ICT Project Guidance

Discovery:  
Common System Capabilities

Version:

V1.3

Author:

Sky Sigal, Solution Architect

## Description

This document outlines standard capabilities expected of ICT systems, describing the value they can offer, to better inform procurement and design discussions and decisions.

## Synopsis

Irrespective of the business reason for an ICT system, mature systems share a relatively standard set of Capabilities to consider as the underlying capabilities required under use-case-specific capabilities.

## Contents

[Description 1](#_Toc147924920)

[Synopsis 1](#_Toc147924921)

[Contents 2](#_Toc147924922)

[Introduction 4](#_Toc147924923)

[Background 4](#_Toc147924924)

[Objective 4](#_Toc147924925)

[Issue 4](#_Toc147924926)

[Resolution 4](#_Toc147924927)

[Capabilities 5](#_Toc147924928)

[System Capabilities 5](#_Toc147924929)

[ System Diagnostic Logging Capabilities 6](#_Toc147924930)

[ Error Recording & Reporting Capabilities 6](#_Toc147924931)

[ System Integration Configuration and System Settings Capabilities 6](#_Toc147924932)

[ System Monitoring Capabilities 6](#_Toc147924933)

[ System Notification Capabilities 6](#_Toc147924934)

[User Capabilities 6](#_Toc147924935)

[ Individual Session and Session Operation Management Capabilities 6](#_Toc147924936)

[ System User Management Capabilities 6](#_Toc147924937)

[ System User Preference Management Capabilities 6](#_Toc147924938)

[ System User Group Management Capabilities 7](#_Toc147924940)

[ System User Group Role Management Capabilities 7](#_Toc147924941)

[ User Onboarding and Offboarding Capabilities 7](#_Toc147924942)

[Module Capabilities 7](#_Toc147924943)

[ Module Management Capabilities 9](#_Toc147924944)

[Reporting Capabilities 9](#_Toc147924945)

[ Report Presentation Capabilities 9](#_Toc147924946)

[Resource Management Capabilities 10](#_Toc147924947)

[Resource Management Capabilities (Basic) 10](#_Toc147924948)

[ Simple BREAD 10](#_Toc147924949)

[Resource Media Capabilities 10](#_Toc147924950)

[ Media Management Capabilities 10](#_Toc147924951)

[Resource Discovery Capabilities 10](#_Toc147924952)

[ Resource Metadata Management Capabilities 11](#_Toc147924953)

[ Resource Discovery Capabilities 11](#_Toc147924954)

[ Resource Linking (& Link Metadata) Capabilities 12](#_Toc147924955)

[Resource Collaboration & Lifespan Management Capabilities 12](#_Toc147924956)

[ Resource specific Role Management Capabilities 12](#_Toc147924957)

[ Resource Versioning and Routing Management Capabilities 12](#_Toc147924958)

[ Resource Workflow Management Capabilities 12](#_Toc147924959)

[Information Resource Generation 12](#_Toc147924960)

[Resource Integration Capabilities 13](#_Toc147924961)

[ Audited Application Programming Interfaces (API)s for Authenticated and Authorised Access 13](#_Toc147924962)

[Resource Interfacing Capabilities 13](#_Toc147924963)

[Basic Resource Viewing capabilities 14](#_Toc147924964)

[ Context Description Capabilities 14](#_Toc147924965)

[ Context Searching and Navigation Capabilities 14](#_Toc147924966)

[ System and User Status Description Capabilities 14](#_Toc147924967)

[Accessible Resource Viewing Capabilities 14](#_Toc147924968)

[ Visually and motor impaired Viewing Capabilities 14](#_Toc147924969)

[ Varied Device Display Capabilities 15](#_Toc147924970)

[Device Integrating Capabilities 15](#_Toc147924971)

[ Device Service Integration Capabilities 15](#_Toc147924972)

[Advanced Resource Viewing & Development Capabilities 15](#_Toc147924973)

[ Charting Capabilities 15](#_Toc147924974)

[ Colour and Texturing Capabilities 15](#_Toc147924975)

[ Text Placement Adjustment Capabilities 15](#_Toc147924976)

[ Layering and Layer Ordering and Dis/Enabling Capabilities 15](#_Toc147924977)

[ 2D cartographic Capabilities 15](#_Toc147924978)

[ 3D Displaying Capabilities 15](#_Toc147924979)

[ Timeline based displaying Capabilities 15](#_Toc147924980)

[Automation Selection and Display Capabilities 16](#_Toc147924981)

[Conclusion 16](#_Toc147924982)

[Appendices 17](#_Toc147924983)

[Appendix A - Document Information 17](#_Toc147924984)

[Images 17](#_Toc147924985)

[Tables 17](#_Toc147924986)

[References 17](#_Toc147924987)

[Review Distribution 17](#_Toc147924988)

[Audience 18](#_Toc147924989)

[Structure 18](#_Toc147924990)

[Diagrams 18](#_Toc147924991)

[Terms 18](#_Toc147924992)

# Introduction

Statistically, most ICT system projects fail to deliver to expectations. The reasons are many. A key is the lack of understanding of what capabilities are required of the system, which in turn leads to mistakes being made in the development of requirements, and therefore lead to poor procurement or custom design processes.

