

DTI Template 303A

Solution Design – Complex template

Project name & ID

Document Author: Insert name

Date Created: Insert date

Template Data

|  |  |  |  |
| --- | --- | --- | --- |
| Title | Solution Design - Complex | | |
| Description | The objective of this document is to define explicit instructions for the delivery of the xxx Software/System comprising of functional and non-functional requirements. This is inclusive of application/infrastructure design plus process requirements including policy compliance, testing, and governance.  Provide explicit architecture requirements for solution delivery  Define governance that will be applied over the delivery | | |
| Template ID | TI303A | Classification | Internal Use Only |
| Template Owner | Architecture team | Template Approver | Architecture team |

Document Information

This document is available in two forms, controlled and uncontrolled. The controlled variant is maintained electronically and accessed by authorized persons of Etihad Airways IT Document Library/Intranet. Uncontrolled variants are all other electronic and printed copies.

|  |  |
| --- | --- |
| Title |  |
| Author | Naseem Haq |
| File Ref. | EYIT-EAP-HLD Complex Template v1.0 |
| Document Location | EA Repository |
| Document Template version | 1.0 |

Remove approvers as required according to the Project Sizing Calculator

| Approval Sign-off (For formal issue) | | | | |
| --- | --- | --- | --- | --- |
| Approver | Signature | Name | Date | Version |
| Chief Enterprise Architect (Very Complex Projects Only) |  | Takhliq Hanif |  |  |
| Assigned Lead Architect |  |  |  |  |
| Application Architect |  |  |  |  |
| Information/Integration Architect |  |  |  |  |
| Security Architect |  | Alan Engles |  |  |
| Infrastructure Architect (for new technology platforms) |  | Abdul Rehman Mayat |  |  |
|  |  |  |  |  |
| Partner Chief Architect |  |  |  |  |
| Partner Architect |  |  |  |  |

| Revision History | | | | |
| --- | --- | --- | --- | --- |
| Version | Status | Date | Author / Editor | Details of Change |
| 1.0 | Approved | 03/05/2017 | Naseem Haq | Revised HLD for Complex Size projects. Changed approvers list |

The latest approved version of this document supersedes all other versions, upon receipt of the latest approved version all other versions should be destroyed, unless specifically stated that previous version (s) are to remain extant. If any doubt, please contact the document Author.

Table of Contents

[1. Table of Figures 6](#_Toc487101797)

[2. Document Control 7](#_Toc487101798)

[Distribution List 7](#_Toc487101799)

[References 7](#_Toc487101800)

[Terminology 8](#_Toc487101801)

[3. Purpose & Background 9](#_Toc487101802)

[Executive Summary 9](#_Toc487101803)

[Design Objectives 9](#_Toc487101804)

[Design Assumptions 9](#_Toc487101805)

[Design Dependencies 9](#_Toc487101806)

[Design Risks 10](#_Toc487101807)

[Design Issues 10](#_Toc487101808)

[In-Scope 11](#_Toc487101809)

[Out-of-Scope 11](#_Toc487101810)

[4. Solution Design 12](#_Toc487101811)

[Current Application Design 12](#_Toc487101812)

[Use Case Model Diagram 12](#_Toc487101813)

[Proposed Application Design 14](#_Toc487101814)

[Use Case Model Diagram 14](#_Toc487101815)

[Use Case Realisation Diagrams 14](#_Toc487101816)

[Current Information and Data Design 15](#_Toc487101817)

[Data Dictionary – In Scope 15](#_Toc487101818)

[Information domain decomposition 15](#_Toc487101819)

[Data Entity Model 15](#_Toc487101820)

[Data Input and Output 19](#_Toc487101821)

[Proposed Information and Data Design 20](#_Toc487101822)

[Data Dictionary – In Scope 20](#_Toc487101823)

[Logical Data Entity Model/Diagram 20](#_Toc487101824)

[Information domain decomposition 20](#_Toc487101825)

[Data Entity Model 21](#_Toc487101826)

[Proposed System Data Input and Output 25](#_Toc487101827)

[Current Integration Design 26](#_Toc487101828)

[Service Overview 26](#_Toc487101829)

[Technical Integration Flow Design 26](#_Toc487101830)

[Proposed Integration Design 29](#_Toc487101831)

[Service Overview 29](#_Toc487101832)

[Technical Integration Flow Design 29](#_Toc487101833)

[Current Technology Design 30](#_Toc487101834)

[Proposed Technology Design 31](#_Toc487101835)

[Technology Consumption Model 31](#_Toc487101836)

[Proposed Infrastructure Design (for on premise deployment only) 32](#_Toc487101837)

[Virtual Machines 32](#_Toc487101838)

[Network Design 34](#_Toc487101839)

[Overall Network Diagram 34](#_Toc487101840)

[Citrix (for on premise deployments only) 35](#_Toc487101841)

[Database (for on premise deployments only) 35](#_Toc487101842)

[Topology Diagram 36](#_Toc487101843)

[Monitoring and Alerting 39](#_Toc487101844)

[Shared Storage (NFS or CIFS) – On premise deployment only 39](#_Toc487101845)

[5. Proposed Security Design 40](#_Toc487101846)

[Security zones are an area of the architecture that comprises of systems and services that have similar security requirements and characteristics. 40](#_Toc487101847)

[Security Zones 40](#_Toc487101848)

[Security Zone Design 40](#_Toc487101849)

[Component Placement 41](#_Toc487101850)

[Inter Zone Communications 41](#_Toc487101851)

[Security Sub System Definitions 42](#_Toc487101852)

[Flow Control 42](#_Toc487101853)

[Flow Control Devices 42](#_Toc487101854)

[Trusted Credentials 43](#_Toc487101855)

[System Actors 43](#_Toc487101856)

[Authentication (Process) 43](#_Toc487101857)

[Single Sign On Configuration 43](#_Toc487101858)

[Directories (Naming and Structure) 43](#_Toc487101859)

[Access Control 44](#_Toc487101860)

[Authorisation 44](#_Toc487101861)

[User Administration 44](#_Toc487101862)

[Data Security 45](#_Toc487101863)

[Data Classification 45](#_Toc487101864)

[Data Loss Protection Controls 45](#_Toc487101865)

[Data Encryption 45](#_Toc487101866)

[Digital Rights Management 45](#_Toc487101867)

[Logging and Monitoring 46](#_Toc487101868)

[Security Information and Event Monitoring 46](#_Toc487101869)

[Security Audit 46](#_Toc487101870)

[File Integrity Monitoring 46](#_Toc487101871)

[Integrity 47](#_Toc487101872)

[Data/Media Handling (if applicable) 47](#_Toc487101873)

[Build Standards 47](#_Toc487101874)

[Encryption 47](#_Toc487101875)

[Digital Certificates 47](#_Toc487101876)

[Anti-Virus 47](#_Toc487101877)

[Hardware & Commissioning (for on premise deployment only) 47](#_Toc487101878)

[Decommissioning 47](#_Toc487101879)

[Compliance 48](#_Toc487101880)

[Assurance 48](#_Toc487101881)

[6. Operability 49](#_Toc487101882)

[Upgrade and Patch Management 49](#_Toc487101883)

[Backup and Restore 49](#_Toc487101884)

[Systems Management 49](#_Toc487101885)

[Orchestration & Automation (for on premise deployments only) 50](#_Toc487101886)

[7. Capacity & Sizing Methodology 51](#_Toc487101887)

[Capacity 51](#_Toc487101888)

[Sizing 51](#_Toc487101889)

[8. Non-Functional Design Considerations 52](#_Toc487101890)

[Availability 52](#_Toc487101891)

[Recovery Point Objective (RPO) in hours 52](#_Toc487101892)

[Recovery Time Objective (RTO) in hours 52](#_Toc487101893)

[9. Testing Considerations 53](#_Toc487101894)

[Test/Development Environment 53](#_Toc487101895)

[Pre-Production Environment 53](#_Toc487101896)

