System Architecture Definition

SYSTEM: <SYstem>

DOCUMENT ID: <12345>

Document Control

|  |  |
| --- | --- |
| Title | <Name> |

Contact(s) for inquiries and proposed changes

For information regarding this document or if you have any questions or suggestions regarding the content, please contact the following:

| Name | Title |
| --- | --- |
|  |  |
|  |  |

Revision History

| Version | Date | Revised by | Brief outline of changes |
| --- | --- | --- | --- |
| 0.1 |  |  |  |
| 0.2 |  |  |  |

Related Documents

| Title | Location |
| --- | --- |
| System Architecture |  |
|  |  |

Reviewers

| Name | Title | Date | Evidence |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Approvals

| Name | Title | Date | Evidence |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

TABLE OF CONTENTS

1. Introduction 5

1.1. Background 5

1.2. Purpose 5

1.3. Scope 5

1.3.1. This document applies to: 5

1.3.2. This document does not address: 5

1.4. Approach 5

1.4.1. Document Structure 5

2. Goals and Requirements 5

2.1. Quality of Service Goals 5

2.2. Technical Requirements 6

2.3. Constraints and Dependencies 6

2.3.1. Development Process and Team 6

2.3.2. Environment and Technology 6

2.3.3. Delivery and Deployment 6

2.4. Assumptions 6

3. Risks and Architecturally Significant Epics 6

3.1. Risks Addressed 6

3.2. Architecturally Significant Epics 6

4. High Level Solution Overview 7

5. Application Layer 7

5.1. Structure View 7

5.1.1. Client Tier 7

5.1.2. Access Tier 7

5.1.3. Presentation Tier 7

5.1.4. Services Tier 7

5.1.5. Domain Tier 8

5.1.6. Integration Tier 8

5.1.7. Resource Tier 8

5.2. Configurations View 8

5.3. Domain View 8

5.4. Behaviour View 8

5.5. Security and Identity 8

5.5.1. Identity Management 8

5.5.2. Access Management 8

5.5.3. Encryption / Tokens / Certificates 8

5.6. Data Entities Across Tiers 8

5.7. Evolutionary Considerations 8

6. Software Platform Layer 8

6.1. Incorporated Mechanisms 9

6.2. Custom Mechanisms 9

6.3. Configuration View 9

7. Shared Services Layer 9

7.1. Incorporated Mechanisms 9

7.2. Configuration View 9

7.3. Evolutionary Considerations 9

8. Operating Platform Layer 9

8.1. Evolutionary Considerations 9

8.2. Configurations View 9

9. Compute and Network Layer 9

9.1. Evolutionary Considerations 9

9.2. Configurations View 9

10. Facilities Layer 9

11. Qualities of Service Review 10

11.1. Availability 10

11.2. Scalability 10

11.3. Performance 10

11.4. Usability 10

11.5. Agility, Extensibility and Flexibility 10

11.6. Security 11

Introduction

Background

This document

Purpose

The primary purpose of this document is to communicate the essential elements of the overall solution so that business implications can be assessed and understood, and that the detailed design and build phases can proceed. As such it achieves the following:

Scope

This document applies to:

This document describes the architecture of the technical solution that addresses <ORGANISATION> requirements for a <system>. It details application level information that is not elsewhere documented.

This document does not address:

To ensure clarity, conciseness and information currency, the <system> Architecture Definition document does not unnecessarily duplicate detailed information that is available and maintained elsewhere.

Approach

The architecture will be described using a multi-tiered and multi-layered approach driven by Qualities of Service (QOS) capabilities, known as the Cube. The approach is described in the POS Channes Reference Architecture Document and readers should refer to this for further elaboration. The Reference Architecture also describes the Principles, Practrises, Patterns and Guidelines that are the basis for this architecture.

Document Structure

The document begins with a description of the goals and constraints around which the architecture is built. The document is then structured using the Cube; the layers and the views are described, and optionally, the tiers and services within those. Finally, a Quality of Service (QOS) view ties everything together. This QOS view describes how the architecture achieves the required business goals for each relevant quality or set of related qualities QOSs.

Goals and Requirements

The Architectural (Quality of Service) Goals, Technical Requirements, Constraints and Assumptions are documented below.

Quality of Service Goals

The top four architectural goals relating to QOS are summarised below.

* **Usability** – the customer experience is a key motivator for the <platform>;
* **Performance –** meet the required performance levels and do not compromise on usability;
* **Scalability and availability –** the application should scale cost effectively to meet availability needs.
* **Security** – Customer data should be maintained or transacted in a secure manner and all their details kept private and confidential. Audit logs should be maintained and the integrity of data ensured.

These goals and the others are discussed in more detailed in the final section of the document.

Technical Requirements

To achieve these qualities the following Architectural / Technical requirements need to be considered.

* Requirements

Constraints and Dependencies

The contsraints on the system are described under three categories.