## Background

System Capabilities provide an overview of what a software system is *capable of*, while System Functionalities expand on this, and describe the specific functions are provided to enable actions that contribute to those Capabilities. Both capabilities and functionalities are important considerations when designing, developing, and using software systems – but it is important to first understand required Capabilities.

## Objective

The objective is to procure or develop services that have the Capabilities required to be of value to stakeholders and users of the system.

## Issue

A majority of ICT services have some but not all the Capabilities required to deliver the service effectively. Additionally, depending on whether purchased or developed (and if so, how it was developed), missing capabilities may be difficult or impossible to add in later without adding significant risk.

## Resolution

To resolve this omission, it is important to have available a checklist of common capabilities expected of most systems to better inform the development of project specific requirements developed to guide procurement and/or development of the service.

# Capabilities

Irrespective of the business use of systems, mature systems share a relatively common set of Systems Capabilities can be outlined using a pyramid layering, each tier depending on the tier below.

At the bottom are capabilities that may not be visible to end users but are foundational for developing a next tier of capabilities that in turn provide culminate to a solid base for providing capabilities that end users expect of systems.

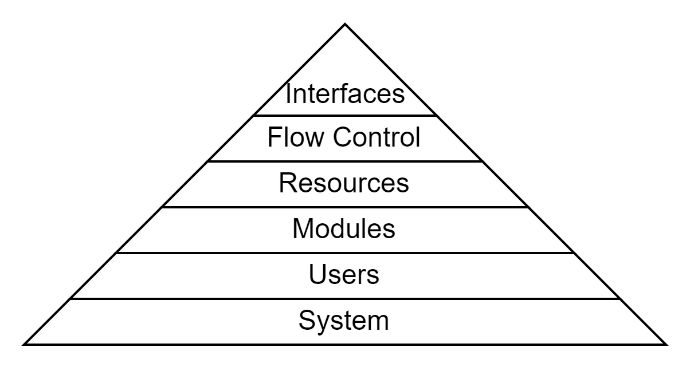


Figure : Typical Organisation of Capabilities

## System Capabilities

Systems are expected to have common capabilities required to manage the system itself, irrespective of the Users who will access the system or the Resources they will work with.

While not accessed by service consumers, these capabilities are required by Maintenance stakeholders to keep the system available and performant.

The capabilities in this tier include the following:

#### System Diagnostic Logging Capabilities

, to better understand why the system is behaving the way it is,

#### Error Recording & Reporting Capabilities

, to record unexpected behaviour for later remediation and service improvement,

#### System Integration Configuration and System Settings Capabilities

, to customise the integration with underlying infrastructure and service provider’s environment,

#### System Monitoring Capabilities

, to measure and inform the prioritisation of work to iteratively improve the service’s availability, performance, resource consumption[[1]](#footnote-2),

#### System Notification Capabilities[[2]](#footnote-3)

to notify responsible parties about undesired qualities of the system. This almost always involves using email at least.

## User Capabilities

A system that lets users interact with the system needs to have the following common capabilities to manage them:

#### Individual Session and Session Operation Management Capabilities

, to manage individual threads of activity, recording whom changed what for later use as required (i.e., generally for both security and auditability reasons),

Note that at this early stage, without further work, sessions can be tracked individually, but only anonymously and not associated to actual Users.

#### System User Management Capabilities

, to enable service consumers to accept terms & conditions and begin using the system (by connecting them with 3rd party digital identities[[3]](#footnote-4), etc.).

At this point, Sessions can now be associated to actual authenticated Users.

#### System User Preference Management Capabilities

, for enabling users to configure their personal workspace for accessibility and efficiency reasons.

At this point, each user can set and forget their preferred interface language, appearance, layout.

#### 

#### System User Group Management Capabilities

, to organise users into groups,

At this point, users can be organised into groups to reflect organisation structure (departments, classrooms, practices and projects, etc), with every person having equal roles.

#### System User Group Role Management Capabilities

, to provide collaborating users with Permissions specific to the Group, as opposed to the whole system.

