## Appendix C – Quality Objectives and Requirements

## Target Quality Objectives

The following table catalogues key quality objectives referenced by the subsequent Quality Requirements section.

| Category | Subcategory | Value |
| --- | --- | --- |
| Minimum Users: | Estimated number of potential users  800,000 learners, 55,000 teachers | 855,000 |
| Minimum new Users/year | New (and therefore leaving) students | 62,000 |
| Minimum Users requiring IdP changes | Changing school/year | 110,000 |
| Minimum User Groups: | Sector groups and organisations:  2500 schools,  5000 ECE providers,  4000 home schools,  600 Tertiary organisations  850  playgroups  Unknown number of sports, learning, etc. groups (1000): | 14,000 |
| Minimum Media Resources | Estimated number of individually managed media resources:  5,000 documents  17 drafts spread over  3 published versioned | 100,000 |
| Minimum Storage (Gb): | Assuming  100,000 Resources, x 10 versions, x 0.25Mb  2.5Mb of media x 100,000 Resources | Max 375 Gb |
| Minimum Concurrent Capacity: | Method 1:  10% of Max Users online,  10% being concurrent over 25 seconds | 8,000/4=2000 users |
| Method 2:  121,000 users/year,  10% being concurrent over 25 seconds:  NCEA 1: 140,000  NECA 2: 88,0000  NCEA 3: 33,000 | 12,000/4 = 3000 users |
| Method 3:  55,000 teachers, 10% being concurrent over 25 seconds: | 5,500/4 = 1375 users |
| Throughput: | Surge Average Concurrent Requests/sec | 3000 requests per sec. |
| Response Time: | Peak  Concurrent Requests/sec | 12000 requests per sec. |
|  | Average *Completed* Response Time | <500ms |
| Max *Completed* Response Time |  | 2 secs |
| Maximum Tolerable Downtime (MTD) | Assuming part of a lunch break. | 20 minutes |
| Minimum Downtime per fortnight | 99.9% | 20 minutes per fortnight. |
| Disaster Recovery Times (mins) | Recovery Point Objective (RPO) | 15 mins |
|  | Recovery Point in Time (RPT) | 60 mins |
| Vendor Support Response Times (mins) | P1 |  |
| P2 |  |
| P3 |  |
| Integration Systems | Hosting | Organisation Cloud Accounts |
| Identity | (I)Commodity Cloud based Identity Providers (IdPs)  (I)Cloud based Organisation specific IdPs  (I)Organisation specific Identity Provider (ESL) |
| Reporting | (I/O)Reporting Database |
| Data | (O) Data warehouse |
| Other | (I/O) Te Rito (via APIs)  (O) Student Management System (SMS)  (I/O) Learning Management System (LMS) |
| Integration Methods: | Protocols |  |
| Identity | OIDC, OAuth, SAML |
| Education specific | cmi5, xAPI, SCORM |
| Standards | JSON, REST, OpenAPI, ODATA, |
| Tech Preferences: | Host Environment | Cloud. Azure preferably |
| Database/Store | SQL Server preferably |
| OS | Windows preferably |
| Service Framework | .NET Core preferably |
| Service Language Type | Compiled |
| Service Language Framework | .NET Core |
| Service Language | C# |
| Service Client Framework | Vue.js preferably |
| Service Client Language | Typescript preferably |
| Accepted Hash Algorithms | SHA-3 |  |
| Accepted Encryption Protocols | TLS 3+ |  |
| Max Length of Certificates | 90 days |  |
| Data Classification | IN CONFIDENCE | The system is intended to hold UNCLASSIFIED resources, but the volume of users demands that the information be treated as IN CONFIDENCE. |
| Systems that may be replaced |  | CoL Tool |
| Backups | Full | Every 24 hours |
| Incremental | Every 15 minutes |
| Duration kept | 31 days |

## Quality Requirements

The quality requirements outline the required or preferred capabilities of the solution by priority (Must, Should and Could).

1. **Must** - the respondent must propose a solution that, as a minimum standard, meets the requirements of the core functionality identified as “Must”.
2. **Should** - functional requirements that have been identified as “Should” are highly desirable functionality that should be included or workarounds may be available for this functionality.
3. **Could** - represents requirements that are ‘a nice to have’ (time and resources permitting), but not considered necessary functionality for the final solution.

### Statement Details

When responding to the Quality Requirements keep in mind the Details column as a target state and answer addressing what can be achieved, and what cannot.

### DELIVERY

| ID | Statement | Rationale | Details |
| --- | --- | --- | --- |
|  |  |  |  |
| Start | The solution’s configuration and/or custom code MUST be delivered by automation to the production environment for end users to access within a short delay defined in the *Target Quality Objectives* table. | It is important to not let a long start up and delayed Go Live Date mentality become common place. Results, regularly reconsidered, must be the norm in order to decrease the risk of non-delivery. | * The solution’s required system security is dependent on components and security rating of the contents within the system. * When a custom project starts, when there is no editing capability, and therefore no data, the focus MUST be on developing an automated secure deployment process -- even if the system is just a bare bone “Hello World” product. * Once a rudimentary automated pipeline is developed, it, like any code, is iteratively developed further, hardening it, making it more resilient, maintainable, flexible, configurable, secure – and by doing so, making the system it deploys more secure, modifiable, installable. |
| Cadence of Delivery to Prod | The solution MUST be made available to end user stakeholders at a high cadence, as defined in the *Target Quality Objectives* table (approximately every 2 weeks) | Deliver early and often to require reliance on automation to be able to manage compilation, deployment and testing in the short time frame.  Automation solves DR and testing IP retention over the project's lifespan. requirements  High cadence permits regular testing of assumptions as to what is most important to stakeholders, putting results in front of them, allowing them redirecting subsequent effort based on new or simply refined requirements. | * The solution is delivered at a cadence matching changes to requirements (e.g.: every fortnight) using a constantly and iteratively developed automated pipeline.   Note: in many ways the pipeline is the actual deliverable to the sponsor, while the system it in turn delivers is the deliverable to business users. |
| Tests Definitions First | The solution’s work items MUST be completed with test scripts before any development on the work item begins. | Developers must know what they are aiming for, rather than developers and testers both interpreting the intent of Business Analysts separately, putting in the effort to do their individual tasks, only to find out at the end they disagreed as to the BA’s intent. | * Work items are developed by converting user desires into structured definitions, as either requirements or use cases. * Definitions are only defined as ready for pickup for development when the use case is completed with a set of one or more target test definitions. * Developers first convert the test definitions into automated pipeline tests. Since there is no code, the automated pipeline fails subsequent check-ins. * Developers then develop the code to pass the automated test definitions. |
| Automated Deployment | The solution MUST include automated deployment functionality to meet Operations Specialists operability expectations. | In order to support Testability, Availability quality objectives, the solution must be able to be deployed rapidly after quality testing has been ascertained, also rapidly. | * The solution uses an automated pipeline to: * Compile code * Run unit tests * Package Artefacts into deployable packages * Optionally create the target environment using Infrastructure as Code instructions * Deploy the package to a Build Test environment * Optionally   + run Data restoration (if PROD),   + develop change notification * or   + Run Integration tests,   + Determine whether to accept & merge the check-in code |
| Requirement Volatility | The solution’s functional and quality requirements MUST be able to evolve by mutual consent. | As the cone of uncertainty inherent with the beginning of projects decreases, it is natural that requirements need adjustment. The process of coming to an agreement should be rapid and straightforward. | * Requirements and work items are managed using an online backlog and work item tracking service. * An automated process is used to track changes (by whom and when). |
| Application Lifecycle Management Suite | The solution SHOULD use am integrated Application Lifecycle Management (ALM) suite to integrate Work Item Management, Configuration and Customisation code repositories, Workflow Management, Integration and Deployment services, testing services. | It’s difficult enough to integrate a system that is to be delivered to prod.  The long-term maintainability of the service should not also depend on having to keep integrating and testing the tools used to manage the discovery, definition, design, development, deployment, defect fixing. | * The solution is developed using this organisation default Microsoft DevOps ALM Suite account. |
| Expertise Ratio | The vendor MUST ensure that at least 1/3 of the staff provided for development and ongoing maintenance and improvements are senior developers rated by a 3rd party as being technically proficient in the technologies and tasks being committed to. | In most cases, the effort and time required to perform a task by a certified senior is less than a junior or intermediate developer, or even another senior who is not proficient in the technologies used in a system. | * If the product is developed using .NET Core, the developers should be certified as Microsoft Certified Software Developers (MCSD), etc. * If certification is not available, expertise should be determined in a way acceptable to both parties. * If the rate of progress is inconsistent or diminishes, this organisation can request and obtain different developers. * This organisation can request and retain specific developers. |
| Security Training | All supplied production team members MUST be trained to work in a secure manner prior to beginning to work on the project. | Insecure solutions must not be deployed. | * Topics covered include:   + Agreement as to the extent of the data that requires consideration:     - Code Repositories,     - Production Data,     - Production Data Backups, Configuration settings,     - project Work Items   + Limiting access to Production Data:     - to only service accounts     - Using test data only to develop logic   + Data in Transit security:     - There is no distinction between internal and external components – they are all at risk and require encryption and or other controls to protect communication in between them.   + Data at Rest security:     - Always use specialised secure storage for confidential configuration credentials.     - Always ensuring that confidential information remains within a system and not passed around as messages (in other words, limiting recording of confidential information in emails or attachments and passing them around between developers, etc.). |
| Quality Training | The resources made available for a project MUST be trained to ensure the quality of the solution’s logic and data remain consistently high and incrementally improves over time. | Garbage in, garbage out.  The cost of correcting outweighs the cost of doing things right the first time. | * The solution’s developers are trained on:   + Code quality:     - code repository management via branching     - check-in review processes     - Component design     - OO Development     - Coding standards   + Data quality:     - Integrating systems via API – and avoiding SQL based ETL -- to ensure validation occurs on any migration data. * The solution’s testers are trained on:   + developing test scenarios as acceptance criteria of Work Items (User Stories, etc.)   + expecting automated tests being developed for them by developers, before they get on with developing code for other stakeholders. |

### SECURITY (ISO-25010/Security)

Security is the practice of maintaining the confidentiality, privacy, integrity and accountability of data changes by controlling authorised access, use and disclosure, while preventing unauthorised use, disruption, modification or destruction.

