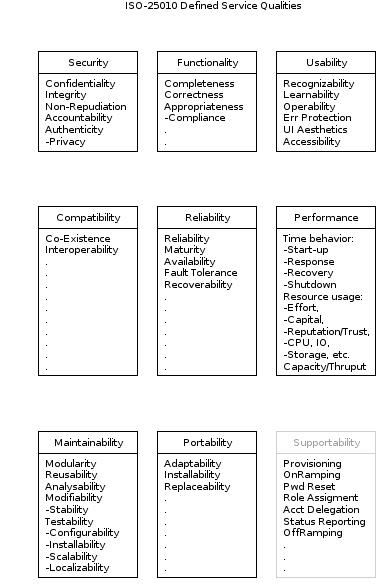
## Synopsis

The solution addresses to a high level all of the System Qualities Objectives recommended within ISO-25010.

## Quality Objectives

Although the terms Functional Requirements and Non-Functional Requirements are common well-known industry terms, they can produce a negative impact on system delivery. Functional Requirements developed by business stakeholders are too often perceived as first-class requirements, leaving all other stakeholders functional requirements as second-class requirements. This observation of human behaviour, and its impact on delivering services is the basis of ISO-25010’s recommendation to not use either term, and instead use the term Qualities.

As we are a Standards based organisation, the term Qualities is used within this View, to summarize the High Level Functional and Non-Functional Requirements, and how the solution meets these Quality Requirements.



## Security Qualities

* The Core system is designed to be deployed to Microsoft Azure Infrastructure which meets ISO27001, ISO27018 Certifications and undergoes regular SOC2 reporting, addressing:
  + ISO-25010/Security/Integrity qualities.
  + ISO-25010/Security/Accountability qualities. Access to physical premises is Limited, and Accounted.
* **ISO-25010/Security/Confidentiality:** the design of the system allows for Core to provide itself and any additional Modules that rely on it, the following features:
  + Communication channels between Client and Service are only functional over HTTPS.
  + HTTPS encryption is negotiated between Client and Server, preferring TLS 1.2 (IETF, 2008) or better.

Note that at this point in time (Q4 2017) Azure PaaS infrastructure limits negations between Client and Server to agreeing to a minimum of TLS 1.0, which could be used by a dedicated malicious agent as a building block for a ‘POODLE ‘or ‘Heartbleed ‘exploit. The risk of a successful exploit remain Highly Unlikely.

On January 9th, 2018 Microsoft announced it will meet the PCI Data Security Standard (PCI DSS) for safeguarding payment data existing deployments 30 June 2018 deadline, by transitioned existing app services to a minimum of TLS 1.1 or 1.2 by April 30th 2018. All new App Services deployed after June 30th 2018 will be configured to agree to a minimum of TLS 1.2.

* + Ccommunication initiated by Core to cloud hosted services it relies on (eg Malware Detection Services, external IDPs, etc.), are over HTTPS.
  + All HTTP communication between Service Clients and the Service are redirected to HTTPS.
  + HTTP Cookies round-tripped between Clients and Service are marked as Secure only (ie, they are not issued by the server over HTTPS, and the Client User Agent is instructed to not transmit them over unsecured HTTP).
  + HTTP Cookies round-tripped between Clients and Service are marked as HTTPOnly instructing the Browser to not allow Client-Side access or manipulation of cookies, effectively ensuring they are only used for round-tripping Server state, and not used as a manipulatable instruction Message, even if well intentioned.
  + Communication between Core and its relational databases use IntegratedSecurity in order to not transmit the Solution’s Service Account in the clear.
  + **ISO-25010/Security/Authenticity** the design of the system allows for the Core to authenticate Principals (ie human Users or other Services or Systems) using the Open ID Connect (OIDC) protocol to communicate with external Identity Providers (IdP).
  + **ISO-25010/Security/Non-Repudiation** is provided by Core to itself and any dependent Module by having all incoming Operations permanently recorded against the current Session, for future Auditing purposes.

**ISO-25010/Security/Integrity** is provided by Core to itself any dependent Module by multiple means:

* + - Azure Storage Service Encryption (SSE) is applied to any Data at Rest stored on Azure Storage Service, whether it be Blob, Queue, File storage (Microsoft, 2017) or relational database (Microsoft, 2017). All data is encrypted using 256-bit AES encryption.
    - Digitally Signage of Operation Audit records is done using the host’s TLS certificate’s private key, significantly increasing the difficult of tampering with audit records.
  + **ISO-25010/Security/Accountability** is provided by
    - The system is deployed to Azure infrastructure which (as part of its ISO27001, ISO27018, SOC2 obligations) limits, monitors and audits all access to the infrastructure and changes thereto.
    - by Core to itself and all dependent Modules by having all incoming Operations recorded against the current Principal’s current Session, for future Auditing purposes.

## Functionality Qualities

* Refer to the [Functionality Qualities View](#_Functional_Quality_View) for a high level view of the Functionality designed to be in Core, for use by the various User groups, including but not limited to:
  + Infrastructure Support Specialists
  + Application Support Specialists
  + Security Specialists
  + Business Support Specialists
  + Business Users
  + Public Users
* **ISO-25010/Functionality/Appropriateness:** exceeds the immediately required functionality.
* **ISO-25010/Functionality/Compliance:** the system meets:
  + Privacy Requirements: the system keeps records in-system for the duration, and no longer, the records are needed to meet functional requirements while meeting regulatory obligations.
  + Auditing Requirements: the system performs logical deletes and auditing of records to meet its security and archiving obligations.

## Usability Qualities

Core functionality is exposed via

* **ISO-25010/Usability/Learnability** is provided by
  + the Core adheres to Well-Known REST URL patterns aiming towards Semantic URLs (Wikipedia, 2018),
  + adheres to the easily learnable ODATA (OData, 2017) query syntax.
  + Self-documents itself using the OpenAPI Specification (Open API Initiative, 2017).
  + Provides a human readable and testable Swagger (Swagger, 2017) page to investigate API functionality.
* **ISO-25010/Usability/Error Protection** the Core and any dependent Module by capturing Unexpected Errors and logging them along with their associated Stack Trace. The service can be configured to publish the error itself (but of course only in Debug/Non Production mode) or replace the error with a general error message, with the unique reference identifier of the Error Record, for subsequent analysis by Customer Support.
* **ISO-25010/Usability/Accessibility**
  + Language accessibility: the design of the system allows for Core to provide for itself and any dependent Module the ability to switch languages where appropriate (eg: from English to Maori, and back again).  
      
    Note that this functionality is expected in order to meet compliance objectives and obtaining accreditation to operate from an accreditation review board.
  + Visual Impaired accessibility issues are generally associated to visual user interfaces (which Core does not provide), and not APIs. That said, the APIs can be utilized by Accessible client user agents.
* **ISO-25010/Usability/User Aesthetics** is associated to visual user interfaces, and not APIs, and therefore are largely irrelevant. That said, if a Decision to include interfaces within the System, they will be bound to the [Principles](#_Principles), which constrain the system being developed while addressing these constraints.
* **ISO-25010/Usability/Operability** is associated to user agents, and not APIs directly.

## Compatibility Qualities

* **ISO-25010/Compatibility/Interoperability:** all resources are appropriately accessible via secure confidential APIs that conform to NZ Government API standards:
  + New Zealand Government API Standard and Guidelines (ICT, 2016). Specifically Part B (Technical), which recommends the use of Representational State Transfer (REST) (Fielding, n.d.).   
    REST is a industry accepted assembly of well-established standards such as URI, HTTP, HTTPS, etc. to deliver the following characteristics:
    - Client-Server Architecture
    - Statelessness
    - Cacheability
    - Layered System
    - Code on Demand
    - Uniform Interface
      * Resource Identifiers in Request
      * Resource Manipulation through representation.
      * Self-Description Messages
      * HATEOS
* **ISO-25010/Compatibility/Interoperability:** The information is serialized to:
  + ISO/IEC 21778:2017: JSON (ISO, 2017) (default format)
  + XML (W3C, 2006) (optional format)
* **ISO-25010/Compatibility/Interoperability:** API data is queryable using:
  + Open Data Protocol (ODATA)
    - ISO/IEC 20802-1:2016 : Core (ISO, 2016) and
    - ISO/IEC 20802-1:2016: Part 2:JSON Format (ISO, 2016)
* **ISO-25010/Compatibility/Coexistence:** the Core service and its dependent Modules are deployed to Azure PaaS infrastructure shared with other systems.
  + the service does not read or write any user media from the underlying file system, mitigating against loss.

Note: It is an Azure feature that systems are deployed to different infrastructure as and when needed based on load. In rearranging system instances, Microsoft cannot guarantee systems are deployed to the same hosts. This in effect means that any files written to the underlying Filesystem will be “left behind” when moves happen.

## Reliability Qualities

The Service is

* Deployed to Standard Azure Infrastructure which:
  + Has an overall Service Level Agreement (SLA) of 99.9% per billing month (Microsoft, 2017).

Note:  
Microsoft surpassed the above SLA in 2016 by almost 100x, having consistently had global uptime upwards of 99.9979% for Compute in the past 12 months alone (Microsoft, 2017).

* Is deployed to Standard Azure App Services (again, with an SLA of 99.9% per billing month), to handle traffic surges by automatic horizontal scaling to and from additional PaaS host instances, triggered when customizable CPU minimum and maximum average thresholds are reached.

Although largely irrelevant when Horizontal Scaling is implemented to provide instances as needed, each individual Standard instance can be expected to be able to handle 4000 sustained client-server requests per second.

* **ISO-25010/Reliability:** the Core provides to itself and any Modules that depend on it Reliability by the combination of multiple solutions:
  + the system uses client side caching instructions to minimize unnecessary duplicate requests for Media (Images, documents, etc.).
* **ISO-25010/Reliability/Availability:** the system is deployed to Azure’s Standard PaaS AppService infrastructure which by default
  + Is backed with an SLA providing 99.9% availability per month billing cycle.
  + Provides dynamically horizontally scalability to handle sudden peaks in demand.
  + the Core service implementation protects itself and any reliant Module by providing protection from Denial of Service (DoS) attacks at two levels:
    - within the System level: by allowing only a customizable number of similar requests from the same IP within a period of time (eg: 30 resource requests /sec to allow for a page of data).
    - at the PaaS infrastructure host level: Azure offers a baseline Web Application Firewall (WAF) protection of its infrastructure and their clients applications deployed to it based using active AI.

An Additional Client-Customizable WAF Service is available from Microsoft Azure that can be subscribed to, but should not be implemented until there is observed risk requiring active control.

* **ISO-25010/Reliability/Fault Tolerance:** the system is stateless, captures unexpected exceptions, logs them, and returns a meaningful but sanitised error message to end users, before releasing resources in order to process another operation.
* **ISO-25010/Reliability/Recoverability:** the system recovers from operational errors. See ISO-25010/Reliability/Fault Tolerance above.

## Performance Qualities

* **ISO-25010/Performance/Time behaviour/Startup performance:** the startup time required by the system after a new deployment is dependent on the Database as Code (DbaC) operations that are scheduled – but in general new deployments will take less than a minute to perform. During this startup time
  + Client User Agents making API calls will receive a recoverable timeout error code if the transition takes several minutes – otherwise simply take longer to complete their operation.
  + If any Business User facing MVC Interfaces are developed (there are absolutely no plans to) Users will instead be presented with an offline.html page explaining the delay.
* **ISO-25010/ Performance/Time behaviour/Response times:** the system is expected to be optimizable so that basic Core API Operations complete on average in <150ms. More complex operations could be optimized to complete on average will complete in <800ms.   
    
  Note:   
  By design no Core API operation will take longer than 1sec. “caching closest to source, in the format closest to use”, is a design principle extensively used to improve responsiveness and therefore user experience.
* **ISO-25010/Performance/Time behaviour/Recover times:** any time required to recover from an unexpected exception raised during an API operation is negligible (in most cases would be expected to less than the time required to complete the requested Operation successfully).
* **ISO-25010/Performance/Time behaviour/Shutdown time:** after a shutdown command has been issued, queued parallel thread operations may take up to 60 seconds to gracefully complete activities before complete shutdown. This is a feature of how C# Threading works.
* **ISO-25010/Performance/Resource Utilisation/CPU & Memory:** the Core system and its dependent Modules are deployed to Standard Azure AppService infrastructure, where available CPU and Memory is for all intents and purposes simply a question of how much one is willing to pay for. A basic Standard package of memory and CPU is sufficient.

Note that an average CPU usage is used as the trigger to elastically scale horizontally to meet ISO-25010/Availability Quality objectives.

* **ISO-25010/Performance/Resource Utilization/Capacity & Throughput:** the Core system and its dependent Modules can handle 400 concurrent Operations per host instance, elastically horizontally scaling as required to meet peek and surge needs.

Note: this dynamic scaling is (economically) untestable but this feature is a key driver behind the Decision to using Azure PaaS infrastructure to host the system upon. Testing should be limited to reproducing escalation from 1 server to 2, based on a CPU trigger artificially low for proof of logic reasons. Soak testing the system for long durations with high loads is specifically out of scope.

## Maintainability Qualities

* **ISO-25010/Maintainability/Modularity:** the component architecture follows Domain Driven Design (DDD) component design patterns, which promotes clean separation of domain functionality into distinct logical deployment components.
* **ISO-25010/Maintainability/Reusability:** the code base is hosted in Git, can be cloned in order to develop clones. The system functionality allows for the reuse of executing systems for additional tenancies.
* **ISO-25010/Maintainability/Analysability:** the system records Exception Report Records that can be accessed via API by analysis tools of the organisation’s choice. In addition the system is hosted on virtual infrastructure for which free Monitoring tools are available (such as App Insights).
* **ISO-25010/Maintainability/Modifiability:** Development of the code within BASE is stipulated to follow base practice SOLID development patterns.
* **ISO-25010/ Maintainability/Stability:** the system is deployed to dynamically horizontally scalable virtual infrastructure, is monitored, records error logs, recovers from unexpected errors and is by design stateless.
* **ISO-25010/ Maintainability/Testability:** fully automated testing is managed within the delivery pipeline. Automated testing is used to cover Static testing, logical Code Tests, Integration Testing, Functional Testing, Horizontal Scalability.
* **ISO-25010/ Maintainability/Configurability:** the system follows industry conventions for configurability (namely deployment pipeline management of deployment target specific web.config settings).
* **ISO-25010/ Maintainability/Installability:** the system is deployed via a workflow based automated pipeline.
* **ISO-25010/ Maintainability/Scalability:** by the components used to develop the system are both dynamically horizontally scalable to handle operations requests, and deployed to infrastructure that can be vertically scalable as required.

Note:   
The system is expected to be able to be optimizable to handle the execution 2000 operations second within a 1 second delay, on a single instance of standard virtual infrastructure.

* **ISO-25010/ Maintainability/Localizability:** the system incorporates cached resource localisation services that Modules can use for the purpose of localizing text resources within User interfaces.

## Portability Qualities

* ISO-25010/Portability/Adaptability
* ISO-2510/Portability/Installability
* ISO-25010/Portability/Replaceability

## Supportability Qualities

* ISO-25010/Supportability/Principal Provisioning
* ISO-25010/Supportability/
* ISO-25010/Supportability/Principal Password Reset
* ISO-25010/Supportability/Principal Role Assignment
* ISO-25010/Supportability/Principal Delegation
* ISO-25010/Supportability/Principal Status Reporting
* ISO-25010/Supportability/Principal Offramping

# Security Qualities View

As per the Detailed [Principles](#_Principles), no organisation service may be deployed in an unsecured manner.

This view summarizes key aspects that ensure the information managed by the system is secure at rest and in transit.

## Synopsis

The system is deployed to a secure near-shore data center with audited access, using physically inaccessible encrypted datastores, that are backed up for a full month, also encrypted, to the same physically inaccessible data center. Communication between all components and tiers is encrypted using using accreditable encryption (TLS 1.2 or better). Deployment Secrets/Credentials are not divulved to Developers, Deployers, or Testers, nor transmitted in cleartext even when within confidential communication channels. Instead relying on AzureAD to authenticate. Operations on Resources by Principals is controlled by RBAC based Access Control Model. Access to Resources by Principals is additionally controlled by DataClassification (see [Access Control Decision Flow](#_Access_Control_Decision)).

## ISO-27001:2013

* Information Security Policies
* Organisation of Information Security
* Human Resources Security
* Asset Management
* Asset Cotrol
* Crytography
* Physical and Environmental Security
* Operations Security
* Communications Security
* System Acquisition, Develoment and Maintenance
* Supplier Relationships
* Information Security Incident Management
* Information Security Business Continuity Management
* Compliance

## Execution Environment

* Deployed to restricted access data centres managed by processes that conform to ISO/IEC 27018:2014 and ISO/IEC 27001.

## Relational Database Security

### Access Control

Beyond the Portal that is used to create the database in the first place, access to relational Databases is restricted to

* The Azure AD managed service account of the System’s App Instance, using Sql Server’s IntegratedSecurity feature.  
  Note:  
  the *IntegratedSecurity* feature provides a secure means to use a secure ConnectionString that does not transmit cleartext service account username and password.
* Database Firewall Rules on the Database Server restrict access to the database from a specific IP.
* Note: As part of the connection process, connections from Azure App Services to TCP 1433 are redirected to a different IP address and port (between 11000 to 11999), unique for each worker role.

### In Transit

Communication between the System and its Relational Databases uses OLE DB over TCP port 1433.

If IntegratedSecurity were not used, the Username would be sent incleartext, the password encrypted.   
Note that *IntegratedSecurity* removes the need to pass either the UserName or Passord.

By default database communications unencrypted and vulnerable to sniffing: the recommendation is to use Encrypt=true;Trust Server Certificate=true; in the connection string.

Note:  
Adding Encryption does impact performance -- adding an extra network roundtrip at connect time -- but is appropriate when using Shared Standard Azure Services.

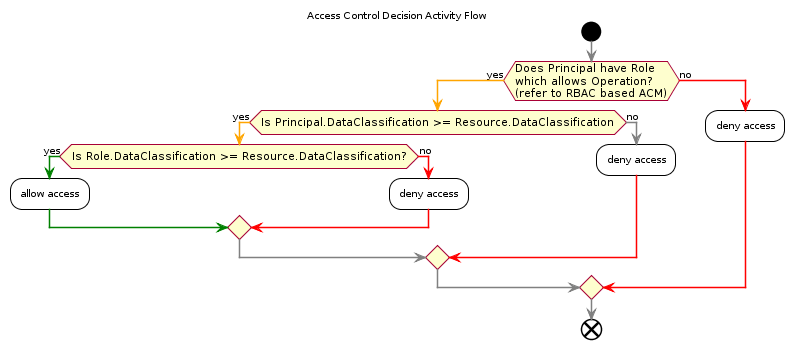
### At Rest

* Within the ISO/IEC 27018:2014 compliant data centre, the system persists relational data to encrypted hard drives using Transparent Data Encryption(TDE) using FIPS 140-2 256 bit AES encryption, which meets HPAA and PCI compliance.
* The databases are not physically removable from the ISO/IEC 27018:2014 ISO/IEC 27001 compliant data centre.
* Transaction Log Files are also encrypted using Transparent Data Encryption, to the same level.
* The encrypted relational databases are backed up to encrypted backup media.
* The backup media is kept for a month, during which time it too is not physically removable from the ISO/IEC 27018:2014 ISO/IEC 27001 compliant data centre.

## Access Control Decision Flow

Operations on target Resources by Principals is based on the following constructs:

* Operational access is granted based on the Principal’s Roles.
* Access to the Resource itself is based on the Data Classification of the Role, Principal, and target Resource.



## Operation Access Control Model

In order to whom has access to the different resources exposed via APIs (Tenants, Users, Roles, Sessions, Exceptions, etc.), and what operation they are allowed to perform (Create, Read, Update, LogicallyDelete), Role Based Access Control (RBAC) is applied.

### Core ACM

Permissions are assigned to the Core defined Roles as per the following table:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Resource + Operation on Resource | User | Authenticated User | Sys | AuthenticatedUser | SuperAdmin | SysAdmin | SecuritySpecialist | BusinessUser | BusinessSpecialist |
| ExceptionRecord/Read |  |  |  |  | X |  |  |  |  |
| ExceptionRecord/Delete |  |  |  |  | X |  |  |  |  |
| Notification/List | X |  |  |  |  |  |  |  |  |
| Notification/Read | X |  |  |  |  |  |  |  |  |
| Notification/Write |  |  |  | X |  |  |  |  |  |
| ConfigurationRecord/Read |  |  |  |  | X | X |  |  |  |
| DataClassification/Read | X |  |  |  |  |  |  |  |  |
| Tenant/Read |  |  | X |  |  |  |  |  |  |
| Tenant/Write |  |  |  |  | X | X |  |  |  |
| Session/Read |  |  |  |  |  |  | X |  |  |
| SessionOperation/Read |  |  |  |  | X | X | X |  |  |
| SysRole/Read |  |  |  | X |  | X |  |  | X |
| SysRole/Write |  |  |  |  | X | X |  |  |  |
| BizRole/Read | X |  |  |  |  |  |  | X | X |
| BizRole/Write |  |  |  |  |  |  |  | X |  |
| Principal/List |  |  |  |  | X |  | X |  |  |
| Principal/Read | X |  | X |  |  |  |  |  |  |
| Principal/Write |  |  | X | X |  |  |  |  |  |
| Principal/LogicalDelete |  |  |  |  |  |  | X |  |  |
| Principal/AddSuperAdmin |  |  |  |  | X | X | X |  |  |
| Principal/AddSysAdmin |  |  |  |  |  | X |  |  |  |

### Extension Module ACM

Extension SADs will rely on the above, and extend it by defining:

* Resources: the business domain entities exposed by the extension Module. Eg: Schools in one case, or Students, Grades, in another, etc.  
  Includes:
  + Reference Resources: any reference data on which the Module’s primary Resources rely on (Gender, Region, etc.)
* Roles (eg: SchoolAdminstrator)
* Rules (any custom access logic, over and above straight forward Role Based Access Control RBAC controls. It is expected that most modules will not have require this level of control and will simply rely on defining an RBAC based ACM).

For example, an Extension Module SAD for Education Count School Directory might have an short addendum to the above ACM table that looks as follows:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rource + Operation on Resource | User | Authenticated User | Sys | AuthenticatedUser | SuperAdmin | SysAdmin | SecuritySpecialist | BusinessUser | BusinessSpecialist | SchoolPrincipal |
| EdCountSchoolDirectory/Read | X |  |  |  |  |  |  |  |  |  |
| EdCountSchoolDirectory/Write |  |  |  |  |  | X |  |  |  |  |

Note:  
Extension Module ACM tables cannot loosen Core ACM.

## Resource Data Classification

Whereas Roles control access to Operations applicable to Resources, Data Classification control access to Resources.