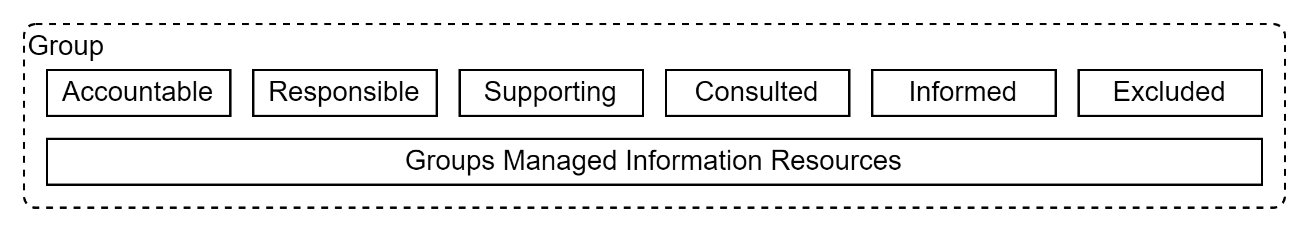


Figure : Group Resources and Roles

At this point, Users can be organised into Roles within the Groups, that reflect RASCIE expectations (Accountable, Responsible, Supporting, Consulted, Informed, Excluded). Roles given in a parent Group trickle down to Groups under it.

Often forgotten is the process of getting users on to the system, hence the system should also include:

#### User Onboarding and Offboarding Capabilities

, for users by either importing a list into a screen or API endpoint, or – more preferably – a mature email based notified Application/Invitation/Acceptance/Expiration process.

At this point, a manager of a group (e.g., school, department, classroom, etc.) can invite another person to become a supporting and contributing member. An external person can also Apply to be invited as well. With very little extra effort, any person can invite another and Approval from a person of authority can be given before or after their Acceptance.

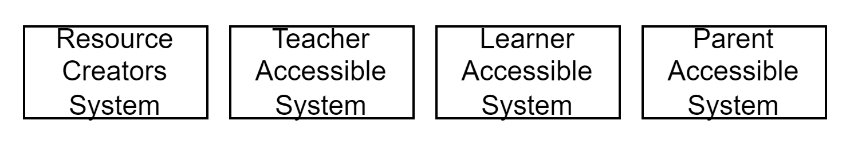
## Module Capabilities

Most enterprise systems will be envisioned and developed to deliver a single service, in a single business domain, catering to a single type of user base. Therefore, Module management capabilities is not often a requirement.

But some systems *are* expected to be used by vastly different types of users.

While it is possible to provide visibility to all parts to all users, moderated by rights, this can be a cluttered experience for most users. This is especially true if the system is meant to be used by both very young people (e.g.: primary school learners) as well as more educated people (e.g., teachers).

An approach is to consider multiple systems aimed at different stakeholders and integrating them together.



Warning:  
The cost of developing Integrations between different systems, is often underestimated for multiple reasons, including focusing primarily on the flows of business information, while omitting short and long term planning for other required supporting aspects: integration of diagnostics, monitoring, permissions, roles, security, accountability, user and role provisioning.

A more integrated approach is to develop different distinct websites, to share common backend logic and services.

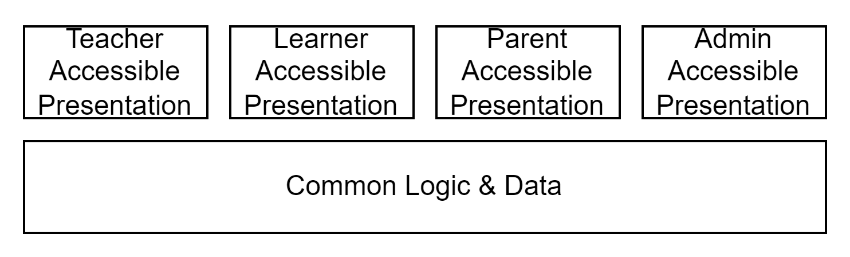


Figure : Distinct Presentation Web Sites

Warning:  
A risk with this approach is that the common logic and data, in trying to meet the needs of multiple different contexts, becomes convoluted, difficult to change without impacts, and hence resists needed improvement.

A higher value approach, one that requires less entanglement as well as duplication of development, quality assurance, and testing, and fragmentation of externally visible designs, is the development of distinct modules within the same system, sharing common back and front-end capabilities.

