IT Project Design

Solution Architecture Description Development Guidance

# Document

## Description

Title: {title}

Author: {author}

Date: {date}

## Distribution

Distribution/Review List:

* {name}, {role}, {org}

References

* “Default Project - Baseline ICT Terms & Acronyms Glossary”, latest version.

## Diagrams Index

[Figure 1 Traditional Architecture Model Plans, Elevations, Sections 7](#_Toc141176196)

[Figure 2: ISO-42010 Components 7](#_Toc141176197)

## Tables Index

[Table 1: Catalog of recommended ISO-42010 Views 8](#_Toc141176203)

# Purpose

The purpose of a SAD is to diminish the risk of non-delivery of an expected business service.

Risk is reduced by making available a common resource for domain specialists to review the plans of systems and their delivery to raise concerns before work commences. Early detection of issues avoids later re-litigation or mistakes from lack of forethought and resolution of interdependencies.

Risk is further reduced by maintaining the common resource up to date to provide different stakeholder groups current trustable plans and assembly and/or delivery instructions.

# Dependencies

## Stakeholder Map

The development of a SAD describes how the system has the capabilities expected by a wide range of stakeholders.

A Stakeholder Map is the artefact used to demonstrate that wide set of stakeholders that were considered.

## User Map

A SAD’s functional View describes how the system provides the functionality expected by a wide range of users.

A User Map is the artefact used to capture the wide range of users that use a system over its operational service lifecycle. It will include at the very least:

* Business Service Consumers: the intended recipients of the service. End Users are categorised into Roles that best describe their role in their own organisation (Principal, Admin, Accountant, Teacher, etc.)
* General Support Specialists: provide general assistance to Service Consumers that is not specific to Business functions. This may include helping with provisioning new user groups, authenticating users, etc.
* Business Support Specialists: providing business assistance to Service Consumers. This may include onboarding groups of new users, assigning them with initial roles, setting up preferences, etc.
* Operations Specialists: providing support to Business Support Specialists.
* Training Support Specialists: providing Business Service Consumers, General Support Specialists, Business Support Specialists and Operations Specialists introductions to service capabilities, and knowledge on how to use the service’s provided functionality.
* Etc.

## System Requirements

Stakeholder Analysts use the Stakeholder and User Map to guide the collection of Desires, which they in turn use to develop System Requirements.

System Requirements is not a deliverable in its own right, but simply a logical grouping of two different deliverables – the system’s Quality Requirements and the system’s Functional Requirements.

### Qualities Requirements

A System’s Quality Requirements describe how a system is expected to operate – regardless of its Functionality.

A set of subjects are provided in ISO-25010, ISO-25012, ISO-25022’s to guide the development of a set of Requirements that do not omit important aspects.

It is common but not obligatory that a SAD includes a matrix demonstrating how the desired Qualities are met by system design. The Qualities Matrix may be presented either as part of dedicated Qualities View or relegated to an Appendices.

### Functional Requirements

Functional requirements are developed by stakeholder analysts querying the stakeholders listed in a Stakeholder and User Map.

The design of a system described by a SAD should attempt to provide functionality that meets most of these Functional Requirements.

It is common but not obligatory that the SAD tracks via a matrix which Requirements are met by which designed Functionality.

## Transitional Requirements

In addition to referencing System Requirements, Architects should reference Transitional Requirements.

Transitional Requirements outline requirements for transitioning between states – generally ‘now’ and ‘release’.

Transitional Requirements will include the development of comms, conditions (T&Cs, Privacy & Tracking Statements, expected Comportment/behaviour, processes (onboarding, supporting, offboarding), system initialising & configuration, provisioning (reference data, users), etc.

An architect developing a design should design a solution that provides a means of meeting these transitional requirements.

# Platform

SADs must be accessible by both internal business and delivery stakeholders, but also 3rd party consultants and collaborators without their needing to be members of an organisation.   
As most project are tendered for proposals, SADs must also be accessible by unauthenticated RFP respondents.

