ICT Project Guidance

Glossary of ICT Specific Terms:   
Solution Design

Author:

Sky Sigal, Solution Architect

Version:

0.3

## Description

A Glossary of common ICT Terms related to system design, to establish a common understanding, while reducing duplication of effort in downstream documents.

## Synopsis

Included are the meanings of acronyms and industry terms used to describe aspects of solution design.

## Contents

[Description 1](#_Toc150844142)

[Synopsis 1](#_Toc150844143)

[Contents 2](#_Toc150844144)

[Introduction 3](#_Toc150844145)

[Objective 3](#_Toc150844146)

[Terms & Acronyms 4](#_Toc150844147)

[System Design Terms & Acronyms 4](#_Toc150844148)

[4+1 4](#_Toc150844149)

[Layer 4](#_Toc150844150)

[DDD 4](#_Toc150844151)

[Domain Driven Design 4](#_Toc150844152)

[Rozanski & Woods 4](#_Toc150844153)

[SAD 5](#_Toc150844154)

[Solution Architecture Description (SAD) 5](#_Toc150844155)

[UML 5](#_Toc150844156)

[Unified Modelling Language 5](#_Toc150844157)

[Appendices 6](#_Toc150844158)

[Appendix A - Document Information 6](#_Toc150844159)

[Versions 6](#_Toc150844160)

[Images 6](#_Toc150844161)

[Tables 6](#_Toc150844162)

[References 6](#_Toc150844163)

[Review Distribution 6](#_Toc150844164)

[Audience 6](#_Toc150844165)

[Diagrams 6](#_Toc150844166)

## Introduction

## Objective

To develop a common understanding of terms used to deliver services with an ICT component.

# Terms & Acronyms

## Acronyms

#### API

: Application Programming Interface.

#### CaaS

: acronym for *Containers as a Service*.

#### DDD

* : acronym for *Domain Driven Design*.

#### FaaS

: Acronym for *Functions as a Service.*

#### FURPS+

: An acronym to define qualities desired of a system. Stands for Functionality, Usability, Reliability, Performance, Supportability, plus Constraints, Interface, Rules.

#### PaaS

: acronym for Platforms as a Service.

#### IaaS

: Infrastructure as a Service. where achievable, Avoid in favour of using *CaaS*, or even *PaaS*.

#### UI

: User Interface

#### UX

: User Experience

## Terms

#### 4+1

: a term to describe the deprecated 90’s structure for *SAD* documents. A 4+1 *SAD* was comprised of:

* *Logical View*, describing the logical functionality made available by a system,
* *Process View*, describing dynamic sequence flows between system aspects,
* *Development View*, describing how the system logic is packaged into discrete elements,
* *Physical View*, describing how the system packages are deployed to target devices,
* *User Scenarios*, describing a series of indicative examples of how the service is used.
* *Note: superseded by the structure advocated within ISO:42010:2011, popularised by Rozanski & Woods seminal industry book.*

#### Acceptance Criteria

: the test definitions that complete a *user story* to be automated to in a delivery pipeline as acceptance tests, to demonstrate the criteria are met.

#### Application Programming Interface (API)

: an interface for another authorised system to interact with the system, as opposed to end users using a *User Interface*.

#### ClickOps

: manual, often undocumented, unrepeatable, unautomated configuration of infrastructure services and integration via their visual user interfaces (compare with *DevOps*).

#### Cloud Architecture

: a design paradigm that uses infrastructure services (e.g., *FaaS*, *PaaS*, *CaaS*, *IaaS*), supporting services (e.g., caching, storage, identity, mail), and cross cutting services (e.g., diagnostics, monitoring) and provided by a cloud service.   
The approach provides a reduction of effort at the expense of more latency and deployment and integration complexity that decreases maintainability if DevOps automation is not used.

#### Cross Cutting Services

: infrastructure services used across system layers. Generally including temporary Diagnostic tracing, permanent Auditing, Authorisation, Routing, notification, messaging, queue and workflow processing, etc. excluding business domain specific services. See *DDD*.

#### DevOps

: the encoding of instructions for infrastructure creation and configuration, then code compilation, static testing, packaging, deploying, configuration, data migration and restoration, dynamic testing, and notification. Contrast with *ClickOps*.

#### Domain

: a logical bounded area of related concepts, practices, and information, that can range from being defined with specific - to aid recognition - terms to intentionally abstracted ones - to provide flexibility of use.

#### Domain rules

: the algorithms, decision trees, computation rules, workflows and use cases specific to a domain.

#### Domain Driven Design

* : a design approach to develop moderate to complex systems (most enterprise services fall in this category) using mature design decisions to deliver a service that remains modular, modifiable, enhancable and maintainable.

#### Interface

: may be a physical or software User Interface (UI), or an Application Programming Interface (API).