Core controls access to Resources in the following way:

* A Role has a DataClassification
* A Principal is assigned one or more Roles
* A Principal can have a DataClassification assigned directly, above and beyond the DataClassification of any Role assigned to the Principal.
* All Resources have a Data Classification.

The result is a Principal’s Role may be allowed to Operate on a Resource, if the specific Resource has been DataClassified to a higher level than the highest DataClassification available to his applicable roles, then the user cannot access the Resource.

## Testing Data

Application Development must be performed with limited test data developed specifically for development test purposes, applied by automation.

# Functionality Qualities View

The section of the document describes the functionality of the Solution.

Although Qualities, outlined in the [Qualities View](#_Qualities_View), include Functionality, it is a convention that [Functionality Qualities](#_Functional_Qualities_View) of a system are described within a distinct SAD View.

Note:   
As Core is a Base addressing System Domain functionality common to additional Business Modules, it does not have Business Domain schemas or entities within it. For Business Domain Schemas and Entities (eg: Students, Schools, Resources, Procurement, etc.), refer to other SAD’s describing Modules that rely on this Core SAD.

## Synopsis

The solution provides functionality to meet the functionality of multiple stakeholders, starting from the the functionality expected by manage access to the system, continuing onwards to the stakeholders expected to support end users, through to application and infrastructure support services.

Bussiness Functionality expected by Business Stakeholder users is addressed in Module SADs.

## System Capabilities

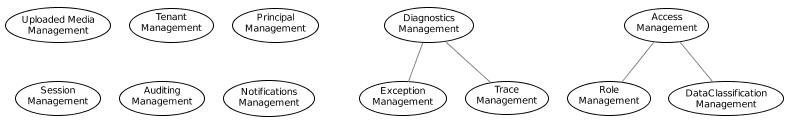
Listed below are any key Capabilities of the system, which underpin the Functionality delivered.

* Access management
* System Record Management (Principals, Roles, System Error Reports, Diagnostics Messages, Notification Messages)
* Notification Distribution
* Queryable API data exposure.

## System Functionality

Listed below are any key system functionality groups.

* Principal Management
* Role based Access Management:
* System Error Report Management
* Diagnostics Trace Message Management
* Notification Management
* Uploaded Media Management



The functionality is provided by API. Core does not provide any UI for these concerns -- Modules can.

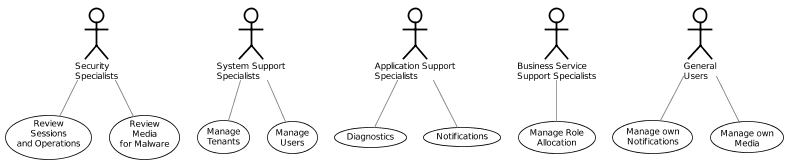
### Out of Scope Functionality

Listed below are any key system or business functionality that will **not** be included in the current Solution.

* Management of Business Concerns.

## Role Functionality

The System Functionality is listed below, grouped by Stakeholder group:

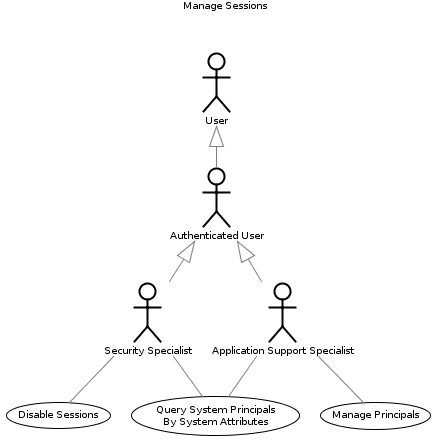
s

## Security Specialist Functionality

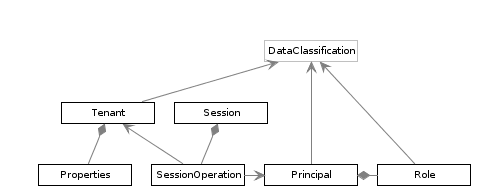
### Session and Operation Query Functionality

Core provides Functionality to assist Security Specialist stakeholders to investigate audited Session Operations as well as assess current Session Operations.

Core provides authenticated and authorised access to Core APIs to query current and past Principal Sessions and the Operations that were requested as part of those sessions.



The data returned will be JSON serializations of DTOs the Projections of one or more of the following Core Entities:



Notes:

A Principal (ie User) can have more than one Session open at one time. This is a common scenario: a user may have a Chrome and Edge browser open, as well as a cell phone regularly poling from their pocket.

A Principal will initiate multiple Operations per Session: \*all\* API Operations are audited, including single Resource View Operations and List Operations (a List operation, returning multiple Resource Summaries counts as a single Operation).

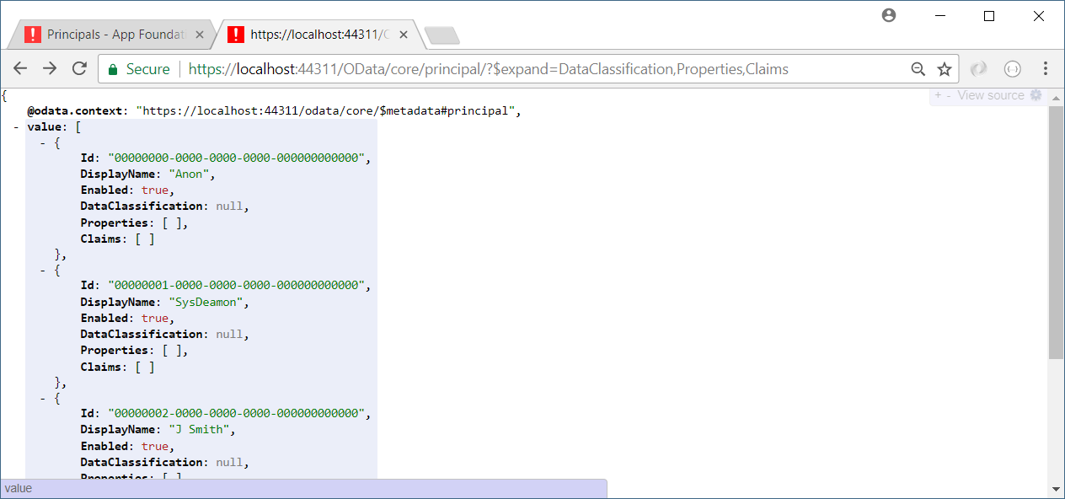
Active Session can be disabled, although it is expected Principal Management APIs would be used to disabled Principals from all Sessions.

A Session is associated to a single System Tenant at a time.

Multiple Sessions can span multiple Tenants.

### Examples

A query from an appropriately authorised Principal (refer to the [Access Control Model](#_Access_Control_Model)) would provide a response similar to:

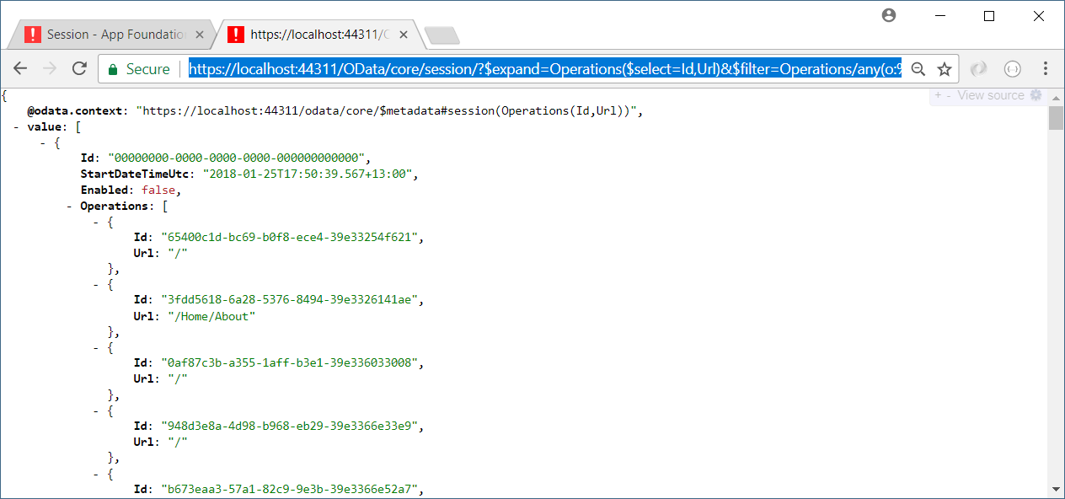


Users can bring up one or more specific Sessions, and the Operations associated to it.



Security specialists can use OData queryability to home in on a specific Operations, from a specific IP:

<https://localhost:44311/OData/core/session/?$expand=Operations($select=Id,Url)&$filter=Operations/any(o:%20o/Url%20eq%20%27/Home/About%27)>



{

@odata.context: "https://localhost:44311/odata/core/$metadata#session(Operations(Id,Url))",

value:

[

{

Id: "00000000-0000-0000-0000-000000000000",

StartDateTimeUtc: "2018-01-25T17:50:39.567+13:00",

Enabled: false,

Operations:

[

{

Id: "65400c1d-bc69-b0f8-ece4-39e33254f621",

Url: "/"

},

{

Id: "3fdd5618-6a28-5376-8494-39e3326141ae",

Url: "/Home/About"

},

{

Id: "0af87c3b-a355-1aff-b3e1-39e336033008",

Url: "/"

}

...etc...

],

Operations@odata.nextLink: "https://localhost:44311/odata/core/session(00000000-0000-0000-0000-000000000000)/Operations?$skip=100"

}

]

## System Support Specialist Functionality

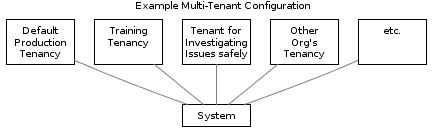
### System Tenant Management Functionality

Core provides Functionality to manage multiple Tenants within the same Service.

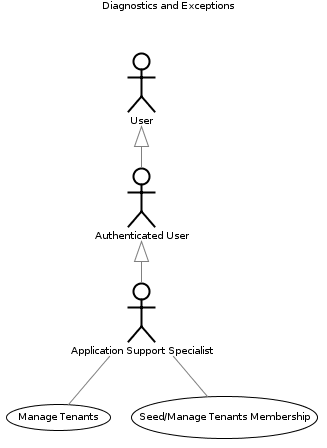
System administrators can configure the Default tenant for this Organisation, and provide additional Tenants for other purposes. Common scenarios for additional Tenants isolated from other Tenants include Training, Support Investigation and even other Organisations/Agencies.

Note:

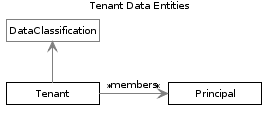
This last aspect is key to provide services to others in the sector in very little time, at minimal cost (effectively, a new, isolated, certified and accredited system in less than a morning of effort).



As needed, new Tenancies can be activated or disabled, and Users added/removed from them:

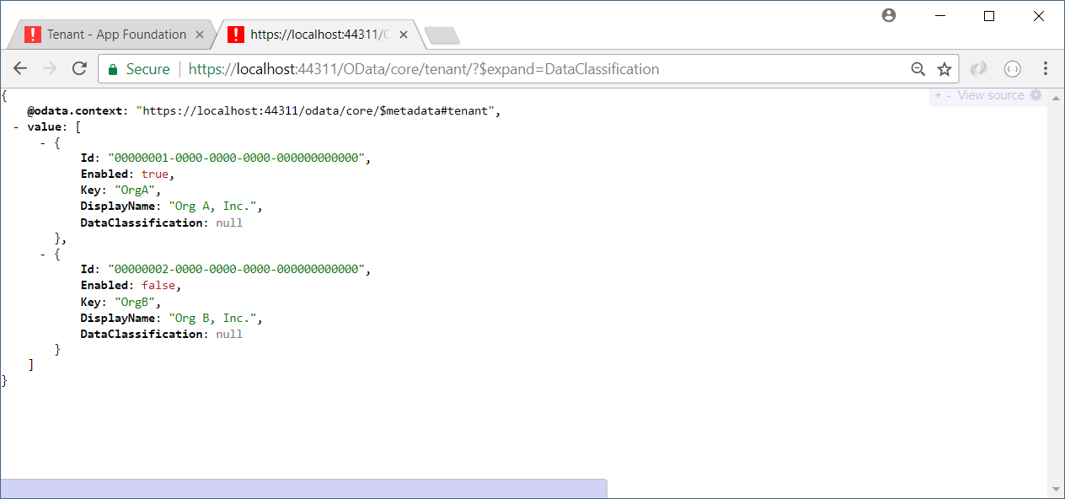


The data returned will be JSON serializations of DTOs the Projections of one or more of the following Core Entities:



#### Example Response

A query response from an authorised user (refer to the Access Control Model) would result in a JSON encoded response similar to the following:



{  
 @data.cntext: "https://lcalhst:44311/data/cre/$metadata#tenant",  
 value:   
[  
 {  
 Id: "00000001-0000-0000-0000-000000000000",  
 Enabled: true,  
 Key: "rgA",  
 DisplayName: "rg A, Inc.",  
 DataClassificatin: null

},  
 {  
 Id: "00000002-0000-0000-0000-000000000000",  
 Enabled: false,  
 Key: "rgB",  
 DisplayName: "rg B, Inc.",  
 DataClassificatin: null  
}  
]  
}

#### Example Use Cases

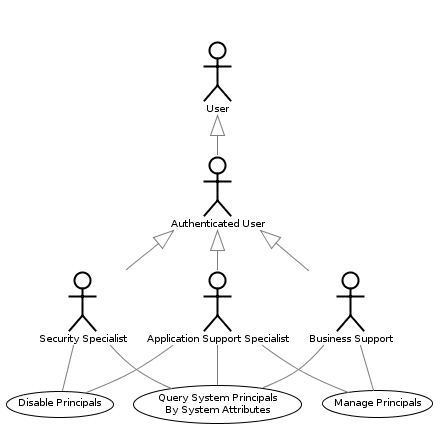
The API endpoint can be update remotely by System Administrators, to create a new tenant as required, (eg: create a Training Site, add another Sector agency, School, Partner, etc.) logically or physically separated.

### System Principal Management Functionality

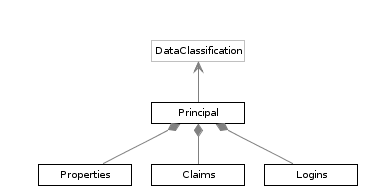
Core provides Functionality to assist Security Stakeholders and Support Stakeholders.

The systems provides APIs to query and manipulate Principals and Roles in an authenticated, authorised and audited manner.

These APIs can be used by authenticated users – principally Business Support resources and System/Application Support Specialists – to manage invited users.



The data returned will be JSON serializations of DTOs the Projections of one or more of the following Core Entities:



Notes:   
Core does **not** Authenticate Users – but delegates Identity Verification to external accredited IDP services.

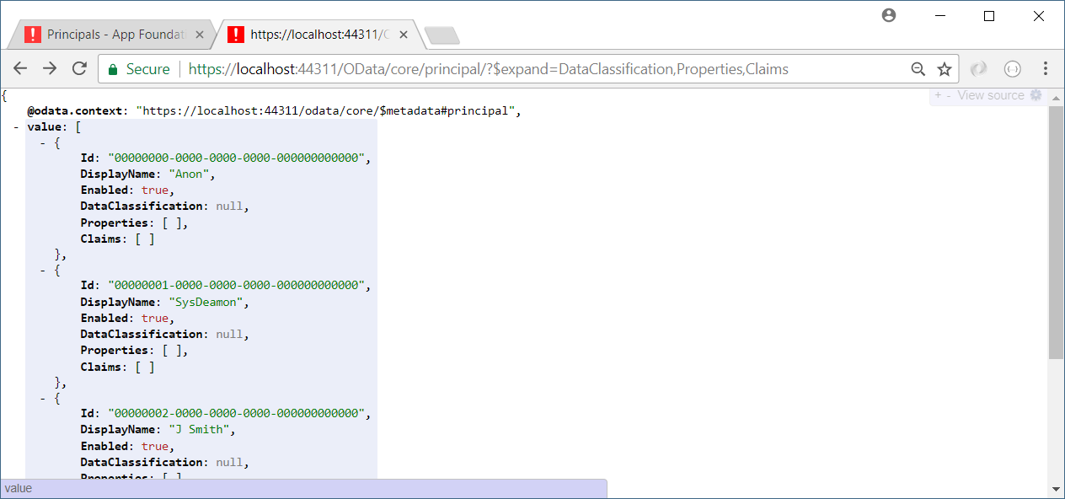
Each Principal can have multiple Principal Properties (also known as Profile Settings) to persist preferences and individual settings. Claims not suitable for external storage can be stored in-system.

A Principal can be related to one or more external IDs Services (Logins).  
A Principal is only permitted to access Resources with an immutable DataClassification Reference Type equal to or lower than the DataClassification assigned to the Principal (or the DataClassification of the Roles assigned to the Principal – see Business Support Functionality, below).

Principals can be Disabled (the process disables all Sessions to that User).

#### Example Response

A query from an authorised Principal (refer to the [Access Control Model](#_Access_Control_Model)) would result in a response similar to the following:



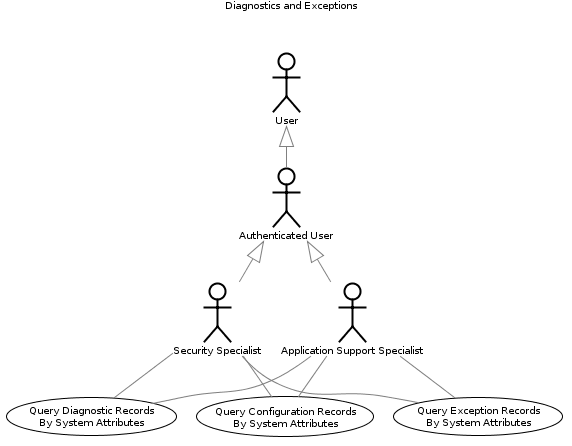
{  
@odata.context: "<https://localhost:44311/odata/core/$metadata#principal>", value:   
[  
{  
Id: "00000000-0000-0000-0000-000000000000",  
DisplayName: "Anon",  
Enabled: true,  
DataClassification: null,  
Properties: [ ],  
},  
{  
Id: "00000001-0000-0000-0000-000000000000",  
DisplayName: "SysDeamon",  
Enabled: true,  
DataClassification: null,  
Properties: [ ],  
},  
{  
Id: "00000002-0000-0000-0000-000000000000",  
DisplayName: "J Smith",  
Enabled: true,  
DataClassification: null,  
Properties: [ ],  
},  
…etc…  
]  
}

#### Example Use Cases

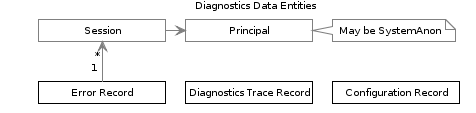
The above API can be used by user interfaces to render a searchable list of users, in order to provide functionality to authorised users. Example updates would include changing a Principal’s DisplayName within the system, adding new Properties, Disabling, etc.

### Diagnostics Functionality

Core provides the functionality required by Application Support Specialists and Support Specialists to investigate records of Handled and Unhandled Exceptions, double check how the system is Configured, and retrieve Diagnostic Trace Records.



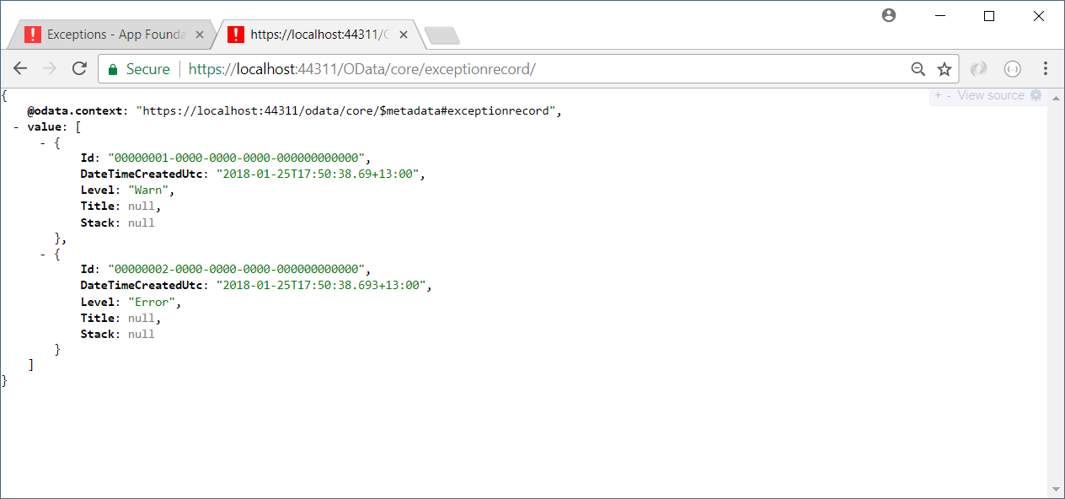
The data returned will be JSON serializations of DTOs the Projections of one or more of the following Core Entities:



Note:  
ConfigurationRecords are in-memory records of a system’s startup operations, and can be queried to ensure that correct deployment configuration steps were undertaken. Common examples of verifications that can be obtained by querying these records are ensuring in-use Certificates are valid for the next 6 months, HTTPS is enforced when communicating between Client and Service, and again between Service and dependency services, ConnectionStrings are using IntegratedSecurity as opposed to sending clear-text Credentials over 1433, etc.   
Verification of configuration in this manner is right-sized for most systems, reducing uncertainty, and therefore the requirement for expensive testing of dubious purpose.

#### Example Request

A simple query from an appropriately authorised Principal (refer to the [Access Control Model](#_Access_Control_Model)) for errors that have occurred lately would be similar to the following:



{

@odata.context: "<https://localhost:44311/odata/core/$metadata#exceptionrecord>",

value:

[

{

Id: "00000001-0000-0000-0000-000000000000",

DateTimeCreatedUtc: "2018-01-25T17:50:38.69+13:00",

Level: "Warn",

Title: null,

Stack: null

},

{

Id: "00000002-0000-0000-0000-000000000000",

DateTimeCreatedUtc: "2018-01-25T17:50:38.693+13:00",

Level: "Error",

Title: null,

Stack: null

}

]

}

### Example Use Cases

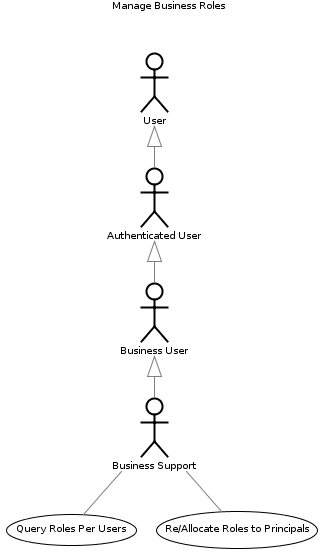
The above queryable information provides the functionality to filter the view of errors down to a specific time rnage, error level, unique identifier, error type -- or combination thereof.