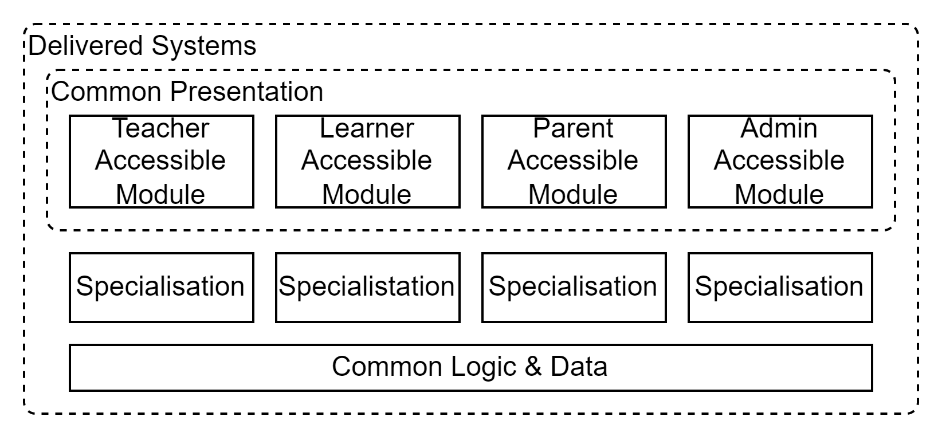


Figure : Common Approach to Distinct Presentation Modules

An example might be a quiz development and taking service that may require most of the following distinct modules:

* A test Item development and calibration module
* A test assigning module, for teachers to design a test from the items, then invite learners to take the test,
* A test sitting module, for learners to use to go through the test
* A test reporting module, used by teachers and school admins.

To manage access to modules by different users, a system needs:

#### Module Management Capabilities

, to control access to different groups of users, presenting to each a focused and optimised grouping of functionality.

## Reporting Capabilities

Most systems are developed to provide information back to users via digital Graphical User Interfaces (GUI) views (“screens”).

But developing printable reports is an important capability expected by stakeholders who are not users. This list may include sponsors, governance, and delivery managers on both the service provision and service consummation side (e.g., school admins, teachers, etc.).

#### Report Presentation Capabilities

, to develop printable Excel workbooks, CSVs, PDFs and reports[[4]](#footnote-5) that can be formatted and paged for accessible and efficient understandability and use.

At this point, a project sponsor can be provided a report that charts change to the number of users over time, what are their most preferred activities, the number of errors, issues that occurred – all usable to inform investment and work decisions.

*Important:  
without careful consideration and controls in place printable Reports can easily become a security risk as they can permit viewing printed confidential information viewable outside of the monitored and audited system’s control.   
Permitting custom development of reports amplifies this risk.*

Note:  
Depending on the use case, the system may benefit from the use of a separate reporting database, so that reporting operations do not impact the responsiveness of the system to other users.

## Resource Management Capabilities

Systems that manage resources – most systems –require Resource Management Capabilities.

The Capabilities required depend on what rigour is needed around the creation and release of new resources, and the ease and efficiency desired to find these resources later.

### Resource Management Capabilities (Basic)

Most systems being internal to an organisation provide sufficient access to information with just basic capabilities:

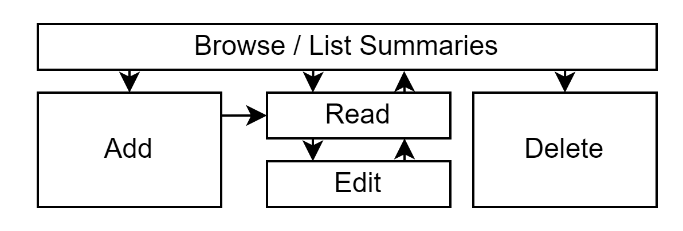


Figure : Basic BREAD UI View Flows

* Simple BREAD**[[5]](#footnote-6)** based Resource Management Capabilities for each resource type the system manages. This provides for listing of records[[6]](#footnote-7), to select a record to view, edit or delete – or add a new one.

At this point, users have the basics to input new resources and find them again.

### Resource Media Capabilities

#### Media Management Capabilities

[[7]](#footnote-8) : to permit Users to upload media (images, video, sounds, etc.) that is malware checked, to build information resource.

At this point, the system has the basics for end users to develop engaging information resources.

### Resource Discovery Capabilities

While basic BREAD functionality is a core requirement for managing records, more advanced capabilities are expected by users – especially service consumers outside of an organisation -- to find information easily and efficiently.

Better discoverability of resources relies on

* Resource Metadata Management Capabilities, to permit tagging and categorisation of resources by ad hoc labels or agreed Code Sets[[8]](#footnote-9).

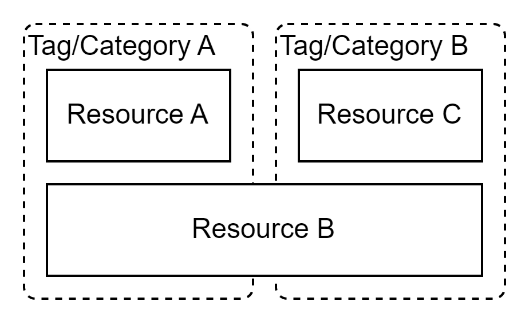


Figure : Resource Collection & Tag based Categorisation

* Resource Discovery Capabilities, to permit Users have a more forgiving Search experience, one that permits misspellings and even phonetic spelling.