#### Internationalisation

: the process of designing systems so they can be adaptable to various languages and regions without engineering change. See *Localisation*.

#### ISO-9126

: *deprecated* standard of qualities desired of an IT system.

#### ISO-25010

: International standard for defining and measuring desirable system qualities, composed of agreed topics and subtopics. An improvement over earlier ISO-9126 and defunct FURPS+. Covers: Functional Suitability, Performance Efficiency, Compatibility, Interaction Capability, Reliability, Security, Maintainability, Flexibility, Safety.

#### Layer

* : to improve maintainability correctly designed services are developed into isolated layers. The most common stack of layers is Presentation, Interface/Validation, Logic, Technical Integration (a subset of which is Data Storage). See *Tier*. See *DDD*.

#### Localisation

: the process of making a previously *internationalised* system available to a local by translating images and text and optionally adding locale-specific capabilities.

#### Micro Services

: a deployment led architecture that may decreases system development effort by leveraging *Cloud Services*, at the expense of potentially increasing latency and deployment complexity, decreasing maintainability. **Important:** Avoid using until there is a business use case (as opposed to a *technology-led* use case). A *Business* use case for it would be if two different business units provide capabilities for the same service and are unable to coordinate delivery using the same scheduling and automation. Recommend avoiding using for custom systems delivered by vendors, limiting its use to internally developed systems.

#### Modulithic

**Architecture**

: TODO . Prefer over *MicroServices*.

#### Monolithic

: systems that are developed to provide the capabilities needed by distinct domains in a system. Most business systems require being iteratively developed to provide capabilities within the following logical domains: System, Individual, Social, Resource and Scheduling, Processing domains.

#### Performance

: commonly referring to measuring Capacity and the meeting of Response times while meeting both Throughput and desired Resource usage, startup and shutdown times.

#### Rozanski & Woods

* : the authors of a seminal industry book on how to structure *SAD*s in accordance with *ISO-42010*. They popularised SADs comprised of the following views:
* [Service] Context View, describing the business context of the service,
* Functional View, describing the functions provided by the system,
* Information View, describing the HL entities the system is managing,
* Integration View, describing systems the system integrates with, and messages transmitted to and from,
* Infrastructure View, describing the target devices and zones the system components are deployed to,
* The following Perspectives could be appended as sub sections within the above Views -- or developed as additional Views:
* Security
* Privacy

#### SAD

: see *Solution Architecture Description*.

#### Solution Architecture Description (SAD)

: a coherent set of Views describing aspects of a complex model, as described within *ISO-42010*. Depending on the scale of the project expected Views will include several or all the following:

* [System] Context View, covering Background, Objectives, Constraints (Regulations), Obligations (Agreements, Principles, Requirements and Governance)
* Delivery View, covering Deliverables, Expectations, Methods of Working, etc.
* Functional View, covering how the service meets its functional requirements, illustrated by Use Cases by various Stakeholder Roles
* Integration View, covering Components and their integration
* Interoperability View, covering how Components are accessible to other services
* Qualities View, covering how the system meets its Quality Requirements
* Development View, covering expected development practices,
* Quality Assessment View, covering how quality is assured,
* Privacy View, covering how the service adheres to its (legal) privacy obligations
* Security View, covering how the service adheres to its security obligations
* Deployment View, covering automation of quality assurance and delivery

Note that a *Description* does not have to be a *Document*. A SAD can be developed in Word or Confluence, or any medium that provides both sufficient access to contributors and consultants, and versioning sufficient to support the accountability required of an *Accreditation* process.

#### Tier

: a physical device containing one or more system *layers*.

#### UML

: acronym for *Unified Modelling Language*.

#### Unified Modelling Language

: an *ISO* defined diagramming standard for modelling structural, behavioural, architectural aspects of systems. See *ISO-19505*-2.

#### Use Case

: a technique for capturing, modelling and specifying requirements for enabling user interactions with a system. A use case involves *users* with a system *role* (actors) within a *scenario* initiating actions to receive a response.

#### Use Case Model

: the diagram used to describe a Use Case, permitting the visualisation of actors, flow of interactions, and system behaviours & messages.

#### User Story

: a general explanation of a feature written from the perspective of a user with a specific role. *User Stories* are not complete without accompanying *Acceptance Tests*.

Appendices

Appendix A - Document Information

### Versions

* 1. Initial Draft
  2. Minor corrections
  3. Minor changes

### Images

**No table of figures entries found.**

### 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 |
| Sandy Britain, Enterprise Architect |  |
| Amy Orr, Data Architect |  |
| Roger Govind, Security Architect |  |
| Archana Sahani, Business Analyst |  |
| Dijana Sneath, Business Analyst |  |
| Vincent Weirdsma, Lead Developer |  |

### Audience

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

### 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.