ISO-25010 recommends considering the following qualities and their descriptions when considering a solution:

* Confidentiality is the degree to which the solution ensures data is accessible only by those authorised to do so.
* Integrity defines the degree to which the solution prevents unauthorised access, modification of systems and the information they manage.
* Non-Repudiation and Accountability are associated in that non-repudiation defines the degree to which the solution can prove that actions have been taken, and accountability is being able to associate the non-repudiatable (audited) activity to a specific user.
* Authenticity defines the degree to which the identity of a user can be claimed.
* Availability – which is a prerequisite for Security is treated separately, under Reliability.

Note:  
Security and Privacy risk assessments, and matching Statement of Applicability listing required controls for the solution will be conducted during the solution’s design phase.   That process will further augment the security requirements listed below.

| ID | Statement | Rationale | Details |
| --- | --- | --- | --- |
| Integrity/Encryption in Transit | The solution MUST protect user-submitted information in transit between devices. | Information MUST be stored and transmitted as per the GCSB New Zealand Information Security Manual (NZISM) v2.4 (or later versions) controls to meet the requirements specified for data with a security classification as per the *Target Quality Objectives* table. | * The solution’s channel encryption is achieved using TLS based HTTP (HTTPS) communication between all internal and external components. * The solution encrypts any other channel (e.g., database connections use the *secure* flag). * Message Encrypting is appropriate if channel encryption is not achievable. * Digital signatures may be relevant on a case by case basis when messages pass through tiers which remove or replace channel encryption (e.g., corporate firewalls). |
| Integrity/Encryption at Rest | The solution MUST protect user-generated information at rest. | Information MUST be stored and transmitted as per the GCSB New Zealand Information Security Manual (NZISM) v2.4 (or later versions) controls to meet the requirements specified for data with a security rating as per the *Target Quality Objectives* table. | * Databases are encrypted. * Blob storage is on encrypted storage. * Diagnostic trace files on encrypted storage.   Note:  encryption at rest depends on the data information security classification specified in the Target Quality Objectives table, so in case the Data Classification rises during development, the database chosen SHOULD be able to be encrypted at database, table, or column granularity if needed. |
| Integrity/Physical Security Controls | Any location where Production data is persisted MUST adhere to ISO-27001 level 2+ constraints on physical access. | Systems can be secured, but users should not be able to directly access, clone or remove physical or virtual hard-drives which contain production data or backups of production data, especially if the access is not audited. | * The solution’s environments are hosted within an ISO-27001, Level2+ based data facility, which limits, monitors and audits physical and virtual access to devices. * The solution is hosted within an accredited cloud provider. * Physical and virtual media devices containing production data are not removed from their environment, and when disposed of are sanitised first then disposed of securely. |
| Integrity/External PII Management Services | The solution SHOULD persist Personally Identifiable Information (PII) in an external specialised and hardened storage service. | The less PII information maintained in a solution, the less attack surface a solution has. | * The solution sources PII data from an external PII service. * PII data cloned to the system's user profile is limited in scope. * Cloned PII data is persisted in an external data store separate from the system's primary database.   Note:  A key reason PII information is persisted separately is to minimise risk of data leakage when data is copied to a reporting database and/or data warehouse. |
| Integrity/External Identity Provider Services | The solution MUST use external Identity Provider Services to persist User Credentials. | The less secrets maintained in a solution, the less attack surface has. | * The solution is capable of using current protocols to connect to external identity providers (OIDC, OAuth, SAML). * The solution is capable of authenticating both UI and API security principles (users). |
| Integrity/External Identity Provider Services/ESL | The solution MUST be capable of using standard protocols to connect to the sector's Education Sector Logon (ESL) service. | This organisation's default identity service is Education Sector Logon (ESL). | * Some Identity Protocols are not standardized. The solution is capable of being customised to meet an IdP’s small variations to standard protocols (version, validation, etc.). |
| Integrity/Clear Text Credentials | The solution MUST not transmit clear text credentials. | Even if the channel and message within are encrypted, credentials SHOULD not be transmitted in clear text. | * Where technically feasible, the solution is integrated with external services without transmitting clear text credentials. |
| Integrity/Encrypted at Rest/Sensitive Configuration Parameters | The solution MUST encrypt any persisted sensitive configuration information. | If configuration secrets cannot be persisted elsewhere than in the system configuration files, databases or similar, they must deter tampering by being rendered cost prohibitive to read or modify.  Note:  Such variables **must** be registered on the project's risk register. | * Configuration credentials are persisted in secure key storage services. * Access to production key storage services is restricted to the automated deployment pipeline agent’s service account. * Configuration credentials and settings which cannot be sourced from a separate secure storage service are encrypted in the system's configuration files. * Credentials that are persisted in the system's config file, but cannot be encrypted, are recorded in the project's risk registry. |
| Integrity/Encrypted at Rest/Sensitive Configuration Parameters/Setting | The solution MUST use deployment automation to automatically source configuration credentials from confidential storage services and inject them into config files. | Code repositories are not secure as they can be copied, duplicated for use anywhere. Credentials in source code repositories are compromised credentials as code repositories are not secure. | * Configuration credentials are not sourced from source control, and instead are sourced from secure storage services. * The automated deployment agent injects the retrieved credentials into the deployed system's configuration files. * The configuration credentials are encrypted within the config file. |
| Integrity/Source Code | The solution’s source code repository MUST be protected from committing any security credentials or environment specific information. | Source code is used for production environments, therefore must be protected from becoming a means by which a nefarious person can discover a way to bypass a solution's security controls. | * Code Branch Policies are effective against checking in bugs, but not effective against checking in confidential credentials. * Project members with write access to code repositories must be trained as to the importance of not committing credentials and reporting when such an error has happened. |
| Integrity/Source Code/Contamination | If credentials and/or environment specific properties are checked in, steps must be taken to:   1. remove the credentials from the code repository and 2. rotate the credentials so the information is no longer a potential risk. | Environments, whether production or non-production environments, must be protected.  Publicly accessible source code must not become a means of discovering means to bypass a solution's security controls.  Note:  Until the source code repository is cleansed the incident must be registered on the project's risk register. |  |
| Integrity/Certificate Lifespans | The solution MUST limit the duration of certificates to short lifespans, as per the *Target Quality Objectives* table. | The shorter the duration certificates are used, the less time bad actors have to crack the certificates. | * The solution includes automated mechanisms to regularly commission and deploy new certificates, at every deployment. * The solution ensures that deployments are done frequently (approximately every fortnight), regardless of new features having been developed or not, to ensure certs are rotated often (no more than 90 days). |
| Integrity/Service Accounts | The solution’s components MUST run under individual service accounts, either developed by automation or by the organisation's operation specialists. | Use of a single service account, creates a single point of failure that if compromised provides access to multiple components. | * Where technically feasible, service accounts are automatically developed.   + If deploying to Azure, consider Microsoft Secure Identities (MSI).   + Service accounts are distinct per tier. * Service account credentials are securely deployed by the automated delivery pipeline to service consumers. |
| Integrity/Client Credentials | The solution MUST not deploy credentials to external clients. | Credentials that are deployed and persisted outside a system’s infrastructure are compromised. | * The solution avoids using integration approaches that require the installation of certs on service clients. |
| Integrity/Data Classification | The solution MUST use NZISM recommended security and risk mitigations controls appropriate to the system's NZ government data classification, as defined in the *Target Quality Objectives* table. | A system must apply applicable NZISM recommended safety and risk mitigation controls to safely persist user provided data. | * The solution adheres to NZISM security controls.   References:  <https://protectivesecurity.govt.nz/home/information-security-management-protocol/new-zealand-government-security-classification-system/> |
| Integrity/Authorisation/Permission based | The solution MUST be Permission based, using Roles to collect several Permissions together. | Roles are descriptions of a person's activities, but do not allow fine grain control. | * The solution allows Assigning or Removing Permissions to Roles. * The solution allows assigning Roles to Users. * The solution allows adding or removing Permissions directly to Users. * If a Permission has been assigned multiple times, either directly or via assigned Roles, and one of the assignments is a Removal of the Permission, then the Permission is not given to the target user. |
| Integrity/Authorisation/Obligations based | The solution SHOULD present Responsibilities & Obligations to Users for acceptance, before they are assigned a Roles Permissions. | Roles are a combination of accepted Responsibilities (SHOULDs) & Obligations (MUSTs) in return for Permissions (MAYs).  Adding them to Role definitions makes Invitation to Group or Resource Roles by Notification possible to be automated. | * The solution allows assigning: Responsibilities, * Permissions, * Obligations. * The solution allows assigning Responsivities & Obligations to Roles, which are in turn assigned to Users. * The solution allows fine tuning by adding or removing Obligations directly to Users. |
| Integrity/Authorisation/Group specific Roles | The solution MUST be capable of assigning group specific roles. | A user may belong to different groups, in different capacities/roles (Owner in one, Member in another, etc.). | * The solution provides the capability of inviting the same users to different groups, in different Roles. For example, in one group, a user can be an Administrator, and in another, be a Member or Guest. |