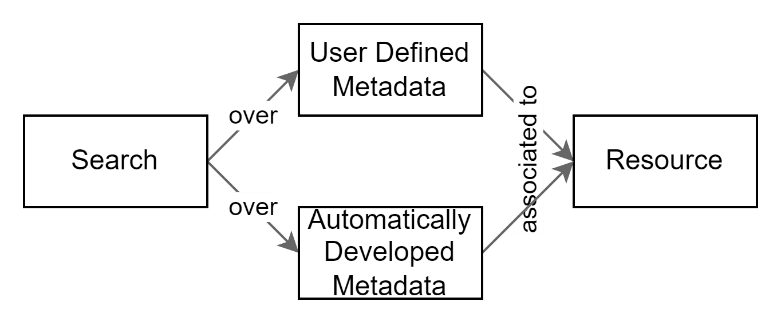


Figure : Resource Discovery over Metadata instead of Resource itself

At this point, Users can retrieve information just by searching, much like using a web search engine such as Google.

More advanced searching still can be offered if the following capability is also offered by the system.

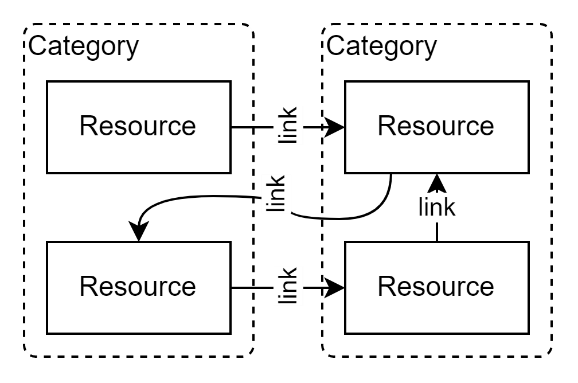


Figure : Linking and creating paths across Categories.

* Resource Linking (& Link Metadata) Capabilities, permitting categorisation by synonym, antonym, relationship, context, etc.

### Resource Collaboration & Lifespan Management Capabilities

Information that requires audited collaboration by multiple people, and full lifespan management of resources over a long duration, requires the following capabilities:

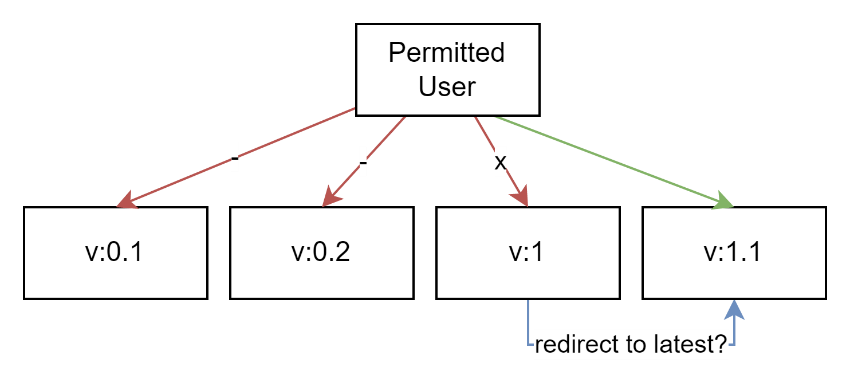
* Resource specific Role Management Capabilities (as opposed to only all-of-Group or all-of-System) permits creating roles specific to single resources, such as Creator, Collaborator, Approver, Maintainer, Commentor roles.
* Resource Versioning and Routing Management Capabilities, permits replacing Resources with new versions, while retaining previous versions for accountability reasons and the ability to determine the changes made between versions. Public government documents are examples of resources that are never deleted, but older copies of resources redirect users to the latest version of the resource.  
   

Figure : Resource Versioning, and associated Routing Capabilities

* Resource Workflow Management Capabilities (for orchestrating resources through various states such as: Created, Draft, For Review, Rejected/Approved, Released, Updated/Replaced, Merged, Removed, Restored).

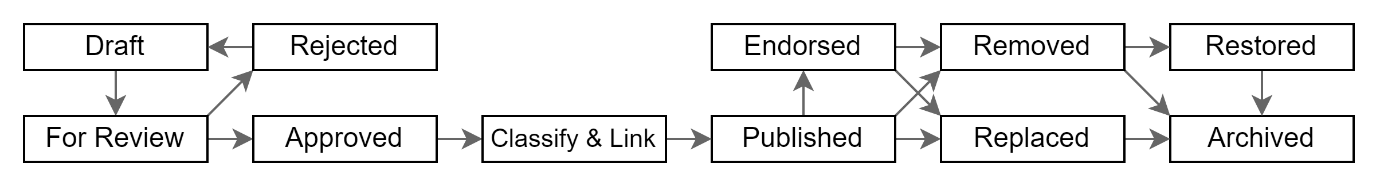


Figure : Indicative Information Resource Workflow

At this point, Users have the tools in place for a mature, auditable, handling of Resources through a multi-year life cycle, permitting collaboration before the resource is published and taking on feedback by consumers on how it can be improved.