The information can then be used by a user interface, or monitoring tool, alerting system admins of specific ones they are looking for.

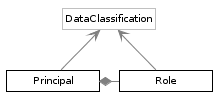
## Business Support Specialist Functionality

Core provides Functionality to assist Business Support Stakeholders to respond to their Users and Clients without requiring Application or Customer Support services for all operations.

Core provides authenticated and authorised access to APIs to query and manipulate the mutable Business Roles (as opposed to System Roles, such as SysAdmin, SupportAdmin) and their allocation to Security Principals (ie Users).



The data returned will be JSON serializations of DTOs the Projections of one or more of the following Core Entities:

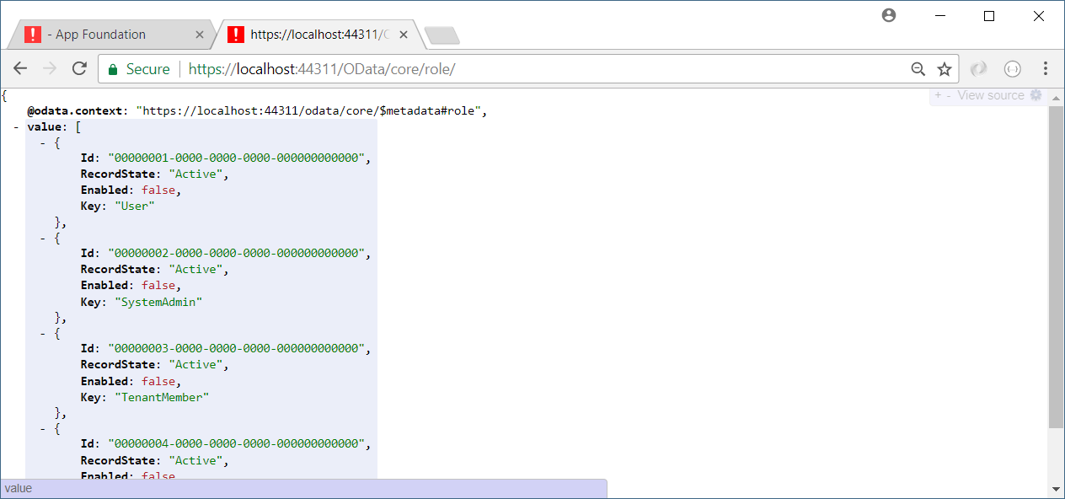


Notes:   
A Principal is only permitted to access Resources who’s DataClassification immutable Reference Type is equal to or lower than the higher of his/her own DataClassification or that of the Roles assigned to the Principal.

### Example Query Responses

A query from an authorised Principal (refer to the [Access Control Model](#_Access_Control_Model)) would result in a response similar to the following:

<https://localhost:44311/odata/core/role>



{

@data.cntext: "https://lcalhst:44311/data/cre/$metadata#rle",

value:

[

{

Id: "00000001-0000-0000-0000-000000000000", RecrdState: "Active", Enabled: false,

Key: "User"

},

{

Id: "00000002-0000-0000-0000-000000000000", RecrdState: "Active", Enabled: false,

Key: "SystemAdmin"

},

{

Id: "00000003-0000-0000-0000-000000000000", RecrdState: "Active", Enabled: false,

Key: "TenantMember"

},

{

Id: "00000004-0000-0000-0000-000000000000", RecrdState: "Active", Enabled: false,

Key: "TenantAdmin"

}

…etc…

]

}

### Example Use Cases

User interfaces can be developed provide to Business Support Stakeholders the ability to assign or revoke Module specific Roles to users specific to their Module.

## Notification Functionality

Core provides the underlying functionality to enable Notification between Users.

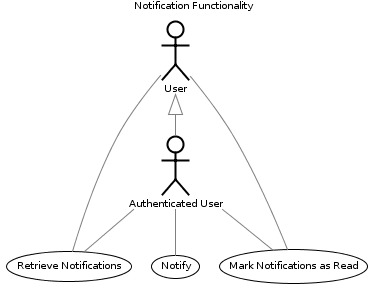
Common use cases include but are not limited to:

* Welcoming new Users, providing them with Terms and Conditions
* Notifying Users of changes to the Terms and Conditions
* Notifying Users of a Long running task being completed,
* Notifying Users of a task failing to complete,
* Notifying Users of a new release and features to expect
* Collaboration between Users.
* Notifying Specialists of Abnormal Behaviour by the System or Users (optional, as there are other means to do this)

Note:  
There are other means of doing this (eg: Azure ApplicationInsights), but do require extra integration effort.

* Notifying Support specialists of system alerts (expiring Certificates, expiring Signatures).

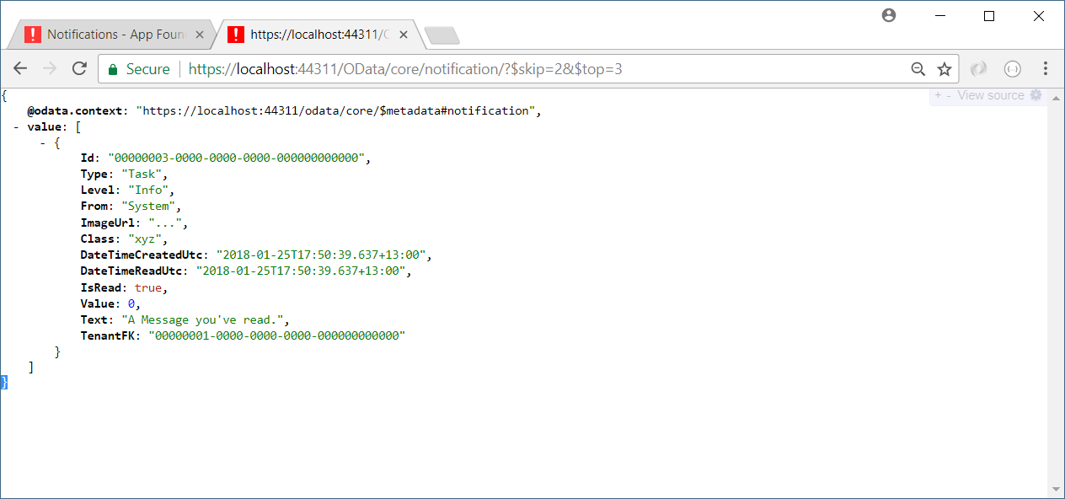
Note:   
There are other means of doing this, but do require extra integration effort.



The data returned will be JSON serializations of DTOs the Projections of one or more of the following Core Entities:

PlantUML Graph

### Examples



{

@odata.context: "<https://localhost:44311/odata/core/$metadata#notification>",

value:

[

{

Id: "00000003-0000-0000-0000-000000000000",

Type: "Task",

Level: "Info",

From: "System",

ImageUrl: "...",

Class: "xyz",

DateTimeCreatedUtc: "2018-01-25T17:50:39.637+13:00",

DateTimeReadUtc: "2018-01-25T17:50:39.637+13:00",

IsRead: true,

Value: 0,

Text: "A Message you've read.",

TenantFK: "00000001-0000-0000-0000-000000000000"

}

]

}

### Use Cases

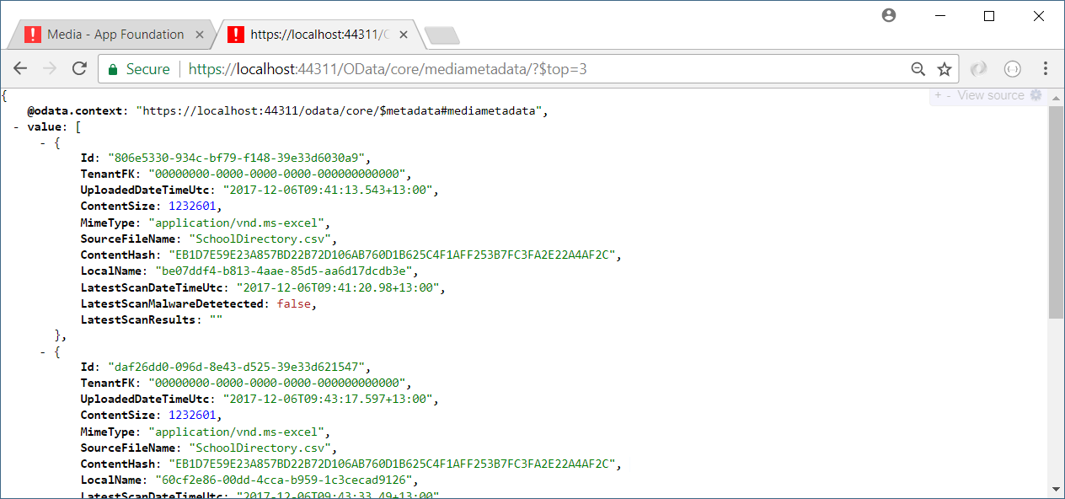
The above user specific feed is used by user interfaces to notify end users as to common events, including but limited to:

* Terms and Conditions updated
* System Maintenance Alerts (“system will be offline on Sunday, starting at 11pm, duration 2 hours”).
* Etc.

## Media Functionality

### Example

An example of an query response from an authorised user (refer to the [Access Control Model](#_Access_Control_Model)) for a list of Principals who use the system is shown below:



{

@data.cntext: "https://lcalhst:44311/data/cre/$metadata#mediametadata",

value:

[

{

Id: "806e5330-934c-bf79-f148-39e33d6030a9",

TenantFK: "00000000-0000-0000-0000-000000000000",

UpladedDateTimeUtc: "2017-12-06T09:41:13.543+13:00",

CntentSize: 1232601,

MimeType: "applicatin/vnd.ms-excel",

SurceFileName: "SchlDirectry.csv",

CntentHash: "EB1D7E59E23A857BD22B72D106AB760D1B625C4F1AFF253B7FC3FA2E22A4AF2C",

LcalName: "be07ddf4-b813-4aae-85d5-aa6d17dcdb3e",

LatestScanDateTimeUtc: "2017-12-06T09:41:20.98+13:00",

LatestScanMalwareDetetected: false,

LatestScanResults: ""

},

{

Id: "daf26dd0-096d-8e43-d525-39e33d621547",

TenantFK: "00000000-0000-0000-0000-000000000000",

UpladedDateTimeUtc: "2017-12-06T09:43:17.597+13:00",

CntentSize: 1232601,

MimeType: "applicatin/vnd.ms-excel",

SurceFileName: "SchlDirectry.csv",

CntentHash: "EB1D7E59E23A857BD22B72D106AB760D1B625C4F1AFF253B7FC3FA2E22A4AF2C",

LcalName: "60cf2e86-00dd-4cca-b959-1c3cecad9126",

LatestScanDateTimeUtc: "2017-12-06T09:43:33.49+13:00",

LatestScanMalwareDetetected: false,

LatestScanResults: ""

},

{

Id: "60831184-4d1c-7fc3-ed50-39e33d6422ad",

TenantFK: "00000000-0000-0000-0000-000000000000",

UpladedDateTimeUtc: "2017-12-06T09:45:32.107+13:00",

CntentSize: 1232601,

MimeType: "applicatin/vnd.ms-excel",

SurceFileName: "SchlDirectry.csv",

CntentHash: "EB1D7E59E23A857BD22B72D106AB760D1B625C4F1AFF253B7FC3FA2E22A4AF2C",

LcalName: "adcadb1f-faf6-4a50-8ee7-5c4d8afbf77b",

LatestScanDateTimeUtc: "2017-12-06T09:45:34.213+13:00",

LatestScanMalwareDetetected: false,

LatestScanResults: ""

}

]

}

# Information View

This section of the document describes how the Solution stores, manipulates, manages, and distributes both static information structures and dynamic information flow.

## Sumary

The system is capable of handling **IN-CONFIDENCE** information.

Information transmitted between all components using HTTP is done using confidential communication controls (HTTPS).

All access to – including Views of -- information via APIs is audited and associated to individual Sessions, in turn associated to [Principles](#_Principles), whether authenticated or Unauthenticated.

Session Operation audit records are cryptographically signed against tampering.

The system is horizontally scalable for high availability.

Both Non-relational and relational data at rest is encrypted (Microsoft, 2017).

The System is backed up regularly, keeping backups for a month.

The Service is deployed to a near-shore data-environment that is geo-replicated to another near-shore data-centre for Disaster Recovery.

Physical and virtual remote access to the data centres is limited and audited.

Authentication of Principals accessing the data is handled externally to the system by trusted Identity Provider (IDP) Services, using the Open ID Connect (OIDC) Protocol, over TLS 1.2 Secured HTTPS confidential communication channels.

Note:   
The system does **not** persist Password or Pin Code credentials.  
The system will persist user a principal’s email address (for Notification purposes), a preferred Display Name (not a Username).

The system may persist the principal’s phone number, if available.

The system will persist the IDP’s identifier for the user, which may or may not be unique per system (it depends on the IDP, not Core).

A Module dependent on Core may persist additional info about the user, appropriate to the Business Domain of the Module.

* Deployed to restricted access data centres managed by processes that conform to SO/IEC 27018:2014 and ISO/IEC 27001.
* Communication between Clients and Service are confidential (over HTTPS).

## Data Classification

Core will be accredited and certified to handle **SENSITIVE** rated resources.  
  
Extension Modules may be rated lower.

Note:  
The highest data classification of the data which will be handled by the first BASE Extension – the Education Count School Directory API BASE Extension -- will be **UNCLASSIFIED**, which the BASE system will be automatically rated to handle when it becomes certified to handle SENSITIVE information.

## Data Ownership and Stewardship

The system data contained within Core is owned by the Organisation which deployed the system.

The business data contained within the system is owned by the Tenant to which the data is associated, stewarded by the Business unit managing the Extension Module.

Certain individual Information attributes managed by the Core module are Stewarded by the Organisation, but owned by the end user.

In other words, data that is managed by Module XYZ, within Tenant ABC, is owned by ABC, Inc, but the responsibility of XYZ – unless we’re talking about Personal Identifying Ifnformation which may actually belong to the Person itself.

## Data Sources

* **IDP Attributes:**The system manages copies of Attributes it initially received from external Identity Providers (IDPs).

Attributes received include:

* + Display Name
  + Business Role (optional)
  + Email Address (for Notification purposes).
  + Identifier given to the Principal by the IDP.

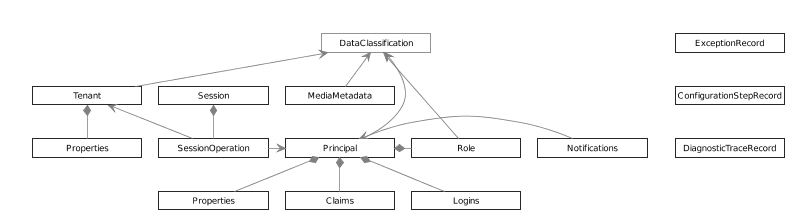
Note:   
Whether the IDP issued Identifier is unique per system depends solely on the trusted IDP’s design.

* **External Databases:**
  + Not applicable from Core Module.  
    Refer to other SADs that are in turn referencing this SAD.

## Data Schemas and Entities

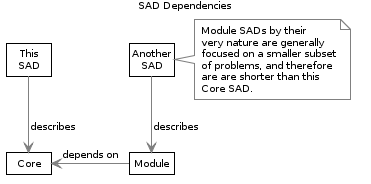
Listed below are any key architectural significant logical data entities.

* **System Entities:**
  + DiagnosticsRecord
  + ExceptionRecord
  + ConfigurationStepRecord
  + Reference Data:
    - DataClassification
  + Tenant (and Properties)
  + Role
  + Principal (and PrincipalProperties, PrincipalClaims, PrincipalLogins)
  + Session
  + SessionOperation
  + Notification
  + MediaMetadata
* **Business Entities:**
  + Not Applicable in Core.



As Core is a Base system for Business Modules, it does not have Business Domain schemas or entities within it.

For Business Domain Schemas and Entities (eg: Students, Schools, Resources, Procurement, etc.), refer to other SAD’s describing Modules that rely on this Core SAD.



## Data Security

Physical, Access, Channel, and Control plans are discussed in the [Security View](#_Security_View).

## Testing Constraints

Application Development will be performed with limited test data developed specifically for development purposes.

Data Migration and Data Quality development will be developed with obfuscated data.

Data Migration and Data Quality verification runs using non-obfuscated Production Data will be limited to Pre-Production/UAT environments.

Note:   
For Compliance reasons, development is not to be done against copies of Production data.  
Additionally, under no circumstances is un-obfuscated Production Data be made available outside of Organisation controlled environments.

## Data Provisioning

Specifically excluded from scope, and design, is any database pre-provisioning process.

External data needed by Core or Modules dependent on Core is to be developed as an idempotent, paged import process where external data is passed through the Application Layer’s validation logic.

Note:   
although a common provisioning pattern in the previous decade, provisioning the database directly is not advisable. To import data directly into the database, bypassing the validation logic, would increase the risk of introducing unintended consequences impacting user experience.

Consider using Data Factory if a connection to a database is required, in order to develop Message Queues that can be passed through the API Service Facade. Consider using the MediaUpload functionality if the import is via a CSV.

## Data Access Auditing

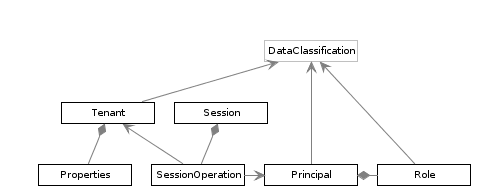
Although most databases can provide database level auditing, it generally is a low value security feature for auditing end user activity (all activity is logged against the system’s service account – not the actual user). Hence Data retrieval and manipulation auditing is performed at the System tier.

All API operations are intercepted and logged within the SessionOperation table, which is associated to a current Session, which is in turn associated to the Principal in question, whether authenticated or not (Anon user).

The SessionOperation table records the UTC date and time, IP of the client user agent, the request URI, and the outcome of the Operation.

The SessionOperation table record can be embedded with a digitially signature of the record’s attributes, using a certificate’s private key, to ensure non-tamperability.

Note:   
User Agent IPs are recorded in the SessionOperation table as Sessions can last a configurable time (eg: up to 30 days) during which time a mobile device can be reissued different IPs as they move around. It is common that a single Session from a Mobile device have multiple IPs.



A Principal (ie User) can have more than one Session open at one time. This is a common scenario: a user may have a Chrome and Edge browser open, as well as a cell phone regularly poling from their pocket.

A Principal will initiate multiple Operations per Session: \*all\* API Operations are audited, including single Resource View Operations and List Operations (a List operation, returning multiple Resource Summaries counts as a single Operation).

Active Session can be disabled, although it is expected Principal Management APIs would be used to disabled Principals from all Sessions.  
A Session is associated to a single System Tenant at a time.

## Data Retention

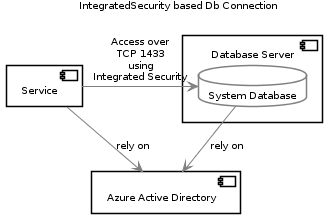
All information within the system, including Personally Identifiable Information (PII), is kept only as long as is required to legally meet its regulatory obligations.

If visible to a Search Engine, upon request a person may request their own PII information to be removed from the system.

Non- PII is retained for the duration of the system, logically-deleted and logically archived as required.

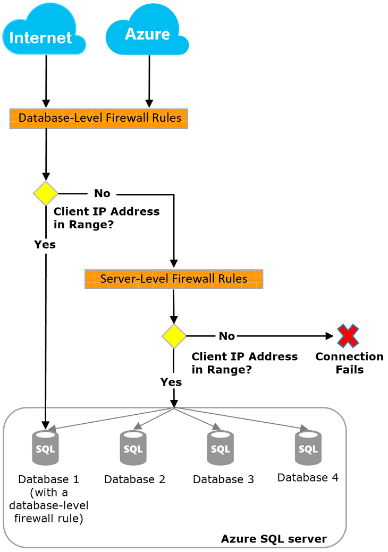
## Database Access

Access to the encrypted-at-rest database itself from the System is done over TCP port 1433, using Integrated Security.



Note:   
Integrated Security is a security feature that allows the Service to communicate with the database server, without divulging the service account’s credentials.  
Note that the Account used is the System’s Service Account.

Access to the database can be controlled (using Database Firewall Rules) to allow only access to the database server from the service, and/or access from support specialists using a SQL Server Management Service (SSMS) User Agent. This access should be temporary.



## Database Users and Logins

TODO: List the DB System Users (generally includes

\* a Srv Account from the System, with full rights in order for the System to be able to deploy Db Schema changes as CodeFirst,   
\* a Group for Analysts to analyse data in the Reporting database (not the Production Database),  
\* a Group for the 2nd Tier Support Personnel to analyse issues using Read Only,   
\* a Group for 3rd Tier Support Personnel to analyse using Read/Write.  
\* a Group for Temporary access from App Developers to Read only, to assist 3rd Tier.  
DBUSER-xxxx

Within the database, The Application Layer accesses the Data Storage Layer using secure connection strings, using Integrated Security.

The Integrated Security relies on a dedicated Service Account based Database Users being provisioned, along with a dedicated Database Login within the target database.

The list of Users within the database are:

* SrvAccount, with rights to create a database on the server, and rights to update schemas within the database (required for the System to be able to deploy Db Schema changes as CodeFirst),
* a Group for Analysts to analyse data in the Reporting database (not the Production Database),
* a Group for the 2nd Tier Support Personnel to analyse issues using Read Only,
* a Group for 3rd Tier Support Personnel to analyse using Read/Write.

## Database Security

The database is encrypted by Storage Service Encryption (SSE).

Being hosted in Azure, the database is not physically removable by either the PAS vendor, or Organisation resources.

## Operation Log Policies

The PAS service includes operation logging by default.

## Data Backup Policies

The PMS Service includes regular full and incremental backups.   
It is the PMS Service vendor’s responsibility to regularly ensure backups are usable.

* Backups are separate per Tenant.

## Data Archiving

The PMS Service provided by the vendor includes regular backups that are remotely available for download by the Organisation for additional manipulation as needed.

Records can be soft-archived to improve query performance.

## Data Integration

TODO: Restate the systems mentioned under Context to which the Data will be synchronized.

The Core element does not need to synchronize data with other systems, unless we are talking about the initial recording of Attributes provided by an external IDP, at first logon.

## Data Exportation

TODO: Do not proceed with committing to a design or service from which there is not a process by which data can be extracted.

Provided sufficient Permissions, all data within Core can be projected to entities serialized in JSON format, in such a way as can be manipulated and imported into another system. The Query Capabilities which ODATA provides makes the export process highly customizable.

Key Data that can exported from Core using APIs include, but are not limited to:

* Tenant
* Roles
* Principal (and their Properties, Claims, and relationships to other IDPs)
* Sessions (and the Operations that occurred within them)
* Diagnostics
* Media

Note:   
In addition to API access, Data can be exported via direct query to the database.

