ICT Project Guidance

Baseline Capabilities View

Version:

0.1

Author:

Sky Sigal, Solution Architect

## Description

This document provides an integrated perspective on the core functional groupings that bridge user needs and detailed system functions. This document outlines the essential capabilities that ensure the overall design is robust enough to deliver comprehensive business functionality, highlighting critical areas where gaps might compromise system performance and guiding strategic decisions for future enhancements.

## Synopsis

While systems are developed to support a myriad of business services, their success ultimately depends on a thorough understanding of the underlying capabilities common to all. This document lists these essential capabilities, organised by domain, and highlights their roles in coordinating both the management of technology and the dynamics of the social domain, including people and groups. It concludes by explaining how to leverage them to build lightweight, business-specific capabilities.

## Contents

[Description 1](#_Toc191306674)

[Synopsis 1](#_Toc191306675)

[Contents 2](#_Toc191306676)

[Introduction 3](#_Toc191306677)

[Domains 3](#_Toc191306678)

[System Domain Capabilities 4](#_Toc191306679)

[People & Groups 7](#_Toc191306680)

[Account & Feature Management Capabilities 8](#_Toc191306681)

[Extensions to Core Capabilities 8](#_Toc191306682)

[Extension Business Modules 9](#_Toc191306683)

[Considerations 9](#_Toc191306684)

[Conclusion 10](#_Toc191306685)

[Appendices 11](#_Toc191306686)

[Appendix A - Document Information 11](#_Toc191306687)

[Images 11](#_Toc191306688)

[Tables 11](#_Toc191306689)

[References 11](#_Toc191306690)

[Review Distribution 11](#_Toc191306691)

[Audience 11](#_Toc191306692)

[Structure 11](#_Toc191306693)

[Diagrams 11](#_Toc191306694)

[Acronyms 11](#_Toc191306695)

[Terms 12](#_Toc191306696)

[Standards 12](#_Toc191306697)

[Appendix B 12](#_Toc191306698)

# Introduction

Capabilities are high-level groupings of functionality that serve as a bridge of understanding between user requirements (UR) and system requirements (SR) (specifically the functional requirements(FR) subcomponent).

Capabilities act as an intermediate abstraction, clustering related functions into coherent areas of business functionality. This abstraction is vital because it sits between the broad needs expressed by users and the granular technical specifications required to meet these needs. By examining capabilities early in the design process, decision makers can assess whether the overall system design is fundamentally able to meet the intended user requirements.

Evaluating capabilities provides a strategic, high-level check on system completeness. When an entire capability is missing, it indicates a significant gap—an entire block of business functionality that cannot simply be compensated for by combining or reusing isolated functions. In contrast, if a single function is absent within a capability, there may be opportunities to marshal existing information objects or alternative functions to deliver the needed outcome. This distinction is critical; a missing capability means the system lacks a crucial area of functionality, whereas a missing function may be remedied with additional logic or components.

# Domains

Capabilities can in turn be organised into logical Domain, or “areas of concern”.

Almost every IT system starts off being structured around two fundamental domains: technology and people. The development process begins with establishing the capabilities to manage the technology domain, which in turn provides a robust foundation for effectively handling the people and groups domain.

Note:  
these aspects are often addressed concurrently—since discussions about system users inherently involve the people domain.

In addition to these two core domains, and depending on the system’s functional requirements, other domains may also play a foundational role. For instance, in systems designed for multiple, distinct consumer groups, domains such as accounts, subscriptions, plans, and features become important.

However, once these two or three foundational capabilities domains are established and available for reuse, additional logical modules can be integrated. These business modules extend and reuse the underlying base layers while also interacting with one another. For example, one module might provide student profiles tied to the People domain, while another offers school profiles associated with Groups. A third module could extend core concepts—such as capturing enrolments as an extension of a base subscription and attendance as an extension of base service usage—thereby linking the student and school modules.