SADs must be versionable for auditing and accountability reasons, to determine who changed what and what has been presented and accepted by governance steering groups.

SADs benefit from being improvable via Comments being able to be added.

For the above 3 reasons, the development of SADs has traditionally been within Word documents.   
Use of a different platform is acceptable if it both provides additional benefits and meets a similar level of availability, improvability and auditability.

Diagrams

There is no universally accepted diagramming convention. Options include

* + - * Simple arrowed “Boxes and Lines” diagrams
      * UML (ISO-19505-2:2012)
      * Entity Relationship Diagrams (ERD)
      * Archimate
      * C4

While there is no clear advantage or accepted domint style, it is our recommendation to using simple plain boxes and lines to communicate to the widest set of audiences wherever possible, optionally *supplementing* them in specific views with industry specific diagrams, depending on the purpose:

* + Archimate in Business Context Views,
  + Entity Relationship Diagrams (ERD) in Information Views,
  + UML Use Case diagrams in Functional Views,
  + UML Communication diagrams in Integration Views
  + UML Sequence or Activity Diagrams in Sequence Views,
  + UML Deployment diagrams in Deployment Views
  + UML Class diagrams in Development Views

Note: C4 can also be used.

# Structure

## Background

The structure of Solution Architect Descriptions have evolved over time from “4+1” – the dominant structure in the 90’s -- to the one advocated in ISO-42010:2011, popularised by Rozanski & Woods seminal book.

### ISO-42010

[](http://skysigal.com/_media/resources/images/00/sad/views.png)

Figure Traditional Architecture Model Plans, Elevations, Sections

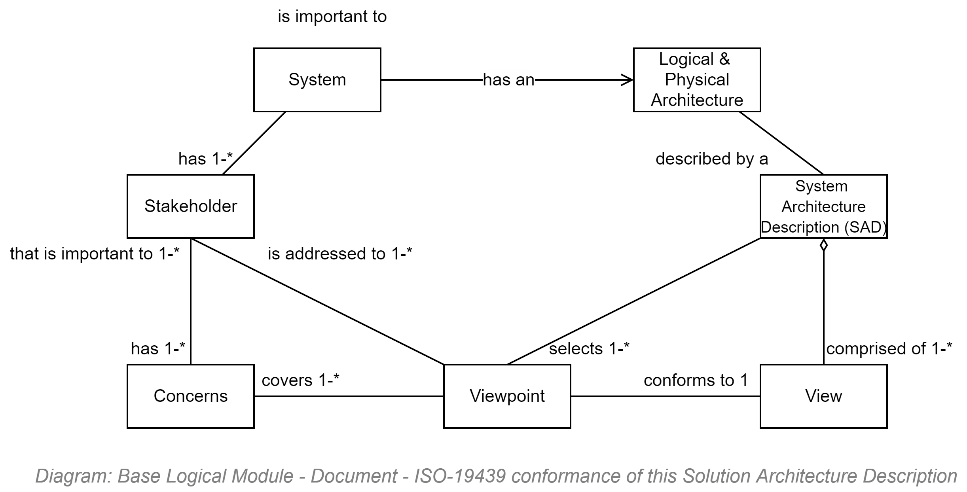
ISO:42010:2011 specifies how complex systems can be described using a a series of interrelated Views (sections) viewed from the perspective of different Stakeholders groups.   


Figure : ISO-42010 Components

The standard does not proscribe what the Views should be, and leaves it up to the architect.

### Rozanski and Woods Recommended Views

To address ISO-42010’s gap (of not proscribing a set of View titles to use), Rozanski and Woods published a seminal book describing how to develop SADs in adherence to ISO-42010.

## Sections

While R&W improved on the sparse guidance provided in ISO-42010, it too had unclarity.

The following is a list of proposed Views to consider and select from when developing SADs.