## Data Quality

TODO: A common key objective is to decrease dual entry and improve data quality. Especially if importing from a system being retired. Describe how the Data Quality during the import phase, and subsequent operations.

DQ-xxxx

### Validation

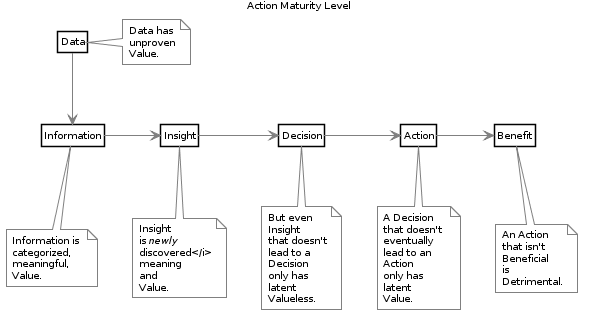
Core provides a common ValidationService that itself and dependent Modules can use at the API Façade to validate incoming messages, based on the submitted entity’s individual Validator.

PlantUML Graph

As per HTTP Code Standards, Clients that submit invalid data receive an error code with -- where feasible – simple to understand information to help the Client determine what is deemed invalid by the Core, or Modules.

## Benefit of Information Collection

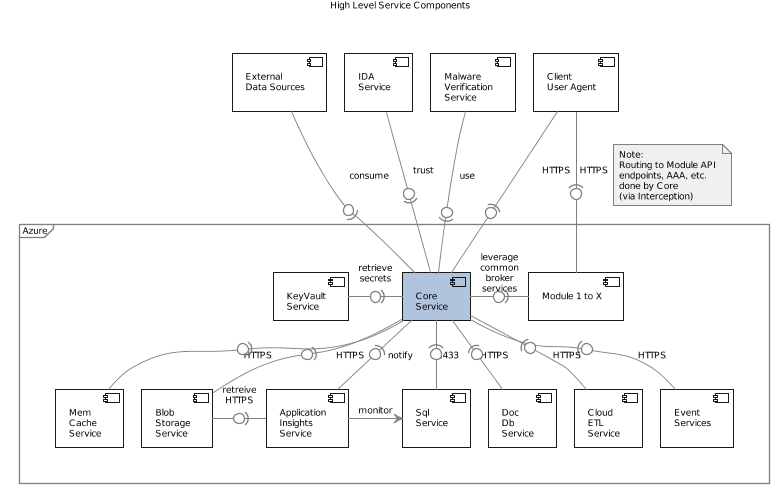
As per the Detailed Principles, information is an asset, where as data is a liability. There’s a little more to the story. When developing Modules, consider what Benefit is to be expected from the expenditure of persisting the data.



# Integration View

## Synopsis

The BASE system integrates with multiple other services, both cloud and on-premise hosted services.

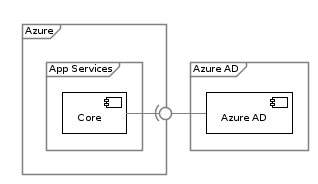


## Azure Active Directory Service Integration

The system is deployed to Azure PaaS App Services, and is automatically assigned an AAD managed Service Account.

The Service Account can be provided access to other Services, such as Azure Key Vault, and private Blob Storage (see below).

Note:  
Persons new to Azure development may be confused as to how to put Azure AAD to best use, and the domain boundary between system authentication, and user authentication. AAD is used to manage service account to other Azure Services (Key Vault, etc.). AAD can \*also\* be used to authenticate users within the same tenancy -- but this is a separate domain, and one can be purely OIDC based.

[](http://skysigal.com/lib/plugins/plantuml/img.php?width=0&height=0&title=PlantUML%20Graph&align=&version=2011-07-16&md5=e1bb474fc70fb155e9f6f577acf9c07c)

## Azure Key Vault Services Integration

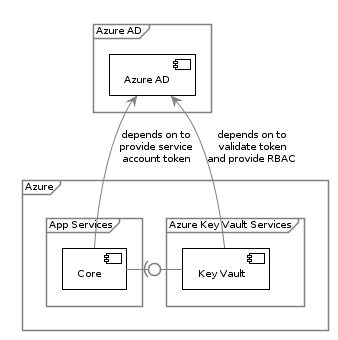
The Azure Key Vault Service is used to secure sensitive information used by applications.

Secrets entrusted to a Key Vault to be retrieved by Core include:

* The Public Blob Service Container *Shared access signature (SAS)*, which is in turn used to generate Blob Item SAS signatures (see Blob Storage below)*.*
* The Client Subscription Id and Client Secret used to authenticate to the anti malware service (see below).

The above secrets are placed in the Azure Key Vault by Infrastructure Support Services.

Access to the vault is then granted to the deployed application’s Azure AD (AAD) issued service identity.

[](http://skysigal.com/lib/plugins/plantuml/img.php?width=0&height=0&title=PlantUML%20Graph&align=&version=2011-07-16&md5=28605a48d968a91e8fd1c154fc5d1552)

## Azure SQL Services Integration

The System uses an Object Relational Mapping (ORM) service to persists relational data records, described within the [Information View](#_Information_View) -- within Azure’s SQL Services.

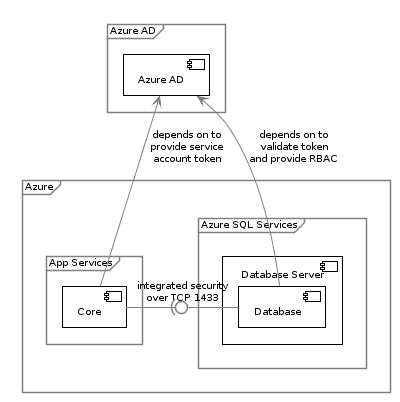
A step of the automated deployment pipeline’s build process is to assemble a default connection string to the environment’s default database.

The target environment’s default configuration string is persisted within the deployed environment’s configuration file (it’s web.config file).

Note:   
the term “Default” connection string is used because depending upon tenancy configuration settings (whether to use logical tenancy separation or physical tenancy separation), the default connection string is used as is by service (dependent on logical tenancy separation), or dynamically updated at runtime to distinguish between physical tenancies.

The connection string must be developed using IntegratedSecurity, where the deployed App Service’s service account is identified.

Note:  
The alternative -- where the UserName and Password are injected directly within the Connection String -- is not accreditable as the Credentials are passed in clear text over unsecure channels, requiring most costly network services to secure the channel by firewall source rules.

[](http://skysigal.com/lib/plugins/plantuml/img.php?width=0&height=0&title=PlantUML%20Graph&align=&version=2011-07-16&md5=c764ab4b2c57a50d6811ee997413a7a1)

## Azure Storage Account Services Integration

TODO: Explain use of retrieval of SAS tokens from Key Vault + rules

Core provides a BlobStorageService to itself and extension Modules.

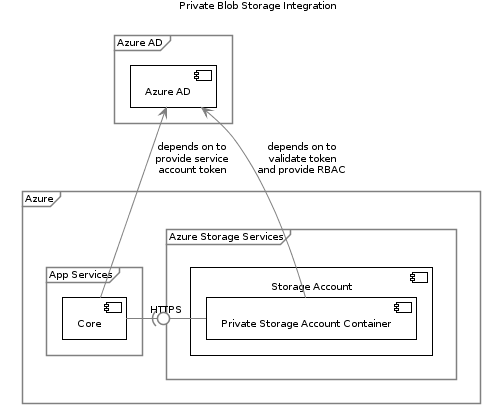
This storage provides managed access to Blob Azure Storage Account Containers for the storage of blobs of data, including uploaded media, and temporary (rolling) diagnostics trace messages.

### Private Blob Storage

Diagnostics Trace Messages are stored in *private* containers, secured by Azure AD, using Role-Based Access Control (RBAC) rules to secure the container for access only the the Service Account under which the system runs.

The items within the blob container – even though they all have an url associated to each and every one -- are not directly available from the web due to the RBAC rules preventing access.

Note:  
It is not accreditable to access the service using Service Account Access Keys within the web.config.   
It is acceptable that the Service Account Access Keys are persisted within Azure Key Vault – but it does add deployment pipeline complexity for no clear gain.  
Prefer to use Azure AD RBAC control to *Private* Blob Storage.

[](http://skysigal.com/lib/plugins/plantuml/img.php?width=0&height=0&title=PlantUML%20Graph&align=&version=2011-07-16&md5=95de7dbe3f53bd8f3f2051b3d514e9d9)

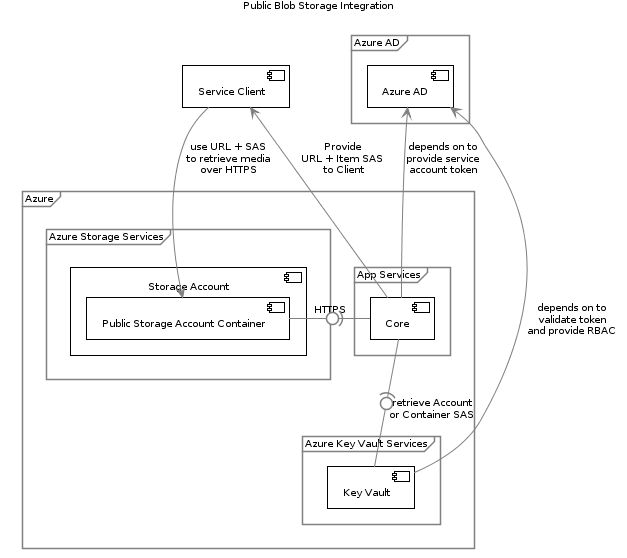
### Publicly Addressable Blob Storage

Media intended to be presented to end users (eg: media uploaded by end users) must be stored in public blob containers, addressable from end user clients (eg: browsers) that are not part of azure AD.

Azure AD RBAC are not appropriate for controlling access to publicly addressable Blob Storage. Use Shared Access Signatures (SAS).

*Shared access signature (SAS)* provides delegated access to resources in your storage account. With a SAS, you can grant clients access to resources in your storage account, without sharing your account keys. This is the key point of using shared access signatures in your applications--a SAS is a secure way to share your storage resources without compromising your account keys.

Note:  
Microsoft’s states: ”Your storage account key is similar to the root password for your storage account. Always be careful to protect your account key. Avoid distributing it to other users, hard-coding it, or saving it anywhere in plaintext that is accessible to others.”

[](http://skysigal.com/lib/plugins/plantuml/img.php?width=0&height=0&title=PlantUML%20Graph&align=&version=2011-07-16&md5=9411596136326477c7b4e046804304bd)

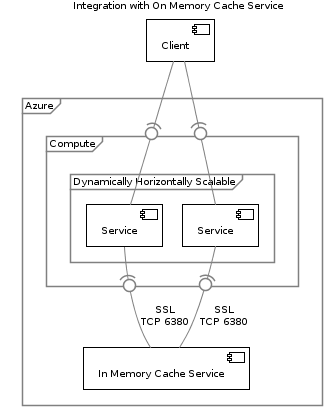
## Azure Document Database Service

The service integrates with the Azure provided CosmosDb service which provides a schemaless document storage service.

## Azure Redis Cache Service

The system integrates with the Azure Redis Cache Service to provide in-memory caching of mutable reference data between dynamically horizontally scalable instances of the service. Cross-host, in-memory caching is used to improve performance by decreasing IO to the server.

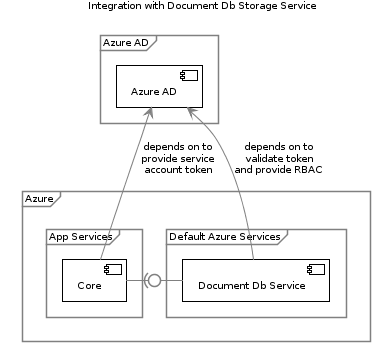
Communication to the service is over a TLS 1.1 (or 1.2) protected SSL channel over TCP 6380.



## Document Db Service

The system integrates with the Azure CosmoDb Service to meet schemaless datastorage.

Communication between the components is confidential, using TLS 1.1/1.2 to secure the HTTP channel.



## AntiMalware Service Integration

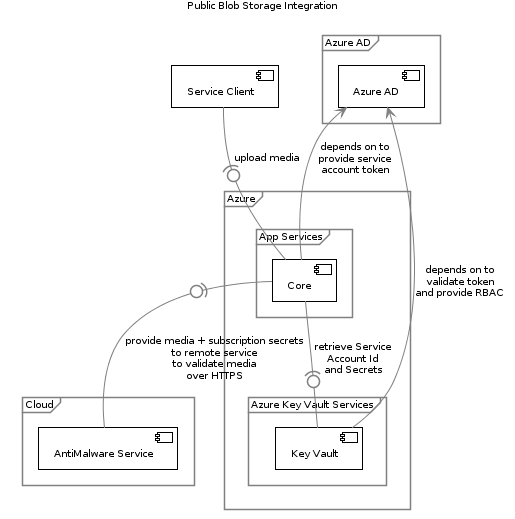
Any media uploaded to the service must be authenticated prior to being persisted, in order to not in turn infect clients downstream.

Core provides an AntimalwareService to itself and dependent Modules which wraps an external malware verification nearshore accreditable subscription service.

The Service requires secrets (subscription id + issued secret token).

The secrets are retrieved from the Azure Key Vault storage.

The secrets are placed there by Infrastructure Service Specialists.

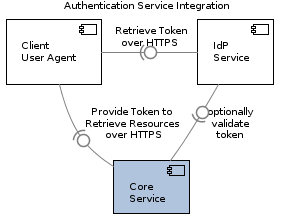
[](http://skysigal.com/lib/plugins/plantuml/img.php?width=0&height=0&title=PlantUML%20Graph&align=&version=2011-07-16&md5=b597cb2b9179d64d8b4a4e326593148a)

## Authentication Service Integration

Authentication of a Person’s Identity Credentials is handled externally, using an Identity [Token] Provider (IdP) using the OIDC Authentication Protocol.

### Open Id Connect (OIDC) Authentication Protocol

It is beyond the scope of this document to explain OIDC. The [Appendices](#_Appendices) contains a short [Summary of OIDC](#_Summary_of_OIDC) of key OIDC concepts to help understand the authentication problem domain this system address, as documented below.



TODO: Explain Token Validation for 3 cases (Fat Stateless, SPA client, and M2M).

Note that this section, like the on-prem connection section is intended to explain in one place identity access in current development. In a non-B2C specific manner (just generic OIDC).

### Modern Client side generated Views

The primary use case for modern applications is Smart Client authentication (eg: SPA applications running in desktop or mobile applications).

TODO: DESCRIBE TOKEN FLOW

Key flow steps are:

* Unauthenticated flow:
  + Person navigates User Agent to a public resource within the application (eg: [https://app.org.tld/#](https://app.org.tld/))
  + Person navigates User Agent to a private resource within the application (eg: <https://app.org.tld/#private>)
  + ...etc...

### Legacy Service side generated MVC Applications

The primary legacy use case is MVC applications.

TODO: DESCRIBE TOKEN FLOW

### Machine to Machine Authentication

TODO: DESCRIBE TOKEN FLOW

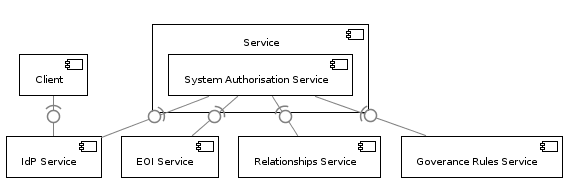
## External Data Sources

Beyond the core services with which Core communicates, any module may communicate with external data sources.

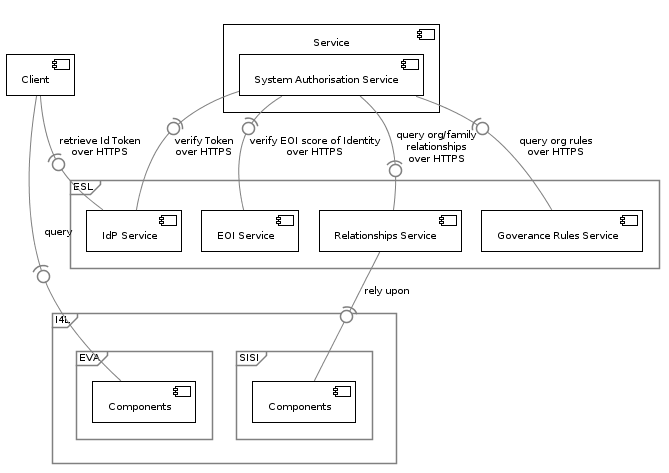
A common source of external data is on premise data which is queried and cached in the cloud hosted infrastructure for faster subsequent responses to clients.

### Identity Attribute Services

A key design principle is implementing Permission decisions closest to the system being accessed – but may need additional attributes from remote services in order to make effective Decisions.

[](http://skysigal.com/lib/plugins/plantuml/img.php?width=0&height=0&title=PlantUML%20Graph&align=&version=2011-07-16&md5=e4eb6c79829ebd37ffd985db85a809cc)

In practice, within this organisation, the above is expected to be implemented as follows:



### On Prem Operational System Access

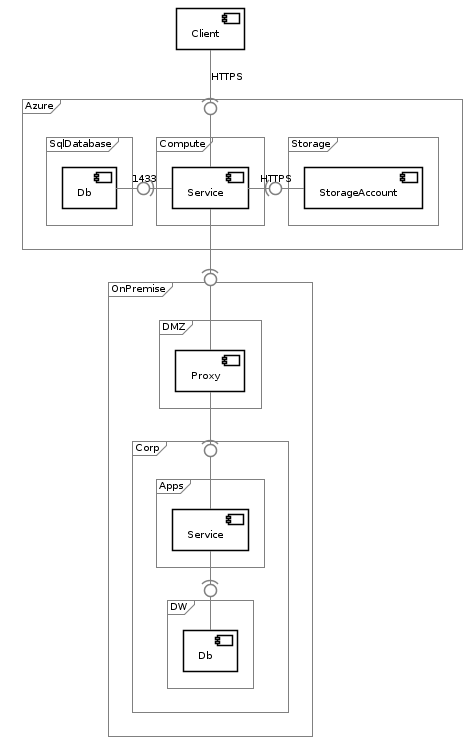
The BASE service in itself does not need to integrate with existing organisation systems. But Modules built on top of it may do so.

This section records design options to achieve these objectives within this document, in order to

* Provide guidance as to advantages and disadvantages to different approaches
* Allow subsequent Module SAD Extension Documents leverage this document’s information.

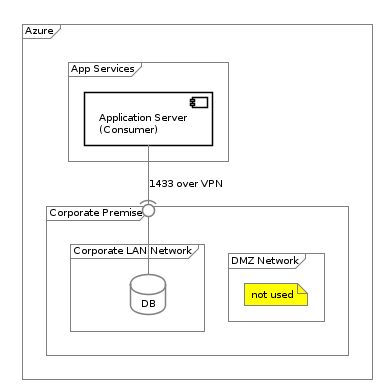
### Synopsis

At a high level, integration between cloud hosted infrastructure and on premise infrastructure can be achieved in one of several ways.

[](http://skysigal.com/lib/plugins/plantuml/img.php?width=0&height=0&title=PlantUML%20Graph&align=&version=2011-07-16&md5=f489f2a1308b66318d0c4214197d96e8)

### Option 1: Reliance on VPN

An option for cloud hosted service to access organisation managed databases is to provide the remote service a database user identity which it transmits as a clear text connection string, protected by a VPN tunnel established between the two networks.

[](http://skysigal.com/lib/plugins/plantuml/img.php?width=0&height=0&title=PlantUML%20Graph&align=&version=2011-07-16&md5=e35e6cd891f4b9da3086939fcdbde186)

Advantages:

* None

Considerations:

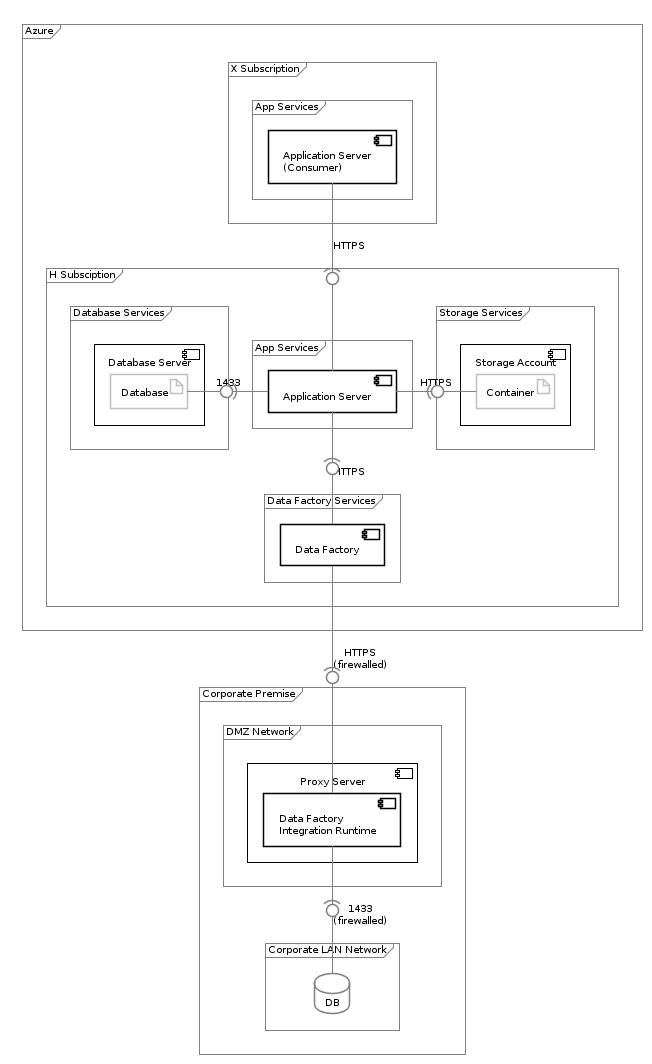
* Reliance or more or less existing skillsets.
* Will lead to a dependency on a vendor service.

Disadvantages:

* + Uses a low value pattern of sending a database username/password over cleartext that creates an expensive and brittle dependency on correct network configurations, password access control.
  + High Coupling between systems due to the lack of DTOs and/or Versioning.
  + May introducing unanticipated bugs when data is written directly to databases.
  + Can lead to duplication of logic with the consumer system to attempt to mitigate the above risks.
* When used to access an existing operational database: extends the current poor value practise of systems reaching directly into other systems data stores, bypassing system defined rich access and validation logic.
* Fails to meet several Operation Design [Principles](#_Principles).

### Option 2: Use of Data Factory

Azure provides a cloud hosted ETL service called Data Factory[[11]](#footnote-11) that can orchestrate calls to on premise agents, which in turn access on premise data stores.