[Production Environment 53](#_Toc487101897)

[10. Design Dispensations 54](#_Toc487101898)

[11. Product Licencing Model, Warranty and renewal process 55](#_Toc487101899)

[Software Licencing Model 55](#_Toc487101900)

Table of Figures

[Figure 1 Example of Logical Data Model 14](#_Toc481595621)

[Figure 2 Example Logical Entity Relationship 15](#_Toc481595622)

[Figure 3 Example Entity Relationship 16](#_Toc481595623)

[Figure 4 Example Entity Relationship Diagrams 17](#_Toc481595624)

[Figure 5 Example Flow Diagram 18](#_Toc481595625)

[Figure 6 Example Data Flow Diagrams 19](#_Toc481595626)

[Figure 7 System XXX current data flow diagram 21](#_Toc481595627)

[Figure 8 Integration Flow Design Example 23](#_Toc481595628)

[Figure 9 Example of Technology Stack Diagram 24](#_Toc481595629)

[Figure 10 Example of Technology Stack Diagram 25](#_Toc481595630)

[Figure 11 Example Network Topology 29](#_Toc481595631)

[Figure 12 Example Logical Infrastructure Diagram 30](#_Toc481595632)

[Figure 13 Example IBM Rational Software Architect Deployment diagram 31](#_Toc481595633)

[Figure 14 Example of Visio Topology diagram 32](#_Toc481595634)

[Figure 15 – Example of Security Zones Levels of Trust 34](#_Toc481595635)

[Figure 16 - Defence in Depth 35](#_Toc481595636)

[Figure 17 Example of Flow Control Diagram 36](#_Toc481595637)

# Document Control

## Distribution List

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Organisation** | **Role** | **Email** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## References

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ref** | **Document** | **Author** | **Status** | **Version** |
| [1] | <project name> - BDV |  |  |  |
| [2] | <project name> - SAW |  |  |  |
| [3] | <project name> - ARS |  |  |  |
| [4] | <project name> - ADD |  |  |  |
| [5] | Business Case |  |  |  |
| [6] | Project Plan |  |  |  |
| [7] | Other |  |  |  |

## Terminology

Note – Please define the terms that are relevant and used in the context of this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Purpose & Background

## Executive Summary

This HLD document provides details of the xxxxx design…

Project or programme specific.

To clearly describe why the IT solution is to be designed and implemented.

## Design Objectives

The objective of this document is to describe the proposed high level design for xxxxxx covering the following:

E.g. :

The objective of this document is to define explicit instructions for the delivery of the xxx Software/System comprising of functional and non-functional requirements. This is inclusive of application/infrastructure design plus process requirements including policy compliance, testing, and governance.

Provide explicit architecture requirements for solution delivery

Define governance that will be applied over the delivery

## Design Assumptions

Table 1 List of Design Assumptions

|  |  |
| --- | --- |
| **Ref** | **Design Assumption** |
| DAS001 | E.g.: Model View Controller (MVC) Architectural pattern applied for the user interface implementation. |
| DAS002 | e.g. Server hardware redundancy is provided through the use of VMware HA and VMware vMotion |
| DAS003 |  |

## Design Dependencies

Table 2 List of Design Dependencies

|  |  |
| --- | --- |
| **Ref** | **Design Dependency** |
| DDP001 | e.g. Specify dependency of the current design ( E.g.: Any third party software integration ) |
| DDP002 |  |

## Design Risks

This section should contain the proposed solutions design risks. A design risk is any specific event which might occur and thus have a negative impact on this proposed design. All design risks should have mitigation against them. Do not include Project risks here!

Note - The standard project risks that have been documented in the standard project documentation (typically populated by a project manager/business) should not be repeated in this section. The above examples should be removed from the table.

Table 3 List of Design Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ref** | **Design Risk** | **Impact**  **(L/M/H)** | **Likelihood**  **(L/M/H)** | **Mitigation** |
| DRI001 | E.G. The solution has been scaled with 5 licences, the utilisation may be greater | L | L | Capacity utilisation will be managed by assigned Application Admin |
| DRI002 |  |  |  |  |
| DRI003 |  |  |  |  |

## Design Issues

A design issue is anything technical which arises within this proposed solution which one has to deal with in order to ensure that the design is successfully implemented into service.

Table 4 List of Design Issues

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref** | **Design Issue** | **Impact**  **(L/M/H)** | **Mitigation** |
| DIS001 | e.g. The deployed software version will become end-of-support within 2 years | M | Develop roadmap to replace solution before end of support |
| DIS002 | e.g. Solution proposes use of non-standard services/components | L | Time bound dispensation |
| DIS003 |  |  |  |

Note – Any design issues should be included in the central Design Issues Register. The above examples should be removed from the table.

## In-Scope

This section, at a high level, should include all the Solution design In Scope activities

* In Scope 1
* In Scope 2

E.g.:   
 In Scope Activities and Interfaces:

Actively track crew through specific zones in the CBC via RFID tracking

Monitor and alert controllers when crew have failed to progress to the next zone within specified tolerances

Monitor and alert controllers when facilities / hardware issues arise

Please also add the current system items that will be de-commissioned by this HLD.

## Out-of-Scope

* Out of Scope 1
* Out of Scope 2

E.g.:

Out of Scope Activities and Interfaces

Non-English language version of application user interface and/or operational manual/guidelines

Support for an airport other than AUH

Land-side and Air-side integration adapters to/from Etihad Integration Platform.

# Solution Design

This section should provide a high-level overview of how the functionality and responsibilities of the system were partitioned and then assigned to subsystems or components

## Current Application Design

Brief description of the application and its current purpose.

### Use Case Model Diagram

Expectation in this section is to show the use case model diagram from system architect showing the actors and use cases supported by the current system. The actors in the diagram must be the same names as the actors in the network concept diagram and the authorisation section in the security chapter.

Example



The actors list and use case list shown below should be exported from System Architect using a report based on the modelled diagram above.

**Actors**

| Actor | Description |
| --- | --- |
| ADGOV Member | This actor represents an Abu Dhabi Government employee who joins the Etihad Guest program. This actor has to prove that they are an employee of an Abu Dhabi government department to be able to join the virtual club for ADGOV. |

**Use cases**

| Use Case | Description |
| --- | --- |
| ON Agent authorization | For ON Agent login, SAML requests will be sent to OKTA Enterprise login instance directly (not via ESB). OKTA will respond with TOKEN, which will be submitted again to OKTA to get SSO initiated for AGENT. |

## Proposed Application Design

Brief description of the application and its intended purpose.

### Use Case Model Diagram

The full use case model diagram for the target state after this design should be included in this section. The actors added/removed and the use cases impacted by this design change should be listed in the same way as the current design sections (impacted objects only)

### Use Case Realisation Diagrams

Where realisation diagrams exist in the design documentation these can be linked here.

* BPMN diagrams
* Sequence diagrams
* State diagrams

Data Feeds that goes as input to the system

Data Feeds that goes out of the system and inputs to the other systems

|  |  |  |
| --- | --- | --- |
| **Inputs** | **System/Application** | **Outputs** |
|  |  |  |
|  |  |  |

Table 5 Current System Inputs and Outputs

## Current Information and Data Design

Information design is a standard and expected component for architecture definitions and design; this guidance applies to both section 3.2 and 4.2.

We require a dictionary that indicates:

The informational payload for all interfaces (of whatever flavour and technical implementation)

The Informational content of any system hosting Etihad or Group data

For such content we expect:

Business semantics for all business data (i.e. what does this attribute mean as defined by the host system?)

Physical characteristics for the data (i.e. type, size etc.)

Explicitly stating this metadata indicates that the SI and system provider fully understand their underlying data and that the design and implementation is based on known facts not guesswork (educated or otherwise).

### Data Dictionary – In Scope

This section should include the Data element level of detail with complete structural and Semantic definitions. Specify this content.