### Context View

: aimed for validation by Business Sponsors and consumption by all stakeholders. the Context View summarises key aspects of the project.   
It is our recommendation to outline the same aspects as would be covered in a BOSSCARD summary. Specifically, it should outline the Background, Problems, Constraints, Opportunities, Assumptions, Stakeholders, Options, selected Option, its Risks, Scope, and expected Deliverables.  
Note that Constraints include Regulations, Policies and Agreements. The Regulations include the applicable laws, for example, and Agreements include both national ones (e.g.: UN) and project-specific Guiding Principles, stating that they align with the Enterprise collection of guiding principles.

### Delivery View

: aimed at Project Managers tasked with organising and prioritising delivery of all artefacts required to deliver an automated business service.   
It outlines the tools used to automate the collection, coordination, prioritisation of user voice to work items to delivered Deliverables expected by different stakeholders. This will include the system of course, but also People, Roles, Decision, Risk and Issue Registries, Development Guidelines, Certificates, Organisation website content media, Deployment Pipelines and their maintenance instructions, Application Support Guides (ASGs) for Support Specialists, Operations manuals, System Media in various languages, etc. Describing the Change Management is a key aspect of this View.

### Qualities View

: describing the ISO-25010, 25012 and 25022 Qualities expected of the service, irrespective of the actual functionality described in the Functional View, and how they are addressed.

### Privacy View

: describing how the system’s information meets privacy expectations

### Security View

* : describing how the system meets and exceeds security (in transit, at rest) and accessibility and auditability expectations.

### Capabilities View

* : describing the capabilities expected of the service. Capabilities are delivered by specific functionality, described in further detail in the *Functional View*.

Describe the capabilities in tiers aimed at different stakeholders. For example, describe business capabilities first, indicating that they in turn rely on underlying generic information service, which is described in a subsequent section.

*Note that most systems will include most of the following capabilities -- Diagnostics, Error recording, Configuration, Settings, Session, Discovery, Routing, Authentication, Authorisation, Search, Versioned Agreements (T&C) Management – on which specific Sector or Business capabilities can be built upon.*

### Integration Information View

: describing the logical elements that move in and out of the system as API messages. Again, describe the information in sections aimed at different audiences to validate, starting with the business entities (e.g.: student, lesson, course, school, etc.), and then generic system entities (diagnostics, errors, configs, etc.) on which they rely on.

### Functional View

: the functionality provided to manipulate the above information. It is common to outline business consumer and support specialist functionality first, and the describe how the expectations of other stakeholders (Support, Operations, Maintenance, Security, etc.) are also met.

### Interoperability View

: describing how the above functionality is made accessible to authenticated and authorised 3rd party systems via audited APIs.

### Sequences View

: covering sequences, such as OAuth and OIDC authentication flows, processing workflows, and state change processes (Draft to Released),

### Component View

: covering the way the system’s logic is broken down to distinct deployable units. Often references DDD.

### Integration View

: describing how the system devices and components are integrated. integrations required to IdPs, Cache, Data stores, Geo/IP Services for geoblocking, and Malware detection within uploaded media.

### Deployment View

* : describing how the system’s components are deployed to target virtual infrastructure.

### Development View

: describing agreed Development Principles and Patterns to use.

### [Datastore] Information View

* : describing datastore elements. Note that while it used to be common to describe in detail a system’s database schema, this perception is dated: data storage is understood to be an internal development concern best left to be described -- if needed -- in a Technical Design Document (TDD) of some form**.**

### Appendices

* : for inclusion or reference to a 3rd party glossary, quality and functional matrices, etc.

Table : Catalogue of recommended ISO-42010 Views

Each of the above views is for reference and review by specific stakeholder groups to provide answers to their queries about Organisation Strategy, Business Services, Security, Privacy, Information, Development, Infrastructure, Deployment, Communications, etc.

Note the following:

* Integration – a deployment aspect – is presented separately to Interoperability – a capability,
* Deployment – a technical view used to describe the use of automated compilation, packaging, deployment and testing pipelines, is distinct from the Delivery view which covers Change Management aspects (Comms and similar transitional requirements).
* Security and Privacy are Views in their own right, rather than view-embedded Perspectives as Rozanski & Wood first proposed, as the subjects have gained importance since 2000, and they need to be prepared for specific stakeholder group audiences to review during accreditation.