[](http://skysigal.com/lib/plugins/plantuml/img.php?width=0&height=0&title=PlantUML%20Graph&align=&version=2011-07-16&md5=a0ab35715c1418cb1e0b788ced36d877)

Advantages:

* + ETL is a well-known development pattern.
  + Service is already built.
  + Does not have a dependency on correct and secure Network pairing (eg: VPN).
  + Meets more current Design [Principles](#_Principles) than a VPN based approach.

Considerations:

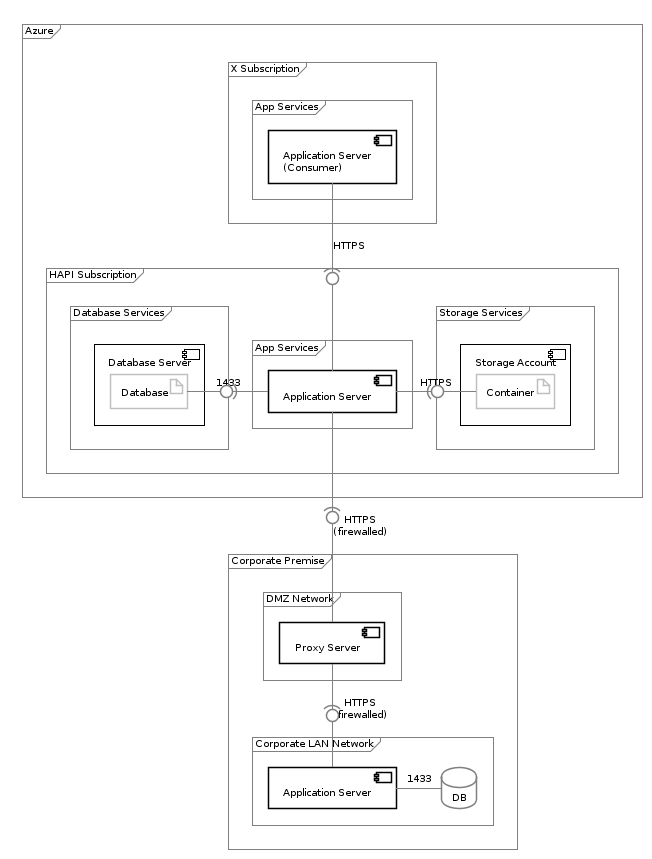
* + Requires ongoing Maintenance by a different team than the consumer's development team.
  + Requires a dependency on ADConnect to pair AAD and WAD.

Disadvantages:

* + Is not a first class development platform, adding development cost
  + RAD development patterns lead to tight coupling by bypassing the creation of DTOs and version API contracts – leading to either breaking changes, or paralysis.
  + Security is minimal, provided by:
    - Firewall rules restricting traffic whitelisted Data Factory clients.
    - Database service accounts (synced using Azure Connect).
  + Monitoring is general in nature, rather than system/use specific.
  + Extends the current poor value practice of reaching directly into other systems databases, bypassing system logic, albeit via HTTPS.
  + Introducing unanticipated bugs when data is written directly to databases.
  + Leads to duplication of logic within Data Factory to attempt to mitigate the above risks.
  + May lead to poor optimisation of cross platform traffic, adding data costs

### Option 3: Custom Service Agent

* Advantages:
  + Optimises data usage compared to direct access over VPN.
  + Ensures development quality required to better maintain security is not spread over multiple teams dabbling with Data Factory, with varying results.
  + Low Coupling between systems is enabled by providing DTOs and Versioning.
* Considerations:
  + API Service can be a shared service, or per solution as required.
  + Note: if installed as a modular shared service, Accreditation costs can be optimized.
  + Can integrate with Data Factory if that becomes necessary.
  + Does not require Data Factory to continue functioning.
  + Requires ongoing Maintenance by a different team than the consumer's development team.
  + Will meet more of the organisation’s design [Principles](#_Principles).
* Disadvantages:
  + Requires more development effort than Data Factory development.
  + Requires installation of a Proxy and Service on Prem.

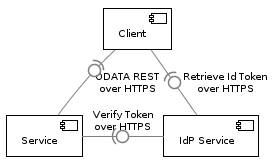
[](http://skysigal.com/lib/plugins/plantuml/img.php?width=0&height=0&title=PlantUML%20Graph&align=&version=2011-07-16&md5=ab4f7d7be423a1982fe8a93ebbfa2c64)

### On-Premise System Recommendation

The Recommendation is to use option 3, falling back to option 2 for tactical objectives.

## Client Service Integration

Finally, it goes unstated that Clients will integrate with the Service’s OData REST based APIs.

[](http://skysigal.com/lib/plugins/plantuml/img.php?width=0&height=0&title=PlantUML%20Graph&align=&version=2011-07-16&md5=27cace9b69927170239556aab1ac83cb)

# Delivery View

## Synopsis

Prior to DevOps and Cross-Team Functionality delivery approaches, it was common for Systems to be developed first then delivered, effectively putting Customers and their feedback last.

Flipping the approach and putting effort into producing an Automated Delivery Pipeline as the the First Deliverable ensure customers can delivered to rapidly and iteratively from the start.

Development and Testing are sub aspects of Delivery, and are addressed in greater detail in their own subsequent Views ([Development View](#_Development_View), [Testing View](#_Testing_View)).

## Sequence

In order to meet the Delivery Principles defined within the [Principles](#_Principles):

* + PRINC-Y-DVRI: Deliver Validated Value Rapidly and Iteratively
  + PRINC-Y-APBP: Automated Pipeline Before Product
  + PRINC-Y-DCOT: Automatable Delivery Contracts over Tests
  + PRINC-Y-AABF: Appropriate Accessibility before Functionality

Initial development effort is expended to develop a Development and Delivery Pipeline prior to commencing development on the delivered system itself.

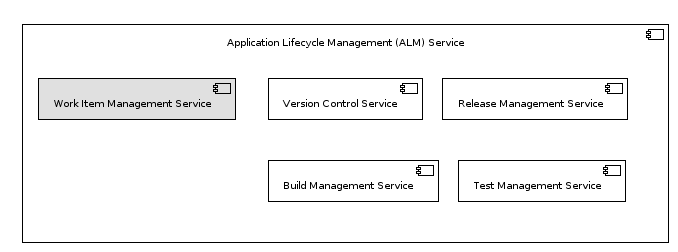
The build pipeline is developed using existing organisation provided services, which deploys to defined cloud hosted target PaaS environments, again provided by the organisation.

## Application Lifecycle Management (ALM) Suite

Delivery is coordinated using the Organisation’s ALM suite.

The organisation’s ALM Suite is Visual Studio Online, which provides several coordinated Services, including:

* **Work Item Management Service:** an online service available to organisation, sector and consultant users, to appropriately add, view and update Work Items – Agile Epics, Features, User Stories, Test Definitins .
* **Version Control Service:** a Git based Code Repository that can be cloned and added to by organisation, sector and consultant users.
* **Build Management Service: an online service to retrieve code from the Version Control Service, and build deployable artefact packages.**
* **Release Management Service:** an online deployment service that can publish built artefacts to target cloud based PaaS environments.
* **Test Management Service:** an online service for legacy testing approaches.   
  Note: we won’t be using this functionality.



## Organisation’s ALM Account

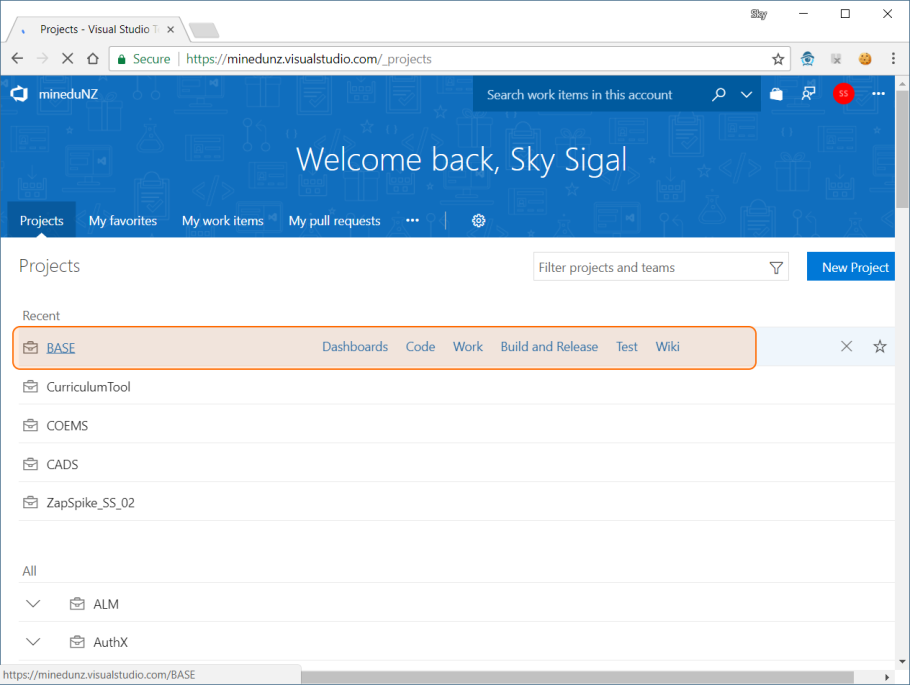
The organisation’s ALM Account is appropriately accessible at the following Url:

* <https://minedunz.visualstudio.com>

Note:   
an ongoing issue to the Organisation is the fact that the ALM Account was originally setup using Microsoft Accounts. A migration exercise of existing projects – including this one – to authenticate using Azure ADthat is expected to be required at some point in the future.

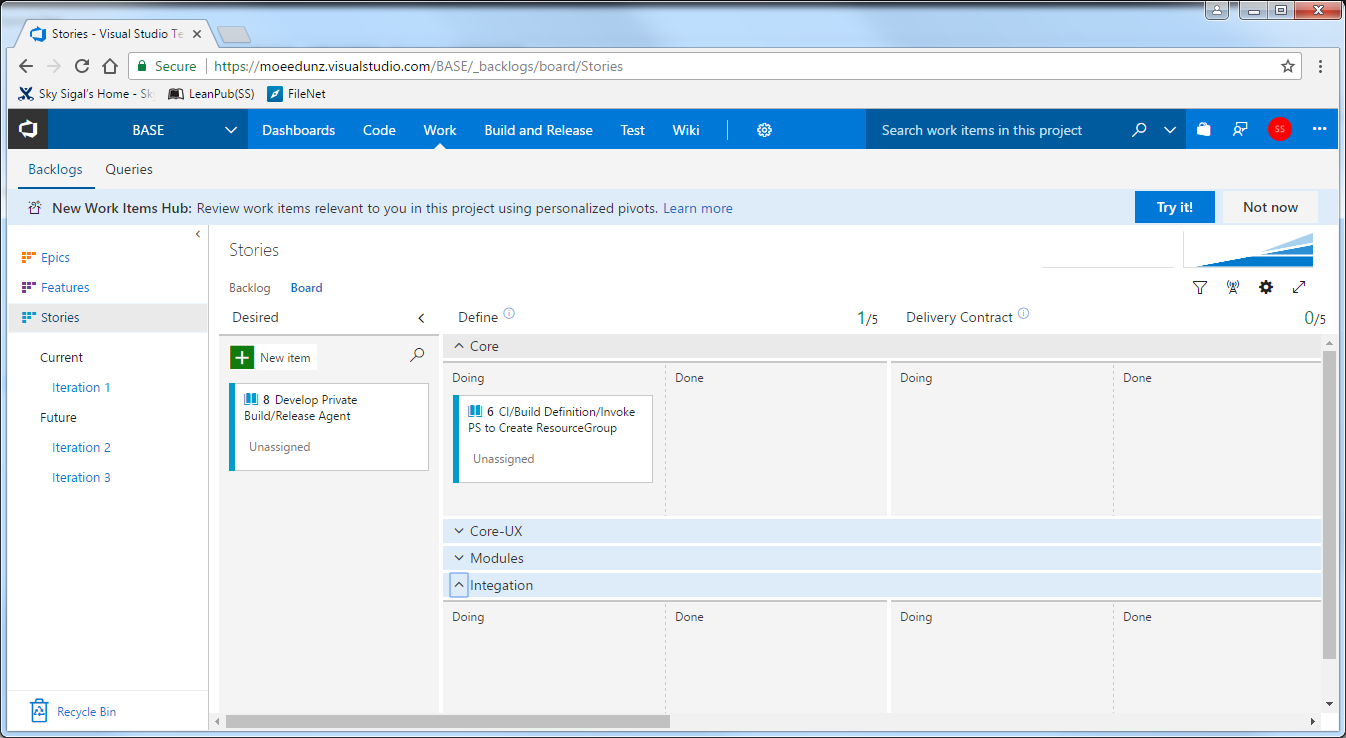
## ALM Project Space

A Project space within the ALM Account has been created specifically for the development and delivery of BASE:



## ALM Work Item Management

The ALM Service provides a common Kanban workspace that all stakeholders can contribute to according to their role.

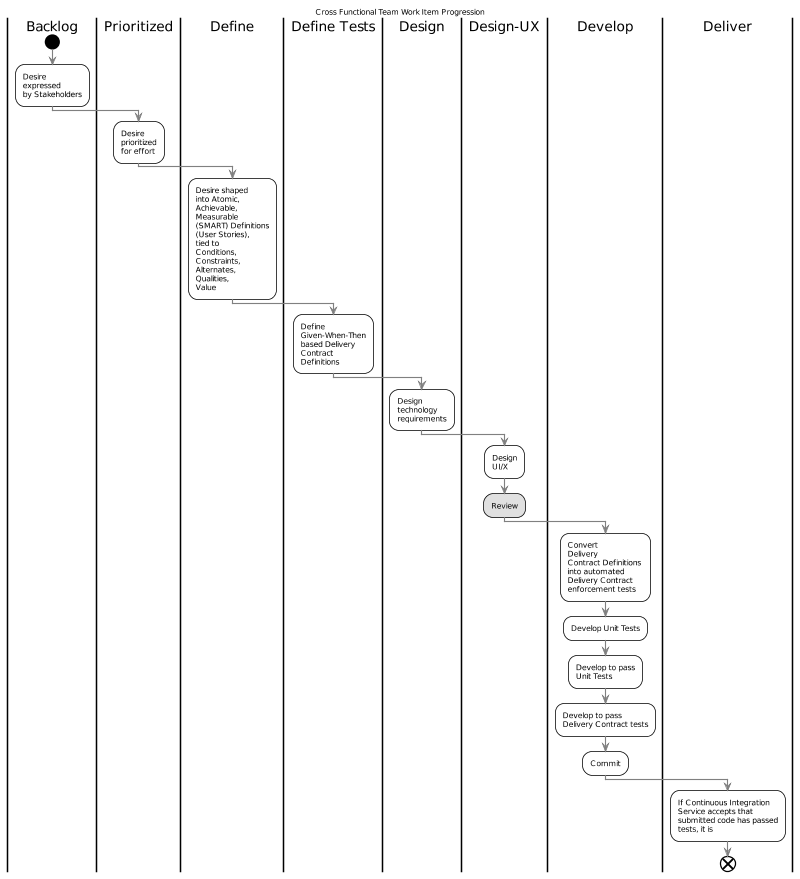


User representatives, business Stakeholders can add Desires as unformed requests, which are subsequently refined into structure Definitions, and progressed through Design, Development, and Delivered stages.

### WIP Limits

WIP Limits are specifically called out as unacceptable controls on effort allocation.

WIP limits are discussed as one method to ensure resources don’t become effort bottlenecks – but there are simpler, methods to address the same issue. Effective alternate solutions include communication regarding the issue, adding more resources to help and/or resource re-allocation if required.



## Source Code Management

The ALM’s Version Control Service provides a web based service to persist versioned copies of submitted code, using Git.

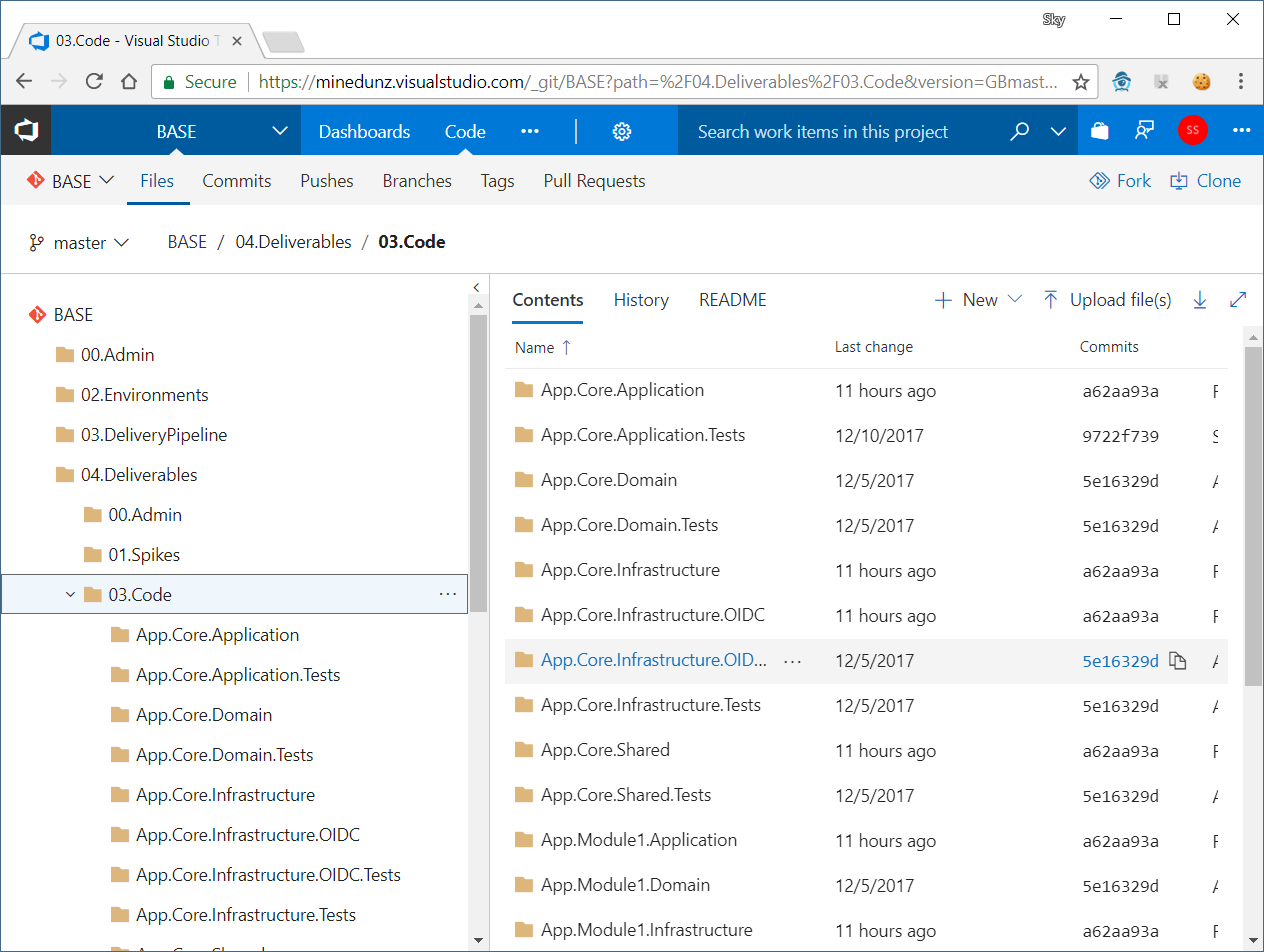
### Multiple Version Control Repositories

The Project contains multiple Git Repositories: one for BASE’s Core, and one for each extension Module.

The Git based Repository of Core code is clonable by Project Team development members from the following Url:

* <https://minedunz.visualstudio.com/DefaultCollection/_git/BASE>

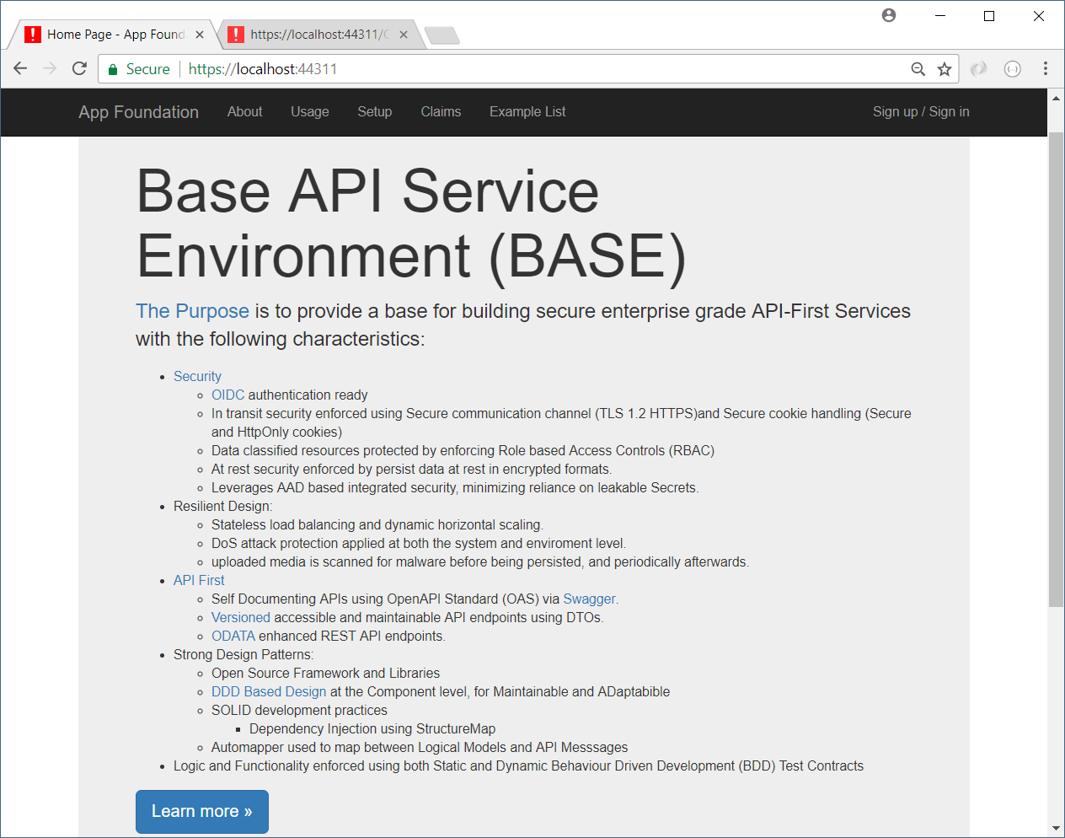
The solution’s \*.sln is available under the *04.Deliverables* folder:



Note:  
Build And Deployment Scripts (see below) are in parent folders.