### Information Resource Generation

As per described earlier, Information Resources can be generated in volume by automation. An example may be the Automated Item Generation (AIG) of assessment items and possible answers.

In such cases the work may moves from concentrating on managing the development of the resources themselves, to managing the development of the *criteria* metadata used to control how the resources are generated by automation:

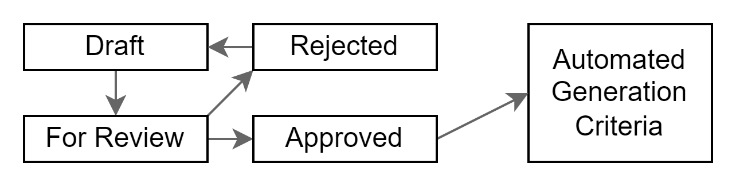


Figure : Managing the Criteria used to automate resource development

Furthermore, where trust in the qualities of the generated results require being established further, appropriate roles can be kept “in the loop” to process information through the name workflows as would have been used if they were developed without automation.



Figure : Qualifying resources developed by automation

## Interface Capabilities

While the is a natural focus on the subject of user interfaces, there are actually several types of interfaces:

* Graphical User Interface(GUI) (usually referred to as the “User Interface (UI)”)
* Reporting Interface (RI)
* Application Programming Interface (API).

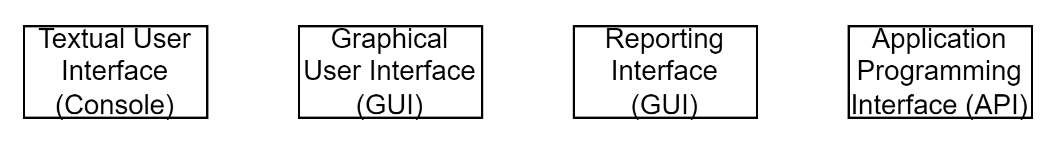


Figure : Common Interface Types

### Application Programming Interface (API)

Under the User Interface (if following best practice of following API-First design principles) is an Application Programming Interface.

The API will support two distinct subtypes of APIs, described next:

* Integration APIs
* User Interface backing APIs.

#### Integration API

Systems that require information to be exportable to other systems will require:

* Audited Application Programming Interfaces (API)s for Authenticated and Authorised Access by 3rd parties, preferably using Standards based protocols.

This capability is not used by system users directly but is the foundation for enabling partners to participate from their own systems. It also contributes to addressing business continuity risks, by allowing resources and data to be exported and thereby avoiding vendor lock-in.

**Important:**Systems that manage confidential and personal information (i.e., most systems) will require careful consideration as to permissions required to provide information over the wire, and/or whether removal of sensitive information is required before transmitting it outside of the system, where access to it can no longer be audited.

#### User Interface Backing API

These are APIs developed to support user interfaces, discussed further down.

User Interface backing APIs differ from integration APIs in that they are generally limited to views of a single user within a single tenant to ensure the user interfaces don’t divulge information unintentionally through programming error.

### Report Based Interfaces (RI)

A system’s reports are a paper-based form of interface discussed earlier under *Reporting Capabilities*.

### Text-based User Interface (TUI)

Text based interfaces -- generally referred to as a “Console” – are far less common in end user systems.

### Graphical User Interfaces (GUI)

While a need for individual interfaces Views was implied earlier when discussing the system’s other capabilities, there are common concerns of User Interfaces that are required in their own right. These capabilities contribute to the system’s accessibility, usability, learnability, etc. qualities[[9]](#footnote-10).

#### Basic Resource Viewing capabilities

* Context Description Capabilities to make it easy for users get their bearings. This is generally about making the top part of screens (the “header”) show information as to which Organisation is offering the service[[10]](#footnote-11), possibly along current Session’s Tenancy/Account, Group and User information (e.g., “you are currently signed in as Joe B., within the context of the Maths Department of Hamilton East School”) and Where the user has navigated to in the program (You are about to start taking a new Maths Test”).

At this point, users (e.g., a support teacher) can be members of multiple organisations and schools, flipping between them to do different work, while always being informed as to which context they are currently working in.

* Context Searching and Navigation Capabilities are the capabilities required for users to choose a target destination area of the system, optionally adding that information to the Context Description information (e.g., via “navigation breadcrumbs”).

At this point, users on mobile, tablets and other small screen devices can avoid using space and time-consuming menus for navigation and use of an easier and lower bar of entry approach.