Below is a list of capabilities associated with these core domains. They represent the standard functionalities expected of a mature system. In this context, “mature” means that while a system doesn’t need to include all these capabilities in a proof of concept (PoC) or minimum viable product (MVP), it is important for system designers, developers, and procurement professionals to recognize them and ensure the system is designed with these clear separations – improving maintainability – as well flexibility and reuse, to add additional capchatabilities later, as needed.

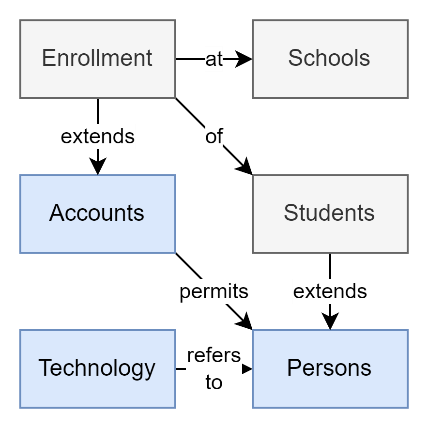


Figure 1: Baseline and example of extension Logical Modules

## System Domain Capabilities

The System domain provides the foundational infrastructure for running and maintaining the platform. It contains core capabilities that support diagnostics, identity, permissions, configuration, and communications—essential for any operational environment. These capabilities are shared across all other domains and establish the base from which the rest of the platform operates.

### Diagnostic Message Handling Capabilities

Supports the transient recording of system events, such as errors, warnings, and performance metrics. These are streamed to a remote store (e.g., cloud-based log service accessible to authorised maintenance specialists) for optional later inspection and correlation. Helps identify issues and monitor health. (Entities: LogEntry, LogStream)

**Important:**   
transient diagnostic logs are not to be confused with permanent auditing tool.

### Metric Management Capabiltieis

### System Configuration Management Capabilities

Holds immutable configuration values that define integration settings, service URLs, feature toggles, and build-time behaviours. These values are locked per deployment and ensure consistent behaviour across environments. (Entities: SystemConfiguration)

### Error Record Management Capabilities

Captures structured, permanent error records to support maintenance and reliability. These errors are analysed over time to understand system quality and failure trends. (Entities: ErrorRecord)

### System Settings Management Capabilities

Allows runtime administrators to modify user-facing settings that alter appearance, behaviour, or metadata of the system. Enables non-developers to control features like branding, SEO, or content discoverability. (Entities: SystemSetting)

### Request Management Capabilities

Handles inbound requests and applies pre- and post-processing logic. Includes middleware handling, auditing hooks, transaction finalisation, and guarantees of change persistence. (Entities: RequestContext)

### System Configuration Management Capabilities

System configuration provides the means to deploy *immutable* integration settings that won’t be changeable by user actions before a subsequent deployment.

Note:  
Immutable configuration capabilities is distinct from mutable System Settings Capabilities listed later.

### Error Record Management Capabilities

Provides the capability to save and manage \*permanent\* error records.   
Purposes include assisting with maintenance, as well as showing increasing qualities as the rate of errors descends.

### System Settings Management Capabilities

Provides the means for users with sufficient permissions to set, update or clear persisted *mutable* settings. While Configuration is about settings device integration values that don’t change between deployments, system settings are values that are persisted is system data stores (e.g.: a database). They permit admins to set the system’s appearance (logos, etc.), discoverability, SEO and other attributes, routing rules, etc. The more that can be managed as settings, rather than configuration values, generally, the more usable and customisable the system.

### Request Management Capabilities

Provides the means to pre-process and post-process requests.

Post processing is a foundational capability to ensure state changes are persisted to the data store, even if the developer hasn’t implicitly programmatically Committed changes (i.e., it just works transparently).

### Request Routing Capabilities

Provides the capability to return to authorised users the current version of resources, as well as previous versions.

Required to serve different versions of a document as it progresses through states.