## First Build

Code can be downloaded, and with a preprovisioned key vault (refer to dev team lead) built:



### Trunk Branch Protection

Key characteristics of the Version Control Serivce are:

* Depends on developers using Git Branches per Development Unit (ie User Story).
* Invokes automated Delivery Contracts (“Tests”) upon newly committed Code prior to accepting to merge new code with existing code (using Visual Studio Branch Policies).
* Code base includes ARM Templates sufficient to define and build a target environment from scratch.
  + Specifically, no human activity is required to deploy to any target environment, including ST, UAT, and PROD.
* Runs Integration Automation Tests on deployed Test instances.

## ALM Build Management Service

A Default Build Definition is defined using the ALM Suite.

The Build Definition defines a series of Build Steps needed to retrieve the code from the Code Repository Service, build the artefacts, test the artefacts using static code testing services, deploy the artefacts to a sacrificial temporary Build Environment, run dynamic tests on it – and only if all these tests pass state the deployment was a success.

Note:  
This combination of static and dynamic testing is a fundamental dependency of Trunk Branch Protection (see above).

Build Steps include:

* Extract the code from the Version Control Service’s repository
* Build the artefacts
* Run Static Unit Tests.
* Execute the ARM Templates to ensure a sacrificial temporary target environment.

## ALM Deployment Management Service

The objective is to deliver a deployment mechanism that is:

* Completely automated
* Can be used to deploy to a new environment (BT, ST, UAT, etc.) with trivial to no changes to the code base, whether it be the app’s, build or delivery scripts.
* The deployment scripts are considered ‘dumb’, and contain little to no project specific parameters or secrets.
* The build and deployment definitions contain no secrets that can be inadvertently leaked via code commits, or other means.

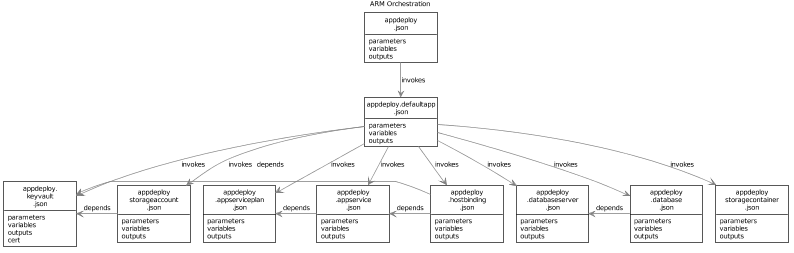
### ARM Templates

Azure Resource Management (ARM) Templates are used to ensure the target environments are available, to which the artefacts will be deployed.

The Build Definition invokes a project specific azuredeploy.json file that in turn invokes other azuredeploy.XXX.json scripts, passing parameters to them, and collecting required outputs.

In this case, the project specific azuredeploy.json sorts out basic parameters (location, preferred price point, name prefix and suffix) then quickly makes the necessary effort to hand off to a generic deployment recipe

Note:  
At present, this project is managing the generic deployment recipe, but it could later be an organisation ESTI managed generic deployment recipe, once ESTI resources are sufficiently skilled to provide such a service.



### ARM Template Parameters, variables and outputs

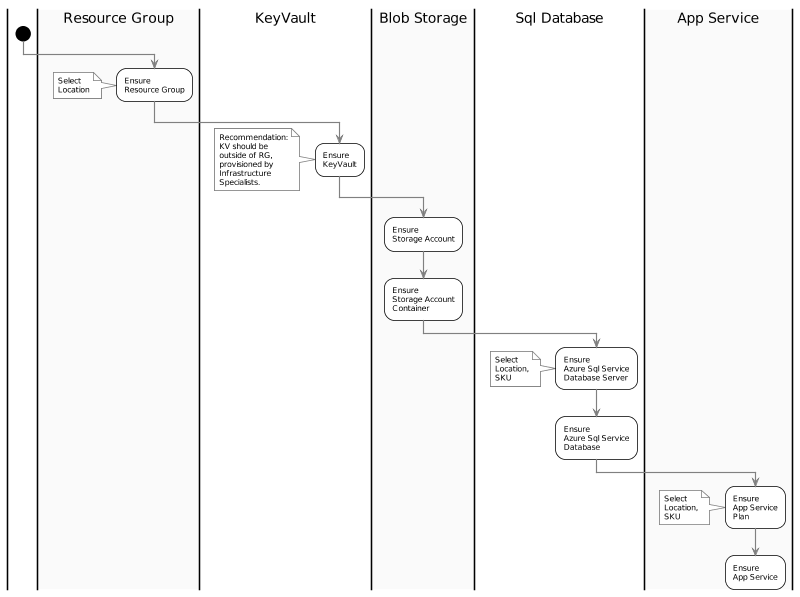
Each script defines a set of parameters, usually provided with default values. The Entry script’s parameters include:

* **DefaultResourceGroupLocation:** the default Location for new Resources.  
  Note that not all Resources can be created in all locals.
* **ScriptBaseUri:** the base path to the location other azuredeploy.xxx.json files can be located.
* **ScriptSAS:** an optional SAS to append to the ScriptBaseUri.
* **Environment, ResourceNamePrefix, ResourecNameSuffix:** variables to build the names of Resources.
* **DefaultDatabaseServer:** there are economic benefits to using a shared Sql Service Database Server for most non-prod servers (and vice versa, there are serious risks associated to using the same server for both prod and non prod).

Note:  
In each script, Parameters can be used to create internal Variables (eg: when building a ResourceName), which can in turn be passed to child azuredeploy.xxx.json scripts as their Parameters. The system is flexible. Albeit…cumbersome to develop the first time around.

### ARM Template Invocation Sequence

The scripts are invoked in the following sequence, ensuring that base resources are created first, before ending up with the AppService resource to which the build artefacts are deployed:



### ARM Design Constraints

Incorrect design of Templates can lead to significant security risks due to over-reliance on Secrets that need to be either added to the ALM’s Build Definition, or worse, within committed code.

Hence specific Design Constraints must be followed.

#### Passing Secrets During Deployment

When you need to pass a secure value (like a password) as a parameter during deployment, you can store that value as a secret in an Azure Key Vault and reference the value in other Resource Manager templates.

#### Deploying Certificates

Deploying certificates stored in Key Vault as part of App Service Certificate offering.

# Development View

This section describes the component architecture and patterns that supports the Development, Maintaining, Enhancing process over the lifetime of the application (not a first Big-Bang delivery), inclusive of the development of static Unit Testing and dynamic Delivery Contract automation.

## Synopsis

* The system’s components architecture follows Domain Driven Design (DDD) principles.
* The system’s code is developed according to SOLID development principles, improving its maintainability by following current development practises.
* To increase maintainability and decrease security risks, the system uses Open Source Nuget-sourced components.
* To increase security by ensuring that Secrecy is not by Obfuscation, the system’s Core logic code is hosted in publicly accessible Git Repositories, after ensuring that Secrets and end user Data are excluded from any code check-ins.

## Constraints

### Development Principles

As a solution deployed for the organisation, the development of this system’s solutions are bound by the Development Principles listed within the Organisation’s [Principles](#_Principles). [DOCR-0002].

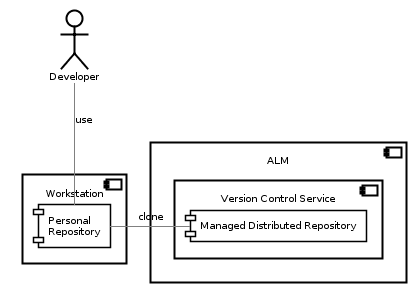
Below are listed additional Development Principles specific to this solution:

* DEVP-0001: Loose Coupling, High Cohesion
* DEVP-0002: Real-time Notification over Delayed Batch Notification  
  where applicable, prefer real-time notification messages between systems of changes, over the use of delayed scheduled updates (ie, no reliance on after hour “fat” ETL operations).
* DEVP-0003: Deltas before Copies  
  where applicable, when querying for changes, prefer retrieving only changed records (eg: seeking records whose LastUpdatedDateTimeUtc Attribute is larger than a specified value).
* DEV-0004: Page versus Book  
  Where applicable, when querying for changes, ensure the data is returned in small batches (eg: a page of 20 records), continuing till no data is returned. This produces a less brittle and more maintainable architecture that often in Near-Time, rather than delayed till after hours.

## Prerequisites

### Use of a Distributed Version Control

The use of a distributed Version control system is a key productivity driver:



### Continuous Delivery Service

The first principle of the Agile Manifesto is "continuous delivery of valuable software".

In order to perform "continuously delivery", a Continuous Integration, Build, Testing and Accreditation and Delivery Service is required.

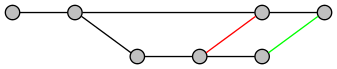
Note:  
The difference between a Continuous Delivery Service and a Continuous Deployment Service is that delivery to target environments with a Continuous Delivery Service is a \*decision\* (by change control, accreditation services, etc.) -- whereas in the second case, every commit to the code base that passes all automated tests is trusted to be deployed to production immediately.   
This organisation -- and this Project -- uses a controlled audited Continuous Delivery based approach.

### Verified Code Branch Integration

The Version Control Service employed to deliver the solution performs \*Continuous Integration\* activities and verifies submitted code feature branches before integrating the code with the protected `master` branch.

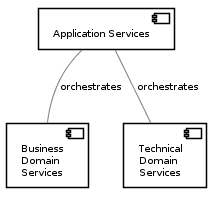


If the Version Control Service rejects the code due to it failing tests -- or a `Pull Request` reviewer (see elsewhere in this View) has manually rejected the submission -- the developer has to fix it and try again before the Version Control Service will allow the submitted feature branch to be integrated with the protected `master` branch:



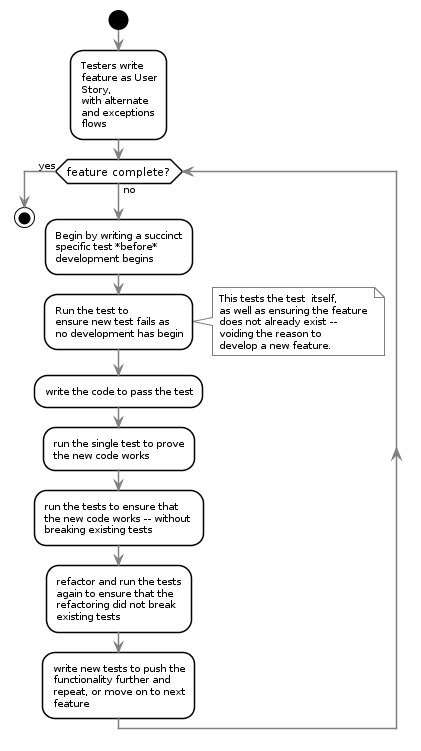
### Domain Driven Design Component Approach

It is a disservice to reduce Domain Driven Design down to only one aspect, but if nothing else, Domain Driven Design based solutions demand that the Business Functional Domain be kept strictly uncoupled from the Technical Functional Domain, and use an Application layer to orchestrate operations between the two:



### Test Driven Development Process

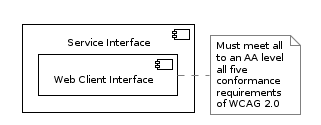
Developers are expected to follow Test Driven Development processes:



## Accessibiliy

API systems do not have interfaces. But systems rely on the APIs providing enough information to meet their Standards obligations.

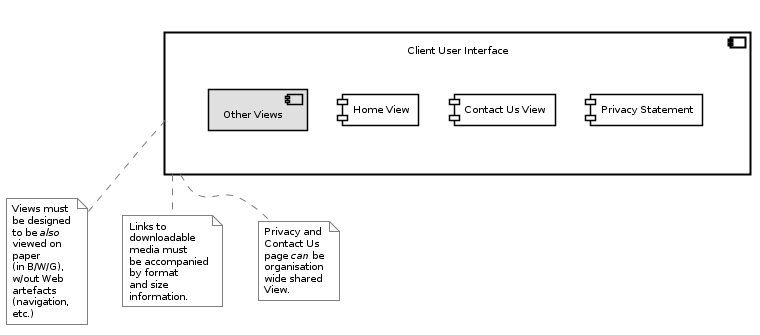
### Acccessibility Obligations



### Usability Obligations

Ensure Modules or the Core has a means to provide the following functionality:

* Privacy Information
* Cookie Usage Information
* Terms and Conditions
* Contact Us page



## Risks

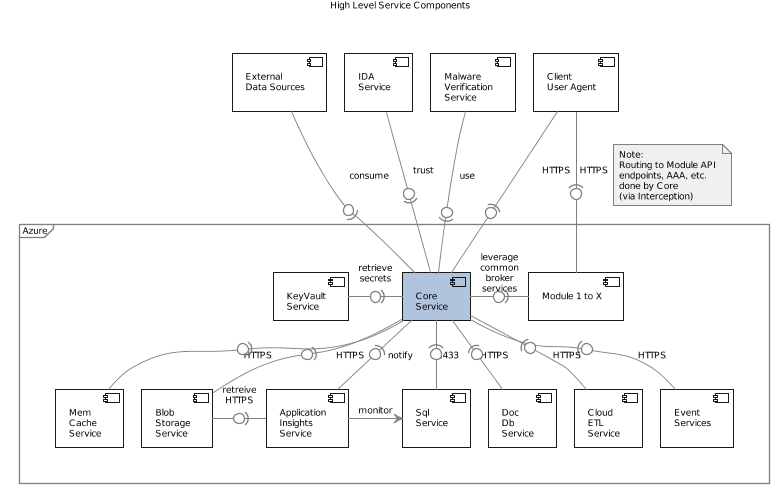
* RISK-0001: Developers and Development Consultants are provided access to production data of any kind. It’s not a matter of if, it’s a matter of when they will inadvertently commit secrets that cannot be extracted from repositories – whether publicly available or not.

Note:   
This risk recognizes that authorised access to Repositories are only illusion of security as any repository can be duplicated and publicly posted at a moments notice. There’s lower risk relying on secure practises than an illusion of security.

* RISK-0002: Reliance on Closed Libraries: closed libraries rely on security by obfuscation.
* RISK-0003: Development and or Maintenance is undertaken by developers unskilled in current SOLID development practises, preferring older, highly-coupled development practise.

## System Components

The solution communicates with several external services:



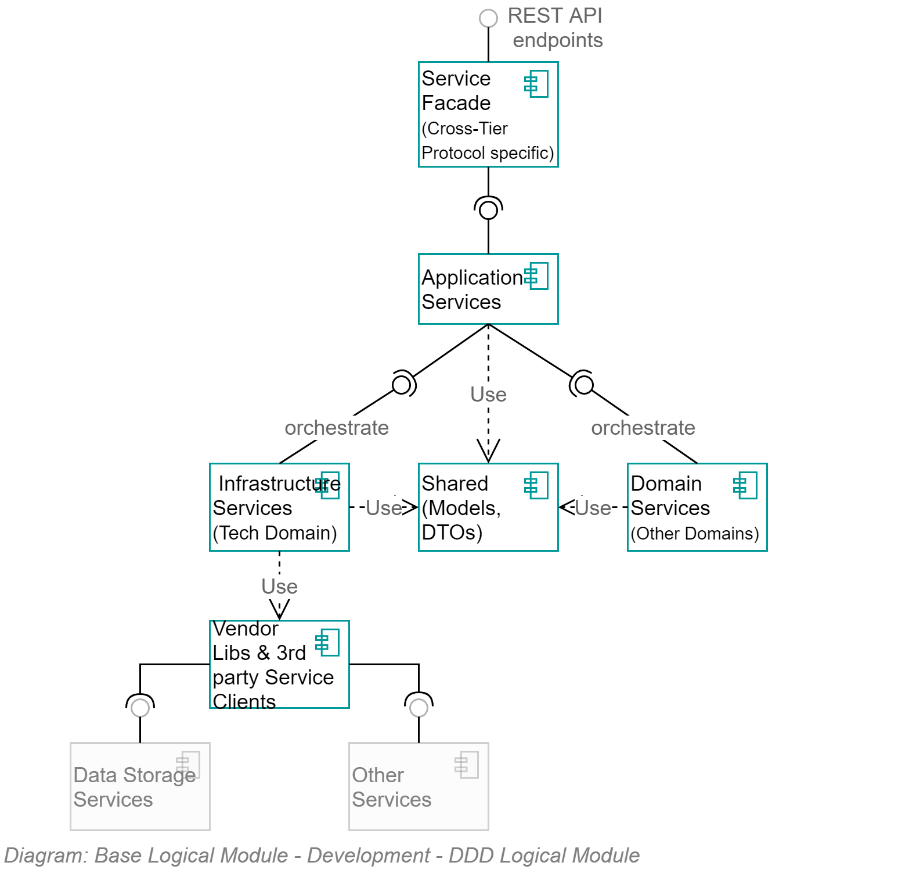
## System Components

Modules are a set of assemblies arranged in the classic DDD formation of the following:

* App.ModuleX.Application
* App. ModuleX.Infrastructure
* App. ModuleX.Shared
* App. ModuleX.Domain

Note:

The Base Module generally has lots happening in its Infrastructure Module and little in its Domain Assembly – whereas other Modules (Business Modules) have generally little to nothing in their Infrastructure Assembly, and more happening in their Business Domain Assembly.



As per Domain Driven Design patterns, Application Service orchestrates calls between the two primary domains: Infrastructure and Business.

The Infrastructure Assembly isolates the rest of the application from dependencies on 3rd party open source and vendor libraries by providing system specific wrappers that provided simplified, app-specific methods, returning to the rest of the system app-Specific messages.

## Service Facade ODATA API Controllers

The Service Façade exposes the following appropriately queryable ODATA REST Controllers:

* ConfigurationStepRecordController
* DataClassificationController
* ExceptionRecordController
* MediaMetadataController
* NOtificationController
* PrincipalController
* RoleController
* SearchController
* SessionController
* SessionOperationController
* SystemDeveloperController
* SystemDocumentationController
* SystemInformationController
* TenantController

Note:  
The apis are appropriately accessible – depending on the controller and data type, information will be made available depending on the Authorisation granted to the current Session’s Principal.

## Core Infrastructure Services

The core infrastructure services offered within Core for use by Core -- and Modules dependent on Core -- follow classic DDD recommendations and include (but not limited to):

* **ContextService:** a service to manage the current operation’s context.
* **ImmutableHostSettingsService:** a service to manage immutable settings of the host device (wrapping access to web.config).
* **DiagnosticsTracingService:** a service to trace rolling diagnostic exceptions.
* **ExceptionManagementService:** a service to manage and record exceptions.
* **LocalizedCachingService:** a service to cache resources closest to use, in the format closest to use, by the user agent’s culture-region code.
* **ObjectMappingService:** service to manage the mapping of application entities to versioned DTO objects, and back again.
* **ValidationService:** a service to manage the object and property validation of objects.
* **AuthorisationService: a service to manage access operations by principals.**
* **BlobStorageService:** provides managed control of private and public Blob Storage.
* **TenantManagementService:** a service to manage Tenants.
* **PrincipalManagementService:** a service to manage System Principal records.
* **PrincipalRoleManagementService:** a Setting to manage relationships between System Principals and System Roles.   
    
  Important:   
  May reley on external services (eg: EOI, Attributes, Relationships) to provide hints as to how to allocate roles – but role allocation is per system (not monolithic/centrally controlled).
* **SessionManagementService:** a service to manage one or more current active sessions per Principal.
* **SessionOperationService:** a service to persist untamperable auditable records of Principal initiated operations within a Session.
* **MutableSettingsService:** a setting to sync across hosts mutable Settings.
* **MediaUploadService:** orchestration service to manage the upload of media.
* **MediaMalwareVerificationService:** validates uploaded media.
* **MediaMetadataService:** develops a metadata object to describe uploaded media (mimetype, extensions, dates, whether contained malware, whether it was persisted, etc.).
* **RepositoryService:** provides managed access to relational storage.
* **UnitOfWorkService:** a service to manage batch persistence across tiers.
* **NotificationService:** a service to manage notifications to and among system users.
* **IUniversalDateTimeService:** a service to return UDT datetimes.
* **ISMTPService:** a service to deliver outward bound email notifications.
* **IDbContectPreCommitService:** a service to intercept database operations to ensure cleanup and logging is handled centrally.
* **IConversionService:** a service to convert values from one type to another.
* **ITelemetryService:** a service to manage the collection and recording of telemetry.

### Service Design

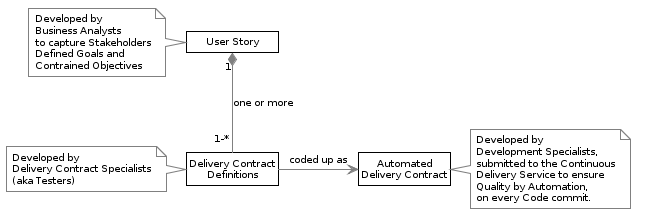
This is a living document: as time permits further technical definitions and design of the above services will be included.

# Testing View

This View summarizes Testing Constraints and Approachs to be employed over the lifetime of the service till decommissioned.

## Synopsis

In order to meet the Delivery Principles of delivering rapidly and iteratively while not sacrificing security, delivery of quality is ensured solely by the automation of defined Delivery Contract Definitions/Scripts.



## ‘Delivery Contracts’ versus ‘Tests’

The choice of the term Delivery Contract versus the more well-known term Test is deliberate.

Testing currently encompasses the concept of Test Script preparation, and (often manual) Test Execution, done by Testers after the Developers have completed their first round of development.   
  
The choice of the term Delivery Contract is deliberate in order to break with this low value approach, emphasising two core concepts:

* Execution of Test Scripts must be Automated, not manually executed, allowing more time for Testers to concentrate on developing Contract Definitions.
* Testing should not be done afterwards, testing what Developers did, but instead Test Contracts should be what Developers aim to do and pass – tested by Automation.

### Driver for term redefinition

“After accelerating other aspects of the delivery pipeline, teams typically find that their testing process is preventing them from achieving the expected benefits of their SDLC acceleration initiative. Testing and the overall quality process remain problematic for several key reasons.  
…  
Iteration length has changed from months to weeks or days with the rising popularity of Agile, DevOps, and Continuous Delivery. Traditional methods of testing, which rely heavily on manual testing and automated GUI tests that require frequent updating, cannot keep pace. At this point, organizations tend to recognize the need to extend their test automation efforts."[[12]](#footnote-12)

## Delivery Contract Sequence

As per Delivery Principles, Automated Delivery Contracts are defined before Development – not after – and are used as development targets.

## Delivery Contract Definitions

Delivery Contract Specialists (often delegated to Testers) are used to convert the Stakeholder defined User Stories into automatable Delivery Contracts Definitions using the industry’s Given [And] When Then (GWT) recommended syntax.

A User Story usually leads to multiple Delivery Contract Definitions being defined.

An example of a Delivery Contract Definition might be as follows:

**Given** a Current User  
**And** User is Authenticated  
**And** User’s Session is valid  
**And** User is in Role SysAdmin  
**When** User requests Tenants  
**Then** the Response Http Code will be 200  
**And** the Response will contain at least 1 item.