## Length

A SAD is not meant to be read in its entirety by any stakeholder group – including quality reviewers.

While not a novel, it is neither a summary[[1]](#footnote-1). It is a reference artefact whose purpose is to reduce unnecessary risks to delivering the expected service by being reviewable by both non-technical and technical stakeholders.

Its length should be the length necessary to decrease the risk of misunderstanding how to implement the Solution. Depending on the project, it can get long[[2]](#footnote-2).

## Duplication

Moderate duplication of information across Views is acceptable where information relevant to one stakeholder group may be also relevant to another stakeholder group.

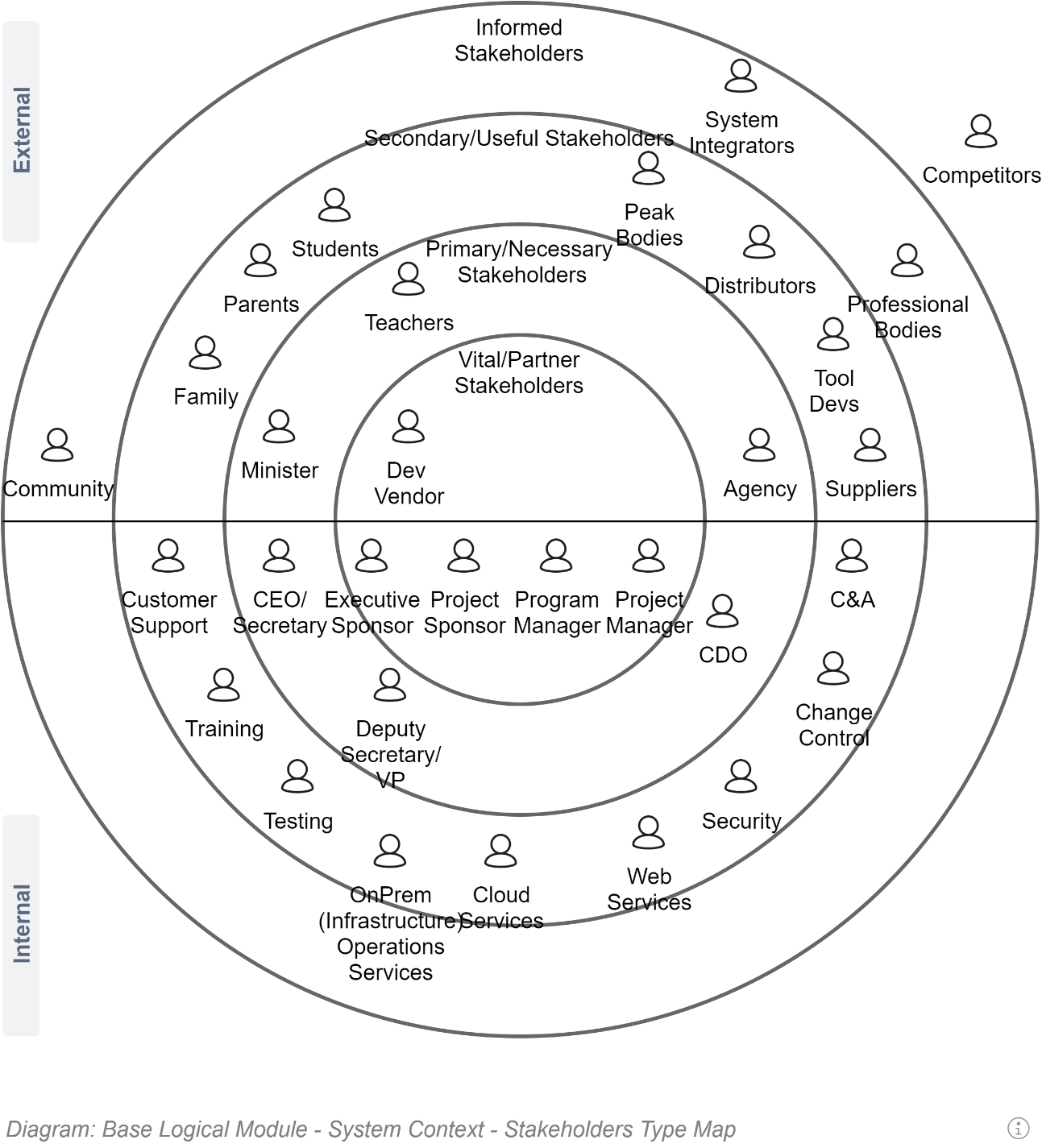
# Appendices

## Diagram Examples

The following are examples of diagrams that can be included in SADs where appropriate.

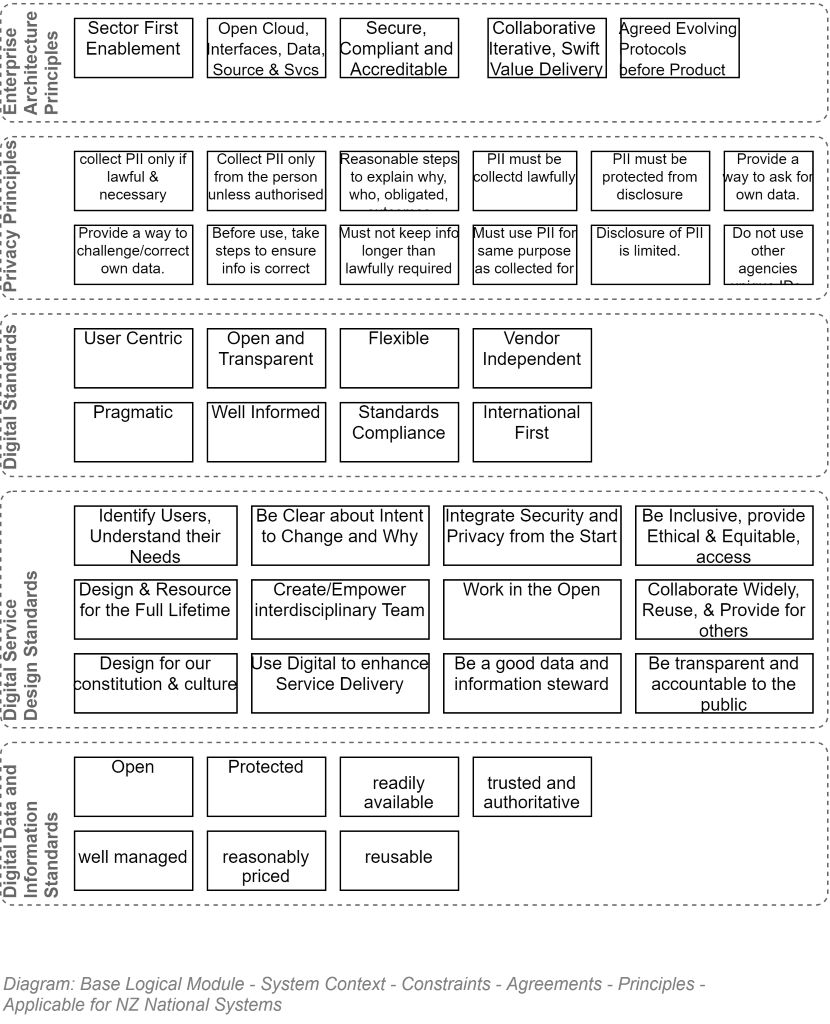
### Business Context View Diagram Example