### Request Authorising Capabilities

Provides the capability of securing resources so they are only accessible to authorised users. Consider using a pre-request Handler to return HTML error messages to unauthorised users.

Required to limit access to resources to authorised public or authenticated users.

### Permissions Management Capabilities

Provides the capability of managing System, Group and Resource management Permissions.

Note:  
Group Permissions include CanJoin, CanInvite, CanRemove, etc, while Resource Permissions include permissions to perform actions on Records & Resources, including: Draft, Collaborate, Submit, Comment, Approve, Publish, Merge, Remove, Restore, Archive, etc.

### System Messaging Capabilities

A system must be able to signal asynchronously to an end user. This is a base requirement for inviting and accepting users to roles in groups, for async processing of workflows, to signal planned downtimes of the service, etc.   
Not that even if signals can be sent to open browsers, SMTP based communication remains the required baseline.

### System Notification Capabilities

Provides the means to signal to users system-wide messages, such as that the system will be taken offline for a specific duration at a specific time.   
Can also leverage the System Messaging Capabilities to reach users that are currently offline.

### Search Capabilities

The menu, a staple of Windows, Icons, Menus, Pointer (WIMP) based systems has for the most part been replaced by spelling-forgiving & phonetic search that return a list of SearchItemSummary items to provide a more efficient manner of searching, discovering, and then navigating to the desired record of whatever type.

### Data Categorisation Capabilities

Provides the capabilities to tag records for improved discovery later.

### Data Classification Capabilities

Related by distinct form Data Categorisation, providing a means to tag data by data classification (UNCLASSIFIED, IN-CONFIDENCE, SENSITIVE, etc.

### User or Principal Management Capabilities

Provides the ability to manage People who are currently using the system.  
Required to manage current and past users of a system.   
Do not conflate User/Principal and Person – they are distinct. A User is usually a “thin” object (It just has an ID, and Enabled flag) that is linked to a Person record, as well as one or more Roles within a system Group.

### User Identity Management Capabilities

Provides the capability of associating 3rd party identities (e.g.: IdP issued identities) to internal system users.   
Used to manage one or more user logon identities authenticated by external systems.

### User System Profiles Management Capabilities

Provides the capabilities for a system user to configure their personal system preferences. Often referred to inaccurately as “user settings”.

### Queue Management Capabilities

Provides the capability of queuing operations to be processed asynchronously.  
Queues are the basis of providing availability at lower hardware cost. They are also the basis of avoiding batch processing which is an anti-pattern.

### Workflow Management Capabilities

Provides the capability of developing workflows to Invite Persons to become Users, to review documents before publishing, etc.

### Scheduling Capabilities

Provides the capability to schedule tasks.  
Enables the scheduling of reviews of expiring Roles.

### Session Management Capabilities

Provides the capability of starting sessions for anonymous users and [optionally] keeping or transitioning to a new session after they authenticate themselves.

### Session Operation Auditing Capabilities

Provides the capabilities of permanently recording operation requests and responses against the session that has been started for the user.   
This is an essential capability required to audit and investigate activity, and provide transparency as required.

### System Configuration Capabilities

Provides the capability of configuring system integrations (cache durations, potentially also integration credentials), system recognisability (logos, etc.).

### Account Configuration Capabilities

Provides the capability of managing configuration settings for different Accounts.

### Media & Media Metadata Management Capabilities

Users upload media (avatars, pictures, documents) to support profiles and records. Like anything that is submitted for storage for a service it requires first validation via a 3rd party media stream malware detection service, then storage, along with its metadata (mime type, dimensions, contents summary) to facilitate searching for later.

## Social Domain

Due in part to the tight relationship between User, Persons, Roles, the domain of People and Groups is often conflated with base System design. However, they remain a distinct logical domain:

### Person Management Capabilities

Provides the capability of managing people, whether they are non-users, have the potential of being users (e.g.: invited people) , are current users, or ex users (their role within the system has expired).  
Required to manage people who are not users (e.g., parents of a child) but could be invited to (e.g., to view a child’s progress report).