Another example of an automatable Delivery Contract Definition might be as follows:

**Given** a Current User  
**And** User is Authenticated  
**And** User’s Session is valid  
**And** User is in Role SysAdmin  
**When** User requests Tenants  
**Then** the Response Http Code will be 200  
**And** the Response will contain no results

## Delivery Contract Development

Delivery Contracts are in turn converted into code by specialised Developers.

The restrained GWT vocabulary, relying only on the terms Given/When/Then/And/Or is easily transposable by a developer to code as logical Parameters/Trigger/Expected Output.

Note:   
It is not the intention of this organisation to turn Testers into Developers.   
Development is a career path that is distinct from test defining that takes an extensive amount of time and significant investment to train resources to produce code to a quality already readily available in the market[[13]](#footnote-13)[[14]](#footnote-14).

Nor is it this organisation’s intention to train testers to automate UI testing via no/low code manual operation recording test process: the results are lower quality than what is available via code based delivery contract development and brittle. The brittleness of manually recorded operations putting unecessary expense for short-term gain.

Ensuring Delivey Contract Definitions are developed by Delivery Contract Specialists – and not Development Specialists -- is a recognition that it takes substantial investment, resources and training to develop valued Delivery Contract Specialists. The skills required of each do not overlap.

Testers, as Delivery Contract Specialists, focused on delivering Delivery Contract Definitions, are essential to Cross Functional Team delivery.

## Delivery Contract Development Scope

A Delivery Contract Definition is not constrained to Functionality.

A Delivery Contract Specialist may create definitions to test Functionality, Access Controls, Throughput, etc.

It is up to to the Development Specialists to find a way to execute the provided Delivery Contract Definitions via tools made available to them, executed by the project’s Automated Delivery Service.

Examples of Delivery Contract Definitions to test various Qualities might include the following examples.

Resileince testing might be defined using a script similar to:

# A test to demonstrate resilience through dynamic horizontal scaling and load balancing  
**Given** a series of Concurrent Requests  
**When** over a Configurable Threshold (for test purposes, 3 is satisfactory to prove the principle)  
**Then** the underlying Infrastructure will ramp up another host.

Responsiveness might be defined as follows:

# Responses should be fast  
**Given** an Authorised User

**And** User is SysAdmin  
**When** requesting the list of the current Session’s Operations,

**Then** the response should be complete in less than 200ms.

Access Control Modeling might be defined using a script similar to:

# A test to ensure that certain resources are RBAC controlled  
**Given** an Anonymous User  
**When** the User tries to query the Principals list,

**Then** the HTTP Code will be 200  
**And** the valid JSON response will contain no entries.

Penetration Testing might be defined as:

# Ensure that HTTPS redirection is in effect, while not forgetting to address Cookie leakage  
**Given** a request made over HTTP  
**When** a request is made over HTTP  
**Then** ensure the service received no System-issued Cookies  
**And** ensure the request isredirected to the same url, but over HTTPS.

Within reason, Developers can automate most Delivery Contract Definitions. It is acceptable for developers to work with Delivery Contract Specialists to rearrange test definitions if it will make the automation easier and more performant to accomplish.

A common example of a need to work together to optimise Delivery Contract Definitions is when Functionality DCD’s that were originally useful in the development phass can now be rolled up into overall tests. It takes an inordinate amount of time for the test tools to setup/test/pulldown each test – whereas doing complete flow tests through a series of pages is faster,and proves the same.

**Given** an Authenticated User  
**And** an Empty Cart  
**And** starts on the Home Page  
**And** Searches for Book Id AAA  
**When** the User Adds the first book found  
**And** Changes the Count to 5 books  
**And** Searches for Book Id BBB  
**And** the User Adds the first book found  
**And** Changes the Count of the second book to 3  
**And** then Removes 4 of the first book  
**And** Removes 2 of the second book  
**And** Enters the Rebate code Of QBD  
**And** set his preferred Currency to NXD  
**And** is not a non-profit  
**Then** the Amount of books in his cart is 2  
**And** the Sum of the books will be XXX  
**And** the Amount Saved will be displayed as YYY  
**And** the Tax Ammount will be displayed as ZZZ

A failed test ensures that errors are highlighted and -- more importantly – new code that is breaking previously working functionality is rejected from the code base.

## Delivery Contract Execution

Delivery Contracts are to be executed solely by automation, via the automated delivery pipeline.

Manual Testing of any kind, including Smoke Testing

Note:  
Manual Exploration, in order to better understand the application and develop Delivery Contracts is absolutely acceptable. But not Testing.

See [Deployment View](#_Delivery_View).

## Security Concerns

No matter the classification of the data (ie, even though it is UNCLASSIFIED), it is an important security concern that this organisation’s testing habits are improved.

For this reason automated testing of logic and integration must not use production data.

This addresses the possibility that if any scope modifications in the future lead to an inclusion of data classified as more sensitive than UNCLASSIFIED, that testing does not become a vector for data leakage.

This also addresses the possibility that a system developed on BASE which includes sensitive information about colleagues (eg, includes Business Unit or Testers Employment Contracts) is not tested with actual data that may cause issue amongst testers.

## Privacy Concerns

Addressing the above Security concern of performing automated tests using only non-production data also addresses privacy concerns during testing.

Note:  
Ensuring that testing with test data only also address a common issue that occurs when testers are using copies of production data: their manual tests may contain screen grabs of a system’s errors – which may contain sensitive or private impacting data – remain in the testers issue tracking software until remediated.

# Operational View

This section describes how the Solution’s Systems will be monitored, administered, supported and operated while running in a Production and/or Training environment.

## Synopsis

Operations are greatly reduced as Deployment is automated to a single button push. Deployment is to a PaaS environment which has an SLA of 99.99% or better. Being a PaaS environment, Infrastructure and Application Support Specialists do not need to worry about underlying OS patching, Execution Environment Patching, etc. Monitoring and Alerting is done via Application Insights, which by default provides insights to throughput and bottlenecks, and can be extended via custom dashboards to integrate more fully with the APIs provided by Core.

## Operating System and Execution Environment Patching

The system is deployed to a PaaS environment. Microsoft takes care of

* OS Patching
* Security Update Patching
* Execution Frameworks Maintenance:
  + .NET (multiple versions)
* Execution Environment Maintenance:
  + Web Server (IIS and/or the more performance .NET Core replacement).
* Realtime and Periodic Malware Scanning of IO to Host hard-drives.  
  Note: Scanning is not done Blob Storage – hence the use of a Malware Service to scan uploaded media before persisting to Blob Storage.

## Domain Name System (DNS) Records

The deployed systems, require Domain Namin System (DNS) entries in order for stakeholders to access the system.

The system may be associated to multiple DNS records, and associated Certificates.

This procurement of DNS and Cert names will be handled as BAU processes.

The information will be configured within the automated deployment pipeline as required.

## System Configuration

System configuration is a Automated Deployment process.

No manual operations are to be performed during or after any deployment.  
  
Refer back to the Project’s Visual Studio Office Project build and deployment process described in the [Delivery View](#_Delivery_View_1).

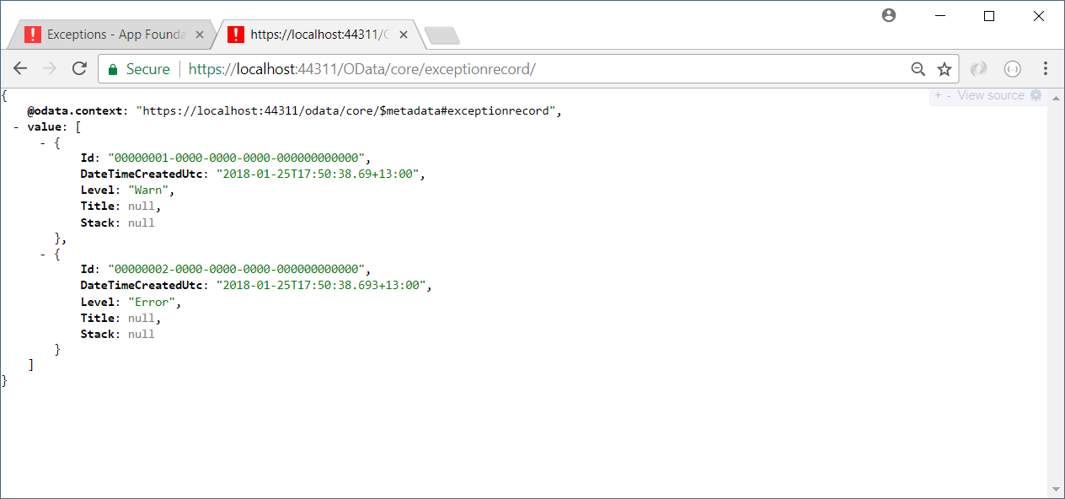
## System Diagnostics Tracing

Diagnostics tTracing is able to be routed to Application Insights such that it can be – when activated – viewed in the Azure Portal.

Error and Warning Messages are routed to the system database, and made available via the APIs such that they can be queried using PowerShell, Postman, curl or similar:

Invoke-RestMethod https://target.tld/api/odata/exceptionrecord/

A simple query from an appropriately authorised Principal (refer to the [Access Control Model](#_Access_Control_Model)) for errors that have occurred lately – without any ODATA based filtering or shaping applied – would be similar to the following:



## System Monitoring

The system is deployed with Application Insights monitoring, which can be configured to collect information on queries received, their average duration, and the average number of errors that occurred.

This type of monitoring requires blob storage to save information for a duration of time.

Note that this collection of data should be temporary, to trace bugs. It should not be on continuously, due to reasons that include the following:

1. It negatively affects performance
2. The storage requirements for the collected metrics do add costs that should be avoided if their advantage is not defined against an actual problem that needs solving.

## System Alerting

TODO: Describe what thresholds trigger alerting, and to whom.

Application Insights can be configured to respond to configurable metrics and alert a support specialist.

Of specific importance are expiration alerts, including:

* TLS certificates backing the HTTPS communication between Clients and Service
* Single Access Signature (SAS) Tokens between the Service and Azure Storage Containers.

[TBD](#_Living_Document): An interface could be developed to connect alerting to Application Insight.

## Operation Auditing

Long term auditing – not to be confused with Diagnostic Tracing – is built into the Solution’s services, recording Operations associated to the Session Principal.

As expressed under the [Functionality Qualities View](#_Functional_Quality_View), Session Operation APIs can be accessed using any of a set of well-known tools, including, but limited to:

* PowerShell (see Invoke-RestMethod)
* Postman, or other well-known REST client tool
* The inbuilt Swagger interface (for simple queries)
* An interface could be developed to connect the view to Application Insight.

## Data Archiving

No external archiving capabilities are required: archiving is achieved using Logical flags within the Operational database.

Records are archived in one of the system’s data stores, including:

* The system’s sequential database.
* The system’s storage accounts:
  + Publicly addressable media storage account.
  + Private diagnostics storage account

## Support Reports

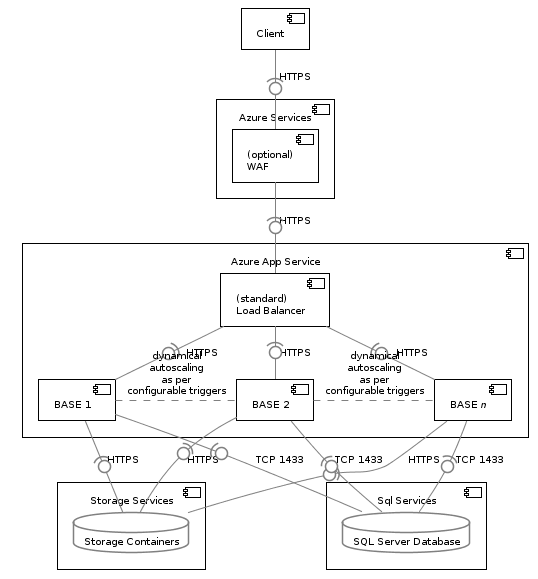
[TBD](#_Living_Document): during Workshops.

## High Availability

The Azure’s Service comes with an SLA of 99.9% per billing month (4.32 minutes/month). That said, last year they achieved a 99.999% availability rating (25 seconds/month).

## High Availability – Sustained and Burst Load Handling

The system’s deployment pipeline deploys to Azure App Services, configured for dynamic horizontal scaling to handle sustained and burst loads.



Note:  
The maximum number of hosts available to a system is dependent on the SKU chosen, but the Standard SKU is largely capable of handling the requirements of the whole sector, if even just a minimum level of development standards are used to address common caching and database issues.

## Failover – Addressing Device Failure

Load Balancing is a default Service provided by Microsoft to Services hosted on its App Service PaaS platform.

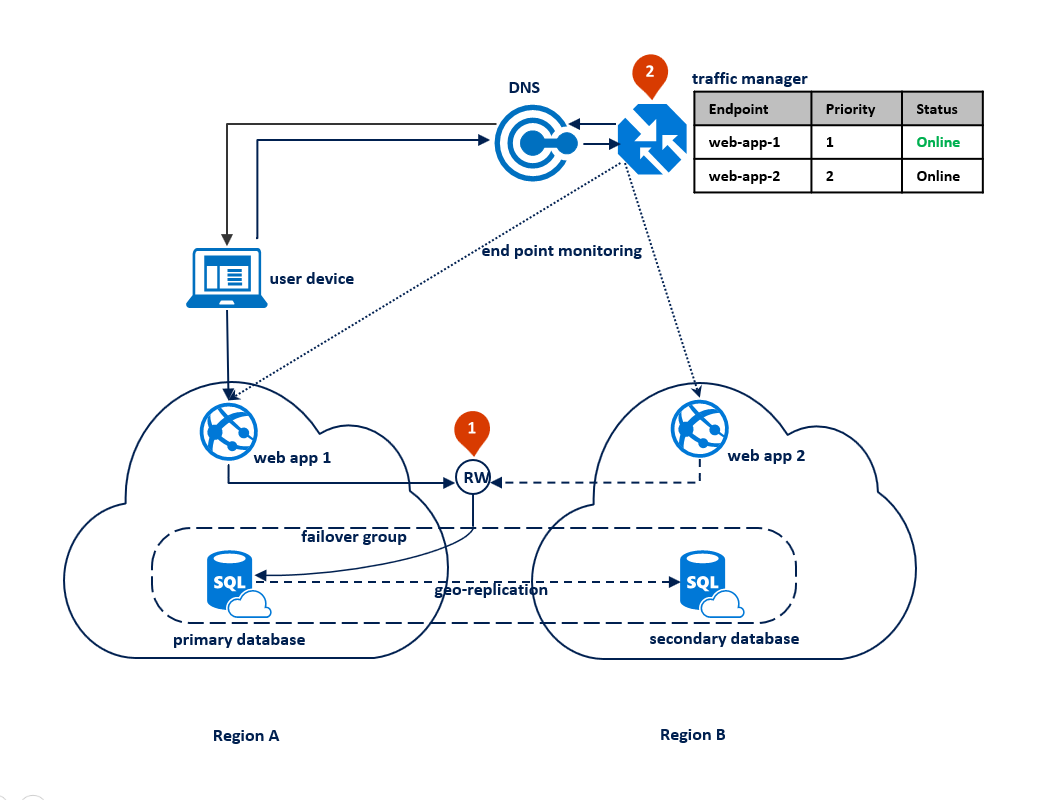
It automatically handles the failure of devices, spinning up replacement devices as needed, and redirecting traffic to the new devices.

The Service is managed by Microsoft, without our involvement, controlled by the default SLAs offered by Microsoft Azure.

## FailOver – Active Geo Replication

Active geo-replication enables you to configure up to four readable secondary databases in the same or different data center locations (regions).

Secondary databases are available for querying and for failover if there is a data center outage or the inability to connect to the primary database.



GeoReplication requires a higher SKU, and therefore monthly charge applied to the project operational expenses.

GeoReplication is not applied by default.

More complex configurations can be developed[[15]](#footnote-15) – but not at this time.

## Fail Over - Testing

Azure’s fail over functionality is a service controlled by SLAs.

There are currently no plans in place to take an Azure data center offline to test the fail over functionality.

## Backup

The deployment pipeline configures the target environment to backup the operational database, and keep these databases for 30+days.

## Restore

Restoration of databases can be manually triggered from the Azure Portal if and as required, from any backup taken within the previous 30 days.

## Backup Restoration Testing

Testing of the Restoration process will be performed regularly, manually, within a non-production environment (eg: ST), and in Production – at larger intervals -- in a manner that will limit impact to end users.

## Disaster Recovery

The deployment pipeline configures the target environment to be geo replicated to an alternate near-shore Azure data warehouse.

[TBD](#_Living_Document): Governance and Process on deciding and activating recovery.

## Business Continuity Plan

TODO: Describe how Businss Services can be continue to be offered even if the system is unavailable. Consider:

\* Developing Browse/List, Read, Edit, Delete (BREAD) View paper alternatives, including:   
 \* Key Reports (TBD) as exportable on a regular basis (TBD) as PDFs that are persisted on non-prod infrastructure, in order that they can be accessed, printed and used until functionality and access to the system is restored.  
 \* Printable based Entry Forms to record New/Udpate/Delete operations (often the same form can be used)

\* Printed Instructions to define the process of using paper based forms.  
BCP-0000

In the eventuality that the System is unavailable the following lists key aspects of the contingency plans.

[TBD](#_Living_Document): Although Core does not contain Business concerns, it is possible that Core could be extended to provide a common way of exporting Extension Modules Data as PDF based reports.

## Disengagement Plan

TODO: It is highly valuable to provide a means to export all of the system’s data on a regular basis to preferably an open protocol format, so that disengagement at the end is no more than a common experience.

* END-0010: Export data to a neutral format -- preferably an industry accepted open standard – that can be used as the basis for a data migration to another system.
* END-0020: Deactivate Service Accounts.  
  It is a security risk to allow service accounts to be left active after the removal of systems.

## Terms and Conditions

Systems are governed by both explicit and implicit Terms and Conditions.

This system, hosted upon infrastructure provided by this organisation has implicit Terms and Conditions for members of the Organisation agreed to upon signing in to the lan network:

“Unauthorised and inappropriate use of the Ministry Technology facilities is against the Ministry of Education Code of Conduct and Information Technology Use Policy and can lead to disciplinary and/or legal action. By clicking Sign On you acknowledge you have read and understood the above.  
  
Please refer to the ICT Use Policy”

[TBD](#_Living_Document): The above covers only Organisation Users. ESL is not yet available for Users beyond the organisation.

## Operational Manual

An Operations Manual is not expected to be required.

That said, if and as needed the following organisation resources will be used to communicate information as needed:

* Visual Studio Online Project Space’s Wiki:  
  https://minedunz.visualstudio.com/DefaultCollection/BASE/\_wiki
* Confluence  
  [TBD](#_Living_Document): Consider developing a Page Area.

## Operational Support

Users of the system can expect to be supported as standard Organisation support channels.

## Operational Management

The deployment pipeline is tasked with configuring the environment, including configuration of DNS, tokens, backups, high availability, replication.

It is the Infrastructure Support Specialists job to leave target environments alone, relying on Alerts to ensure certificates and keys remain valid, along with the validity of service subscriptions.

## Actions

* TASK-xxxx: Develop a process to automate the alerting of Expiring Subscriptions
* TASK-xxxx: Develop a process to automate the alerting of Expiring Certs
* TASK-xxxx: Develop a process to automate the alerting of Expiring SAS Tokens
* TASK-xxxx: Determine Governance and Process on deciding and activating recovery.
* TASK-xxxx: Determine whether a Confluence page area may or may not provide additional value.
* TASK-xxxx: Determine process to activate Business Continuity in a common way.
* TASK-xxxx: Define and pottentially design how Core can provide any Business Continuity assistance.
* TASK-xxxx: Determine whether Support Reports would be of value to discuss operations with Module owners. Or whether widgets would do the same.