Development Process and Team

Environment and Technology

Delivery and Deployment

Assumptions

The main assumptions made are:

* Assumption 1: describe;

Risks and Architecturally Significant Epics

Risks Addressed

Technical risks represent areas of potential difficulty and uncertainty. Early design activity in the project will be focused on mitigating these risks. The following risks have been identified, in order of priority, with their associated mitigation strategy:

|  |  |  |
| --- | --- | --- |
| **Risk ID** | **Risk Description** | **Mitigation Strategy** |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

Architecturally Significant Epics

The architecturally significant Epics are a subset of the Epics whose implementation would necessitate the mitigation of the risks documented above. The selected architecturally significant epics include:

|  |  |  |
| --- | --- | --- |
| **Story ID** | **Description** | **Reason for Inclusion** |
| 1 |  |  |
| 2 |  |  |

**Table 1: Architecturally Significant Epics**

The following table cross-references the selected architecturally significant use cases to the technical risk areas they address. The rows map to the risks identified in the previous section.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Technical Risk ID** | **Epic ID** | | | | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| Risk 1 | X |  | X |  |  |  |  |  |  |  |
| Risk 2 | X | X | X | X | X |  |  |  |  |  |
| Risk 3 | X |  |  | X |  |  |  |  |  |  |

**Table 2 : Technical Risks V Epic**

High Level Solution Overview

**Figure 1** provides a high level logical overview of the architecture of <system>. The figure shows the main actors, external; resources, and the key components at each tier.

< Include Figure here >

**Figure 1 : High Level Overview**

The following sections will examine the system layer by layer and where appropriate tier by tier.

Application Layer

This section describes the architecture of the application and services that are custom built and provide the business functionality specific to the system. This layer is described using the standard Cube views.

Structure View

The structure view describes the structural decomposition of the architecture into services, architecturally significant software packages and components. This view describes the static dependencies and structural relationships among them. The decomposition strategies, principles and reasons for choosing these packages, are described and justified.

Figure 2 shows the system’s application and service components across the tiers and how they interact.

< Include Figure here >

**Figure 2 : Major Applications and Services across Tiers**

Client Tier

The client tier manages the customer experience on the client devices.

Access Tier

The access tier provides access into an organisation as gateway services. The tier routes and proxies messages, provides security and authentication and manages device detection.

Presentation Tier

The presentation tier provides services that assemble, aggregate and format content for the client devices above.

* + - 1. Frameworks

The presentation tier uses:

* Framework list.

Services Tier

The services tier provides a single point of entry to services from the presentation tier. The services tier insulates the presentation tier from the business logic, integration, legacy applications and data resources in the tiers below. The services are re-usable, and may be accessed by other systems either internal or external to the system or organisation.

Domain Tier

The Business Tier provides components that implement the business logic, domain object model, business processes, workflow and business rules, typically using container based technologies and designs.

Integration Tier

The Integration Tier provides services, such as persistence and legacy wrappers, that connect to external resources, insulating the business services from those resources.

Resource Tier

The Resource Tier includes external services, legacy and database resources used by the systems.

The main resources are:

* List( e.g. DBMS, Legacy Systems)

Configurations View

The configuration view describes the dynamic relationships and runtime dependencies between components from the perspective of physical location, deployment, configuration, operation and managing the application. **Figure 3** shows the deployment of components in the production environment.

< Include Figure here >

**Figure 3 : Deployment Diagram**

## Domain View

The main important domain objects of interest are:

* list

## Behaviour View

The behaviour view describes architectural decisions around significant, complex or central epics, stories or use case scenarios, focusing on how functionality is implemented across collaborating components domain objecs and across tiers.

Security and Identity

Security is achieved using a number of approaches.

Identity Management

Access Management

Encryption / Tokens / Certificates

Data Entities Across Tiers

This section describes key architectural decisions that apply to entities across tiers, describing how data is represented and passed from tier to tier, data/object passing, transactions, integrity management and security.

Evolutionary Considerations

Software Platform Layer

This section describes the software frameworks, platforms and standardizing layers that provide vendor-independent access for the application layer.

Incorporated Mechanisms

The <system> includes mechanisms, required by the system, that are not custom built.

Custom Mechanisms

The <system> includes mechanisms, required by the system, that are custom built.

Configuration View

Shared Services Layer

This section describes the platform infrastructure on which the higher layers are built and any mechanism used across the system. Mechanisms are supporting functionality or capabilities, re-used across the application, that require some degree of management.

Incorporated Mechanisms

The <system> includes mechanisms, required by the system, that are not custom built.

Configuration View

Evolutionary Considerations

Operating Platform Layer

The operating platform layer describes describes the operating system, virtualisation platforms and Environments

Evolutionary Considerations

Configurations View

Compute and Network Layer

The compute and storage layer describes the hardware and O/Ss required to run the applications, virtual platfroms and services.

Evolutionary Considerations

Configurations View

Facilities Layer

The facilities layer describes the data centre and system facilities.

Qualities of Service Review

This section describes the QOS requirements and how they have been addressed by the architecture.

Availability

|  |  |
| --- | --- |
| Requirement |  |
| Architectural Solution |  |
| Evolutionary Considerations |  |

Scalability

|  |  |
| --- | --- |
| Requirement |  |
| Architectural Solution |  |
| Evolutionary Considerations |  |

Performance

|  |  |
| --- | --- |
| Requirement |  |
| Architectural Solution |  |
| Evolutionary Considerations |  |

Usability

|  |  |
| --- | --- |
| Requirement |  |
| Architectural Solution |  |
| Evolutionary Considerations |  |

Agility, Extensibility and Flexibility

|  |  |
| --- | --- |
| Requirement |  |
| Architectural Solution |  |
| Evolutionary Considerations |  |

Security

|  |  |
| --- | --- |
| Requirement |  |
| Architectural Solution |  |
| Evolutionary Considerations |  |