#### Stakeholder Map



### Principles Map

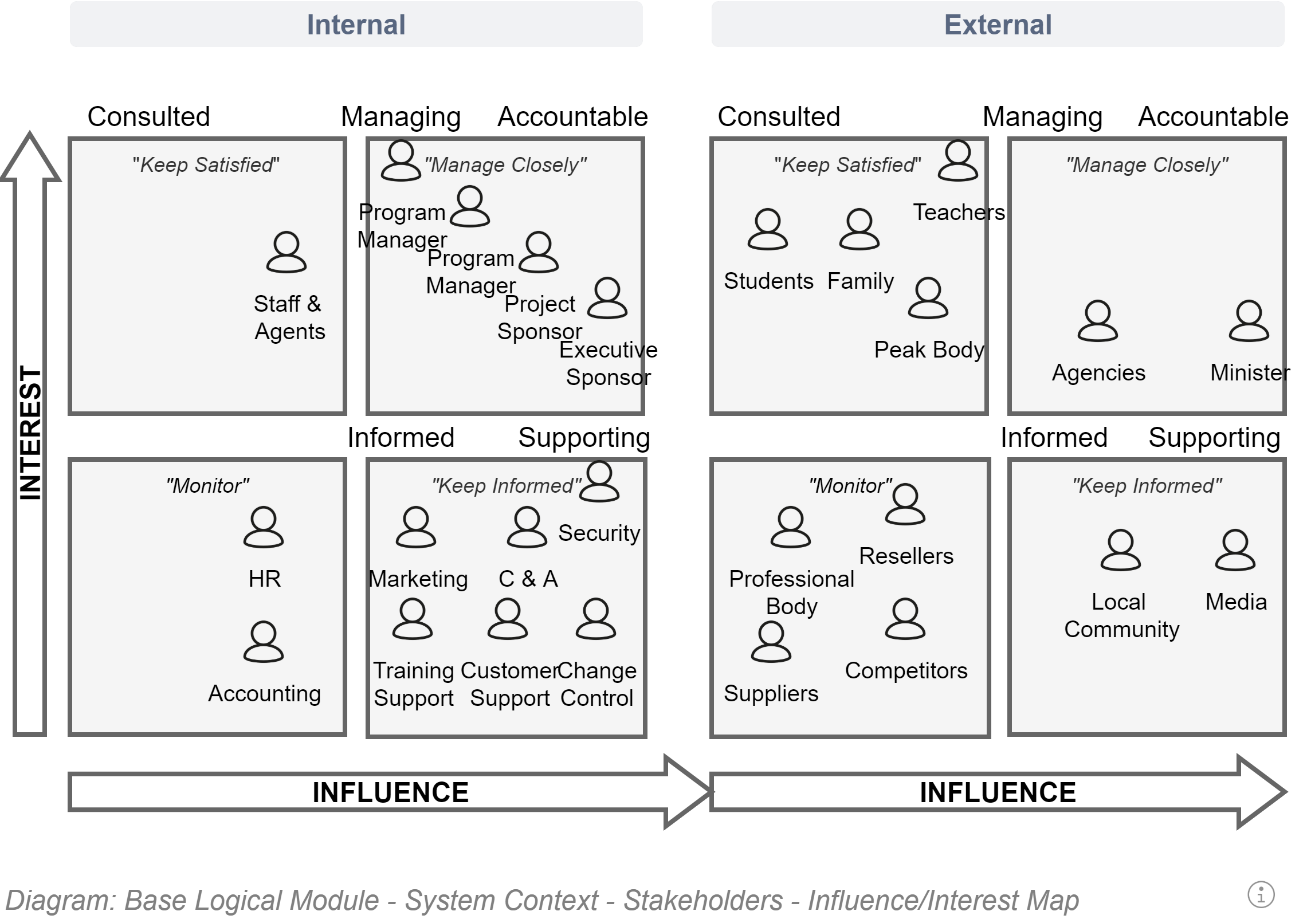
It is optional to include a diagram to summarise Agreed Principles that the team has to follow:

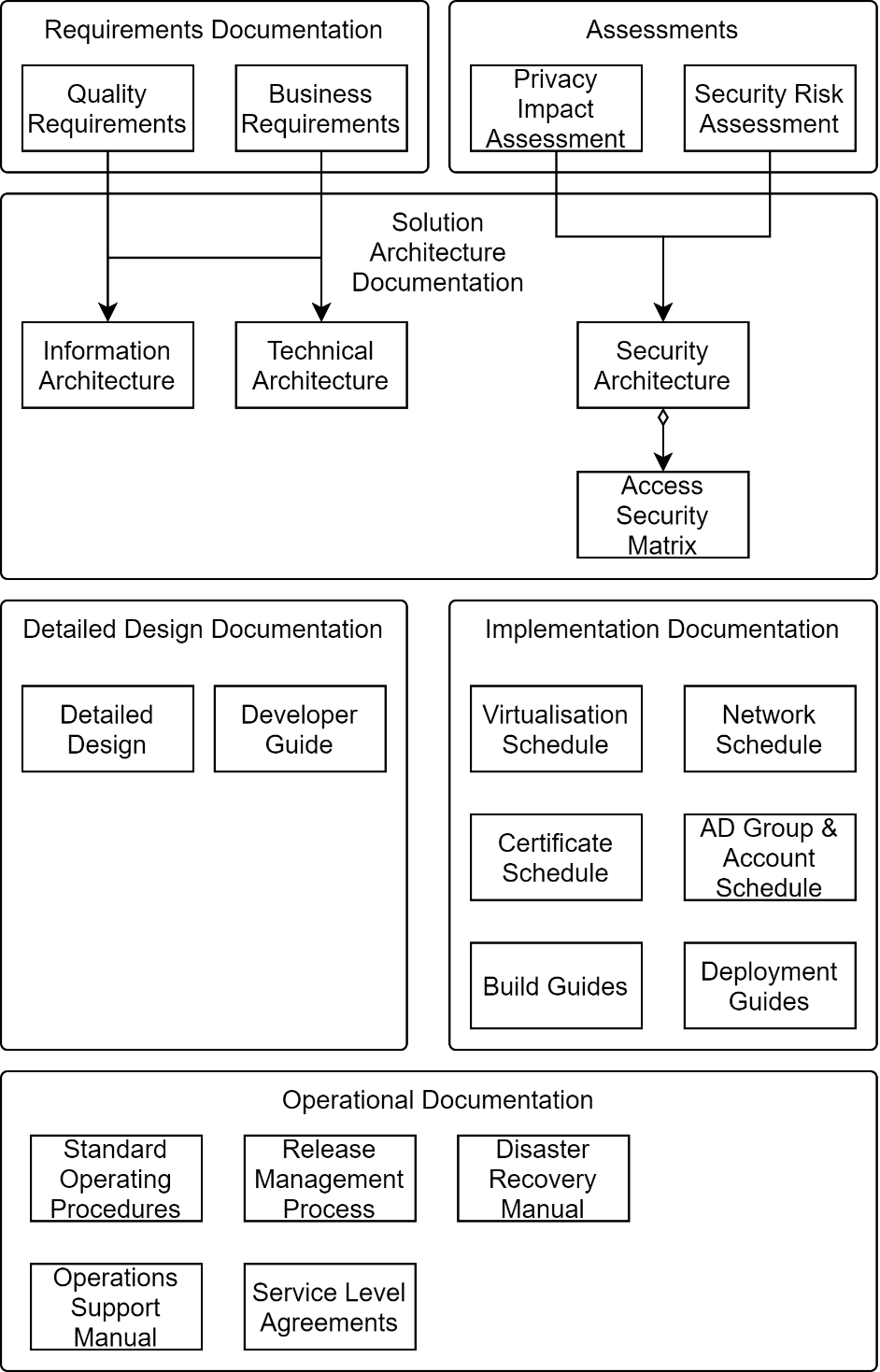


### Delivery View

#### Influence Map

An influence map is an reorganisation of a stakeholder map in terms of their influence on the priority of workitems.