# Appendices

## Glossary

* **[SAD] View:** an industry term for a Section of a SAD used to describe a complex model from a specific audience’s Viewpoint (refer to ISO/IEC/IEEE 42010:2011).
* **ISO/IEC/IEEE 42010:2011:** an authoritative convention of describing complex models from the viewpoint of specific audiences.
* **RaW: Rosanski & Woods:** authors of the industry respected Software Systems Architecture book, describing an example approach to apply ISO/IEC/IEEE 42010:2011 for describing software systems.
* **UML: Unified Modeling Language: a restrained visual modelling language for describing software systems.**
* **Archimate: a restrained visual modelling languages to map drivers to stakeholder desires to objectives to requirements to business services to automated services and processes.**
* **Quality: a positive or negative trait of a service.**
* **ISO-25010:**  the current authoritative standards-based list of positive qualities which stakeholders consider valuable when assessing a sevice. ISO-25010 supersedes the earlier ISO-9126 standard, and earlier-still FURPS acronym, allowing one to leverage a lot of experienced independent thought on the matter of valuable qualities, and allows for a common agreed terminology which removes ambiguity as to what is actually being asked for, or being delivered. The top level expected Qualities are: Security, Functionality, Usability, Compatibility, Reliability, Performance, Maintainability, Portability, Supportabilty.
* **Assessment: a review of an Organisation, Service or System’s** Strengths **and/or** Weaknesses**.**
* **GOST:** an acronym for Goals, Objectives, Strategies, Tactics. Strategies define alignment to Goals, and Tactics align to Objectives.
* **Drivers:** Internal or external Threats **or** Opportunities**.**
* **Goals: unconstrained desires to address Drivers.**
* **Constraint:** a Resource (time/monetary/skillsets assets), Regulatory (Legal), Agreements (Alliances, [Principles](#_Principles)) or Physical limit imposed on pursuing valued opportunities or goals.
* **Objectives:** constrained Measurable (see SMART) Goals.
* **SMART:** Specific, Measurable, Achieveable, Relevant, Time-Bound (Objectives).
* **Requirement:** an Objective statement categorised as per ISO-2119, traceable to ISO-25010 Qualities.
* **ISO-2119:** an authoritative requirement categorisation sysem which defines requirements as either an Obligation (MUST), Recommendation (SHOULD), Permission (MAY), Prohibition (MUST NOT).
* **Functional Requirements:** Requirements for a specific Quality of a Service, usually limited to only Business Functionality (to the exclusion of functionality desired by other stakeholders, such as Security, Support, Maintenance stakeholders).
* **Non-Functional Requirements:** a low value legacy term for an adhoc selection of Qualities that exclude (Business) Functional Requirements. The term is to be considered Obsolete.
* **Service:** a manual or automated service to render assistance to other Persons (commonly in IT via a Service Agent).
* **Service Façade:** an outward representation of the Service. Preferably public facing, appropriately accessible, audited, public facing, access to versioned resources. See API.
* **Service Agent:** a Service-specific optimised agent for a Service. Preferably a SPA/PWA.
* **[Interface] Views:** a User Interface element to elicit a single Action. Commonly built as poor value Web Pages (see MVC).
* **SPA:** Single Page Application. A low value (ie: misrepresentative) industry name for a high value web service client development pattern (see MVC).
* **PWA:** Progressive Web Application: an incremental improvement to the SPA development pattern.
* **MVC:** Model View Controller pattern. A high value development pattern, unless Views are developed on the Server, where there are no Person’s to actually view it. See SPA.
* **Business Service:** a service, that may in turn rely on on or more automated services, commonly referred to a solution.
* **API:** Application Programming Interface. A misleading term, as modern APIs do not provide programmability in any shape or form, and only provide Browse, Read, Add, Edit, Delete (BREAD) access to permitted service resources by appropriate users. Consider prefering the term Service Façade.
* **REST:** Representational State Transfer protocol.
* **Hackable:** a colloquial term used by developers to mean “Predictable/Understandable” when referring to Nagivation or REST. Used in that contect, it does not imply a diminished security profile.
* **Person:** a Natural Person (Individual) or Legal Person (Organisation).
* **PersOrg:** a vernacular equivalent term for Person.
* **Organisation:** a set of Persons working together to meet common Objectives in a Principled manner. Sizes vary from adhoc groups, to distinct Business Groups to Enterprises.
* **Enterprise:** a set of internal Organisations.
* **Sector:** a set of internal and external Organisations.
* **Subject:** any entity that requests access to an object[[16]](#footnote-16).
* **Principal:** A uniquely identifiable Body, Service, or System Subject that accesses a System[[17]](#footnote-17).
* **User:** A subset of Principal.
* **Actor:** a commonly used term for Principal.
* **Attribute:** a Key/Value pair describing a property of an entity.
* **Claim:** an Attribute distributed by a trusted authoritative source. Often cryptographically signed against tamperability. Often distributed by an Identity [Token] Provider (IdP) when issuing Identity Tokens but can also simply be retrieved independently by calling an API endpoint (see OIDC user profile endpoint).
* **Credential:** a secured undisclosed Attribute (eg: Password) managed by an Identity [Token] Provider (IdP) to verify the Identity is who they say they are, before issuing an Identity Token.
* **System Permission:** a logical system permission associated to an Identity Token.
* **System Role:** a logical grouping of System Permissions, associated to an Identity Token.
* **Identity:** a unique set of claims representing one of many identities belonging to a Person.   
  Note: a Person has multiple logical Identities that may be staggered in serial or overlapped in parrallel at any one time (at home, preschool, student, sports group memberships, social group identity, training, tertiary, soldier, employment, 3rd age learning, retirement, friends, aliases, private identities).
* **Identity Token:** a short-lived digital token issued by an Identity [Token] Provider (IDP) to a client, and then transmitted with messages to a Resource Provider/Server, uniquely identifing the Subject sending the message.
* **Identity Service:** a logical service that can manage one or more of the concepts of:
  + manages the relationships of multiple identities to a common Person (eg: soldier, student, social identities, all belonging to a single Person), and providing access to the same resources to one or more of the linked identities.
  + Manage the relationships between identities (parent to children, teacher to children, principle to teachers, etc.)
* **CRUD:** a commonly used industry term for data asset operations (Create, Read, Update, Delete) that is logically inappropriate for meeting common target objectives. Refer to CRUS.
* **CRUS:** an acronym for data asset operations (Create, Read, Update, State) that addresses Logical Deletes as State changes (providing a means to Undo, Merge and optionally Archive if and as required).
* **ORM (Object Relation Mapping):** an abstraction mechanism for managing resources within data stores, without having to resort to data specific vendor control languages (eg: SQL).
* **AD:** Active Directory is a directory service that Microsoft has developed for windows domain networks. The AD domain controller authenticates and authorises all users and computers in a widows domain type networks.
* **AAD:** Azure Active Directory is a cloud hosted directory service for authentication and authorisation of users to systems associated to the organisation’s tenant,
* **ADFS:** Active Directory Federated Services (ADFS) is a software component developed by Microsoft to provide users with single sign-on access to applications hosted across organisational boundaries.
* **WAD: Windows Active Directory:** An alternative term for Active Directory, to differentiate it from AAD.
* **OIDC:** OpenID Connect (OIDC) is an interoperable, simple, secure, flexible, web and mobile friendly authentication protocol extension to the Oauth (an authorisation protocol for authorising the user, and obtaining access tokens to access web APIs and protected resources), superceding the earlier OpenID 2.0 authentication protocol, to provide to client apps basic user profile info serialized using JWT as an id\_token.
* **OD:** Operational Database.
* **RD:** Reporting Database, distinct from the Operational Database to not impact the Operational Database Server’s performance.
* **DW:** Data Warehouse.
* **EDW:** Enterprise Data Warehouse.
* **IAM:** Identity and Access Management.
* **HA:** High Availability. An Architecture implementation pattern to ensure the platform is available for the most time (all the time being the target).
* **SFTP:** Secure File Transport Protocol – Data Transportation
* **SMTP:** Simple Mail Transport Protocol
* **SSIS:** SQL Server Integration Services – Data Transformation
* **Azure:** A cloud computing platform and infrastructure, created by Microsoft for building and managing applications and services through a global network of Microsoft managed data centres. <https://en.wikipedia.org/wiki/Microsoft_Azure>
* **AoG:** All-of-Government contracts establish a single supply agreement between the Crown and approved suppliers for the supply of selected common goods and services purchased across government
* **TRW:** Technical Working Group.
* **ESDA:** Education Sector’s Design Authority
* **Business role - Functional role – Permissions (Access):** Users are assigned business role(s), business role(s) are made up of functional role(s), and functional role are made up of permissions/access.
* **Contract:** An agreement with specific terms between two or more persons or entities in which there is a promise to do something in return for a valuable benefit known as consideration.
* **OnBoarding:** A process to register permanent and non-permanent staff detailing reasons for procurement and permissions for access to Organisation systems and processes
* **Offboarding:** A process which completes the life cycle of permanent or non-permanent staff and disables agreements and permission access to Organisation systems and processes
* **SOW:** Statement of Work is a Schedule attached to a Panel, Project or Syndicated Agreement that suppliers have signed up to undertake separate pieces of work under these agreements
* **FMIS:** Finance Management Information System. The application used by the Finance Service team to enable payments after a contract has been approved and signed off.
* **SaaS:** Software as a Service is a software distribution model in which applications are hosted by a vendor or service provider and made available to a customer over a network.  
  This model is where SaaS providers move the task of managing software and its deployment to third party services. This model tends to reduce the cost of software ownership by removing the need for technical staff to manage, install, and upgrade software, as well as reduce the cost of licensing software.   
  Customisation of SaaS solutions is not recommended but can in some cases be part of the agreement.
* **SPA:** A single-page application (SPA) is a web application or web site that interacts with the user by dynamically rewriting the current page rather than loading entire new pages from a server. ... Interaction with the single page application often involves dynamic communication with the web server behind the scenes. [[18]](#footnote-18)
* **Ajax: Asyncronous Javascript and XML**: is a set of Web development techniques using many Web technologies on the client side to create asynchronous Web applications. With Ajax, Web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behavior of the existing page. By decoupling the data interchange layer from the presentation layer, Ajax allows for Web pages, and by extension Web applications (and SPAs), to change content dynamically without the need to reload the entire page.  
   In practice, modern implementations commonly substitute JSON for XML due to the advantages of JSON being native to JavaScript.
* **Javascript: often abbreviated as *JS*, a high-level, dynamic, weakly typed, prototype-based, multi-paradigm, and interpreted programming language. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production. It is used to make interactive service clients (eg: SPAs).**
* **JSON: JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language.**
* **JWE (JSON Web Encryption):** **represents encrypted content using JSON-based data structures.**
* **JSON Web Signature:** an [IETF](https://en.wikipedia.org/wiki/Internet_Engineering_Task_Force) proposed standard for [signing](https://en.wikipedia.org/wiki/Digital_signature) arbitrary [JSON](https://en.wikipedia.org/wiki/JSON)
* **JWT: JSON Web Token:** a [JSON](https://en.wikipedia.org/wiki/JSON)-based [open standard](https://en.wikipedia.org/wiki/Open_standard) ([RFC 7519](https://tools.ietf.org/html/rfc7519)) for creating [access tokens](https://en.wikipedia.org/wiki/Access_token) that assert some number of claims.
* **SAS:** **Shared Access Signature:** signed access tokens to scope public access to Azure storage accounts, containers, or items within. With a SAS, you can grant clients access to resources in your storage account, without sharing your account keys. This is the key point of using shared access signatures in your applications--a SAS is a secure way to share your storage resources without compromising your account keys.
* **ToC:** **Terms and Conditions**: persons may have to agree to before using a service.
* **ESTI:** Education Sector Technology Infrastructure Program.
* **CAPEX:** Capital Expenditure
* **CSF:** Critical Success Factor
* **DBC:** Detailed Business Case
* **e-asTTle:** Electronic Assessment Tool for Teaching and Learning
* **ECE:** Early Childhood Education
* **EDK:** Evidence Data and Knowledge
* **ELI:** Early Learning Initiative
* **ESDS:** Education System Digital Strategy
* **GCIO:** Government Chief Information Officer
* **ICT:** Information and Communications Technology
* **iEd:** Integrated Education Data
* **IFL: Insights for Learning**
* **ILM:** Investment Logic Mapping
* **IOs:** Investment Objectives
* **IQA:** Independent Quality Assurance
* **LMS:** Learning Management System
* **MOE:** Ministry of Education
* **NCEA:** National Certificate of Educational Achievement
* **NGO:** Non-governmental Organisation
* **NPC:** Net Present Cost
* **NZQA:** New Zealand Qualifications Authority
* **OPEX:** Operating Expenditure
* **PaCT:** Progress and Consistency Tool
* **PATs:** Progressive Achievement Tests
* **PBC:** Programme Business Case
* **PQM:** Price Quality Methodology
* **QRA:** Quantitative Risk Assessment
* **SIF:** System Interoperability Framework
* **SIS:** School Information System
* **SISI :** Student Information Sharing Initiative
* **SMS:** Student Management System
* **SRT:** Student Record Transfer
* **TEC:** Tertiary Education Commission

## Summary of OIDC

### Description

OpenID Connect (OIDC) is an interoperable, simple, secure, flexible, web and mobile friendly authentication protocol extension to the Oauth (an authorisation protocol for authorising the user, and obtaining access tokens to access web APIs and protected resources), superceding the earlier OpenID 2.0 authentication protocol, to provide to client apps basic user profile info serialized using JWT as an id\_token.

The uptake of OIDC is proportional to its simplicity and focus on fewer, current, technologies -- compared to older protocols such as SAML and WS-Fed.

The OIDC specification is a combination of

* Key protocols (HTTPS, JWT, OAuth),
* components to connect together (RP, OP, IdP),
* via specific service endpoints (token endpoint, UserInfo endpoint)
* in order to exchange security tokens (id\_token).
* Via one of several component interaction flows (Authorisation Code Grant Type Flow, Implicit Grant Type Flow, etc) which are chosen based on the types of components involved (eg: server generated stateless web site, client generated stateless apps, mobile app).

Note:

The acronyms and terms used above are described in more detail below.

### Terms

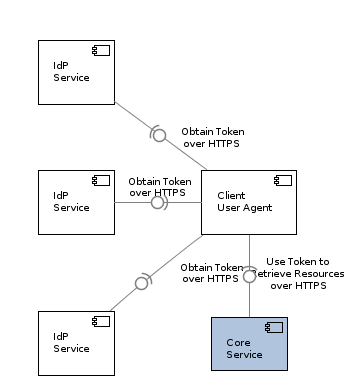
Key Terms specific to understanding OIDC include:

* **Identities and Claims and Scopes:**
  + **Identity:** a set of Claims belonging to a Person, containing a unique set of Credentials.
  + **Claims:** A claim is any piece of data about a user, such as names, locations, privilege levels, group associations, device types and preferences. The unique set of claims for a user of a specific application is presented as that user’s identity within the application.
  + **Scope:** when an Relying Party (RP) asks a OpenId Provider (OP) to return claims, it can do so individually (email, etc.) or by scope, which define a set of claims. Standard scopes are: email, phone, profile, address
* **Protocols:**
  + **JWT:** Java Web Token: – format of the id\_token returned to Relying Party.
* **OIDC:**
  + **Components:**
    - **Relying Party (RP)**: same as an OAuth Client – the component which relies on the OP.   
      Equivalent to an OAuth Client. Equivalent to a SAML Service Provider (SP).
    - **OpenId Provider (OP)**: Signs the ID Token that is passed back to the RP in a JWT.
    - **Identity Provider (IdP):** the Identity [Token] Provider Service.
  + **OIDC Endpoints:** 
    - **Api:**
      * **token endpoint:** the OP API endpoint the RP sends the Authentication Code to, in order to get back the OAuth accesstoken, and an idtoken.  
        Note: the token endpoint can accept other types of Assertions, including JWT and SAML assertions.
      * **UserInfo endpoint:** an OP API endpoint where the access\_token can be used.
    - **Views:**
      * **authorisation endpoint:** the OP endpoint the client is sent to in order for the user to authenticate (eg: a webpage form in which to enter their credentials).
  + **OIDC Tokens:**
    - **Id Token:** an OIDC implementation of the underlying OAuth specification’s authentication token, in this case developed using JWT, containing Claims “about the Authentication of an End-User by an OAuth Authorization Server, and potentially other requested Claims”.
* **OAuth:**
  + **Principals:**
    - **Resource owner:** the Person accessing the resource server.
  + **Components:**
    - **Client:** most often, the user agent (browser)
    - **resource server:** the service managing the resource that is accessible or owned by the resource owner.
    - **authorisation server:**
  + **Endpoints:**
  + **Tokens:**
    - **authorisation code:**
    - **authorisation token:**
    - **Authentication token:**

### OIDC Integration

OIDC Provides a standard protocol provided by most current Identity Token Provider services, whether enterprise (enterprise Azure AD or Google App Identity Domains) or social (eg: Google, Microsoft Account, LinkedIn, Atlassian, GitHub, etc.) .

Due to a common agreed protocol, the development effort required is more or less identical amongst the various options.

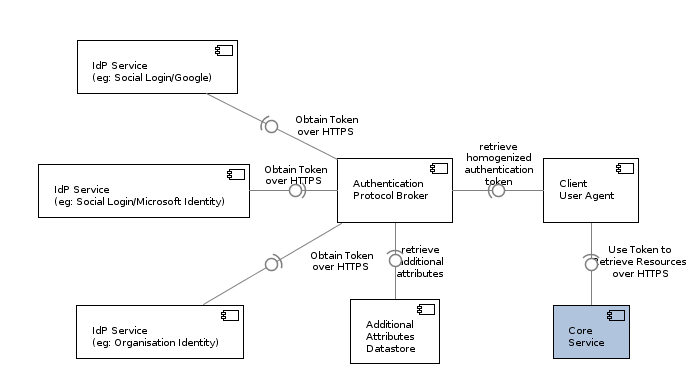


### OIDC Brokered Integration

Although it is more or less the same work, the tokens provided by different sources vary in the claims they contain (one might have Business Role, others might not, and instead have Cell Phone number – most agree on sharing a DisplayName and Email).

The above sometimes leads to the consideration of using a broker to homogenize the claims before returning them within the authentication token (or subsequently via a user profile request).

When Brokered with a component similar to Azure B2C, the topology is as follows:



Note:  
Homogenization comes at a cost (eg: transaction fees) and therefore should probably be avoided if there is not a defined need for homogenized outside of an System (callbacks by the system at initial sign in can achieve the same result).

### Flows

A key consideration when designing authentication (again, not authorisation) using OIDC is the choice of interaction sequence flow. This in turn depends on the use case, which is largely dependent on the type of client components involved.

### Authorisation Code Grant Flow

#### Use Case:

* For authenticating websites developed using the legacy pattern of developing views on the server (eg: ASP.MVC).  
  Not optimal for SPA scenarios.

#### ACiD Analysis:

**Advantages:**

* Supports Refresh Tokens – unlike the **implicit grant flow**.

**Considerations:**

* client\_secret is transmitted securely directly between resource server and authorisation server, and the client never sees it. Note this is not necessarily advantageous or disadvantageous.

**Disadvantages:**

* Uses Cookies – which requires CORS to reconnect with the *authorisation server* (VERIFY).
* If performed from a SPA, the CORS needs to be enabled on on the resource server in order to allow AJAX call to pick up resources (since it is from a different server than the client the SPA belongs to).
* But other services that the SPA contacts, would need to have CORS enabled…so not sure if this is a big deal.

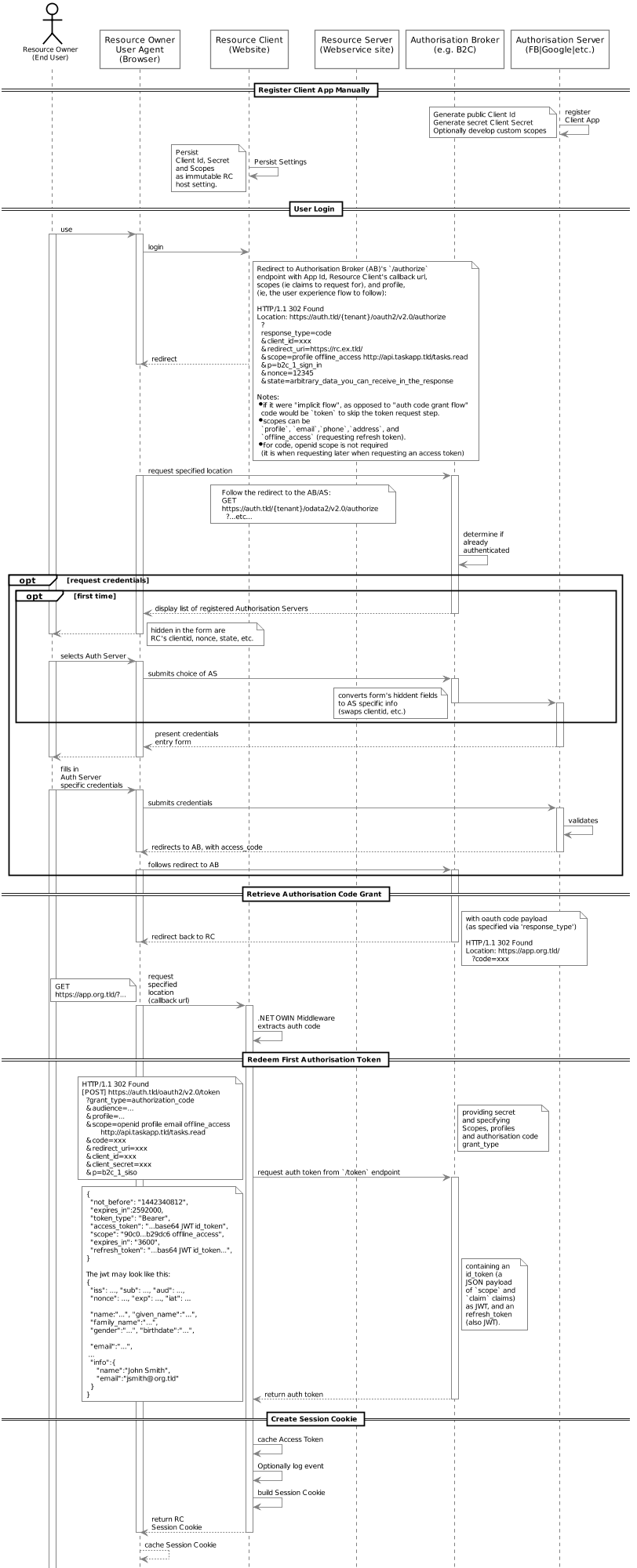
#### Summary

The flow involves two tokens (authorisation code and authorisation token) and endpoints (the SP’s auth code delivery callback*, and* theAS's auth token api endpoint), and goes as follows:

* user clicks login via Facebook|Google|AAD button on website, and is redirected to the appropriate authorisation server.
* upon completion of verification of the resource owner's credential, the authorisation server provides an authorisation code to the resource server (website), via a client (browser) redirection to a website callback endpoint (an httpHandler in asp.mvc).
* the callback processor does a back-channel POST (not entirely sure why POST and not GET) synchronous call over HTTPS back to the authorisation server, providing the short lived authorisation code. An access token (verify: towhat. the AS, or RS?) is returned.

note:   
the reason for the two steps is that it ensures the authorisation token, containing claims, is not passed through the client

Sequence Diagrams

[](http://skysigal.com/lib/plugins/plantuml/img.php?width=0&height=0&title=PlantUML%20Graph&align=&version=2011-07-16&md5=2aba309d807f547090345303059c2c45)

1. https://stackoverflow.com/a/5025140 [↑](#footnote-ref-1)
2. https://en.wikipedia.org/wiki/Principal\_(computer\_security) [↑](#footnote-ref-2)
3. To be ratified with and by ESDA Stakeholders. [↑](#footnote-ref-3)
4. https://www.ict.govt.nz/guidance-and-resources/open-government/new-zealand-data-and-information-management-principles/ [↑](#footnote-ref-4)
5. https://www.gcsb.govt.nz/publications/the-nz-information-security-manual/ [↑](#footnote-ref-5)
6. http://www.odata.org/ [↑](#footnote-ref-6)
7. http://openid.net/connect/ [↑](#footnote-ref-7)
8. To be ratified with and by ESDA Stakeholders. [↑](#footnote-ref-8)
9. https://azure.microsoft.com/en-gb/blog/microsoft-azure-reaches-new-industry-leading-cloud-compliance-milestones/ [↑](#footnote-ref-9)
10. ESTI-provided [↑](#footnote-ref-10)
11. https://azure.microsoft.com/en-us/services/data-factory/ [↑](#footnote-ref-11)
12. https://en.wikipedia.org/wiki/Continuous\_testing|Wikipedia: Continuous Testing [↑](#footnote-ref-12)
13. <https://www.quora.com/How-long-would-it-take-an-average-person-with-no-programming-background-to-become-a-full-stack-developer-I-want-to-be-proficient-in-both-the-front-end-and-back-end-technologies-I-have-a-very-basic-understanding-of-HTML-and-Im-learning-CSS> [↑](#footnote-ref-13)
14. https://www.reddit.com/r/cscareerquestions/comments/r3jy2/how\_long\_does\_it\_take\_to\_become\_a\_senior\_developer/c42npbv/ [↑](#footnote-ref-14)
15. https://docs.microsoft.com/en-us/azure/sql-database/sql-database-designing-cloud-solutions-for-disaster-recovery [↑](#footnote-ref-15)
16. https://stackoverflow.com/a/5025140 [↑](#footnote-ref-16)
17. https://en.wikipedia.org/wiki/Principal\_(computer\_security) [↑](#footnote-ref-17)
18. https://en.wikipedia.org/wiki/Single-page\_application [↑](#footnote-ref-18)