| Integrity/Authorisation/Resource specific Roles | The solution MUST be capable of associating Users in different Roles to Resource. | A resource can be commenced, collaborated on, approved, published and managed by different users. | * The solution associates a user that starts a new resource the owner role. * Other users are invited to work on the resource in one capacity or another (collaborator, etc.) * The solution’s Inheritable Group configuration settings will determine if resources that belong to a group inherit default roles from the group. |
| Integrity/Firewalls & WAFs | The solution MUST be protected by Firewalls and Web Access Firewalls (WAF)s. | Firewalls limit the surface of attach to a single (web) port, and WAFs analyse web traffic for abnormal behaviour and intrusion. | * The solution uses a Firewall to only allow traffic from clients to the service via ports 80 (for HTTP) and 443 (for HTTPS). |
| Integrity/WAFs | The solution MUST be protected by Web Access Firewalls (WAF)s. | WAFs analyse web traffic (over port 80 and 443) for abnormal behaviour and intrusion. | * The solution’s WAF must be configured to allow through media containing malware so that the service’s malware detection service can analyse the upload and return a system error (as opposed to a WAF generated error). |
| Integrity/Log Protection | Solution audit, debug and access logs **MUST** be protected from tampering and loss |  |  |
| Integrity/Key Management | The solution design and operational processes **MUST** demonstrate appropriate encryption key management. |  | * The solution uses a cloud based secure credentials/key storage service to persist keys. * Access to the key storage solution is limited to the deployment pipeline's service account, which uses this access to retrieve credentials and inject them into the deployed system's config file. * Access to the service is monitored and audited.   References: NZISM v2.7 - Section 17.9 Key Management |
| Integrity/Compartmentalised Defence in Depth | The solution MUST remain secure if a component, layer or tier's security is breached. | Multiple layers of security slow the progress of attackers. | * The solution is designed to protect the service tier using firewalls. * The solution is designed to protect the service tier using a WAF configured to monitor and alert subscribed users (operations and security specialists). * The solution’s infrastructure is dynamically horizontally scalable to handle resource DoS attacks. * The solution’s services run on a non-local Service Account that is generated elsewhere. * The solution’s API service facade is protected using authentication and authorisation validation. * The credentials used to access 3rd party storage services such as databases, media storage, etc., are protected within 3rd party secure storage services, or encrypted locally in the host's config file. |
| Integrity/Defence in Depth | The solution MUST protect the information it manages using approaches at least to the level of protection appropriate to the information's highest security classification, as defined in the *Target Quality Objectives* table. |  | * The solution uses NZISM recommended security controls, including:   + Transport encryption   + Firewall protection   + WAF protection   + User challenge and authentication   + Auditing   + Intrusion detection   + Database encryption   + OS specific configuration / hardening   References:  [http://www.gcsb.govt.nz/publications/th e-nz-information-security-manual](http://www.gcsb.govt.nz/publications/the-nz-information-security-manual) |
| Integrity/Least Privilege | The solution MUST allow solution and business administrators to attribute to end users the least amount of privilege needed to perform their tasks. | Avoid the risk of unauthorised activity (accidental or otherwise). |  |
| Integrity/OWASP | The solution’s MUST address and demonstrate application of the Open Web Application Security Project (OWASP)'s recommended mitigation of the latest Top Ten Application Security Risks. | Rely on a reputed source of current knowledge regarding web application vulnerabilities in order to pass a third party validation of the system's security. | * The solution’s service and client systems are hardened against:   + injection   + broken authentication   + sensitive data exposure   + broken access controls   + security misconfiguration   + cross site scripting (XSS)   + cross site request forgery (CSRF)   + insecure deserialisation   + using components with known vulnerabilities   + insufficient logging and monitoring |
| Integrity/security Incidents | The solution **MUST** support Ministry endorsed security incident processes. | Appropriate subscribed stakeholders must be made aware of security incidents in order to ascertain the risk to their business concerns and determine whether it is appropriate to inform end users. | * the solution includes a means of informing appropriate subscribed stakeholders, within stated response delays, keep them informed of progress towards resolution, inform them when resolution is achieved, deliver a summary report. * In the event of a security incident within the solution, processes exist (and are accessible and fit for purpose) to support recovery as applicable.   *Note: Solution design with the successful vendor will include creation of a security incident classification matrix and different classes of security incident will be assigned to the incident resolution requirements defined in the Target Quality Objectives table.* |
| Confidentiality/Production Data | Production Data MUST not be accessible outside of the Production Environment and Production data backup environment. | The less people who have access to information the less chance of data being divulged by accident or intent. | * The solution’s developers develop to meet test-data-based tests and are restricted from direct access to either production data itself, or even full or partial copies of the production data. |
| Confidentiality/Production Data Backup | Production Data Backups MUST be separated from Non-Production Data environments.  Access to Production Data, whether in a PROD data system or a backup, must be restricted to only service accounts. | Backups *are* production data – just from a point in time in the past – and should be protected as completely as production data. | * Backups are encrypted * Access to Backups is limited to Deployment Service accounts * Backup Restoration operations are accessible to Operations Specialists. |
| Non-Repudiation | The solution MUST permanently store Session audit records of all operations in order to correctly ascertain the process by which information was changed. | Discovery of irregular activity may be months or years after the event, or happen sporadically over a long duration, so records should be kept for the duration of the solution.  The solution must audit the activity of authenticated users as well as unauthenticated public users, because irregular activity can start before authentication occurs, and when they sign in, their identity be associated to all their pre-sign in activity as well.  The solution must audit the activity of any background service agents (batch operations, etc.). | * The solution audits operations, how and where the solution keeps these records in order to keep the solution performant, and whether the solution’s audit records are queryable when required. |
| Accountability/Multiple Sessions | The solution MUST allow multiple Sessions per user. | A user may be signing in from a mobile device at the same time as using a desktop or laptop device. | * The solution allows for a user to have multiple distinct sessions.   the solution allows users to open multiple concurrent sessions from multiple devices |
| Accountability/Multiple Digital Identities | The solution SHOULD allow a user to associate multiple digital Identities to their system user's record. | Users belong to multiple organisations (schools, organisations, groups) each of which may have their own Identity Provider Service.  A user should be able to use either identity to sign in.  But also keep them separate in case they prefer to keep their system identities separate. | * the solution uses external IdPs to persist user credentials. * the solution allows integration with multiple social IdPs (Microsoft Accounts, Google Accounts, etc.). * the solution allows easy integration with Azure AD. |
| Accountability | The solution MUST record the system user identity in all session audit records. | Note:  a session's user may be anonymous until authenticated. The act of Authentication does not change the session, it just clarifies who is in operating in the session. | * The solution tracks anonymous users through the process of authentication, ongoing until session termination or timeout. |
| Authenticity | The solution MUST correctly ascertain the identity of system users using external Identity Providers. | The basis of allowing access to users is knowing who they are. | * The solution relies on external IdPs to authenticate users.   Please describe how the solution authenticates system "super users". |
| Sanitisation | The solution MUST sanitise diagnostics and audit records of confidential credential and other secrets. | Persisting credentials in audit or diagnostics records allows Operations and/or Support Specialists access to confidential information they should not have. | * the solution ensures that diagnostics and audit messages are scrubbed of confidential and/or personally identifiable information prior to being persisted. |
| Accreditation | The solution MUST complete and achieve security Certification and Accreditation (C&A) prior to production implementation. | The process incorporates:   * Security Risk Assessment * Security Risk Management Assessment * System Security Plan   The controls defined within the System Security Plan must be mappable to an acceptable detailed control catalogue (such as the latest version of the NZISM). | * the solution has previously achieved C&A from another organisation and what artefacts are available to advance through this process as rapidly and cost effectively as possible. |

### PRIVACY

Privacy is not a defined ISO-25010 concern, but since the last edition, has become a first-class quality.

| ID | Statement | Rationale | Details |
| --- | --- | --- | --- |
| Privacy Act 1993 | The solution MUST adhere to the applicable Acts outlined under the *Target Quality Objectives* Table. | The solution must meet legal obligations before being accreditable to be delivered as a service by this organisation. | Note: Refer to the linked AoG Principles, which outline data and information handling constraints. |
| Minimum PII | The solution MUST collect the minimum amount of PII to meet its functionality expectations. | The more PII information maintained in a system, the more tempting it is, and the more damage can occur from it being accidentally or intentionally leaked. | * The solution actively avoids requesting non-essential PII information. * The solution imports the user’s Given Name (for notifications), Display Name (for rendering views), and identity email credential (for notifications). * The Given and Display Name can be updated by the end user. * The PII is persisted in a secure data store that is separate from the production data. |
| Correction | The solution MUST allow users to self-correct their own information. | Users know themselves best. | * the solution allows users to change or request changes to the PII information imported from an external PII service. |

### USABILITY (ISO-25010/Usability)

Usability is how easy and efficient it is for an end user to correctly, accurately, and safely use the solution. It also ensures constancy with other Ministry applications. It is sometimes called system ergonomics, accessibility and ease of use.

