

Solution Architecture Definition (SAD)

**<Customer Name> - <Project Tile>**

**<Date>**

**<Version>**

**<Author>**

# Document History

| **Version** | **Date** | **Author** | **Comment** |
| --- | --- | --- | --- |
|  |  |  |  |

# Document References

| **Reference ID** | **Documentation** | **Versions** |  |
| --- | --- | --- | --- |
|  |  |  |  |

Table of ContentS

[Document History 2](#_Toc530086752)

[Document References 2](#_Toc530086753)

[3. Introduction 5](#_Toc530086754)

[3.1 Document Purpose 5](#_Toc530086755)

[3.2 Disclaimers 5](#_Toc530086756)

[3.3 Definitions and Abbreviations 5](#_Toc530086757)

[3.4 Business Terms 5](#_Toc530086758)

[3.5 Technical Terms 5](#_Toc530086759)

[3.6 Overview 5](#_Toc530086760)

[4. Assumptions and Constraints 7](#_Toc530086761)

[5. Context View 8](#_Toc530086762)

[5.1 Context View Diagram 8](#_Toc530086763)

[5.2 Users 8](#_Toc530086764)

[5.3 Internal Systems 8](#_Toc530086765)

[5.4 External Systems 8](#_Toc530086766)

[6. Project View 9](#_Toc530086767)

[7. Functional View 10](#_Toc530086768)

[7.1. Use Cases 10](#_Toc530086776)

[7.2. Use Case Groups 10](#_Toc530086777)

[8. Process View 11](#_Toc530086778)

[8.1. <Process 1> 11](#_Toc530086779)

[8.2. <Process 2> 11](#_Toc530086780)

[9. Non-Functional View 12](#_Toc530086781)

[9.1. Performance Efficiency 12](#_Toc530086782)

[9.2. Integration 12](#_Toc530086783)

[9.3. Compatibility 12](#_Toc530086784)

[9.4. Usability 12](#_Toc530086785)

[9.5. Reliability 12](#_Toc530086786)

[9.6. Security 12](#_Toc530086787)

[9.7. Maintainability 12](#_Toc530086788)

[9.8. Portability 12](#_Toc530086789)

[9.9. Accessibility 12](#_Toc530086790)

[10. Logical View 14](#_Toc530086791)

[11. Interface View 15](#_Toc530086792)

[12. Design View 16](#_Toc530086793)

[13. Physical View 17](#_Toc530086794)

[13.1. System Landscape 17](#_Toc530086795)

[13.2. Components 17](#_Toc530086796)

[<Component 1> 17](#_Toc530086797)

[<Component 2> 17](#_Toc530086798)

[13.3. Component Versions 17](#_Toc530086799)

[13.4. Environments 17](#_Toc530086800)

[13.5. Size, Performance and Scaling 17](#_Toc530086801)

[14. Deployment View 18](#_Toc530086802)

[15. Operational View 19](#_Toc530086803)

[16. Security View 20](#_Toc530086804)

[17. Data View 21](#_Toc530086805)

# Introduction

## Document Purpose

This document describes the technical architecture of the solution through different views to expose the concepts, constraints and mechanics behind it. It does not set out a list of responsibilities of the parties or the deliverables of the product.

## Disclaimers

* The architecture specifications captured in this document are not fixed and are likely to change during the project phases. Each version contains the architecture views and decisions for that moment in time, but do not necessarily reflect the same architecture components and integrations that will be delivered for each release. This is mainly due to the fact that the view of the system evolves during its development. This is especially true during the Foundation project phase, as only a high-level view and understanding of the solution is available. The architecture view, key design decisions and integrations are most likely to change during the Engineering phase.
* <Other disclaimers>

## Definitions and Abbreviations

The tables below define the terms and languages used within this document. Business terms will be adopted and used within the development to ensure consistency and common understanding between the implementation team and the customer.

## Business Terms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| <Term> | <Definition> |
|  |  |
|  |  |

## Technical Terms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| <Term> | <Definition> |
|  |  |
|  |  |

## Overview

The Solution Architecture Definition document describes the subsystems and components of the solution by presenting a number of architectural views. Each view shows a different aspect of the system to address different concerns. These are as follows:

* **Assumptions and Constraints** – describes any limitations and factors that must be taken into account when designing the system.
* **Context View** – shows at the highest level how the proposed system interacts with external systems and user groups.
* **Project View** – maps key functionality to releases and milestones.
* **Functional View** – describes the top-level functionality that the system should implement and summarises the architecturally significant Use Cases (i.e. those which influenced the overall architectural design of the system) along with how the key Use Cases will be realized.
* **Process View** – describes the top-level workflows and processes to be followed within the system.
* **Non-functional View** – describes specific system changes that allow meeting the non-functional requirements, which are defined outside of the SAD.
* **Logical View** – defies the functional decomposition of the system into smaller subsystems
* **Interface View** – defines the system interfaces required in the solution.
* **Design View** – this section describes any lower level design detail required to support the implementation of the system.
* **Physical View** – shows the hardware infrastructure on which the software runs.
* **Deployment View** – shows the mapping of the logical components onto the physical infrastructure.
* **Operational View** – describes the elements of the solution that will support the operations and support teams when the system goes into production.
* **Security View** – describes elements and functionality that impact the overall security of the application.
* **Data View** – shows the main data structures and how the entities relate to each other

# Assumptions and Constraints

This section documents significant constraints, assumptions and requirements that influence the architecture of the solution.

# Context View

This view gives a high level representation of the system, the different user types and interactions with external entities. It describes the boundaries of the solution.

## Context View Diagram

The diagram in this section shows the interaction between the key system components, user’s types and external entities.

*<Insert context diagram here>*

## Users

**<Customer>**

*<Description>*

|  |  |  |
| --- | --- | --- |
| **<User Type>** | **Description** | **Site Restrictions** |
| *<User Type>* | *<Description>* | *<Site restrictions>* |
|  |  |  |
|  |  |  |

**<User Type>**

*<Description>*

|  |  |  |
| --- | --- | --- |
| **<User Type>** | **Description** | **Site Permissions** |
| *<User Type>* | *<Description>* | *<Site permissions>* |
|  |  |  |
|  |  |  |

## Internal Systems

**<System Name 1>**

*<Description>*

**<System Name 2>**

*<Description>*

## External Systems

# Project View

This section shows how key functionality relevant to the solution architecture maps to releases and milestones.

For details on the release schedule, project phases and team structure please refer to the Project Management Definition.

|  |  |
| --- | --- |
| **Release** | **Key Functionality** |
| *<Release #>* | *<Description>* |
|  |  |
|  |  |

# Functional View

This section describes the key functional areas of the project. The goal is to provide context around the architecture – all software performs some functionality and the definition of this functional scope is a very important factor to define the architecture.



## Use Cases

The full set of use cases is documented indirectly by the user stories defined for the project. This section highlights those key use cases which influence the architectural decisions and why.

|  |  |  |
| --- | --- | --- |
| **Use Case Reference** | **Reason for Architectural Significance** | **Sections Addressed in** |
| *<Reference #>* | *<Description>* | *<Section>* |
|  |  |  |
|  |  |  |

## Use Case Groups

|  |  |  |
| --- | --- | --- |
| **Use Case Reference** | **Reason for Architectural Significance** | **Sections Addressed in** |
| *<Reference #>* | *<Description>* | *<Section>* |
|  |  |  |
|  |  |  |

# Process View

The intent of the process view is to show how the various processing steps within the system fit together to implement the overall functional requirements. This is necessary if the system relies on workflow processes, forked or parallel processing mechanisms.

The following processes are significant:

## <Process 1>

*<Description>*

## <Process 2>

*<Description>*

# Non-Functional View

This section describes architecturally significant changes that enable the solution to achieve the agreed non-functional requirements (NFRs). Each change is mapped to the corresponding NFR category, which is based on the ISO/IEC 25010-2011 product quality model.

NFRs are documented and maintained in the Non Functional Requirements Definition and will not be repeated here. In case of duplication, the Non Functional Requirements Definition takes precedence.

## Performance Efficiency

*<Description>*

## Integration

*<Description>*

## Compatibility

*<Description>*

## Usability

*<Description>*

## Reliability

*<Description>*

## Security

*<Description>*

## Maintainability

*<Description>*

## Portability

*<Description>*

## Accessibility

*<Description>*

# Logical View

# Interface View

This section describes the interfaces that will be required to the external system integration touch points.

# Design View

This section describes and explains any lower level design concepts if required arising from the solution.

# Physical View

## System Landscape

*<Insert system landscape diagram here>*

## Components

#### <Component 1>

*<Description>*

#### <Component 2>

*<Description>*

## Component Versions

The table below lists the software components and versions that will be used for the solution.

|  |  |  |
| --- | --- | --- |
| **Component** | **Version** | **Notes** |
| *<Component>* | *<Version #>* | *<Notes>* |
|  |  |  |
|  |  |  |

## Environments

This section describes the various environments that will be configured for the solution*.*

## Size, Performance and Scaling

This section describes how the architecture supports the future traffic projections

# Deployment View

This section describes how code will be deployed in test environments, as well as key considerations for the more complex Production go-live deployment.

# Operational View

This section describes how the architecture will support operational processes and activities.

# Security View

This section describes how the architecture addresses the different security aspects.

# Data View

This section describes the important data model changes required to fulfil the requirements and the associated data flows.