### Information domain decomposition

This defines an area of scope or type of information, relevant and in-scope to the project. E.g. PNR structural decomposition, at the level of high level information types (Itinerary)

### Data Entity Model

Logical [entities/classes, attributes, definitions that conform to standards, metadata/data dictionary incorporation]

Note – a formal entity relation diagram or class diagram where a HLD specifies an implementation such as a database or XML schema messages etc. A diagram to help identify scope is optional but recommended



Figure 2 Example Logical Entity Relationship

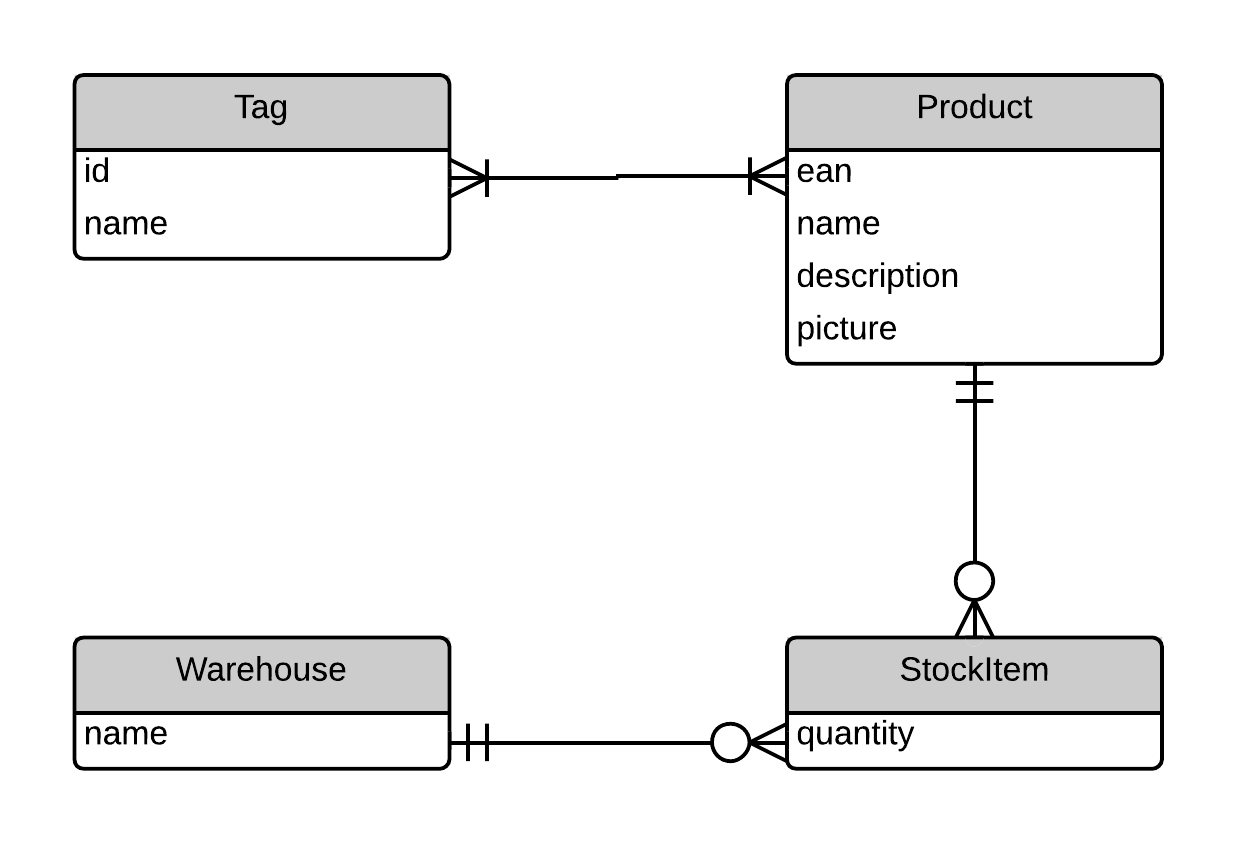


Figure 3 Example Entity Relationship



Figure 4 Example Entity Relationship

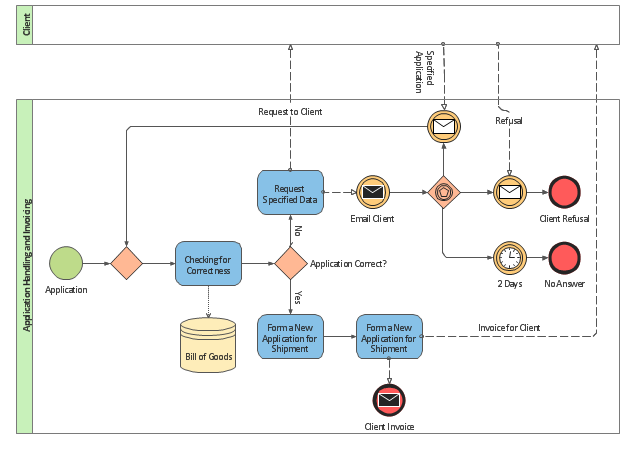


Figure 5 Example Flow diagram

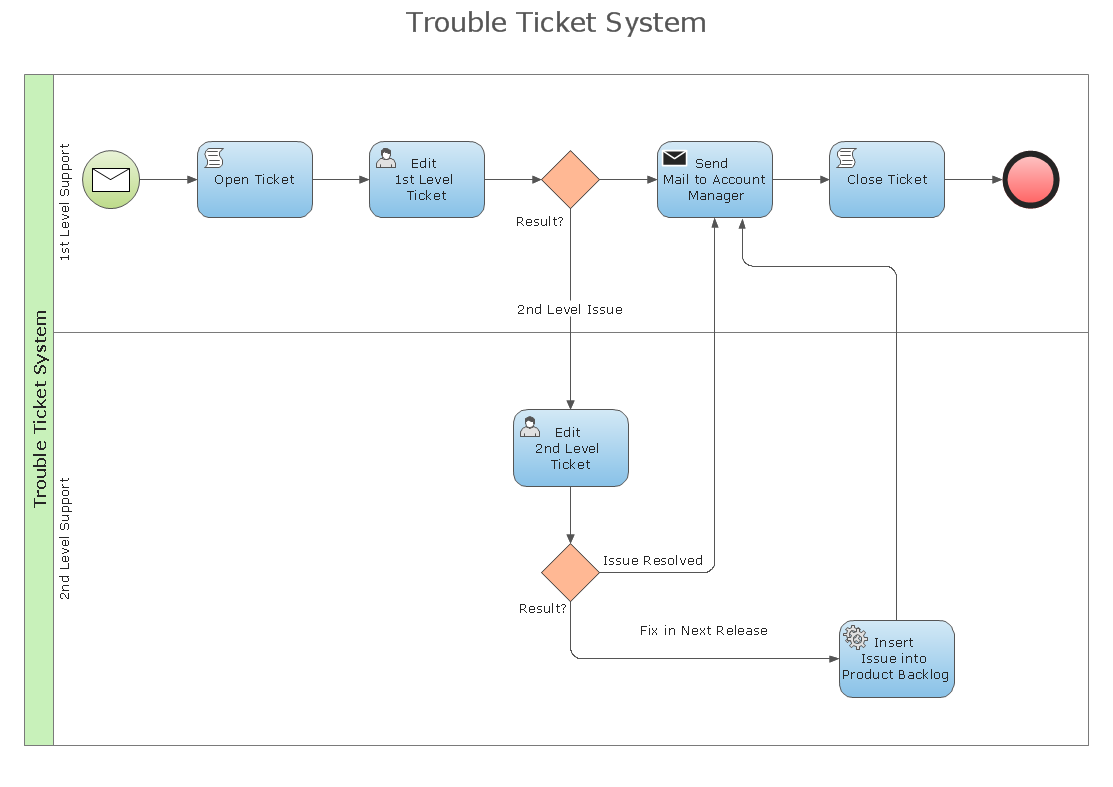


Figure 6 Example Business Flow diagram

### Data Input and Output

Data Feeds that goes as input to the system

Data Feeds that goes out of the system and inputs to the other systems

## Proposed Information and Data Design

### Data Dictionary – In Scope

This section should include the Data element level of detail with complete structural and Semantic definitions. Specify this content. Note that this should document the changed elements only.