ISO-25010 recommends considering the following qualities and their descriptions when considering a solution:

* *Appropriateness recognisability* is the degree to which users can recognize whether a solution is appropriate for their needs.
* *Learnability* is the degree to which a solution enables the user to learn how to use it with effectiveness, efficiency and in the case of an emergency.
* *Operability* is the degree to which a product is easy to operate, control and appropriate to use.
* *User Error Protection* is the degree to which a solution protects users against making errors.
* *User Interface Aesthetics* is the degree to which a user interface enables pleasing and satisfying interaction for the user.

*Accessibility* is the degree to which a solution can be used by people with the widest range of characteristics and capabilities to achieve a specific goal in a specified context of use.

| ID | Statement | Rationale | Details |
| --- | --- | --- | --- |
| Functionality/Completeness | The solution SHOULD meet the expectations for the functional requirements of *all* solution user stakeholder groups, not just business user stakeholders. |  | * the solution meets the needs of multiple groups, including:   + Operations Specialists in charge of ensuring the correct behaviour of the system (using diagnostics trace logs, monitoring, alerts, health dashboards, etc.)   + Security Specialists in charge of investigating suspicious behaviour (querying auditing records, monitoring, alerts, dashboards, etc.)   + Support Specialists, in charge of providing support to end users, (e.g., identity on-boarding via standard APIs, identity credential resets, identity emulation/delegation to do tasks on their behalf, off-boarding, etc.)   + Business Support Specialists in charge of providing business specific support to end users (e.g., invite users to roles, investigate the state of processes, etc.)   + Business Users (e.g., help documentation, on-boarding, etc.)   + 3rd party service client systems (e.g., via API integration, the protocols used, etc.) |
| Recognisability/Layout | The solution SHOULD use standard UX layout approaches. | Use of standard graphical and behavioural design patterns improves usability and decreases input errors. | * the solution uses a common CSS framework such as Bootstrap, etc. |
| Recognisability/NZ WAG | Any custom UX element of the solution MUST meet NZ Government Web Standards - Web Usability Standards. | All NZ Government web services MUST use similar branding layout and structure. | * the solution meets NZ GWS WUG obligations. For example, the service client has a Home Page, Navigation, Copyright notice, etc. |
| BREAD Interfaces | The solution’s custom interfaces MUST be developed according to Browse/Read/Edit/Delete (BREAD) design patterns to improve usability. | Using a commonly recognised pattern throughout a system improves usability and decreases navigation and data entry error. | * the solution uses a common Search/Browse functionality to navigate to Resource specific View/Edit/Add/Delete Views. |
| PWA | The solution’s custom interfaces MUST be web and mobile capable web pages, preferably following Progressive Web App (PWA) Single Page Application (SPA) development patterns. | The solution must be accessible via the most prevalent devices (mobile).  The solution must take advantage of the devices and sensors available in a device to provide a better user experience (based on location, etc.) | * The solution’s user interface is separate from the service. * The solution’s user interface is developed using the most current recommended development practices in order to provide the longest use. |
| Configurability | The solution’s mutable system configuration values MUST be configurable via a Service Client User Interface. | Mutable system configurations must be configurable by the automated deployment process, and in extremis, by Operations Specialist intervention until the next system deployment. | * The solution’s mutable system settings are presented on a System Configuration view. |
| Learnability/Training | The solution MUST be intuitive enough to not require users to undergo training to use the solution. | The cost of training becomes a cost born by the sponsor.  It is inappropriate that a sponsor must bear an additional cost on to resolve an in adequacy in usability that was already paid to vendors to resolve in the first place. | * The solution provides a digital help system that is readily available to both unauthenticated and authenticated system users. |
| Operability | The solution MUST be operable by the widest range of users. |  | * The solution follows AoG mandated rules for Usability & Accessibility * The solution uses graphics suitable for visually impaired users. * The solution uses culturally appropriate graphics. * The style sheet allows for dynamic resizing of the interface in order to make text more easily readable. * The page is structured with * The HTML is decorated with ARIA tags to make it easier to read by blind readers. |
| Operability/Malware Detection | The solution MUST protect users from downloading malware by inspecting uploaded media for malware. | The reputation of the organisation is dependent on users trusting its services to do them and their group or organisation no harm. | * the solution uses a 3rd party service to inspects media uploaded by end users. * The solution can handle checking multiple files in one operation. * The solution does not persist media that is infected. |
| Operability/User Management | The solution MUST not require centralised management to manage users, groups or permissions. | Users closest to work are in the best place to make decisions of whom they wish to work with. | * the solution allows appropriate users other than centralised specialist staff (e.g., support specialists & business support specialists) to manage users, groups, obligations, permissions, roles. |
| Operability/Web Browsers | The solution MUST be operable from the widest range of current browser technologies. | The solution SHOULD be accessible from corporate and popular cell phone devices, using current browsers. | * The solution is accessible via the latest web capable devices (desktop, laptop, tablet, mobile). * The solution is available on the current 2 or 3 most used browsers (Edge, Chrome, Safari), whether on desktops or mobile devices. * The solution is available on the latest 3 automatically deployed/updated versions of the above listed browsers. |
| Universality | The solution MUST by default handle Unicode, UTF-8 transmission, and UTC datetimes. | The solution must persist and transmit universal data. | * The database is developed to persist Unicode. * UTF-8 Character set encoding is used in transmission between components * Validation allows Māori macrons and other variants (ā, ē, ō, etc.) * Search is case insensitive, can handle omissions of Macrons, and/or variations in spelling (‘wh’ / ‘f’). |
| Localisation | All UX text and images other than user submitted media MUST be translatable to multiple languages. | The NZ Realm is composed of multiple cultures, and should be inclusive to all.  The first cultures to address with this requirement are those implicit in The Treaty – both the English and Māori cultures. | * The solution is capable of using language packs to localize the user interface's images and text. * The solution’s text and images are managed in language-culture packs * different cultures may use different imagery * The solution SHOULD be capable of handling the different orientation of text in different cultures |
| Localisation/Layout | The UX SHOULD be able to be different depending on the culture | In addition to changing text and images, some cultures may desire different layout. | * The solution’s service client interface can reference different style sheets depending on the presentation culture. |
| Search | The solution MUST provide a generalised means to search the solution for Synopsis records of users and resources, returned in a filtered and paged manner. | The solution’s purpose is to allow users to easily develop, persist and re-find stored information.  Note:  Paging is a required precaution against DOS resource consummation attacks. | * The solution’s search service is universal in that it searches across all key elements of the system. * The solution’s search is text based. * The solution’s search may be filtered by textually entered constraints, rather than by use of secondary controls (e.g., one can type a date range, rather than have to use two different date controls). * Results are returned as Synopsis records, summarising key information of the found records (much as Google makes summaries of web pages) * synopsis records provide links to the source record being summarised. |
| Reportability/Reports | The solution MUST be capable of developing user refined and filtered Reports on Diagnostics, Errors, Sessions, Users, Groups, User submitted Resources, User submitted Resource Usage, System Resource usage (CPU) and System Responses. | Decisions as to how to spend resources are aided by reports demonstrating current conditions.  Decisions are often done by governance boards whose members do not access the system, or its views (refer to a requirement defined elsewhere in this catalogue, defining that Reports must be printable). | * The solution may leverage 3rd party analysis and reporting services that have access to a duplicate read-only database for reporting purposes only. * The reporting database is separate from the production system to ensure complex reporting operations do not negatively affect system performance. |
| Reportability/Channels | The solution MUST be capable of developing both on screen and printed versions of reports. | Although a need that is diminishing in a digital era, there is still value in having physically distributable artefacts of information. | * The solution generates reports that are for screen display, that include a link to the same information, prepared for printing.  Note: The currently preferred solution is to link to print ready PDF documents, rather than using web print-specific CSS tags (they rarely produce better than rudimentary results). |
| Alerts/Abnormal Behaviour | The solution MUST be capable of alerting to abnormal behaviour | Alerting operations and/or security specialists early can allow them to defend a system against manual or automated access to data. | * The solution SHOULD be able to notice abnormal activity by time, volume, source, illogical flows that may indicate automated searching for authorisation or validation weaknesses. * A WAF, separate from this solution can be used to fulfil most or all these tasks. |
| Alerts/Errors | The solution MUST be capable of alerting subscribed users of errors | Errors should be flagged to stakeholders who can analyse error reports and action any required resolution of the logged problem. |  |
| Error Protection/Meaningful Messages | The system MUST inform the end user of a failure that caused the request to not be successfully completed.  The application displays a generic plain English message to the user advising there has been a unexpected failure.  Messages are sanitised of PII or confidential credentials before persistence or presentation. | Informing users of the category of the error may help inform conversations they may have with support services. | * The   solution catches, logs, sanitizes & reports exceptions and stack traces, for errors including the following:   + Network failure   + 3rd party service failure   + Data storage exceptions   + Application exceptions   + User Authentication failure   + Resource location failure * A unique identifier (UUID) is developed for the report, which is referenced when notifying the end user that an error has occurred, so they can refer to it if they ask for assistance from support specialists. |
| Error Flow | In the event of an error the solution **MUST** provide the instructions describing the correct steps for a user or system. | Informing users what to do next may help users continue with their intended or alternately achievable tasks. | * The solution presents information to the user on how to proceed. |
| Documentation | The solution MUST be accompanied with sufficient documentation for all components to be deployed, maintained and operated by revolving resources over the solution’s service lifespan. | Developers and testers not expected to stick with the project for the service's lifespan.  In order to ensure their knowledge is not lost, documentation is required.  Better yet, automated tests that demonstrate how the system works should be developed prior to beginning work items. | * Developers can rely on automated tests to understand functionality developed by others, rather than having to rely on outdated and poorly maintained documents describing components that should be regularly evolving. * Documentation is electronically stored, preferably within a wiki. * Documentation can be made open and public, because developers do not access to configuration secrets, and therefore run less of a risk of including within documentation compromising information. |
| UI Aesthetics/AoG | The solution MUST meet All of Government (AoG) UI design guidelines. | Reusing AoG aesthetic guidelines improves recognisability and decreases the effort required to learn to use a solution's services. | * The solution’s service client interface can be skinned to align with the aesthetics of this organisation, and/or AoG directives. |