### Personal Management Capabilities

A person has one or more Personas. That they share with different Groups.

### Persona Relationship Management Capabilities

Personas have relationships with othere Personas.

### Group Type Management Capabilities

There are different types of Groups.

A Group of type System issues identiifers to its Members.

### Group Management Capabilities

Provides the capability of developing Nested and Parallel Groups (Organisations, Departments, Schools, Classrooms, Teaching or Interest groups). An organisation is nothing more than a Group that has an Identifier (e.g. a Business Number) between it and a Group representing a 3rd party system (in this case the external Company Registry). Note that Account – defined later - is also simply another *type* of Group.

### Group Role Management Capabilities

Personas invite other Personas to accept Roles within Groups.

Provides the capability of organising Groups of System Permissions (and optionally Resource Permissions) as Roles associated to Groups.

### Group Role Association Capabilities

Provides the capability of associating a Role within a Group to a Person.  
Note that there are long term functionality benefits to being capable of assigning roles to persons who are not yet system Users (it allows setting up Persons before they are invited, for one).   
Note that it is poor practice to Assign Roles. Best Practice is to Invite a Person to Accept the Responsibilities of a Role within a Group. Once accepted, the Role is assigned for a limited time (e.g., employment might be for 1 year) that triggers Reviews and Extensions in the future.

## Subscriptions, Service Plans, Features and Account Management Capabilities

Account management is a requirement to provide a service to more than one organisation.

An account is associated to a Group.

### Tenancy Management Capabilities.

Do NOT implement. Implement instead Account Management Capabilities.

### Account Management Capabilities

Accounts are a more correct approach than Tenancy Management which does not permit the sharing of records among partner organisations.  
Accounts a simply a type of Group.

### Services Plans & Feature Management Capabilities

Permits defining certain features belonging to specific Services which can be Subscribed to to by different Accounts managed by different system Users.

### Subscription Management Capabilities

Permits the management of the associations of Services to Accounts.

**Important:**The benefit of Account based approach over the older Tenancy based approach is the ability to share records across organisations and agencies – rather than requiring a consultant to be enrolled within an organisation’s directory service to access records they have a role in.

## Artefacts/Resources Domain

People come together to access or develop new Resources.

The resources can be database records, or records describing persisted media (text, images, documents).

### Resource Management Capabilities

Management of

### Resource Classification Management Capabilities

### Resource Categorisation Management Capabilities

### Resource Grouping Management Capabilities.

### Resource Groups are often referred to as digital folders.

## Aspirations Domain

A key part of personas is to define objectives.

### Objectives Management Capabilities

A set of nestable objectives.

## Work Domain

The work domain is…working towards Objectives and milestones…

### Task Type Management Capabilities

### Project and Task Management Capabilities

Tasks are assigned to Personas

### Task Completion Management Capabilities

## Extensions to Core Capabilities

While the above list of Capabilities is sufficient to meet most conditions, some other Capabilities could be added if needed to permit more flexibility and less work in next modules.

A few examples might include:

### Communication Channels Capabilities

This capability provides the means to persist information on how to communicate with Persons or Groups (e.g.: Organisation) via whatever channels they provide – from old school, multi-field Post (street, city, postal code, country), to more modern digital single field, two ways channels, both synchronous (Phone, Cell, etc.) and asynchronously (SMS, etc.) or one way asynchronous channels (twitter, etc. )

### Place Capabilities

While it is common to confuse Postal address with location, they are not the same, so recording a Group or Person’s location on a map can provide useful information that can’t be obtained from a postal box address.

One might reuse the base capabilities to make other foundational services, such as:

### User Onboarding Capabilities

which uses both the Notification Capabilities and Workflow Capabilities and Scheduling Capabilities to develop a workflow that permits users to inquire about being Invited to be Apply and be Accepted and Approved to a specific time limited Role within a Group of some kind. This kind of viral but verified onboarding is effective at removing onboarding bottlenecks that happen with centralised onboarding.