Dictionary of Ambiguous and Uncommon Terms

|  |  |
| --- | --- |
| **Business Terms** | **Description** |
| e.g. Ticket | Voucher to board aircraft for a specific journey |
| e.g. Origin | Starting point for buses to begin their journey |

### Logical Data Entity Model/Diagram

Logical [entities/classes, attributes, definitions that conform to standards, metadata/data dictionary incorporation]

Note – a formal entity relation diagram or class diagram where a HLD specifies an implementation such as a database or XML schema messages etc.



Figure 1 Example of Logical Data Model

### Information domain decomposition

This defines an area of scope or type of information, relevant and in-scope to the project. E.g. PNR structural decomposition, at the level of high level information types (Itinerary)

### Data Entity Model

Logical [entities/classes, attributes, definitions that conform to standards, metadata/data dictionary incorporation]

Note – a formal entity relation diagram or class diagram where a HLD specifies an implementation such as a database or XML schema messages etc. A diagram to help identify scope is optional but recommended



Figure 2 Example Logical Entity Relationship

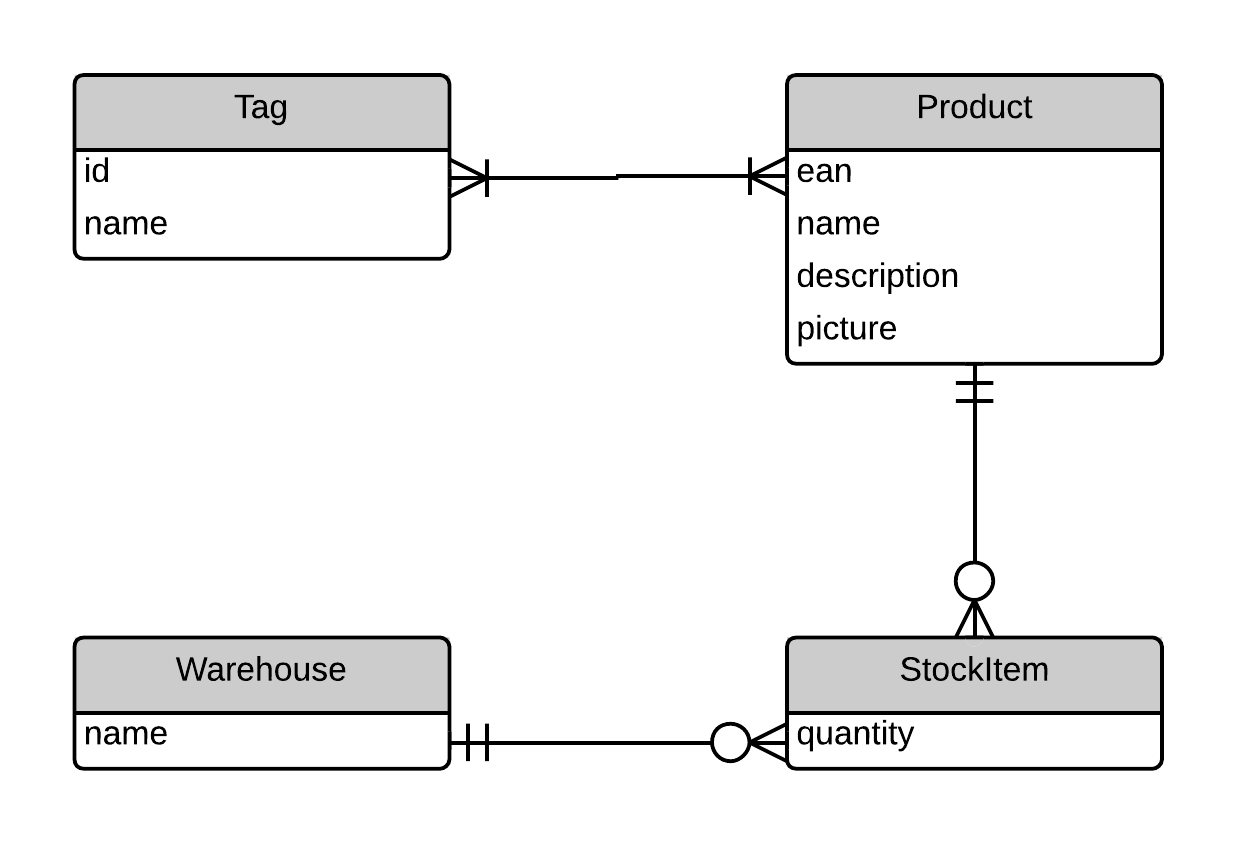


Figure 3 Example Entity Relationship



Figure 4 Example Entity Relationship Diagrams

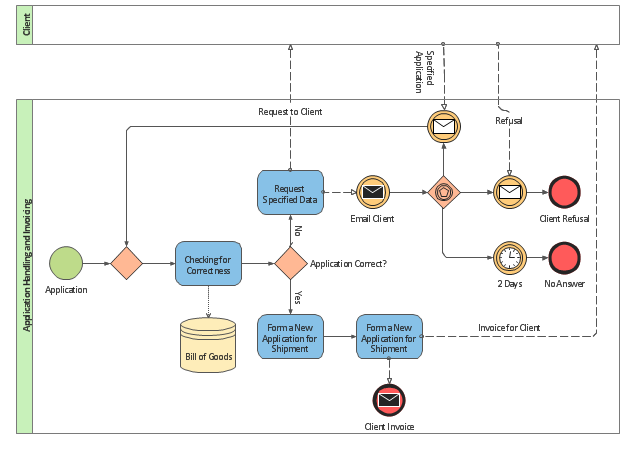


Figure 5 Example Flow Diagram

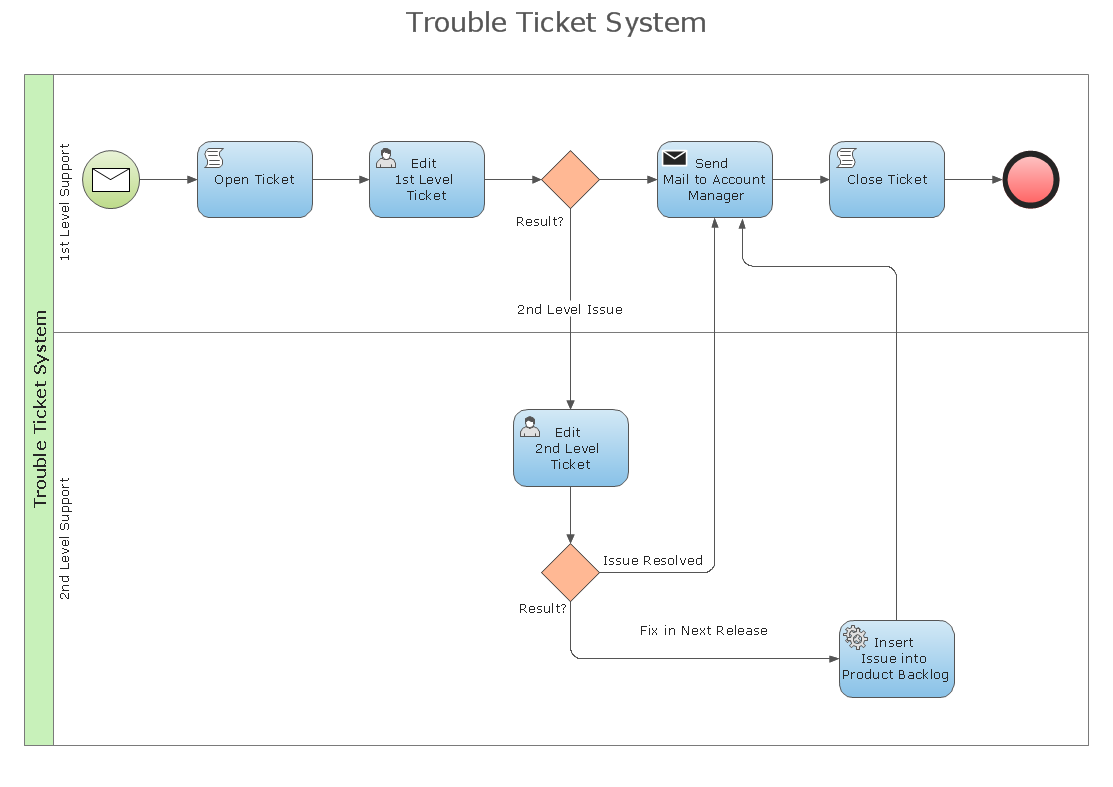


Figure 6 Example Data Flow Diagrams

### Proposed System Data Input and Output

Data Feeds that goes as input to the system

Data Feeds that goes out of the system and inputs to the other systems

|  |  |  |
| --- | --- | --- |
| **Inputs** | **System/Application** | **Outputs** |
|  |  |  |
|  |  |  |

## Current Integration Design

### Service Overview

This section contains a brief description of the services used or provided by this solution.

The list of services here should be extracted via a report from the data flows matched with the interface list in System Architect. It is important that the services name provider and consumer are all using names aligned with the System Architect names. Interfaces listed here should be web services and file.

|  |  |  |  |
| --- | --- | --- | --- |
| **Service name** | **Provider** | **Consumer** | **Synch./Asych./Batch** |
|  |  |  |  |
|  |  |  |  |

Table 5 Current List of Integration Services

### Technical Integration Flow Design

This section should describe the general integration flow for the solution.

This section should contain the data flow diagram from System Architect followed by a table of the data flows in place.

Use sequence diagrams, BPMN diagrams, and text to describe the solution.



Figure 7 System XXX current data flow diagram

| **Name** | **Provider** | **Consumer** | **Description** | **Feed Type** | **Format Type** | **Transmission Frequency** | **Transmission Mechanism** | **Data Type** | **Data Classification** | **PII** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| EY0360-EY0950 (Re-credit Miles, WS, on demand) | Etihad Guest Reward Card | Group Loylogic ON Payment | If the same amount of points is re-credited, the original transaction and re-credit are not to be displayed to the member on their statement/s unless it has already been shown before the time of re-credit. | Text - other | XML | 10 On Demand/On Request | WS - Web Service | Re-credit Miles | Confidential | F |

## Proposed Integration Design

### Service Overview

This section contains a brief description of the services used or provided by this solution.

This section should only list new/changed services.

|  |  |  |  |
| --- | --- | --- | --- |
| **Service name** | **Provider** | **Consumer** | **Synch./Asych./Batch** |
|  |  |  |  |
|  |  |  |  |

Table 6 List of Proposed Integration Services

### Technical Integration Flow Design

This section should describe the general integration flow for the solution.

Use sequence diagrams, BPMN diagrams, and text to describe the solution.