### COMPATABILITY (ISO-25010/Compatibility)

Interoperability is the ability for the solution to interact with other systems by providing services to, or accepting services from, the other systems. This interaction could be any combination, as appropriate, of systems that are internal or external to the Ministry, existing or proposed. Interaction should demand minimal work on the new system and rework on existing systems. It should be isolated so that it is not 100% dependent on such interaction. It is also known as system integration, interaction and compatibility while maintaining independence.

ISO-25010 recommends considering the following qualities and their descriptions when considering a solution:

* *Coexistence* is the degree to which a product can perform its required functions efficiently while sharing a common environment and resources with other systems, without detrimental impact on other systems.
* *Interoperability* is the degree to which two or more solutions, systems, components can exchange information and use the information that has been exchanged.

| ID | Statement | Rationale | Details |
| --- | --- | --- | --- |
| Co-Existence | The performance of The solution SHOULD NOT negatively affect the performance of other systems on the same device. | The solution’s resource consumption must not be so onerous as to negatively affect other services on the same device. | * The solution consumes less than 33% on average of the CPU of any device it is installed on. |
| APIs | The   solution MUST expose all core elements and functionality by externally & appropriately accessible APIs. | The solution must be in accordance with this organisation's API-First design Principles and enable flexibility around future integration and channel strategy requirements. | * The solution makes available APIs to manage key services, including the following:   + Diagnostics   + Errors Reports   + Settings   + Users   + User Profile/Preferences   + Groups   + Organisations   + Group Responsibilities, Obligations, Permissions   + Roles   + User Role Assignments   + User Permission Assignments   + Group Workflows,   + Resource Metadata,   + Resource Rights,   + Resource Roles   + Resource Responsibilities, Obligations, Permissions   + Resource User Role Assignments   + Resource Permission Role Assignments,   + Resources   + Resource Collections   + Search   + Validation |
| APIs/DTOs | APIs MUST use DTOs that do not expose the internal schema of entities or data storage schemas. | Security requires that internal structures are not exposed.  Maintainability requires that changes to internal logic and entities does not cause changes to integration messages – DTOs – which would break contracts established integrated systems. | * The solution maps entities to DTOs, and back again, at the API Service Façade layer. |
| APIs/DTOs/Versioning | APIs MUST be versioned | Maintainability requires that changes to DTOs does not cause force breaking contracts with established integrated systems. | * The solution provides versioned endpoints for all its APIs. |
| APIs/Queryability/ODATA | APIs MUST be queryable using ODATA. | To decrease development effort and complexity, Versions of APIs must be made available that are queryable, using an internationally accepted (OASIS) protocol (ODATA). | * The solution makes available ODATA capable REST APIs. |
| APIs/Queryability/GraphQL | APIs MUST be queryable using GraphQL | To decrease development effort and complexity, Versions of APIs must be made available that are queryable, using a protocol that is gaining international acceptance (GraphQL). | * The solution makes available GraphQL capable REST APIs. |
| APIs/SDKs | APIs **MUST** be supported with Software Development Kits (SDKs) developed using agreed technologies, invokable from a Command Line Interface (CLI). | APIs require testing by first from the command line, in order to develop automated integration tests that can be run by the automated deployment pipeline. | * SDKs are required to develop automation to ensure APIs remain accessible. * SDKs may be useful to 3rd party consumers to inform them how to connect to the service. |
| Interoperability/NZSIF | Where applicable, the solution MUST be able to exchange data with other systems using the NZ SIF format for its DTOs. | The solution is intended to operate within the NZ education sector, which is integrated using the NZ SIF interoperability standard.  DTOs are used to decouple internal logic from consumer requirements. | * The service exposes NZ SIF compatible DTOs. |
| Interoperability/SCORM | The solution MUST be able to exchange data with other systems using the latest SCORM protocol. | The solution must be able to communicate and exchange resources in the most current version of common education formats. | * The solution is either naturally capable or is extended with 3rd party libraries to manage SCORM resources. |
| Interoperability/cmi5 | The solution MUST be able to exchange data with other systems using the latest cmi5 protocol. | The solution must be able to communicate and exchange resources in the most current version of common education formats. | * The solution is either naturally capable or is extended with 3rd party libraries to manage cmi5 resources. |
| Interoperability/Common Cassette | The solution MUST be able to exchange data with other systems using the latest Common Cassette protocol. | The solution must be able to communicate and exchange resources in the most current version of common education formats. | * The solution is either naturally capable or is extended with 3rd party libraries to manage Common Cassette based resources. |
| Interoperability/Common Cartridge | The solution MUST be able to exchange data with other systems using the latest Common Cartridge protocol. | The solution must be able to communicate and exchange resources in the most current version of common education formats. | * The solution is either naturally capable or is extended with 3rd party libraries to manage Common Cartridge resources. |
| Compatibility/LTI | The solution' service client interface MUST be able to act as an LTI client. | The solution’s interface must be able to embed in other education sector tools. | * The solution’s service client is embeddable in an LTI host, such as a Learner Management Service (LMS) Client. |
| Interoperability/MTA | The solution MUST be able to connected to an Mail Transfer Authority (MTA). | The solution must be capable of distributing Notifications at least by one external channel. | * The solution uses Email to notify subscribed stakeholders of alerts. * The solution internally also provides a Notification area. |
| Interoperability/Malware Service | The solution MUST be capable of invoking a 3rd party cloud based anti-malware service. | The solution must be capable of validating that user uploaded media is free of malware. | * The solution validates user uploaded media. * The solution validates user uploaded media on a regular basis to retest uploaded media against newly discovered virus definitions. * Media that contains malware is not allowed on the system. * Media that has been uploaded, then found out to contain malware will be physically removed from the system. |
| Separate Server & Client | The solution’s service client interface MUST be developed in a separate tier from the service. | When the interface is developed as a server-rendered part of the service, APIs can be bypassed and therefore there is no impetus or guarantee that all necessary APIs will be developed and sufficiently tested. | * The solution is developed such that all information required to make any required user interface is available via versioned & DTO’ed REST APIs. |
| Validation | The solution MUST validate all input. | The solution must use a common validation approach irrespective of the Service Client used. | * The solution validates input at the Service Facade. * The solution duplicates validation logic in the Service Client's user interface in order to provide a better user experience. |
| Data warehouse | The solution MUST be capable to make key information available to a data warehouse. |  | * The solution’s environments export key aspects of their data to a pickup location for pickup by a data warehouse. * The solution can also be queried by a data warehouse using the solution’s reporting APIs (which use the solution’s reporting database to protect the system's performance from complex reporting queries). |

### RELIABILITY (ISO-25010/Reliability)

Reliability of a solution is - a request being processed by that system according to agreed business logic and a valid response being returned by the system in the time expected. The time expected is influenced and constrained by the performance requirements of the system.

Fault tolerance is defined as the immunity of the system in the event of an unexpected internal or external fault. This is evaluated in the context of the likelihood of the fault, the severity of the fault and resulting impact on end users.