* System and User Status Description Capabilities keep users notified to the status of the system (e.g., the status will be unavailable on Saturday 9-11 UTC”), processes (e.g., “Processing…”, and “Done”), and Outcomes (“Your overall score is, your level is”, etc.).

#### Accessible Resource Viewing Capabilities

Making systems accessible to users is the responsible thing to do – and directed by regulation -- for Government services.

* Visually and motor impaired Viewing Capabilities to be accessible ensuring that a wider audience can use the system.

Important:   
government offered digital services are required by both strategic direction and regulation to meet Web Content Accessibility Guidelines (WCAG) targets.

* Varied Device Display Capabilities: to be accessible by a wider age range of users on a wider range of current devices (mobiles, tablets, watches), reducing the dependency on the cost of provisioning specific hardware for end users (e.g.: young learners in a school).

Note:  
At this point, the system is more accessible for a current younger aged workforce and learners, in field, often closer to service consumers where better feedback is collected from.

#### Device Integrating Capabilities

* Device Service Integration Capabilities: it may be important to the service to access mobile devices information. For example, the GPS Location may be a requirement on a nearest shopping suggestions displaying service.

Note:  
at this point, the system can be permitted to collect information by other means than typing, permitting more efficient integration of information, and ensuing decisions.

#### Advanced Resource Viewing & Development Capabilities

Some information is better presented in a visual manner as opposed to only using textual representations.

While listing a full set of such Capabilities is beyond the scope of this document, the following are a small sample of possibilities:

* Charting Capabilities, the capability of converting sets of information into charts,
* Colour and Texturing Capabilities, the ability to change the colour of charts and models,
* Text Placement Adjustment Capabilities, the ability to adjust labels on output,
* Layering and Layer Ordering and Dis/Enabling Capabilities, the ability to select and stack charts, models, diagrams,
* 2D cartographic Capabilities**,** the ability to display geographics information, potentially with overlaid information[[11]](#footnote-12),
* 3D Displaying Capabilities, the ability to display 3d models,
* Timeline based displaying Capabilities, the ability to show animations of evolutions of state,
* Etc.

Note:  
the above are capabilities take considerable effort to develop, therefore their development would require justification to differentiate a service from competing services.

#### Automation Selection and Display Capabilities

While most systems leave it to users to select what they wish to view, there are many cases where the selection process of what to present is controlled by the system. Examples include but are not limited to:

* screen backgrounds, selected at random, but keeping track of what was shown before to reduce the chance of showing previously seen material,
* advertisements presented by algorithmic processing of information about other information viewed by the user,
* self-study quizzes and training, algorithmically selected from calibrated collections of item banks, ensuring that items are for a specific skillset, evenly spread across the full range of aspects of the subject, etc.

## Conclusion

Given time, a mature and managed system, whether arrived at by a waterfall or agile delivery process, and irrespective of the different Resources it manages, or ways they are presented, will develop most of the following capabilities:

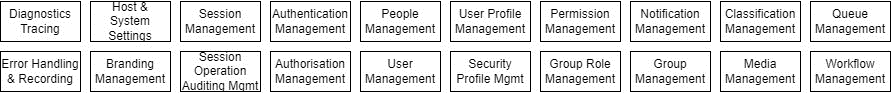


Figure : Common Mature System Capabilities

Depending on prioritisation and planning, they will be developed and delivered in approximately the following order:

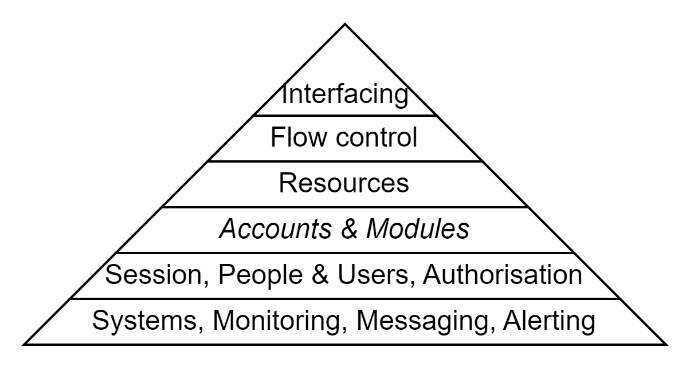


Figure : Detailed Pyramid of Capabilities

Appendices

Appendix A - Document Information

### Images

[Figure 1: Typical Organisation of Capabilities 5](#_Toc147925471)

[Figure 2: Group Resources and Roles 7](#_Toc147925472)

[Figure 3: Distinct Presentation Web Sites 8](#_Toc147925473)

[Figure 4: Common Approach to Distinct Presentation Modules 8](#_Toc147925474)