This section should have the target data flow diagram and table describing the interfaces.



Figure 8 Integration Flow Design Example

## Current Technology Design

Brief description of the technology used to support the application.

Use Technology Stack diagram. It should be possible from the diagram to understand what technology stack is installed on each VM/Physical machine.

Network concept diagrams for topology overview

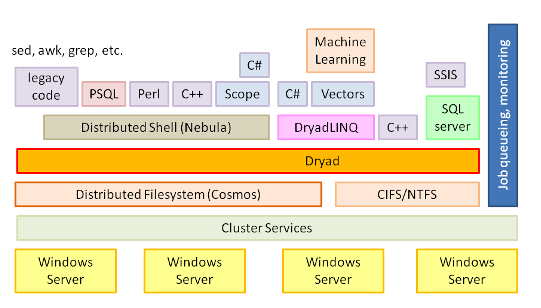


Figure 9 Example of Technology Stack Diagram

E.g.:

Windows based application for automating the check-in process. The application is developed in .NET and the UI is touch based. Apart from the kiosk application there is also a desktop management console.

## Proposed Technology Design

Brief description of the technology used to support the application.

Use Technology Stack diagram

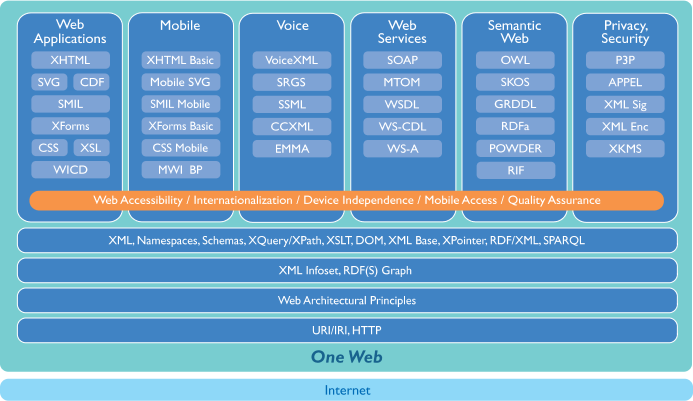


Figure 10 Example of Technology Stack Diagram

Network concept diagrams for topology overview

E.g.:

Windows based application for automating the check-in process. The application is developed in .NET and the UI is touch based. Apart from the kiosk application there is also a desktop management console.

### Technology Consumption Model

* Multi-tenant
* Multi-data centre (Active Active)
* SaaS
* PaaS

## Proposed Infrastructure Design (for on premise deployment only)

### Virtual Machines

| **System** | **Requirements** | **Application Server** | **Database Server** | **Interface Server**  **(SOA/Automation)** |
| --- | --- | --- | --- | --- |
| **System 1 Production Environment** | *Number of virtual servers* | N servers with the below configuration |  |  |
| *Storage Sizing/ Drive* | / - 16 GB |  |  |
| /boot – 2 GB |
| Swap – 16 GB |
| /opt – 10 GB |
| /tmp – 5 GB |
| /oradb – 30 GB |
| /oradbf – 265 GB |
| *vRAM* | 8GB |  |  |
| *vCPU* | 2 x 2.5Ghz Cores or 2 x equivalent vCPUs |  |  |
| *Operating System* | RedHat Enterprise Linux 6.4 (64-bit) |  |  |
| *Context* | *e.g. Common/Corporate/management/Business/PKI/VDI/Apps Services* |  |  |
| *State* | *Standalone/Clustered* |  |  |

Table 7 Proposed Production environment Server configuration

| **System** | **Requirements** | **Application Server** | **Database Server** | **Interface Server**  **(SOA/Automation)** |
| --- | --- | --- | --- | --- |
| **System 1**  **Pre-Production**  **Environment** | *Number of virtual servers* | n |  |  |
| *Storage Sizing/ Drive* | / - 16 GB |  |  |
| /boot – 2 GB |
| Swap – 16 GB |
| /opt – 10 GB |
| /tmp – 5 GB |
| /oradb – 30 GB |
| /oradbf – 265 GB |
| *vRAM* | 8GB |  |  |
| *vCPU* | 2 x 2.5Ghz Cores or 2 x equivalent vCPUs |  |  |
| *Operating System* | RedHat Enterprise Linux 6.4 (64-bit) |  |  |
| *Context* | *e.g. Common/Corporate/management/Business/PKI/VDI/Apps Services* |  |  |
| *State* | *Standalone/Clustered* |  |  |

| **System** | **Requirements** | **Application Server** | **Database Server** | **Interface Server**  **(SOA/Automation)** |
| --- | --- | --- | --- | --- |
| **System 1 Test Environment** | *Number of virtual servers* | n |  |  |
| *Storage Sizing/ Drive* | / - 8 GB |  |  |
| /boot – 2 GB |
| Swap – 8 GB |
| /opt – 5 GB |
| /tmp – 5 GB |
| /oradb – 15 GB |
| /oradbf – 100 GB |
| *vRAM* | 4GB |  |  |
| *vCPU* | 2 x 2.5Ghz Cores or 2 x equivalent vCPUs |  |  |
| *Operating System* | RedHat Enterprise Linux 6.4 (64-bit) |  |  |
| *Context* | *e.g. Common/Corporate/management/Business/PKI/VDI/Apps Services* |  |  |
| *State* | *Standalone/Clustered* |  |  |

Table 9 Proposed Test Environment Server Configuration

### Network Design

Bandwidth requirements

Load Balancing design (if applicable)

Segmentation and segregation (firewall)

External connectivity requirements e.g. Proxied Internet

Network concept diagram should be modelled in system architect. All the actors defined in the use case sections should be represented on the network concept diagram either by name or class or group of actor.