*Recoverability* is a subset of *Reliability, and*is the ability to re-establish the normal functioning of a system after a significant external event, such as a disaster. A disaster implies total loss of access or use of the system by end users (e.g. production) where the outage has, or will, extend beyond the Maximum Tolerable Downtime (MTD).

The MTD and ‘normal functionality’ are agreed between IT (or the vendor) and the business owner of the system in the Service Level Agreement (SLA).

*Recoverability* uses activities such as data protection, Disaster Recovery (DR), and Business Continuity Planning (BCP) to ensure the recoverability of systems

ISO-25010 recommends considering the following qualities and their descriptions when considering a solution:

* *Maturity* is the degree to which a solution needs for reliability, under normal operation.
* *Availability* is the degree to which a solution is operational and accessible when required for use.
* *Fault tolerance* is the degree to which a solution operates as intended despite the presence of hardware, software or user faults.
* *Recoverability* is the degree to which, in the event of an interruption or failure, a solution can recover the data directly affected and re-establish the desired system state.

| ID | Statement | Rationale | Details |
| --- | --- | --- | --- |
| Maturity | The solution’s language, framework and components MUST be accreditable as supportable technologies. | The solution must be developed using a language and framework that decreases the number of errors in a system. | * The solution is developed using components that this organisation's accreditation and security specialists do not raise as risks. |
| Maturity/Development Language | The solution SHOULD be developed using a compiled language. | Compilers can prevent errors by pre-compilation that are not discoverable by scripts languages which can only discover them during run-time. | * The solution is built with a compiled language to limit the number of bugs that make it to a production environment. * The solution’s development team uses strict compilation flags to find and force the fixing of errors early. |
| Maturity/Development Framework | The solution MUST use a development Framework that is usable on multiple platforms. | No OS has complete dominance of the market and it is virtually certain that a solution has to be capable of running on different OSs. | * The solution is built in on a common framework, such as:   + .NET Core (as opposed to the older Windows specific .NET Classic Framework)   + Java (with reservations due to Oracle's licensing costs). |
| Availability | The solution MUST meet the availability expectations defined in the *Target Quality Objectives* table. | The system must be capable of operating during the hours users are most expected to use the system, meeting the demand, throughput and response criteria listed in the *Target Quality Objectives*table.  In the case of a planned or unplanned outage, the solution must be able to be returned to a functionality state in the shortest amount of time. | * The solution is deployed to cloud infrastructure that has a higher uptime SLA than the one described within the *Target Quality Objectives* table. * The solution is protected by a WAF to limit DoS attack traffic. * The solution is horizontally scalable to handle surges of demand. * The number of connections to the database is minimised per request to diminish the chance of the database becoming a bottleneck. * The solution uses caching to limit the number of threads required to communicate across tiers to the database to diminish the chance of it becoming a bottleneck. * Complex queries for reporting are delegated to a separate reporting database. * The database is backed up regularly for DR restoration. * The solution is deployed using Infrastructure as Code, so that a new production environment can be re-developed quickly after a Disaster. * The solution is accompanied with an automated DR recovery solution. |
| Availability/Single Point of Failure | The solution MUST be designed to minimise the number of Single Points of Failure. | If a component of the system fails (e.g., caching) the system must be resilient enough to continue to function until the issue can be addressed. | * The solution is capable of working if caching is unable to work. * The number of connections to the database is minimised per request. |
| Disaster Recovery | The solution MUST include an organisation accepted automated Disaster Recovery plan. | Upon disaster to the system environment and/or production data, the service must be able to be made available to end users within the shortest agreed delay. | * The solution is accompanied with an automated DR recovery solution that is able to redevelop an environment within time delays defined within the *Target* *Quality Objectives* table. |
| Backups | The solution MUST include a regular backup strategy that meets Response Point Objective Time (RPO) and Response Time objectives (RPT) targets as defined in the *Target Quality Objectives* table. | Upon corruption of PROD data, data should be restored from backups. | * The solution’s data is backed up in full every day. * The solution’s data is backed up incrementally at an interval that satisfies the constraints defined in the *Target* *Quality Objectives* table. * The solution’s backups are kept for a delay defined in the *Target* *Quality Objectives* table. * Backups are kept in a resource group separate to the production data's resource group. * Backups are taken, restored and accessed only by the deployment pipeline's agent (to remain in conformance of ISO-27001 requirements). |
| Geo Resilience | The solution’s design SHOULD be ensure it is resilient against unavailability of infrastructure in the default area. | Regional network failures may make services in other countries temporarily unavailable. | * The solution is developed using Infrastructure as Code so can create a new environment in a new location, if directed to. |
| Transactions | All change request operations SHOULD be committed within a single end of request Transaction. | Failure in mid request can lead to only half the information being committed, leading to difficult to trace errors. | * The solution attempts to ensure that write operations are committed within a transaction so that all operations either occur together, or don't. * The solution is developed using an ORM to handle requests (as they generally are wrapped in transactions). * The solution collects and holds a request's write operations and only triggers them at the end of the request to limit hard to trace/resolve "half/writes". |
| Restart | The process of restarting or recovering after a service or component outage or failure MUST NOT require needing to restart the underlying infrastructure of the services. | Minimises the downtime experienced by users by simplifying the process of restarting the solution and automating it to make it more efficient for support staff. | * The solution restarts itself after a system reset upon the first request. * The deployment pipeline has a final step to invoke the service and wake it up, ensuring the first user is not faced with a longer than usual delay. |
| Offline | The solution MUST not be made unavailable for longer than the Maximum Tolerable Downtime (MTD) defined in the *Target Quality Objectives* table. | The system must not be made unavailable for use for longer than necessary. | * The solution uses Infrastructure as Code, Database schema as Code, Automated database backup restoration to permit – in the case of a Disaster – the recreation of a completely new environment within the delay stated within the *Target Quality Objectives* table. |
| Scheduled Downtime | Stakeholders and users MUST be given notice as to when solution capabilities and components will be rendered inaccessible. | Users require foreknowledge, and Support specialists require foreknowledge as to when users may inquire as to why a service is no available. | * The solution notifies end users as to upcoming events that will take the service offline. * System Notifications are available as interface banners, and/or alerting subscribed stakeholders. |
| Scalability/Users | The solution MUST be capable of supporting the projected number of concurrent and absolute users as defined in the *Target Quality Objectives* table. | The solution must remain accessible through the service's expected lifetime.  The system must be capable of running economically when not in demand, while not compromising its ability to meet projected peak demand. | * The solution dynamically scales horizontally based on the CPU being consumed at higher than a specified %, for longer than a specified duration. * The solution dynamically scales back down based on the CPU consumption falling lower than a specified % for longer than a specified duration. |
| Fault Tolerance | The solution MUST trap unexpected errors and capture relevant information in permanent records. | The solution must record error reports for later analysis by operations specialists. | * The solution creates error records and persists them to facilitate analysis later, the solution’s error records storage is filterable and searchable error records are given a unique UUID, which is shared with the system user, for later sharing and discussion with Customer Support specialists. |
| Recoverability | The solution MUST be able to recover from unexpected behaviour. | The solution must be capable of continuing to perform, even if it encounters an unexpected error – just log it for later analysis, conditionally alert operations specialists, and continue. | * The solution traps all unexpected logical and component errors, alerts subscribed stakeholders, and resets state to allow for handling subsequent requests. |
| Recoverability/MTD | The solution MUST be capable of meeting the Maximum Tolerable Downtime (MTD) specified in the *Target Quality Objectives* table. | The reputation of the organisation is associated to the ability of end users being confident that the service they used will be available when needed – and if it goes down, that it is a both a rare occurrence, and that the service re-becomes available in an acceptable amount of time. | * The solution’s cloud provider’s SLAs, combined with the solution’s fully automated DR recovery process ensure that the stated MTD is achievable. |
| Recoverability/DR | The solution MUST meet be able to recreate a new environment and be operational within the delay specified in the *Target Quality Objectives* table. | Cloud infrastructure reduces but does not eliminate the chance of having to recreate a new environment. | * The solution uses automation to recreate environments as needed, including restoring data from backups, within short delays. |
| Recoverability/BCP | The solution SHOULD rely on a capable DR plan to restore business services within delays specified in the *Target Quality Objectives* table. | If cloud services are down, and services are impaired in such a way that a DR operation will not resolve the issue, the organisation will have other problems to consume our attention until the cloud becomes available. | RNR |

### PERFORMANCE (ISO-25010/Performance)

Efficiency, throughput, capacity, response time and resource consumption for Online Transaction Processing (OLTP) and background processes.

The system's ability to complete defined volumes of computer assisted Business Transactions (online and background operations) within acceptable times and with an acceptable consumption of resource.