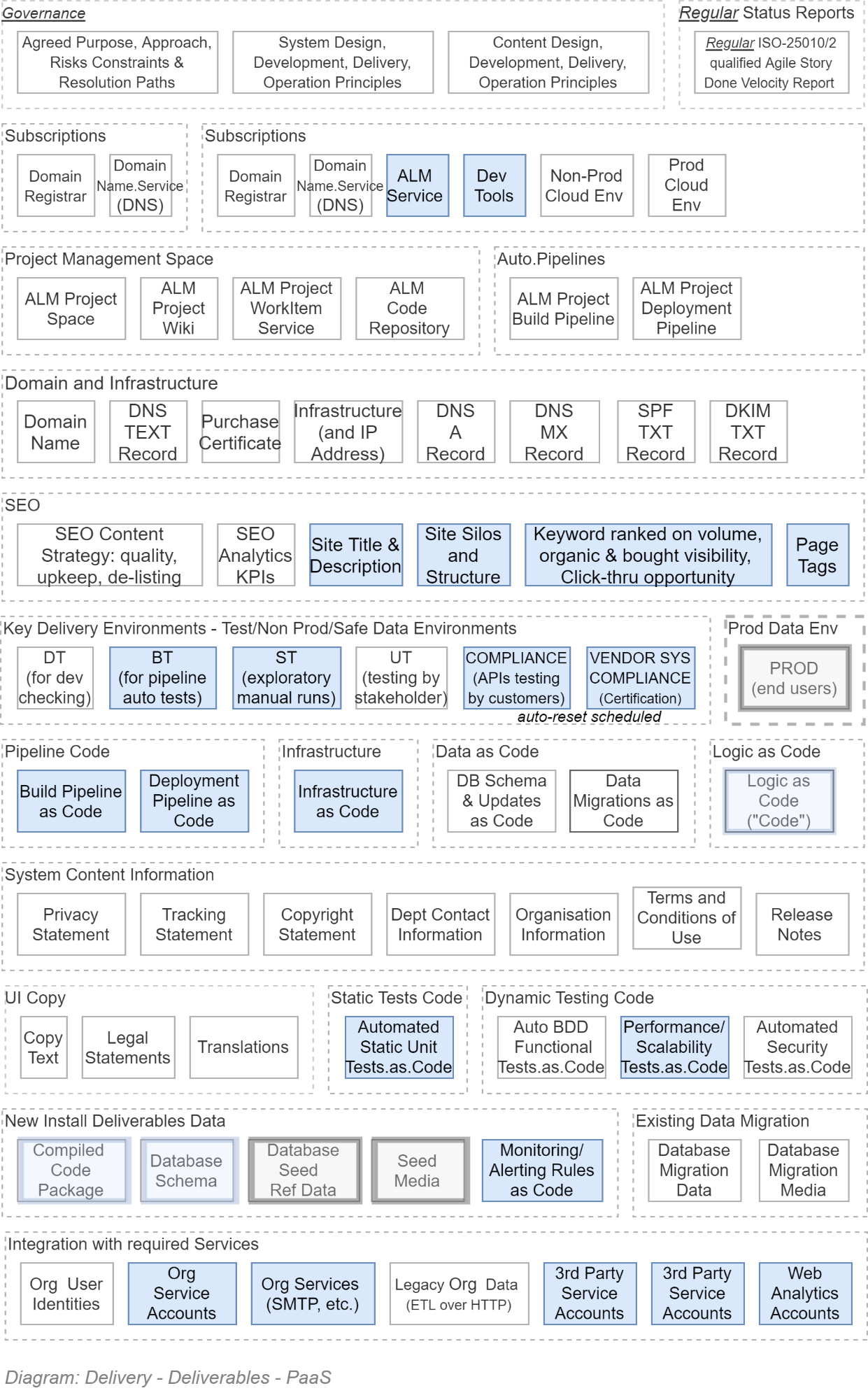


#### Deliverables Map

It is optional to include a diagram that summarises the deliverables that need work item development and tracking by a delivery manager.

The following diagram is an example of a delivery map of technical deliverables required for a simple information system.

It is incomplete as it does not include reference to Transitional Requirements beyond those needed for the system itself.



1. Develop a separate succinct artefact (e.g.: either a PP or A3 based “Architecture on a page”) to summarise to governance groups. [↑](#footnote-ref-1)
2. Making the document a GLAD (Gloriously Long Architecture Document)? [↑](#footnote-ref-2)