### Overall Network Diagram

To provide an overall Network Diagram of the communication.

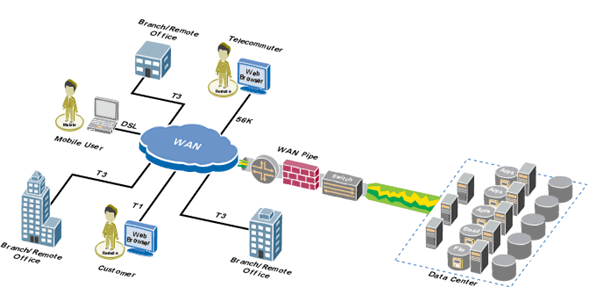
[](http://www.google.ae/url?sa=i&rct=j&q=&esrc=s&frm=1&source=images&cd=&cad=rja&uact=8&ved=0CAcQjRxqFQoTCPiQgcTu0MgCFUF-GgodInQOng&url=http://capitalhead.com/solutions/network-design.aspx&psig=AFQjCNGk-Nx5GHjbw-0uydAMymS-tApXRA&ust=1445424102558196)

Figure 11 Example Network Topology

### Citrix (for on premise deployments only)

Specify if end-user access is as a URL or a published Client App

Client design document E.g. Client backend connectivity details, ODBC, Oracle TNSNAMES.ORA details etc.

Client prerequisites. E.g. Supported OS, System minimum requirement CPU, RAM, HDD, Privileges required to run the client, Network Drive mapping, File Shares, Java versions, supported browser versions etc.

Client license type E.g. Single user license, Multiuser license, MAC based license, Online activation required, etc.

Document total numbers of users and concurrent users expected to use the application

Microsoft Exchange IntegrationE-Mail Integration

Exchange Version

Mail Relay (External / Internal)

Email Notifications

WorkFlows

Calendar integration

### Database (for on premise deployments only)

Database – Oracle/ MS SQL Server 2012

Database Schema details

Database Size and growth Estimates

Initial design estimates

Growth Estimates

Clustering

Database Disaster Recovery Plan

Database Backup Policies

Monitoring Database Details

This section should focus on how the business application will integrate into the underlying Infrastructure. This needs to include the following diagrams:

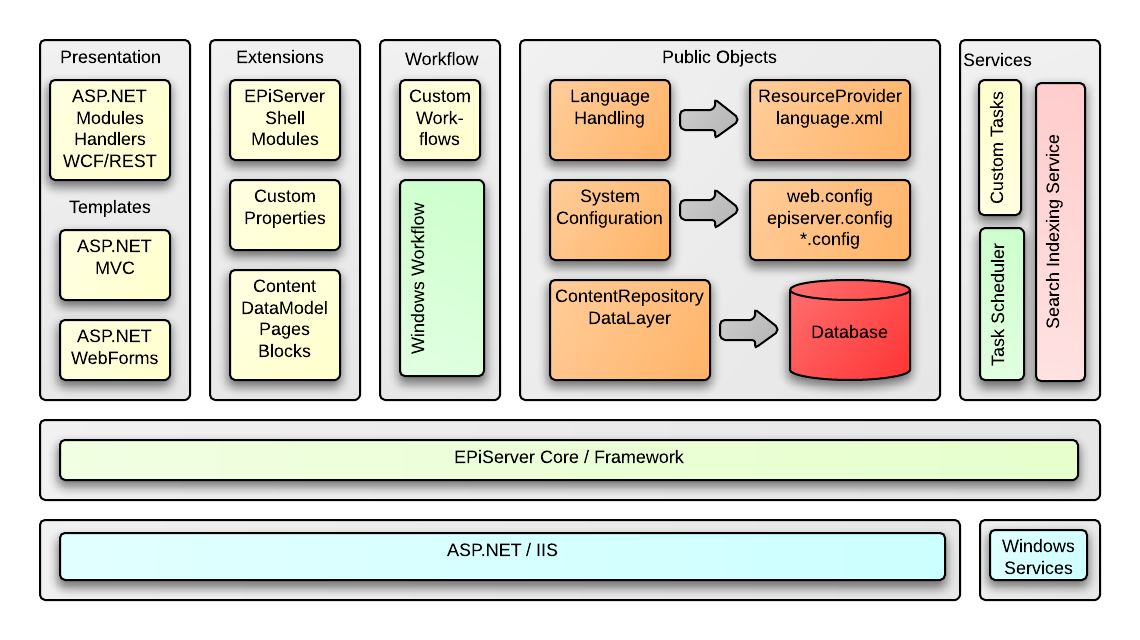


Figure 12 Example Logical Infrastructure Diagram

## Topology Diagram

The following diagram shows the main components in the system.

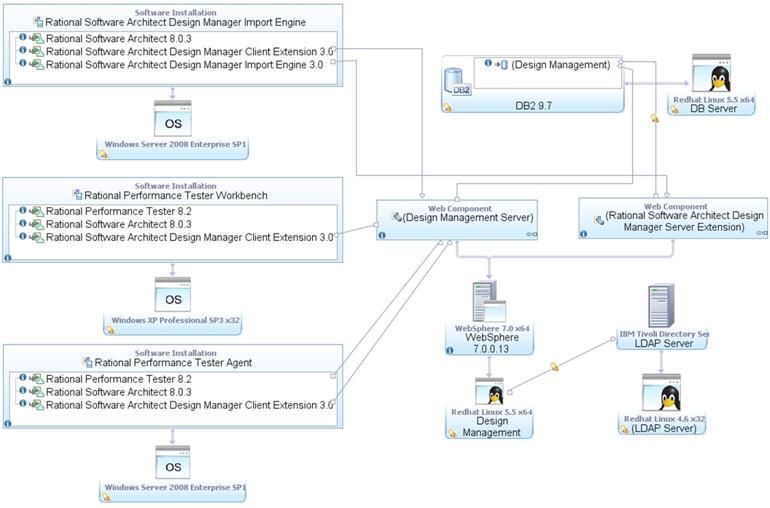


Figure 13 Example IBM Rational Software Architect Deployment diagram

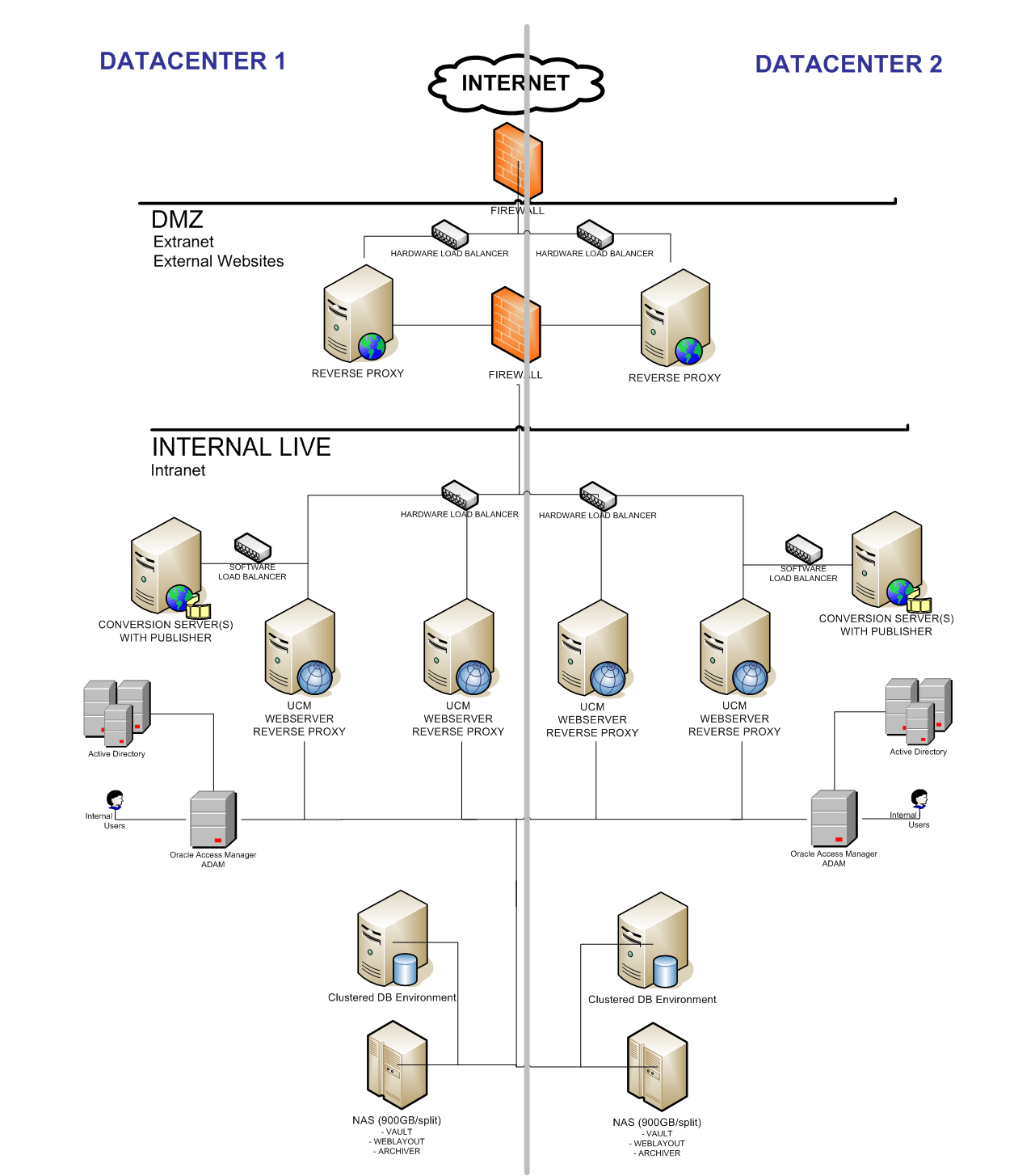


Figure 14 Example of Visio Topology diagram

*In this topology:*

*Forefront UAG is configured as a relying party of the corporate AD FS 2.0 server (Resource Federation server in the diagram).*

*A separate Active Directory Domain Services (AD DS) server is used within the corporation; however, you can configure AD FS 2.0 to run on your AD DS server.*

*A web application has been published through Forefront UAG. You can publish the application using either host address translation (HAT) or alternate publishing name (APN).*

### Monitoring and Alerting

Identify the key Components (Hardware and Software) to be included in pro-active monitoring.

Describe monitoring and alerting services in your environment.

Where the solution is on premise it should cover IT360 and AppDynamics integration.

### Shared Storage (NFS or CIFS) – On premise deployment only

Estimate storage capacity

Estimate storage performance requirements (e.g. IOPS)

# Proposed Security Design

The security design provides an end-to-end view of the proposed solution from a security perspective. It identifies all components within the solution, shows where components are placed in terms of security zones and describes how each component interacts with one another.

## Security zones are an area of the architecture that comprises of systems and services that have similar security requirements and characteristics.

## Security Zones

<Note the security zones are generally already defined within an infrastructure, if this is the case, then document and describe the zones only, however the rules governing the zones will still affect inter zone communications, zoning principles and the connectivity matrix>

### Security Zone Design

Four categories of zone have been used to represent the security requirements of the architecture. These are named and colour coded according to the level of trust associated with each zone as shown in the diagram below:

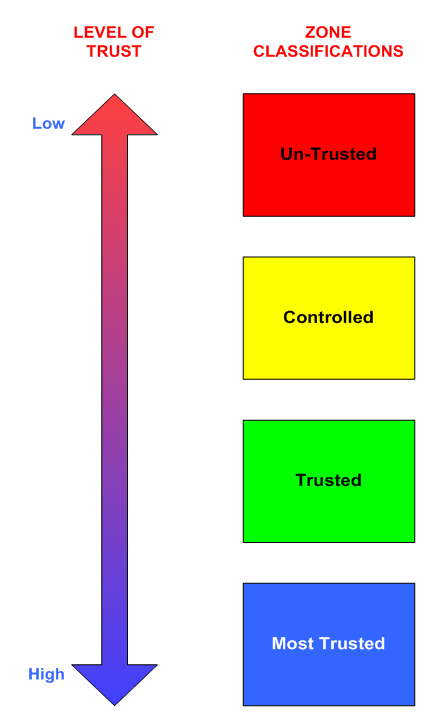


Figure 15 – Example of Security Zones Levels of Trust



Figure 16 - Defence in Depth

### Component Placement

Analysis of components and their locations must be performed in order to understand the environment and to identify/verify security risks. The colour coding of the zones corresponds to different levels of trust in the zones, with the following criteria:

* Un-trusted (Red)– Not under CLIENT control, considered a threat to the CLIENT infrastructure. e.g. Third Parties not controlled by CLIENT
* Controlled (Yellow)– Under CLIENT control, application proxies, Internet DMZ’s, boundaries between trusted and uncontrolled zones.
* Trusted (Green)– Under CLIENT control, and more trusted than Controlled as there are a limited number of threats to the zone coupled with good physical and logical security. However critical business data should not normally be stored in this type of zone.
* Most Trusted (Blue)– Under CLIENT control, contains Critical business data.

### Inter Zone Communications

Any communications that are against Information Security policy must be raised as an exception with IT Security.

The inter zone communications for the project, and in general, between security zones works on the principle that data that flows between the zones must be explicitly permitted to ensure that the data is not accidentally made available to areas of the architecture where the level of trust is low.

## Security Sub System Definitions

## Flow Control

The Information Flow Control Security Sub-system is responsible for gating the flow of information within a computing solution. This can be achieved physically or virtually at the Network, OS or Application level. This section identifies what entities need to communicate with one another, the direction or path that the flow of traffic will follow and information on the protocols/services that will be used. Note that secure protocols (i.e. SSH, SNMPv3, SSL, TLS, HTTPS etc.) must be used wherever possible.

Network flow control components are typically placed at the boundary of security zones; generally this is either built in to network switches, routers or dedicated hardware (e.g. firewalls). Flow control components can also include remote access devices such as VPN concentrators, application proxies/content filtering, and remote dial in servers. The Web Services Security Gateway devices are used as application level flow control devices to control inter zone access

### Flow Control Devices

In order to control access between zones, there is a specific requirement to provide sufficient flow controls between zones and inter zone communications. List flow control devices below.

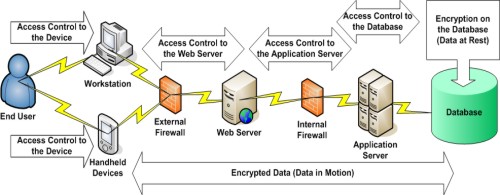


Figure 17 Example of Flow Control Diagram

The functions available via each device type can be summarised as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **Authorisation based on source addresses** | **Event logging** | **User Authentication (application layers)** | **User authentication (network layers)** |
| E.g. Network Firewall | Y | Y | N | **N** |
| E.g. Web Services Security Gateway | Y | Y | Y | **Y** |
| E.g. Web Application Firewall | N | Y | N | N |

Table 13 Flow Control Devices

Details of any rules that need to be configured on firewalls must be provided. Idle sessions should also be configured to timeout after a reasonable period of time. (For On premise deployments)

## 

## Trusted Credentials

A fundamental part of access control systems is that they require a trusted method to provide authentication or identify if a person or entity is who they say they are. Credentials such as token, certificate, password etc. are used to demonstrate this.

This section provides details of the components within the solution that will be authenticated i.e. users, systems etc. and describes how the authentication process takes place for each component within the solution.

### System Actors

|  |  |
| --- | --- |
| **Actor Name** | **Purpose** |
| e.g. Administrator | Management of user accounts and system configuration |
| e.g. Service Account | Machine account to automatically send messages between interfaces |

### Authentication (Process)

### Single Sign On Configuration

### Directories (Naming and Structure)

## Access Control

Once the identity of a user or system has been verified or authenticated, access control systems provide a level of granularity to control what the entity is able to access. This section describes how levels of access to resources are controlled. It is expected that areas such as permissions, access rights, groups etc. will be covered within this section.

### Authorisation

The mechanism to authorise all actors (note this should include all actors defined in the use case model) should be detailed here for all areas of the system (application, database, file transfer, web services etc.)

### User Administration

The mechanisms to provide access to all parts of the solution (application, database, web service, file transfer and service users) should be considered here. This section should include user provisioning, de-provisioning and periodic access right reviews.

## Data Security

The data security sub-system is responsible for ensuring the data within the system is properly classified and controls are applied that are consistent with that classification. This includes data at rest, data in transit and data in use.

### Data Classification

The data classifications used here should be the same as the information architecture section and the data flow diagram table. This report should effectively be an export from system architect.

|  |  |
| --- | --- |
| **Data Type** | **Classification** |
| Employee medical records | Secret |
| Employee contact details | Confidential |
| Bus routes  Bus schedules | Restricted |
| Published flight schedules | Public |

### Data Loss Protection Controls

How is sensitive data loss/leakage prevented?

### Data Encryption

What is the encryption standard applied to protect the data?

### Digital Rights Management

If applicable

## Logging and Monitoring

The logging and monitoring security sub-system is responsible for capturing, analysing, reporting, archiving and retrieving records of events and conditions within the security architecture. This includes both real time and after-the-fact monitoring. This section provides an overview of the monitoring capabilities incorporated into the solution, describes the areas that will be actively monitored and will also identify what will happen to alerts or audit logs that are generated. It also provides information on how clocks will be synchronised, where logging and monitoring information will be stored and for how long.

### Security Information and Event Monitoring

Details of the integration of this solution into the Qradar SIEM system should be provided. If the solution is hosted by an external party, details of whether they will ship us the logs or not are required. If the hosting party will not provide logs then details of how their SIEMs processes will be integrated with the Etihad SOC (security operation centre) will be performed.

### Security Audit

Detail what audit logging of changes is available in the solution and how this data can be extracted/reviewed by independent parties.

### File Integrity Monitoring

## Integrity

The Integrity security sub-system is responsible for maintaining the correct and reliable operation of critical components and processes within the computing solution. An Integrity security sub-system can be a discrete set of components or a coordinated set of mechanisms among the several components in the solution. The Integrity security sub-systems can also include business operating procedures, processes, standards, and integration with existing enterprise solutions not dedicated to the security architecture such as backup services for example.

Other typical areas that are covered by the integrity security subsystem are highlighted below:

### Data/Media Handling (if applicable)

Data/Media handling highlights the type of data will be will be stored within the system such as personal information, credit card details etc. and where data will be stored. It also provides an overview of how this information will be handled in terms of who will have access to it and from what locations i.e. internal/external, describes whether or not there is a need to exchange information or media with third parties and if so how this would take place.

### Build Standards

This provides information on how systems are built. Any specific areas where alterations to current build standards are required, such as making necessary changes to hardened settings for example, will be detailed.

### Encryption

This provides details of any encryption services that may be used within the system to protect data at rest and/or data in transit. It will cover areas such as what will be encrypted and how. This will include information relating to the products or services used to encrypt data, algorithms, key lengths used etc.

### Digital Certificates

This provides details on where and for what purpose digital certificates will be used together with how they will be issued and revoked, what the certificate’s life span is and what to do should a certificate needs to be re-issued should also be included.

### Anti-Virus

This provides information relating to what AV measures have been introduced as part of the systems.

### Hardware & Commissioning (for on premise deployment only)

This provides a summary of the hardware components that will be used, where they will be sourced from e.g. re-use of existing kit, purchase of new kit etc., describe how these components will be commissioned and what process will be followed to populate the system with data?.

### Decommissioning

This describes what areas need to be considered for decommissioning equipment or sending components off for repair , reuse, disposal etc. i.e. what needs to be wiped, how data will be sanitised if appropriate, how data/configuration settings will be wiped, prior to sending equipment away for repair/reuse/disposal

### Compliance

This provides a summary of how the systems achieve compliance e.g. it will provide a summary of the security policies (ITSPols) that have been followed as part of the system design, what best practice/standards have been followed e.g. CIS Benchmarks etc., what legal requirements does the system complies with i.e. data protection, computer misuse, PCI-DSS etc.

### Assurance

This provides a summary what will be assured i.e. identify the assurance boundary.

# Operability

## Upgrade and Patch Management

Server Upgrades

Client Upgrades

## Backup and Restore

System configuration and Data

Backup and Retention Policy

## Systems Management

Installation and upgrade of system software

Reporting the health of system components/services

Stopping and starting system components/services

Maintaining log and temporary files

## Orchestration & Automation (for on premise deployments only)

Provide details of automation and orchestration techniques used to automate deployment of solution for Etihad Airways.

# Capacity & Sizing Methodology

## Capacity

Provide calculations for capacity planning (volumetrics)

Capacity Management (how do we manage capacity – e.g. user based, some standard units etc..)

## Sizing

Provide calculations for environmental sizing (server/storage/network)

Include Design limits of this solution (concurrency, transactions etc..)

# Non-Functional Design Considerations

## Availability

How has this been designed for our planned and unplanned downtime.

Provide Availability SLA in 9’s

## Recovery Point Objective (RPO) in hours

RPO (Recovery Point Objective) refers to the amount of data at risk. It's determined by the amount of time between data protection events and reflects the amount of data that potentially could be lost during a disaster recovery. The metric is an indication of the amount of data at risk of being lost

## Recovery Time Objective (RTO) in hours

RTO (Recovery Time Objective) is related to downtime. The metric refers to the amount of time it takes to recover from a data loss event and how long it takes to return to service. RTO refers then to the amount of time the system's data is unavailable or inaccessible preventing normal service.

# Testing Considerations

All design solutions that will be deployed or will utilise Etihad downstream systems must follow the Etihad route-to-live process of Testing/Development, Pre-Production and Production.

**This section should ONLY define any specific/unique testing considerations within each of the environments**.

## Test/Development Environment

Unit Testing

System Testing

## Pre-Production Environment

System Integration Testing/ Interface testing

User Acceptance Testing

Regression Testing

Non Functional Testing (Load Testing)

## Production Environment

Smoke Testing

# Design Dispensations

This section contains any design dispensations that are required due to not adhering to the Etihad Architecture Principles[X] or Etihad Policies.

Any design dispensation that that are required will need Etihad Architecture Review Board (ARB) approval and will be logged in the central Etihad Enterprise Architecture repository

Table below can be generated from System Architect.

Table 14 List of Design Dispensations

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Title** | **Design Dispensation** | **Owned by** | **Related Org Unit** | **T&I Owner** | **Conformance Deadline** | **Impact** | **Justification** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

# Product Licencing Model, Warranty and renewal process

## Software Licencing Model

License Type

License Consumption

License Validity

License Renewal Process