ISO-25010 recommends considering the following qualities and their descriptions when considering a solution:

* *Time behaviour* is the degree to which the response and processing times and throughput rates of a solution, when performing its functions, meets requirements.
* *Resource Utilisation* is the degree to which the amounts and types of resources used by a product or system, when performing its functions, meets requirements.
* *Capacity* is the degree to which the maximum limits of the solution meet or exceed requirements.

| ID | Statement | Rationale | Details |
| --- | --- | --- | --- |
| Metric Collection | The solution MUST collect and make available key metrics for reports. | Collecting metrics provides information required to direct effort towards continual improvement of the service rendered. | * Average & Max Duration of Transactions * Average & Max Volume of Transactions * Average & Max Transactions per Session * Transactions to Dynamically allocated Hosts * Average & Max Number of Sessions per User * Average & Max Number of Resources accessed per Session * Include collecting metrics for developing the following ascending/descending lists:   + Resources collaborated on per User   + Feedback per User   + Resources Flagged   + Users Flagged   + User's Resources Flagged   + Storage used per User   + Storage used per Group |
| Time Behaviour | The solution MUST be capable of completing peak user requests within delays defined in the *Target Quality Objectives* table. | Users become disconnected from systems that make them wait. |  |
| Time Behaviour/Response Time | The solution’s APIs and User Interface MUST be able to capable of completing Peak surge requests and returning control to users within response times defined in the *Target Quality Objectives* table. | The solution must feel responsive to users, allowing them to be as efficient as they are capable. | Note:  completed User operations specifically means that the first operation has been actioned and completed on the server, and the next operation can be begun on the service client. All images, etc. in the response from the first operation should have completed loading by the time of beginning the actioning of the next operation. |
| Capacity | The solution MUST be capable of completing the requests of the peak user quantity defined in the *Target Quality Objectives* table. | The solution must not cause users to be confused by receiving generic undescriptive system HTTP errors. | * the solution is capable to dynamically scale up and down horizontally to meet peak normal activity. |
| Capacity/Data Locking | The application MUST NOT impede multiple (concurrent) users from using and maintaining the same data. The system will be designed so that data locking does not exclusively lock data in such way that it prevents other business users from using the system in an efficient manner. | One user using data should not impede another user from using and maintaining that same data. Lock management will be designed to support concurrent access to data for read, update and delete operations. | * The solution does not open transactions at the beginning of requests and close them at the end. |
| Capacity/Diagnostics trace message storage | The solution MUST keep diagnostics trace messages for a configurable duration of time before rolling over. | Diagnostics trace messages can be voluminous. Being not required for operations, auditing or archiving purposes, they can be purged as needed to keep storage requirements within reason. | * The solution’s diagnostics trace messages are kept for a configurable duration (by default 31 days). * The solution’s PROD environment’s diagnostic trace messages are conserved even if the environment is destroyed and rebuilt (e.g., using Infrastructure as Code invoked by an automated deployment). * The solution’s diagnostics messages are copied to a target area available to an organisation’s SIEM system to import. |
| Capacity/Storage | The solution MUST be able to persist user supplied information, sised as per described elsewhere, for the service lifespan of the system. | The information system should not physically delete information, and therefore keep it for the lifespan of the system. | * The solution dynamically scale up and out as required to store the suggested quantity of data required by the solution over its service lifespan. |
| Capacity/Archiving | The solution MUST not archive data outside the solution’s designed production data environment. | Archived data is production data at a specific point in time, and therefore has a relative value compared to production data. | * The solution does not export archives. * The solution deletes records using a logical state flag. |
| Capacity/Users & Resources | The solution MUST be able to manage the expected number of users and resources, described elsewhere. | The solution SHOULD be capable of meeting expectations of users to access the system and manage information. | * The solution is capable of dynamically scaling up to manage the projected number of users and resource elements without change to the architectural design. * Please describe any concerns that the solution could not scale as required. |
| Capacity/Throughput | The solution MUST be capable of handling – within max acceptable delays defined in the *Target Quality Objectives* table – the requests of the peak projected number of concurrent users. | The solution SHOULD be capable of handling projected activity surges without frustration of end users, or becoming unavailable. | * The solution is capable of dynamically scaling horizontally to handle the requests of projected peak surge activity. * the solution ensures there are no bottle necks at various tiers (for example database accesses per second). |
| Resource Usage/CPU | The solution MUST be capable of completing peak surge user requests within delays defined in the *Target Quality Objectives* table, while consuming no more than 50% of each virtual device they are on. | The solution must be able to handle requests without putting undue demands on physical or virtual hardware, which in turn shortens the lifespan of physical infrastructure. | * The solution is capable of dynamically horizontally scaling to handle surge activity requests, using CPU activity as a trigger. * The proposed solution will meet performance delays at peak loads as defined in the *Target Quality Objectives* table, with CPU loads being no more than 50%, preferably 33%. |
| Background Operations | The solution’s capabilities, capacity and performance MUST NOT be negatively affected by background operations. | Users should not be affected by background batch, reporting or analysis operations. | * The solution is developed to ensure background operations do not negatively affect users, using a combination of one or more of the following:   + command queues to handle tasks asynchronously,   + using a limited number of background threads to process long running processes,   + etc. |
| Background/Multi-threaded | The solution SHOULD process background operations across multiple threads. | Parallel Multi-threading allows for faster completion that Serial operations, providing processing have plenty of opportunity to complete before the beginning of the next business day | * The solution processes background activity in a paged, multiple threading manner. |
| Background/Timing | The solution MUST process background processes throughout the day. | Processing background tasks after hours is common – but illogical to run important tasks when no support, operations, or security specialists are working. | * The solution processes background activity as a separate service account running a separate process, accessing the same database. |

### MAINTAINABILITY (ISO-25010/Maintainability)

Maintainability is the ability to maintain the system efficiently: find and remove faults, improve performance, carry out modifications and infrastructure upgrades. It is also known as modify-ability, enhancement, fault detection, isolation and repair.

Traceability is the ability to retain details of specified business, transactional, and system activity and data changes, including normal, abnormal, and error conditions with selective control. It is also called auditability and audit trail.

ISO-25010 recommends considering the following qualities and their descriptions when considering a solution:

* *Modularity* is the degree to which a solution is composed of discrete components such that a change to one has minimal impact on others.
* *Reusability* is the degree to which an asset can be used in more than one system or reused to build other assets.
* *Analysability* is the degree of effectiveness and efficiency with which it is possible to assess the impact of a solution, a change, a failure, or determine what requires change.
* *Modifiability* is the degree to which a solution can be effectively and efficiently modified without introducing defects or degrading operational quality.
* *Testability* is the degree to which test criteria can be established, and determination of whether the solution meets them.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Statement | Rationale | Details |
| Reusability | The solution’s components MAY be reusable by other systems. | The cost benefits of reusing components and services must be weighed against the cost of complexity and cross-dependencies. | * The solution’s components are re-using other solution's components, * this solution's components are envisioned as being reusable by other solutions. |
| Analysability/Error Report Logging | The system **MUST** log any exceptions to an appropriate log and make the information accessible to a system administrator role. | Ensure adequate exception logging to support discovery and diagnosis of errors by appropriate support persons. | @@@ |
| Analysability/Auditing | The system **MUST** capture the appropriate level of audit information for all transactions that read, create, update or delete data.  Technical and business users reading audit trail data can determine, for all transactions that read, created, updated or deleted. | Technical and business users should be able to determine:   * What was viewed or changed * By whom * When | * The solution’s auditing logs are protected from tampering, by * Using data store service within an ISO-27001 compliant data warehouse that logs access to physical resources and access to virtual hard drives is limited and audited to prevent tampering   Note:  Signage of the audit log entries is not implemented in this system. |
| Analysability/Capabilities | The solution’s services and service client interfaces MUST support filtering and sorting capabilities using a case insensitive, accept sensitive, culture. | Any user interface must assist users in finding appropriate information in a manner that produces the highest chance of finding the correct information, no matter the culture. | * The solution’s service client interfaces provide a means to search across items, whether AutoComplete is offered, present Summaries of matching records, whether they can be sorted by one or more fields, can be used effectively on both mobile and desktop interfaces (i.e., avoid using table layout strategies better suited to desktop interfaces than mobile devices, etc.)   Note:  Consider matching or improving on Latin\_1\_General\_CI\_AS |
| Analysability/Search Tools | The solution MUST include capabilities of filtering, searching case-insensitively, sorting paged diagnostics, error and audit records. | Support specialists must have the tools to find records to diagnose user and system activity. | * the solution makes system records searchable in an efficient manner, using filtering criteria & response paging. |
| Reportability/Monitoring | The solution MUST include a means of monitoring key metrics. | Metrics are required to be collected in order to compare abnormal behaviour against. | * The solution monitors: * Performance issues * Security issues * Abnormal Behaviour issues * Errors |
| Reportability/Alerting | The solution SHOULD be capable of alerting subscribed stakeholders of atypical/abnormal activity that warrants investigation. | Abnormal/atypical activity should be flagged for review by the appropriate user. | * The solution analyses activity for atypical activity and can alert subscribed stakeholders. * Performance issues * Security issues * Abnormal Behaviour issues * Errors |
| Testability/Automated | The solution MUST perform all testing of configuration or customisation by automation. | Resources, whether permanent resources or contractors, leave with their knowledge of systems.  Documentation of evolving systems are hard to keep current.  The only effective way to ensure quality assurance knowledge lasts the length of the service lifespan, and is rapid enough to not delay deployments of environments, functionality and data is to automate all testing. | * The solution’s capabilities, functionality and qualities are tested by automation initiated by the delivery pipeline. * The solution’s automated pipeline tests the following:   + external system (database, identity provider, caching, malware detection, etc.) integration testing,   + API service facade authorisation, access and functionality integration testing   + automated smoke testing,   + automated quick, high-level functionality testing,   + automated longer, in-depth, functionality testing. |
| Modularity/Reference based | The solution’s customisation SHOULD align with and leverage organisation standards and patterns.  Deviations must be recorded as Project Decisions. | The risk of delivery failure is reduced by:   * reusing where possible * aligning with Ministry and AoG design and data principles and standards * proven reference architectures for custom development * development standards * use of available environment services (reusing enterprise systems, including IdPs, as malware validation, smtp services, etc.) * data management standards * deployment standards, practices and guidelines | * The system's database schema is specific to this application. * The system deploys data to data warehouses via API (not SQL). * The schema of the DTOs used meets data management standards. * Uses deployment standards, practices and guidelines. |
| Modifiability/Configurable Features | The solution SHOULD be configurable to enable and disable different functionality depending on user group. | Certain features are required to be released in stages, starting with only a subset of users. | * The solution allows the dynamic disabling of features that are in the code base, but not ready for use by end users. * The solution allows for the dynamic association of features to certain groups of users, for A-B feature testing. |
| Accessible Code Repository | Custom code, configuration or test scripts written for this solution **MUST** be maintained in an organisation accessible and clonable code repository. | Code written for this organisation must be analysable for code security and quality. | * The solution uses a Git based repository * The solution uses a GitHub based repository unless otherwise defined |
| Modifiability/Lockout | Any development or ongoing support contract prepared to deliver this solution MUST NOT exclude this organisation from actively participating in development and operations. | This organisation must not accept locking or lockout contractual conditions. | * The solution’s automated pipeline, running unit tests, integration tests, behaviour tests are there to catch errors in logic before checkin branches are accepted and merged. * Any fixes required are the whole project team's responsibility, allocated to the group most knowledgeable as to the cause and resolution, dependent on availability. |
| Modifiability/Loosely coupled | The solution SHOULD prefer being developed using loosely coupled systems and components. | The solution SHOULD allow modification to one component or group of functionality with the least chance of affecting other parts of the solution. | * the solution is developed as a loosely coupled set of systems, components, logical domains.   Consider describing the use of external 3rd party subscriptions services for Identity services, PII storage services, confidential storage services, structured and unstructured storage services, caching, malware detection, content validation, etc.  Consider describing using DDD for system component development.  Consider describing using SOLID development patterns. |
| Web Standards | The solution MUST be developed and maintained to current Web Standards. | The solution is more maintainable and secure when the standards used are mainstream and current. | * the Solution’s Service Client Interface is developed using current web standards.   + HTML5+   + CSS Levels 4+   + All W3C Modules   + Document Object Model level 3+   + ECMA Script 5.1+ * The solution’s service client does not rely on platform or browser extensions (eg:  ActiveX Adobe Flash, Flex or AIR Microsoft Silverlight, Microsoft ClickOnce, SmartClients, Java or JavaFX, Embacadero DataSnap, etc.) |
| Status Reporting | The solution SHOULD be able to develop both web interface and printable reports of the status of key aspects of the system. | Senior stakeholders are not always users of the system yet require reports on which to base decisions. | * The solution has web view interfaces and associated printed reports to describe key aspects of at least the following key elements of the solution:   + User Concerns:     - Users (# of users, # of new and leaving users, etc.)   + Groups (# of groups, # of new and closing groups, etc.) * Resource Concerns:   + Resources (# of resources, # of new, retired resources, etc.)   + Resource Collections (# of groups, activity metrics, etc.) * Support Concerns:   + User Feedback (#ratings, # comments, etc.)   + Support Tickets (#open, #closed, etc.) * Security Concerns   + Security Events: (# of reviewable events, reviewed events, issues, etc.)   + Operations Concerns:   + Unexpected behaviour (# of errors, # of resolutions, # of reviewed and discounted events, etc.) * Infrastructure Concerns:   + Environments (# of environments)   + Services   + Costs (per month, new, closed) * Development Concerns:   + Issues, Bugs (# of open/closed issues, # of open/closed requests, # of open/closed bugs, etc.)   + Work Item velocity |
| Quality Assurance | The solution’s functionality and qualities MUST be quality assured before significant change releases. | Business and support specialists should have confidence that the solution automates testing to support a quicker delivery of confidence to system certifiers and accreditors. |  |
| Source Control | All configuration, documentation and source code developed for the solution will be persisted in a distributed source code repository accessible by this organisation. | Vendor support lock is not valued by this organisation. | * The solution uses an ALM managed repository for configuration, scripts, etc. required to configure a new environment and deploy the solution’s code to it. |
| Change Notification | Feature changes to the solutions MUST be communicated to subscribed Stakeholders. | Uses should be able to prepare for changes to the system. | * the system notifies users of system changes.   Note:  use organisation Change Control provided processes to notify stakeholders other than users. |
| Change Control | Large change releases MUST be coordinated as per this organisation's Change Control process. | Impact to end users and other stakeholders (e.g., Support) must be minimised by letting them know of upcoming changes to the system. | The solution’s deployment process:   * Is compatible with a formal change control processes. * Uses automated pipeline workflows and email based workflows to notify and gain approval for deployments from one or more approvers prior to beginning automated deployments. |

### PORTABILITY (ISO-25010/Portability)

ISO-25010 recommends considering the following qualities and their descriptions when considering a solution:

* *Adaptability* is the degree to which a solution can effectively and efficiently be adapted for different and evolving hardware, software, or environments.
* *Intstallability* is the degree of effectiveness and efficiency in which a solution can be successfully installed/uninstalled in a specified environment.
* *Replaceability* is the degree to which a solution can replace another system for the same purpose in the same environment.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Statement | Rationale | Details |
| Provisionability | The solution MUST be able to provision users and groups via a standards based API. |  | * the solution uses a protocol (SCIM or equivalent) to provision and deprovision groups and users within. |
| Group Invitations | The solution MUST allow appropriate users to invite by notification other users to one of their groups, in one group role or another. | Centrally controlled group and user provisioning requires unwieldy processes that do not scale efficiently.  Letting appropriate users organize groups as they see fit improves the speed with which users can begin to benefit from using a system. | * the solution allows for a non-centralised process of inviting users, whether internal or external organisation users, to join a group within the solution. * The solution allows requiring approval in some cases of invitations. |
| Deliveryability/Rollback | A deployment of the solution MUST be able to be automatically rolled back upon failure or other reason. |  | * the system's automated deployment processes ensure that a failed deployment can be reverted. |
| Deliverability/Environments | The solution MUST include a number of non-PROD data environments. | In addition to a PROD environment, stakeholders must be able to inspect the functionality of the system, and test integration, in environments that contain non-PROD data (data that is not derived from production data, even if obfuscated). | * the solution is deployed to non-PROD environments, such as DevTest (DT), System Test (ST), User Test (UT), Integration (INT) for external clients to test their API integration.   *Note: Refer to the Deployment View for proposed environments.* |
| Deliverability/Automated Environment Creation & Deployment | The solution MUST be able to create and deploy to any new environment, solely by automation. | Upon a disaster, a solution must be able to be rebuilt quickly from scratch using   * Infrastructure as Code, * Data base schema as Code, * Automated Backed up data. * Automated Data Restoration. * Automated Migration of Data to other systems (either drop/pickup areas, or APIs of external Data warehouse, SIEMs, etc.) | * the solution uses an automated pipeline to recreate environments from scratch and deploys logic and data to them. * the solution can be deployed purely by automation to a new environment of any choice (including BT, DT, ST, UT, PP, IT, PROD, etc.). * the solution’s credentials are persisted external systems for safe management. * the solution is deployed to non-PROD environments, such as DevTest (DT), System Test (ST), User Test (UT), Integration (INT) for external clients to test their API integration.   *Note:* *Refer to the Delivery View for proposed environments.* |
| Deliverability/Automated Integration Deployment | The solution’s testing processes COULD include deploying agreed 3rd party API clients. | In order to verify that API changes do not break remote key Service Clients, consider validating the changes by connecting to the API as intended. | * NRN: Unit-testing and integration testing are performing the same functionality. This requirement is not required. |
| Adaptability | The solution MUST be able to adaptable by configuration, and not require another deployment. | Operations Specialists must be able to reconfigure the system to meet changing use case or constraints. | * the solution has apis and a user view to manage system configuration. |
| Adaptability/Hardware | The solution SHOULD NOT require specialised hardware or service license or subscription. | Specialised hardware may not be available for the duration of the service. | * NRN |
| Replaceability | The solution SHOULD allow the consolidation of the capabilities of one or more existing systems. | Consideration of New solutions should be considered based on reducing complexity, improve maintainability of the organisation's system, decreasing cost. | * NRN: this requirement cannot be answered by respondents. |
| Installability/Dependencies | The solution SHOULD be portable to different operating systems. | Solutions should not be limited to commercial operating systems (e.g., Windows). | * the solution does not depend on a particular device, OS, Framework or Database – although it is acceptable that the solution is dependent on a specific cloud provider’s services. |