[Figure 5: Basic BREAD UI View Flows 10](#_Toc147925475)

[Figure 6: Resource Collection & Tag based Categorisation 11](#_Toc147925476)

[Figure 7: Resource Discovery over Metadata instead of Resource itself 11](#_Toc147925477)

[Figure 8: Linking and creating paths across Categories. 11](#_Toc147925478)

[Figure 9: Resource Versioning, and associated Routing Capabilities 12](#_Toc147925479)

[Figure 10: Indicative Information Resource Workflow 12](#_Toc147925480)

[Figure 11: Managing the Criteria used to automate resource development 13](#_Toc147925481)

[Figure 12: Qualifying resources developed by automation 13](#_Toc147925482)

[Figure 13: Common Mature System Capabilities 16](#_Toc147925483)

[Figure 14: Detailed Pyramid of Capabilities 16](#_Toc147925484)

### Tables

### References

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

### Review Distribution

The document was distributed for review as below:

|  |  |
| --- | --- |
| Identity | Notes |
| Sandy Britain, Enterprise Architect |  |
| Amy Orr, Data Domain Architect |  |
| Rodney Snell, Technical Lead |  |
| Josh Xie, Business Analyst |  |

### 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.

### Terms

##### API

: acronym for Application Programming Interface -- one of the 3 types of common interfaces (UI, API, Reports), it is the interface that permits machine to machine interactions without requiring an end user to broker information flows between two systems.

##### BREAD

: an ICT domain acronym for the development of User Interface Screens/Views to interact with data. It stands for:

* Browse: a View to list *summaries* of records, selected by one or more filter criteria, ordered in an ascending or descending order of one or more attributes, organised into pages, with the ability to Add a new record, or select a single summary row to in see the full record in a Read View:
* Read: a View to view a single record, with the ability to return to the previous summary list View, or proceed to either Editing the record, or Deleting it.
* Edit: a View that permits updating the Record’s fields.
* Delete: an action to remove the record from further User viewing or access, logically and not physically[[12]](#footnote-13).
* Add: reuses the Edit View to create new records and add it to the system.

##### Capability

: a logical grouping of what a software system is capable of doing or delivering.

##### CRUD

: a well-known ICT domain acronym for describing the operations that databases apply to data. Create, Read, Update, [logically] Delete. It is our recommendation to prefer the BREAD acronym, as it includes a Listing (Browse) activity not referred to in CRUD.

Delete: an operation that can be done to database records either “physically”, or “logically”, via a flag of some form to indicate the record is Removed from further discovery and display. The advantage of logical deletes is that deletions done in error can be reversed. There are no advantages to physical deletes other than space storage.

##### Functionality

: describe the specific actions (usually closely aligned to BREAD view actions) that contribute to those Capabilities.

##### ICT: the

acronym for “Information, Communication Technology”, the field of system engineering for managing and transferring information using digital technology.   
*Tip: the term indicates a certain hierarchy of importance by putting “Information” in the fore, and “Technology” at the end.*

##### *ISO-25010,*25012*,25022*

: the primary International Standards Organisation defined list of qualities of a system, its data and the experience user have of using the system.

##### Qualities

: a description of the different aspects expected to be experienced when using a system (see ISO-25022), partly based on the qualities of the information provided by the system (see ISO-25012), and qualities of the underlying system itself (see ISO-25010).

Refer to a project Glossary for further terms and acronyms.

1. See ISO-25010 for the list of qualities a system should have and iteratively improve on [↑](#footnote-ref-2)
2. To notify someone about abnormal behaviour [↑](#footnote-ref-3)
3. An important implementation tip to build in flexibility is to solve for Persons, some of which can be Users. [↑](#footnote-ref-4)
4. Often developed form lists and summarised as charts. [↑](#footnote-ref-5)
5. Browse, Read, Edit, Add, Delete (BREAD) [↑](#footnote-ref-6)
6. possibly even with some basic form of filtering, ordering and paging capabilities. [↑](#footnote-ref-7)
7. Note that the addition of Media Upload Capabilities will be needed surprisingly early if users are expected to be able to modify their avatar image via the earlier described User Preferences Capabilities. [↑](#footnote-ref-8)
8. Code Sets are sector wide (versus a single organisation) agreed reference data usable for categorisation. [↑](#footnote-ref-9)
9. Refer to ISO-25010 for the list of qualities a system should have. [↑](#footnote-ref-10)
10. using Branding Capabilities [↑](#footnote-ref-11)
11. An example might be something like QGIS. [↑](#footnote-ref-12)
12. Logically deleting records, rather than permanently deleting them improves the error forgiving qualities of the system, by permitting undoing deletes by mistake – this in turn improves user enjoyment of the system while decreases support system support costs. [↑](#footnote-ref-13)