Listing every remaining permutation of combination and/or projection of base capabilities is beyond the scope of this summary document.

### Extension Business Modules

On top of the above core logical modules one can build most any time of business service module: Student, Schools, Teachers, Curriculum, Schedules, Resources, Tasks, Assessments, Achievements, Support, etc.

## Considerations

Even if a system boasts a complete suite of core capabilities, careful consideration remains essential when selecting or designing it. For instance, enterprise architects may mistakenly assume that a platform featuring both technology and people management capabilities will automatically meet all needs. This misconception often arises from an incomplete analysis of specific requirements.

Organizations typically manage distinct roles, such as employees and customers. To avoid conflicts of interest, these roles are frequently maintained in separate data structures, making it challenging for a single entity to assume multiple roles simultaneously—a scenario commonly seen in the education sector, where an individual might function as a student, parent, or teacher.

Moreover, the relationships among these roles differ significantly. In most organizations, employee relationships are hierarchical, with each person reporting to a single superior. In contrast, commercial applications treat customers as independent individuals without inherent familial ties. The education sector, however, is more complex; a child may have siblings, one or more parents, and even extended family members, like aunts and uncles, who contribute to caregiving.

These fundamental oversights during the analysis phase can lead to a system that, despite having a full set of capabilities, ultimately fails to support the intended business needs.

## Conclusion

A system’s long-term qualities—including maintainability and extensibility—benefit immensely from a thoughtfully designed architecture that employs Domain Driven Design principles. By organizing the system into distinct, interrelated logical modules, each providing specific capabilities, developers and procurement professionals can build a robust foundation that supports both current needs and future growth. However, even with a comprehensive technical framework, it is crucial to conduct a thorough analysis of the business context and requirements to ensure that the design remains truly fit for purpose. Balancing a strong technical foundation with a nuanced understanding of roles, relationships, and functional requirements ultimately determines the system’s success in meeting its intended objectives.

Appendices

Appendix A - Document Information

### Images

[Figure 1: Core and Extension Logical Modules 4](#_Toc191291445)

### Tables

**No table of figures entries found.**

### References

**There are no sources in the current document.**

### Review Distribution

The document was distributed for review as below:

|  |  |
| --- | --- |
| Identity | Notes |
| Russell Campbell, Project Manager |  |
| Amy Orr, Enterprise Architect |  |
| Gareth Philpott, Solution Architect |  |
| Vincent Wierdsma, Lead Developer |  |

### Audience

The document is technical in nature, but parts are expected to be read and/or validated by a non-technical audience.

### Structure

Where possible, the document structure is guided by either ISO-\* standards or best practice.

### Diagrams

Diagrams are developed for a wide audience. Unless specifically for a technical audience, where the use of industry standard diagram types (ArchiMate, UML, C4), is appropriate, diagrams are developed as simple “box & line” monochrome diagrams.

### Acronyms

Refer to the project’s Glossary.

ISO

: the [International Standards Organisation](#_International_Standards_Organisatio), to which NZ is a contributing member.

IT

: acronym for Information, using Technology to automate and facilitate its management.

ICT

: acronym for Information & Communication Technology, the domain of defining Information elements and using technology to automate their communication between entities. IT is a subset of ICT.

### Terms

International Standards Organisation

: organisation that manages international standards, to which most countries – including this one - belong and contribute to.

Logical [Module]

A logical module is a defined set of interrelated components or system parts only bound by commonality that collectively address a common cause or domain within an overall system architecture.

### Standards

The following are key ISO’s to consider when developing solutions:

ISO-25010

: data *system* qualities

ISO-25012

: system *data* qualities

ISO-25022

: system *user* experience qualities

Appendix B

In mature service delivery conditions, the Capability Requirements are developed prior to defining Functional Requirements, as per below:

