|  |
| --- |
|  |



|  |
| --- |
| **Application Architecture**  Audit Online  Client: Global Audit Technologies - Princeton, NJ  Author: Jignesh Patel  Version: 3.5  Date: 21 May 2020  File Name: Detailed Architecture - Audit Online v3.5.docx |

##### **Document Control Information**

|  |  |
| --- | --- |
| Document Name | Application Architecture |
| Project Name | Audit Online |
| Client | Global Audit Technologies - Princeton, NJ |
| Document Author | Patel, Jignesh |
| Document Version | 1.0 |
| Template Version | 1.1 |
| Status | Draft |
| Architecture Approval |  |
| Dev. Manager Approval |  |
| Date Released |  |
| Electronic File Name | Detailed Architecture - Audit Online v3.5.docx |
| TPT/TPX Code | TP |

##### **Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Additions/Modifications | Prepared/Revised By |
| 1.0 | 05/21/2020 | Initial Draft | Jignesh Patel |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

##### **Distribution of Final Document**

The following people are designated recipients of the final version of this document:

|  |  |
| --- | --- |
| Name | Organization |
|  |  |
|  |  |
|  |  |

##### **Approval of Final Document**

These signatories confirm that this document will serve as the Application Architecture Document for the project.

|  |  |
| --- | --- |
| Name | Signature / Date |
| <Architect> |  |
| <Architect Channel Lead> |  |
| <Product Owner> |  |
| <Project Manager> |  |
| <Development Lead> |  |

****

[1. Introduction 5](#_Toc513123024)

[1.1 Executive Summary 5](#_Toc513123025)

[1.2 Business Value 5](#_Toc513123026)

[1.3 Scope 5](#_Toc513123027)

[1.4 Definitions Acronyms and Abbreviations 6](#_Toc513123028)

[1.5 References 6](#_Toc513123029)

[1.6 Architectural Views 6](#_Toc513123030)

[2 Architectural Goals 8](#_Toc513123031)

[2.1 Architectural Goals 8](#_Toc513123032)

[2.2 Architectural Drivers 8](#_Toc513123033)

[2.3 Technology and Infrastructure 9](#_Toc513123034)

[2.3.1 Design and Development Technologies 9](#_Toc513123035)

[2.3.2 Production Environment Technologies 9](#_Toc513123036)

[3 Secure Software Development Lifecycle (SSDLC) requriements 10](#_Toc513123037)

[3.1 Security Requirement at Application Level 10](#_Toc513123038)

[3.2 Security/Controls Design Guideines 10](#_Toc513123039)

[3.3 Security Testing Results 11](#_Toc513123040)

[4 Conceptual View 12](#_Toc513123041)

[4.1 Conceptual System Diagram 12](#_Toc513123042)

[4.2 Data Flow Diagrams 13](#_Toc513123043)

[5 DevOps 14](#_Toc513123044)

[5.1 CI/CD Detail 14](#_Toc513123045)

[5.2 Web Analytics 14](#_Toc513123046)

[5.3 Instrumentation and Logging 14](#_Toc513123047)

[5.4 Instrumentation Design 18](#_Toc513123048)

[6 Use Case View (Conceptual View) 20](#_Toc513123049)

[6.1 Personas 20](#_Toc513123050)

[6.1.1 Persona Definition 20](#_Toc513123051)

[6.1.2 Persona Hierarchy 20](#_Toc513123052)

[6.2 Use Case 22](#_Toc513123053)

[6.2.1 Use Case Definition 22](#_Toc513123054)

[6.2.2 Use Case Diagram 22](#_Toc513123055)

[7 Logical View 24](#_Toc513123056)

[7.1 Subsystem Decomposition 24](#_Toc513123057)

[8 Process View 25](#_Toc513123058)

[8.1 Services Decomposition Model 26](#_Toc513123059)

[9 Infrastructure View 27](#_Toc513123060)

[9.1 Deployment View 27](#_Toc513123061)

[9.2 Infrastructure Diagram (Physical View) 28](#_Toc513123062)

[9.3 Business Continuity Planning (BCP) 29](#_Toc513123063)

[9.3.1 BCP Model 29](#_Toc513123064)

[9.3.2 BCP Standards 29](#_Toc513123065)

[10 Data View 31](#_Toc513123066)

[10.1 Entity/Logical Data Model 31](#_Toc513123067)

[10.2 Data Integration and Classification 32](#_Toc513123068)

[10.2.1 Data Domain 32](#_Toc513123069)

[10.2.2 Data Source – Core System 32](#_Toc513123070)

[10.2.3 Data Destination 34](#_Toc513123071)

[10.2.4 Data Transfer Method 34](#_Toc513123072)

[10.2.5 Data Retention 35](#_Toc513123073)

[10.2.6 Data Classification 35](#_Toc513123074)

[11 Non-Functional Attributes 37](#_Toc513123075)

[12 Glossary 39](#_Toc513123076)

# Introduction

The goal of this document is to provide the architecture and design for Audit Online v3.4.

## Executive Summary

Engagement Management System (EMS) is widely adopted by all the member firms and is being used by most of Deloitte’s audit practitioners. EMS will be the audit platform for next couple of years (after which a next-gen Audit platform is expected to commence wider deployment). Audit Online extends the current EMS capabilities and provides interim solution to business needs per Audit Transformation.

## Business Value

The main objective of Audit Online is to provide an interim audit transformation solution to address current needs on the EMS platform until the next-gen Audit platform is available for Global member firms.

## Scope

This subsection specifies the scope of the project and architectural changes. Reflects what the team has agreed to deliver.

GRA Improvements

Levvia-Audit Online Integration for RADC

## Definitions Acronyms and Abbreviations

This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the Application Architecture Document.  This information may be provided by reference to the project’s Glossary.

|  |  |
| --- | --- |
| Definition | Description |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Acronyms | Description |
|  |  |
|  |  |

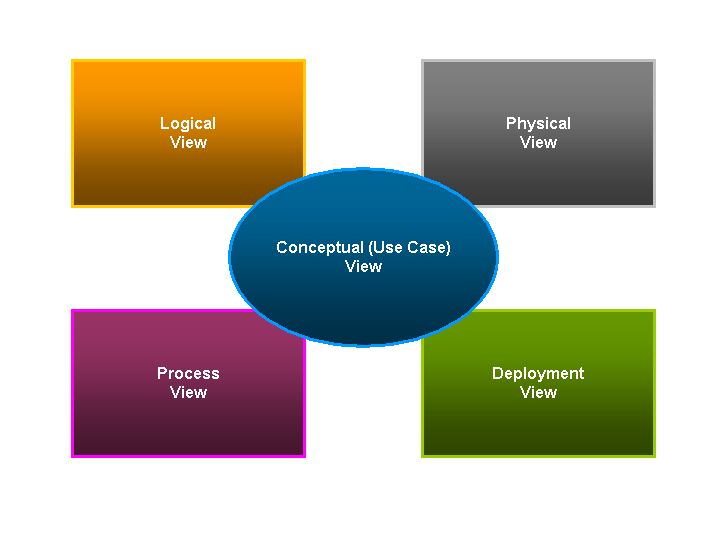
|  |  |
| --- | --- |
| Abbreviation | Description |
|  |  |

## References

This subsection provides a complete list of all documents referenced elsewhere in the Application Architecture Document. Identify each document by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.

|  |  |
| --- | --- |
|  |  |
| * + 1. Proposed Architecture Review Template | <https://applicationstandards.deloitte.com/app/core/governance/proposed> |
| Cyber Assessment Questionnaire | <https://deloittenet.deloitte.com/sites/MyTechnology/Pages/Security/SSDLC.aspx> |
| Application Standards | <https://applicationstandards.deloitte.com> |
|  |  |
|  |  |

## Architectural Views



The Views describe architecture from the Viewpoint of different stakeholders. Four primary Views are described (Logical, Development, Process, Physical) and augmented with a Scenario (Use Cases) "plus 1" View which are described in below sections 6-9.

# Architectural Goals

The architecture goals section provides a description of the goals of this solution architecture and highlights the constraints for which it must operate under.

## Architectural Goals

|  |  |
| --- | --- |
| Functionality | e.g. “Provide basic functionality to get started as a pilot project with consulting and advisory groups and extend it over a period of time to entire Deloitte US businesses.”   * e.g. Acceptance on UI mockups and navigation flow * e.g. User Acceptance Criteria from user stories * e.g. Unit testing - covering all major & important functionality * e.g. Customer engagement at the end of every sprint (Demo + UAT). * e.g. Scoping of each functionality has been thoroughly reviewed with clients and UX team. |
| Performance | * System is able to perform its work within the expected timing requirements. |
| Usability | * User is easily able to accomplish a desired task with the system * Responsive web application across supported web browsers   + Microsft Edge   + Google Chrome * Designed by standard Deloitte UI practices. * Built with focused UX expert’s reviews & suggestions. |
| Modifiability | * Aligned to application standards. * Components will be well-defined and extensible. * Designed with future releases in mind * Changes or enhancements can be made to the system at reasonable risk and cost |
| Portability | * Browser based solution to have targeted browser support. * Standard HTML 5 and other web standards for cross browser and cross device compatibility. |
| Reusability | * Makes use of high reusable components such as Deloitte Core, header, navigation, SSO etc. |

## Architectural Drivers

NFRs which significantly influenced your design decisions

|  |  |
| --- | --- |
| e.g. User Interface | Provide effective responsive user interface design to help users have seamless experience across Desktop, Laptop & Tablet. |
| e.g. Framework Upgrade | Upgrade to .Net 4.6.2 to take advantage of Debugging APIs (CRL) |
| e.g. SQL Server upgrade | Upgrade to SQL2016 to take advantage of Always-Encrypted |

## Technology and Infrastructure

This describes the hardware and software technologies that will be used to design, develop, deploy and operate the system under consideration, and will have an impact on the architectural decisions. Example is provided as follows.

### Design and Development Technologies

|  |  |
| --- | --- |
| Hardware | Description |
| e.g. 2 VMs with 2 CPU, 8GB of RAM and ~150GB of hard drive space each. Running Windows Server 2012 R2 | Dual Core Processor |
| e.g. F5 | The Load balancer |

|  |  |
| --- | --- |
| Software | Description |
| Windows Server 2012+ |  |
| SQL Server 2014+ |  |
| IIS Web Server/ Kestrel |  |
| NODE 12 |  |
| .Net Core 3.1 |  |
| ASP.NET Core Web API |  |
| Angular 9 |  |



### Production Environment Technologies

|  |  |
| --- | --- |
| Hardware | Description |
| e.g. ~4 VMs with 4CPU, 8GB of RAM and ~150GB of hard drive space each. Running Windows Server 2012 R2 | Quad Core Processor |
| e.g. F5 | The Load balancer |

|  |  |
| --- | --- |
| Software | Description |
| e.g. Windows Server 2012 |  |
| e.g. SQL Server 2014 |  |

# Secure Software Development Lifecycle (SSDLC) requriements

This section describes the Secure Software Development Lifecycle (SSDLC) that must be completed throughout SDLC processes to integrate security into development

The SSDLC approach consists of seven phases such that all control activities referred to a phase must be completed before moving to the next phase of the lifecycle. Strengthened security is achieved through consistent use of this approach and appropriate reviews at defined security gates (checkpoints for validation of security controls). Secure SDLC (SSDLC) mandates security checkpoints that must be completed throughout SDLC processes to integrate security into development and minimize risks related to solution implementation.

More information on the SSDLC can be obtained from <https://deloittenet.deloitte.com/sites/MyTechnology/Pages/Security/SSDLC.aspx> which also contains the “Cyber Assessment” questionnaire referred to elsewhere in this document.

## Security Requirement at Application Level

This subsection includes:

* Submit Cyber Assessment questionnaire
* Review and understand security requirements including confidentiality and privacy

Please refer to below link: <https://deloittenet.deloitte.com/sites/MyTechnology/Pages/Security/SSDLC.aspx>

## Security/Controls Design Guideines

This subsection includes:

* Must Pass Static Code Analysis

Example tools for Static code analysis - Veracode, HP Fortify, SonarQube, Code analysis ruleset

<https://itsteams.deloittenet.com/sites/ISRC_Services/AppSec/Veracode.aspx>

* Must Pass Dynamic Code Analysis and Penetration test

<https://itsteams.deloittenet.com/sites/ISRC_Services/AppSec/Pentest.aspx>

* TLS 1.2 and higher – Yes/No
* .NET Config Protection with Rivest, Shamir, and Adelman (RSA) key – Yes/No

*Engage DevOps to implement (RSA) encryption/decryption for your application.*

* Total solution meets Segregation of duty control requirements: Yes/No

[Example – app teams do not have access to Prod environment and data]

Please refer to below link: <https://deloittenet.deloitte.com/sites/MyTechnology/Pages/Security/SSDLC.aspx>

* Appropriate at rest encryption has being implemented: Yes/No

Encryption at rest is required if you Data classification has CC, PHI, Sensitive PII then encryption should be required. For guideline please refer URL:<https://deloittenet.deloitte.com/sites/MyTechnology/Pages/Security/SSDLC.aspx>

* Application meets Privileged Access Management(PAM) guidelines: Yes/No

It is required to access privileged and elevated accounts through the Privileged Access Management (PAM) solution as standard.PAM Email: [uspcs@deloitte.com](mailto:uspcs@deloitte.com). PAM link:

<https://wt.deloitteresources.com/solutions/IAM/PAM%20Pages/PAM_Overview.aspx>

* Hardware Controls, OS and Software meet Cyber guidelines: Yes/No

Please refer to below link: <https://deloittenet.deloitte.com/sites/MyTechnology/Pages/Security/SSDLC.aspx>

* Security Architecture and Design/Role Matrix

This subsection includes security architecture and role matrix

* Authentication with MFA- e.g. Azure AD with MFA
* Authorization
* Role Matrix

## Security Testing Results

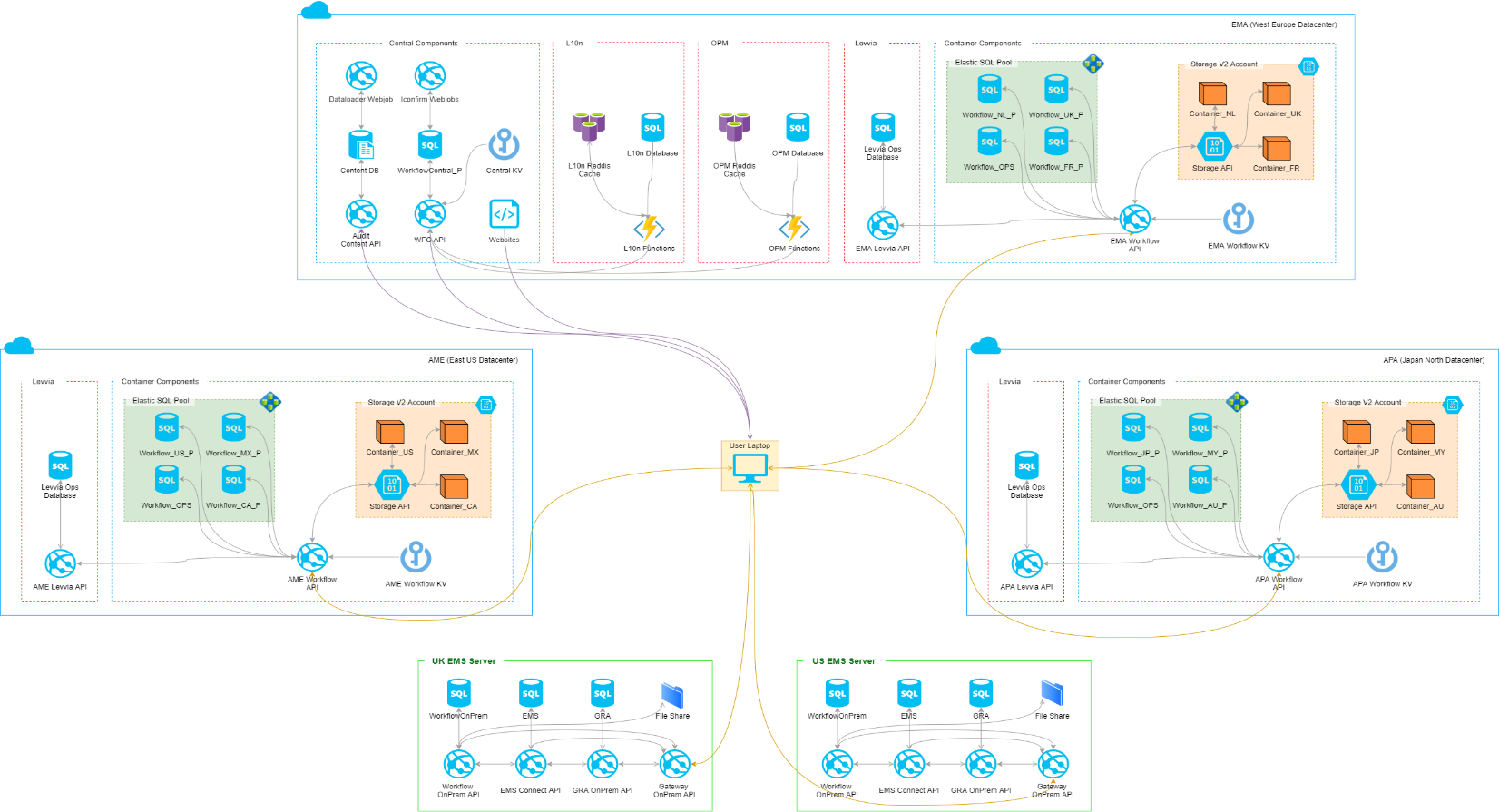
This subsection includes security testing results

* Static Code Analysis (Veracode)
* Dynamic Code Analysis
* Penetration testing

# Conceptual View

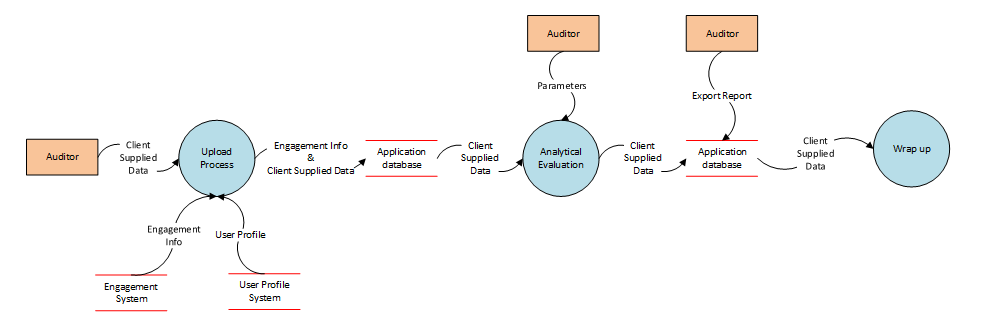
## Conceptual System Diagram

This subsection includes a conceptual system diagram:



## Data Flow Diagrams

This subsection includes a data flow diagram. An example is provided below:



# DevOps

## CI/CD Detail

This sub section describes CI/CD process and tool used. Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early. With Continuous Delivery (CD), any commit that passes the automated tests and security scan can be considered a valid candidate for release

* + VSTS (Azure Pipelines)
  + Source Control – Git Repositories
  + Automation Server – TOSCA
  + Unit Testing Framework – MSTest, XUnit, MOQ

## Web Analytics

This sub section describes how Web analytics is leveraged toward measurement, analysis and reporting of web data for purposes of understanding and optimizing web usage.

* + Capture Web and Mobile analytics (e.g.Omniture)
  + Capture Mobile Crash logs (e.g.Hockey App)

## Instrumentation and Logging

**Controllability and Observability**

In context of computer programming, instrumentation refers to an ability to monitor or measure the level of a product's performance, to diagnose errors and to write trace information. Application teams implement instrumentation in the form of code instructions that monitor specific components in a system (for example, instructions may output logging information to appear on screen).

This sub section describes the areas that needs to be incorporated into development in order to control the issues and observing them through effective measures. These are usually termed as Instrumentation in Software Application Development and Maintenance Life Cycle and typically works around logging, diagnosis, tracing and application infrastructure performance.

**Application Instrumentation Approaches**

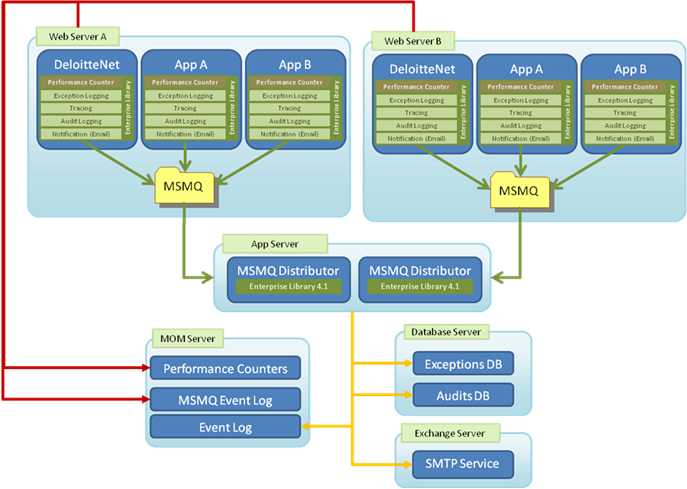
| Activity Item | Type | Category | Level | Retention | Suggested Approaches |
| --- | --- | --- | --- | --- | --- |
| Error Logging | Diagnostic | Logging | Medium | Per Quarter | Logging of detailed errors to any persistent store is highly essential. This could be a custom logging or proven logging mechanisms like Microsoft Enterprise Library for logging, Log4J, etc |
| Information | Diagnostic | Logging | Low | Per Quarter | It is not recommended to utilize logging for information logging |
| Business Transactions | Audit | Logging | Medium | Per Quarter | This is driven from business needs and records important events in the system that is necessary for auditing purposes  For example, authentication events, entity management events, etc.  No logging of critical information |
| Visualization | Diagnostic and Audit | Logging | High | Per Quarter | While logging of events to a persistent store with established archival mechanisms is one part of the need, an effective visualization of these logs for the key stakeholders is equally important. Information on the logs with its key attributes, trends is highly important. A scalable system that is dedicated for these purposes will be of high level in effective trouble shooting |
| Access Control | Diagnostic and Audit | Logging | High | Continuous | Access to logs and the log visualization tools should be retricted by using mechanisms like AD groups etc; |
| Connectivity Checks | Application Monitoring | Monitoring | Medium | Continuous | Monitoring and problem remediation can be automated by creating effective health checks that pre-define problems  Resolution Actions can be automatically triggered once the health checks notify  Heath checks can be in the form of system connectivity dashboards etc.  This is typically done by the application development team |
| Performance Alerts | Application Monitoring | Monitoring | High | Continuous | The ability to observe performance metrics captured in real time while the application is running in production  The captured metrics are compared to baselines based on past performance  A proven monitoring tool like SCOM is recommended for monitoring and sending alerts to the application operations teams. |
| Service Alerts | Application Monitoring | Monitoring | High | Continuous | A proven monitoring tool like SCOM is recommended for monitoring and sending alerts to the application operations teams. |
| Remediation | Troubleshooting | Troubleshooting | High | Continuous | This activity is about the actual troubleshooting exercise that is carried out after any critical issue is identified.  Reports from Logging and Monitoring mechanisms are received from appropriate parties like health check reports, application logs, infrastructure performance reports, request and response snapshots  An established stakeholder connectivity and agreement standards should be in place to ensure and authorize the normal stage. |

**The level above indicates the importance of that activity item. Retention plans are driven from business needs.**

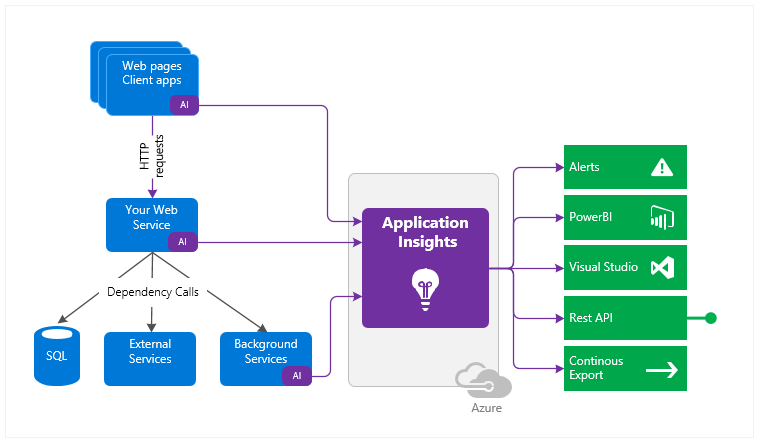
## Instrumentation Design

This sub section describes the overview all design on how instrumentation is implemented in the application.

*e.g.*



e.g.



# Use Case View (Conceptual View)

*This section lists use cases or scenarios from the use-case model if they represent some significant, central functionality of the final system, or if they have a large architectural coverage—they exercise many architectural elements or if they stress or illustrate a specific, delicate point of the architecture.*

## Personas

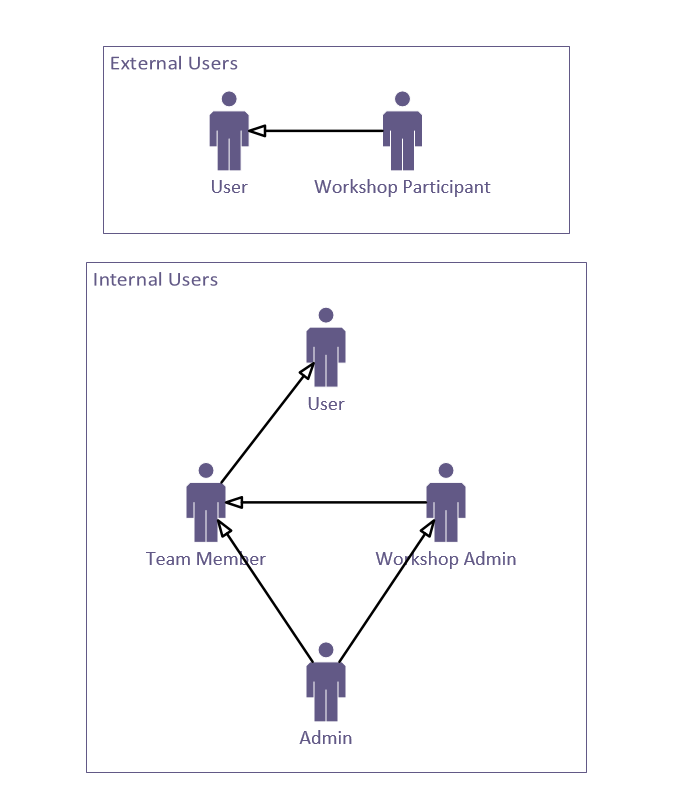
A persona is a role that a user plays in the system. All the architecturally significant personas in the system are identified in this section. Example is as shown below.

### Persona Definition

|  |  |
| --- | --- |
| Persona Type 1 |  |
| Persona Name 1.1 |  |
| Persona Name 1.2 |  |
| Persona Name 1.3 |  |
| Persona Type 2 |  |
| Persona Name 2.1 |  |
| Persona Name 2.2 |  |

### Persona Hierarchy

*e.g.*



## Use Case

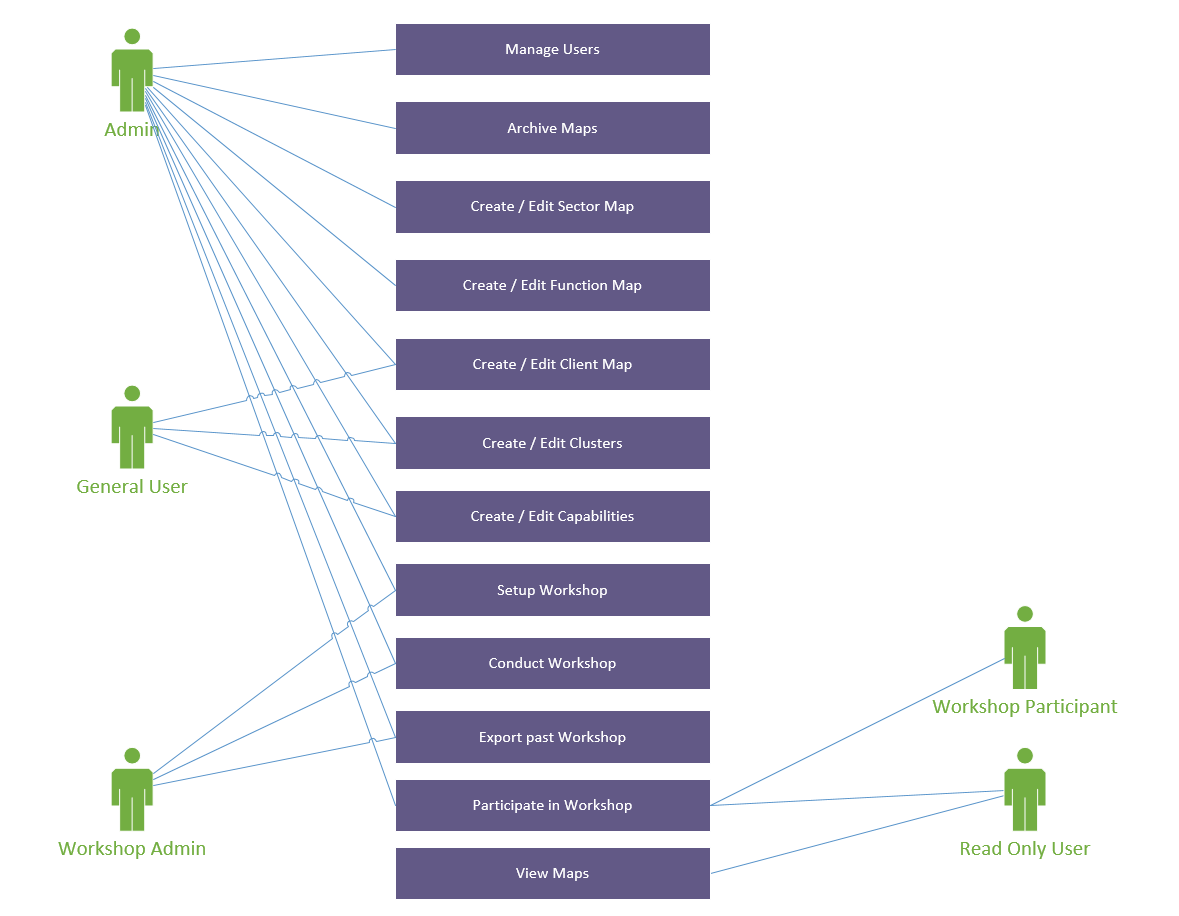
This section outlines the system behavior organized by various subsystems in the application and significant central functionalities.

### Use Case Definition

|  |  |
| --- | --- |
| Use Case Id | Use Case Description |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### Use Case Diagram

*e.g.*



# Logical View

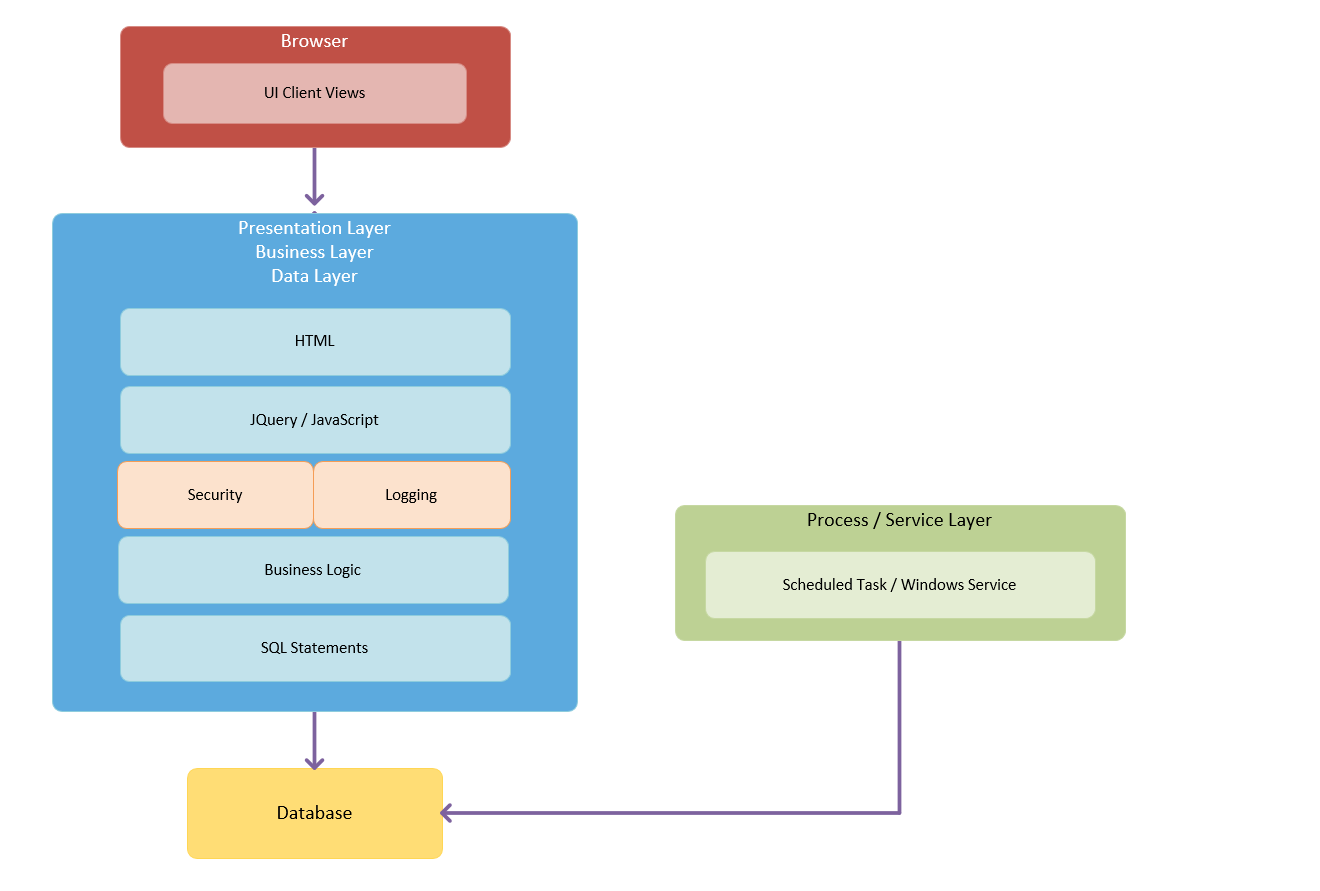
*The logical view primarily supports the functional requirements and the functionality the system should provide to its end users. Designers decompose the system into a set of key abstractions, taken mainly from the problem domain. These abstractions are objects or object classes that exploit the principles of abstraction, encapsulation, and inheritance. In addition to aiding functional analysis, decomposition identifies mechanisms and design elements that are common across the system.*

*Audience: End Users, Business Analysts*

## Subsystem Decomposition

This subsection describes the subsystem decomposition of the design model in terms of its design element hierarchy and layers.

e.g.



# Process View

*The process view mainly focuses on the business processes that the target application is to support. Interestingly enough, it clearly serves as the “service bridge” between the Logical View (Functionality) and the Physical View (Implementation Technology). At its core, the basic principle of this architecture phase is the technological agnosticism. In other words, business processes are clearly identified without much deliberation of what technological solutions may be implemented. Then, business process analyses are used to ascertain business contexts and patterns, which serve as the roadmap for abstraction and decomposition thereof into business services.*

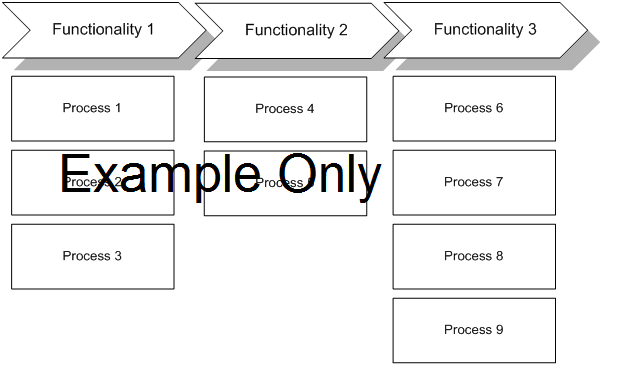
*In addition, the process view takes into account some nonfunctional requirements, such as performance and system availability. It addresses concurrency and distribution, system integrity, and fault-tolerance. The process view may also specify which thread of control executes each operation of each class identified in the logical view.*

Organize the section by groups of processes that communicate or interact. Describe the main modes of communication between processes, such as message passing, interrupts, and rendezvous.

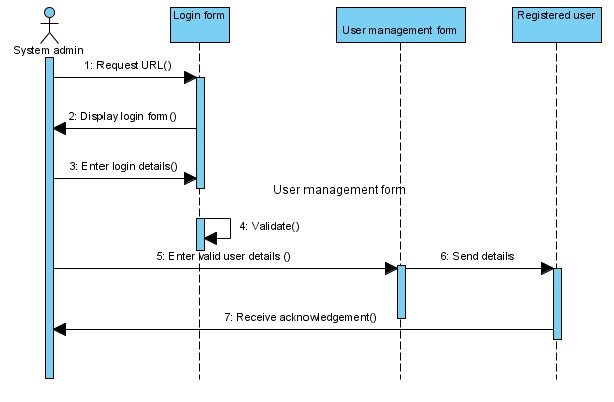
*Audience: Business Analysts, Technical Leads*

Following diagram depicts an example of the representation of the process view.

e.g.

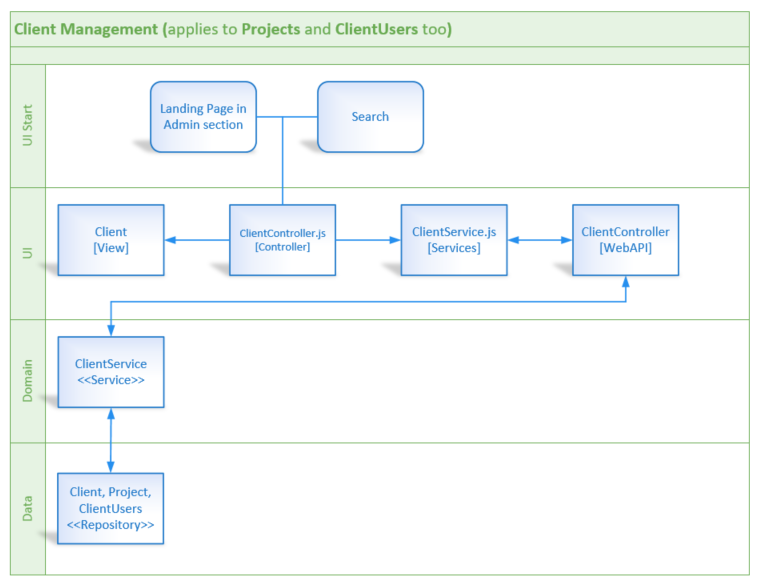


e.g.



## Services Decomposition Model

This section describes the major service areas within the system.

e.g. 

# Infrastructure View

Design (Deployment View)

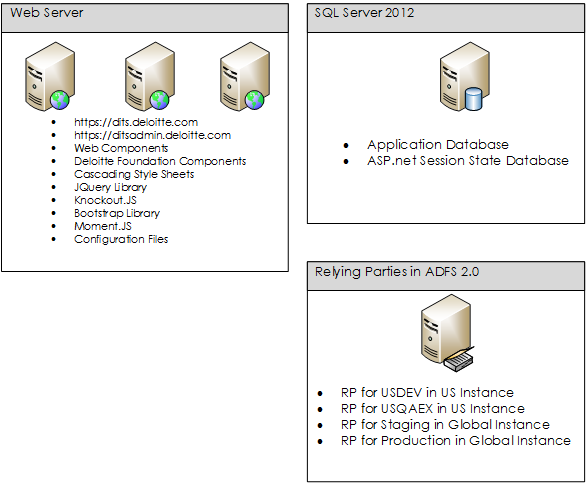
The Infrastructure view takes into account the system's nonfunctional requirements such as system availability, reliability (fault-tolerance), performance (throughput), and scalability. The software executes on a network of servers (the processing nodes). The various elements identified in the logical, process, and physical views—networks, processes, activities, and objects—must be mapped onto the various nodes. Several different physical configurations can be used—some for development and testing, some for quality assurance, and others for system deployment at various sites or for different customers. The mapping of the software to the nodes must therefore be highly flexible and have a minimal impact on the source code itself. Location transparency is often a major driver in distributed systems that span multiple network nodes.

Audience: System Engineers and Infrastructure Personnel

This section includes deployment and infrastructure diagrams. Examples are provided below:

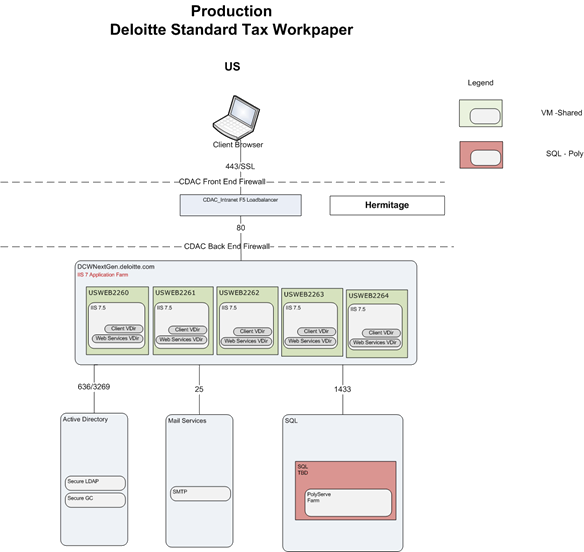
## Deployment View

Include server details for all environments.



## Infrastructure Diagram (Physical View)

This diagram is produced by the infrastructure team.



## Business Continuity Planning (BCP)

IT Service Continuity Management is an integral part of Deloitte's approach to effectively managing risk. It ensures the required IT technical and services facilities including computer systems, networks, applications, telecommunications and technical support services can be recovered within required, and agreed upon business timescales.

### BCP Model

|  |  |
| --- | --- |
| Tier | e.g. {Tier 2} |
| Recovery Time Objective(RTO) | e.g. {24 hours} |
| Recovery Point Objective (RPO) | e.g. {24 hours} |
| Recovery Method | e.g. {SQL AlwaysOn} |

### BCP Standards

| Tier | **Tier 0\*** | **Tier 1\*** | **Tier 2** | **Tier 3** | **Tier 4** |
| --- | --- | --- | --- | --- | --- |
|  | Base service required for the support of other applications, communications, restoration of files, et. (WAN, LAN, TSM, Exchange, etc.) | Extremely high levels of service availability required to support a critical business process. The operational, financial or reputational risk to the business resulting from a service delivery failure would be **high** | High levels of service availability required to support an important business process. The operational, financial or reputational risk to the business resulting from a service delivery failure would be **high to medium.** | The service is important to the business but the operational, financial or reputational risk to the business resulting from a service delivery failure would be **medium** to **low** | The service is not mission critical and the operational, financial or reputational risk to the business resulting from a service delivery failure would be **low** to **none** |
| Recovery Time Objective(RTO) | 6 hours | 12 hours | 24 hours | 72 hours | 20 days |
| Recovery Point Objective (RPO) | 6 hours | 6 hours | 24 hours | 72 hours | 72 hours |
| Recovery Method | Redundant  High-Availability | Redundant  High-Availability | Redundant | Restored\*\* | Restored\*\* |
| Plan Validation Frequency | Every 12 Months | Every 12 Months | Every 12 Months | Every 12 Months | Every 12 Months |
| Plan Testing Frequency | Every 12 Months | Every 12 Months | Every 24 Months | Every 24 Months | Every 24 Months |

For BCP guideline Link: <https://itsm.deloitte.com/SD/SitePages/ITServiceContinuity.aspx>

BCP Email: ITS BCP Manager

\* Tier 0 and Tier 1 are reserved for infrastructure / platforms that require extremely high levels of service availability

\*\* Typical Backup Model  
 Mirroring to offsite location  
 Online | Transaction log backup to Virtual tape library (VTL) every 15 minutes  
 Offline | Alternate between Full and Delta backup to Virtual tape library (VTL) every 24 hours

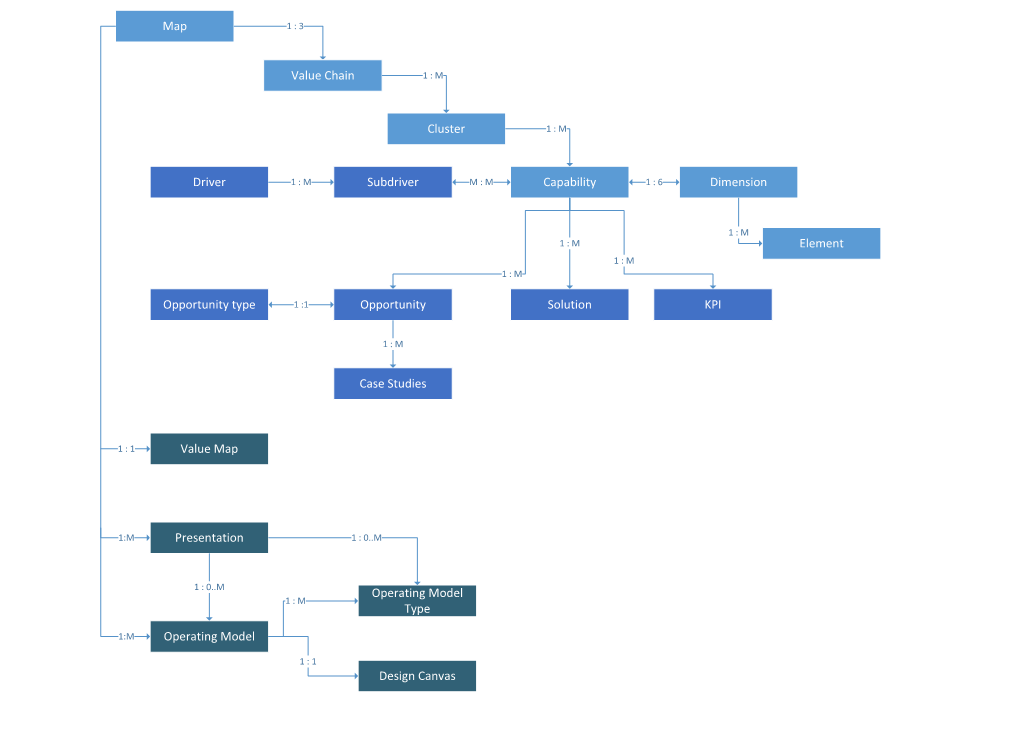
# Data View

A description of the persistent data storage perspective of the system. This section is optional if there is little or no persistent data or the translation between the Design Model and the Data Model is trivial.

## Entity/Logical Data Model

This subsection contains the logical data model.

e.g.



## Data Integration and Classification

This subsection contains the documentation about data that will be acquired from other sources in addition to the data collected from the user in the application. The line for Application data is required. Refer to the following example to complete the Data Integration and Classification with various Firm Systems.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Domain | Data Source / Firm System | Destination | Transfer Method | Retention Plan | Data Classification |
| *e.g. Finance* | *HANA Analytics* | *Application* | *API - oData* | *Not Retained* | *Confidential – Deloitte - Financial Information* |
| *e.g. Employee Data* | *MDR* | *Application Database* | *SSIS* | *Refreshed Daily* | *Public - Deloitte*  *Confidential - Deloitte Data - Personal information (PI)*  *Confidential - Client Data - Competitive intelligence*  *High-Risk Confidential - Deloitte - Sensitive PI*  *High-Risk Confidential - Deloitte - Sensitive Information* |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Data Classification

* Public - Deloitte
* Confidential - Deloitte/Client - Intellectual property
* Confidential - Deloitte/Client - Personal information (PI)
* Confidential - Deloitte/Client - Financial information
* Confidential - Deloitte/Client - Competitive intelligence
* Confidential - Deloitte/Client - Other (based on contractual terms, regulations, CIMP)
* High-Risk Confidential - Deloitte/Client - Sensitive PI
* High-Risk Confidential - Deloitte/Client - Sensitive Information
* High-Risk Confidential - Deloitte/Client - Government Information
* High-Risk Confidential - Deloitte/Client - Other (based on contractual terms, regulations, CIMP)

### Data Domain

Summarize the data domain for the application required for data consumption.

Examples:

Talent, Finance, WBS, Cost Center, Profit Center, Talent, Payroll, Clients, etc.

### Data Source – Core System

This subsection contains the documentation about the core systems to reference in the **data source** section above**.**

Firm Systems and reference architecture can be found here:

<https://atlas.deloitte.com/Pages/ApprovedTechnologies.aspx>

For all SAP Systems, developer access and application accounts can be requested through SAP Governance, Risk and Compliance (GRC).

<https://uswasgdq.deloitte.com/sap/bc/ui5_ui5/ui2/ushell/shells/abap/FioriLaunchpad.html>

US ITS SAP Security [USITSSAPSecurity@DELOITTE.com](mailto:USITSSAPSecurity@DELOITTE.com)

|  |  |
| --- | --- |
| Core Data Sources |  |
| **MDR – Master Data Repository** | Current snapshot of U.S. master data in one SQL Server for individual and application use. E.g. Client, Employee, WBS, Profit Center, Cost Center  Refer to the MDR Team site for the process of submitting a data consumption request/team.  <https://deloitteteams.deloittenet.com/sites/MDRSP2007/default.aspx> |
| **US HANA Analytics** | In-Memory Enterprise Data Warehouse for U.S. Master Data and Transactional Reporting and Analytics.  e.g. Finance, Talent, Client, Employee, WBS, etc.  Refer to the HANA Integration standards for information on how to connect to HANA.  <https://applicationstandards.deloitte.com/app/core/data/databases>  Standards\_Data\_HANA\_Integration.docx  US DAS HANA Data Management [hdm@deloitte.com](mailto:hdm@deloitte.com) |
| **SAP S/4** | Core Global Financial System (General Ledger, Time, Expense, Projects, Clients, Vendors, Profit Centers, etc.)  SAP ERP system contact information:  Robbrecht, Ivo A. (US - Hermitage) [irobbrecht@DELOITTE.com](mailto:irobbrecht@DELOITTE.com)  Horsman, Erik (US - Hermitage) [ehorsman@deloitte.com](mailto:ehorsman@deloitte.com) |
| **SAP ECC** | Core U.S. Human Capital Management system (People/Payroll) + Legacy Financials  SAP ERP system contact information:  Robbrecht, Ivo A. (US - Hermitage) [irobbrecht@DELOITTE.com](mailto:irobbrecht@DELOITTE.com)  Horsman, Erik (US - Hermitage) [ehorsman@deloitte.com](mailto:ehorsman@deloitte.com) |
| **MDR – Master Data Repository** | Current snapshot of U.S. master data in one SQL Server for individual and application use. E.g. Client, Employee, WBS, Profit Center, Cost Center  Refer to the MDR Team site for the process of submitting a data consumption request/team.  <https://deloitteteams.deloittenet.com/sites/MDRSP2007/default.aspx> |

### Data Destination

Describe the target destination for data integration.

Examples: Application UI, Application Database, External Database, Other Core System, etc.

### Data Transfer Method

This subsection contains the documentation about the core data transfer technologies to reference in the transfer method section above.

|  |  |
| --- | --- |
| Data Transfer |  |
| **API Management** | Provides a middle layer between Deloitte applications and Backend Resource in order to create APIs to retrieve the data. It is an on premise version of Apigee Edge rebranded by SAP as SAP API Management. The primary usage is to create restful based services against SAP HANA (oData, xsjs).  <https://apidev.deloitte.com>  SAP API Management Support (US) [sapapimanagementsupport@deloitte.com](mailto:sapapimanagementsupport@deloitte.com) |
| **SAP Data Services (BODS)** | Data management software for data integration, quality, cleansing, and more. Data Services is used primarily for batch scheduling of ETL jobs between SAP and non-SAP systems. <https://applicationstandards.deloitte.com/app/core/integration/etl>  Standards\_Data\_BODS\_OnBoarding\_Guide  US ITS Data Services Admin [USITSDataServicesAdmin@DELOITTE.com](mailto:USITSDataServicesAdmin@DELOITTE.com) |
| **SAP SLT** | SLT is the standard real-time replication tool for big data for SAP ABAP-based source systems to SAP HANA, to all DBs supported by SAP, to the SAP Business Suite and to SAP applications, but also supporting non-SAP sources as covered by the SAP Product Availability Matrix.  US ITS SLT Admin [USITSSLTAdmin@deloitte.com](mailto:USITSSLTAdmin@deloitte.com) |
| **Other** | Other methods of data transfer include:  SSIS, Globalscape (sFTP), Manual Upload, Manual Entry, etc. |

### Data Retention

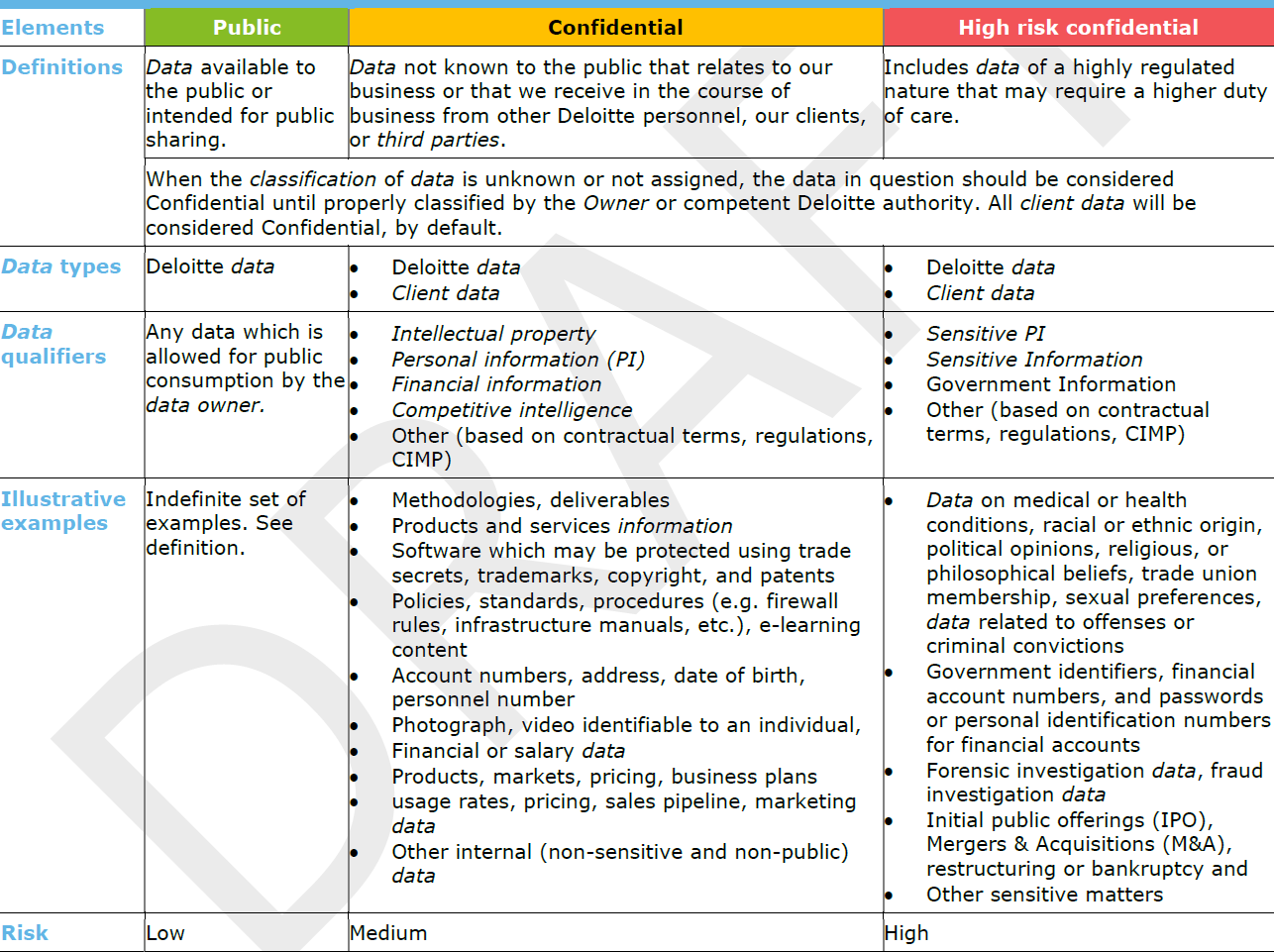
Describe the data retention plan for the data.

Examples: Not Retained, Cached, Rolling 2 years, etc.

### Data Classification

Please follow the data classification guidelines.

Minimum documentation should be the data element (Public / Confidential / High Risk Confidential), the Data Type (Deloitte data vs. Client data), and the Data Qualifier for each data domain.



# Non-Functional Attributes

A description of how does the system address various architectural attributes (non-functional requirements): extensibility, reliability, portability, and so on. If these characteristics have special significance, such as safety, security or privacy implications, they must be clearly delineated.

|  |  |  |
| --- | --- | --- |
| Availability |  | This subsection lists the architectural approaches that will be used to meet the business availability needs. |
| Scalability |  | This subsection lists the architectural approaches that will be used to meet the business scalability needs. |
| Performance |  | This subsection lists the responsiveness of a system to execute any action within a given time interval - response time, utilization, and throughput behavior of the system |
| Security |  | This subsection lists the architectural approaches that will be used to measure system’s ability to resist unauthorized attempts at usage or behavior modification, malicious actions outside of the designed usage, prevent loss of data or information while still providing service to legitimate users. |
| Usability |  | This subsection lists the architectural approaches that will be used to meet business requirements by being intuitive, effiecient, ease of use and of training the end users of the system and resulting in a good overall user experience. |
| Interoperability |  | This subsection lists the architectural approaches that will be used for an interoperable system makes it easier to exchange and reuse data internally as well as externally |
| Modifiability |  | This subsection lists the architectural approaches describing the ability of a system to undergo changes with a degree of ease |
| Portability |  | This subsection lists the ability of a system to run under different computing environments. The environment types can be either hardware or software, but is usually a combination of the two |
| Reusability |  | This subsection defines the capability for components and subsystems to be suitable for use in other applications and in other scenarios. Reusability minimizes the duplication of components and the implementation time |
| Integrability |  | This subsection lists the architectural approaches which defines the consistency and coherence of the overall design. This includes the way that components or modules are designed, as well as factors such as coding style and variable naming. |
| Testability |  | This subsection lists the architectural approaches to ease with which the software can be made to demonstrate its faults |
| Supportability |  | This subsection lists the architectural approaches used to provide information helpful for identifying and resolving issues when the system fails to work correctly |

# Glossary

|  |  |
| --- | --- |
| CC | Client Confidential data |
| CIMP | Confidential Information Management Plan. |
| PII | Personally Identifiable Information |
| PHI | Protected Health Information |



**About Deloitte**  
Deloitte refers to one or more of Deloitte Touche Tohmatsu Limited, a UK private company limited by guarantee (“DTTL”), its network of member firms, and their related entities. DTTL and each of its member firms are legally separate and independent entities. DTTL (also referred to as “Deloitte Global”) does not provide services to clients. Please see [www.deloitte.com/about](http://www.deloitte.com/about) for a detailed description of DTTL and its member firms. Please see [www.deloitte.com/us/about](http://www.deloitte.com/us/about) for a detailed description of the legal structure of Deloitte LLP and its subsidiaries. Certain services may not be available to attest clients under the rules and regulations of public accounting.  
  
Copyright © 2018 Deloitte Development LLC. All rights reserved.  
36 USC 220506  
Member of Deloitte Touche Tohmatsu Limited