Medicaid Management Information System Replacement (MMISR) Project

SIPLT1: System Design Document

HSD Deliverable Owner: Vladislav Vilensky

System Integrator (SI) Deliverable Owner: Satya Govindu

Configuration Number v1.1

June 14, 2019

Section 508 Compliant

Confidentiality Notice: This document, including any attachments, is for the sole use of the intended recipients and may contain confidential and privileged information, which must be protected from public disclosure. Any unauthorized review, use, disclosure, or distribution is prohibited. If you are not the intended recipient, please contact the sender and destroy all copies of the original document

Table of Contents

List of Figures

List of Tables

1 Introduction

The System Design Document (SDD) provides the design and architectural choices of the System Integration (SI) Platform of the New Mexico (NM) Human Services Department (HSD) Medicaid Management Information System Replacement (MMISR) project. This platform facilitates the overall State initiative of Health and Human Services (HHS) 2020 vision of creating a modular and extensible integration platform.

1.1 Overview

The NM HSD has adopted the HHS 2020 vision, a transformational, enterprise approach to the health and human services business. HHS 2020 will move service delivery from a program-centric approach to a stakeholder-centric approach. NM HSD will migrate away from program and technology silos into an integrated, flexible framework that supports service delivery and stakeholder interaction across HHS programs and organizations. The HHS 2020 is technology-enabled but includes rethinking the organizational design, redesigning and streamlining business processes, and reducing barriers between organizations within the HHS enterprise.

The HHS 2020 framework emphasizes modularity coupled with interoperability, achieved through a standards-based approach to implementation, widespread adoption of a Service-Oriented Architecture (SOA), and the delivery of micro-services. Please see “Section 1: Introduction” in the Project Management Plan for a detailed MMISR Project Overview.

The SDD follows the holistic approach to solution development laid out in the MITA Technical Management Strategy. This SDD adheres to the Preferred-Acceptable-Discouraged-Unacceptable (PADU) framework-based approaches to the technical problems, as listed in the Reference Architecture (RA).

Based on the Reference Architecture guideline that every architectural and design decision must be traceable to at least one corresponding requirement, this SDD has elicited Architecturally Significant Requirements (ASRs). The ASR is defined as the core architectural building block requirement for every design element. These ASRs drive the high level integration patterns and design elements for every subcomponent of the platform. These ASRs are sourced from the reference architecture library, including, RA, Concept of Operations (ConOps) and listed in a tabular format in Appendix I.

This document follows the guidelines provided in the Centers for Medicare and Medicaid Services (CMS) for system design: CMS System Design Document Template. References used to create this document are provided in Appendix D.

1.2 Intended Audience

The intended audience for this document includes:

- Application developers.
- Quality Assurance (QA) testers
- Project managers
- Infrastructure and operations support
- System security managers
- Architecture Review Board (ARB)
- Enterprise and Data Architects
- MMISR participant vendors

- NM HSD Information Technology Division (ITD)
- NM HSD Medical Assistance Division (MAD)
- Subject Matter Experts (SMEs) and data stewards of legacy data sources.
- Data stewards and technical stakeholders of MMISR modules.
- Business, technical, governance, and MMISR project management stakeholders, especially those working with the SI Platform.

1.3 Scope

The purpose of the SDD is to define the architecture and system design of the SI Platform's components and sub-components. The development teams can effectively use this design as an interactive blueprint. The SDD ties architectural and design decisions to ASRs to form a cohesive and consistent design evolution from the Medicaid Information Technology Architecture Technical Management Strategy (MITA TMS) and Reference Architecture (RA) documents.

The scope of this document includes architecture and system design of the following components:

- Hardware design including system, server, and network configurations.
- SOA and Enterprise Service Bus (ESB) platforms, which modules and external systems, other than SI Platform, use to interact and integrate.
- Information architecture and data strategies for migration, integration, and Master Data Management (MDM).
- Service orchestration interfaces, and partner integration services, which the SI Platform will provide to enterprise modules that consume and transact with external systems.
- Enterprise shared services, which the SI Platform provide to other modules of the enterprise to centralize resources and promote reuse.

This document also includes the following architecture and design concerns, which cut across all components and sub-components of the SI Platform:

- The security architecture of the overall platform as it conforms to CMS mandates and at each component level.
- Performance architecture and design, which allows the SI Platform to scale and meet the stipulated performance requirements.

1.4 Roles and Responsibilities

The SI Contractor development team will work closely with the NM HSD ITD team and other HHS 2020 enterprise solution stakeholders, to establish the roles and responsibilities associated with the design of SI Platform.

The following table outlines the key personnel and other individuals who are responsible for the system design of SI Platform.

Table 1-1: Roles and Responsibilities

Role	Responsibility
Business Analyst (BA)	Develops the business and architecturally significant requirements, system attributes, and other supporting requirements.
Enterprise Architect	Responsible for the definition and execution of cross-program, cross-module architectural strategies for the HHS 2020 vision.
Data Architect	Responsible for the design and use of data in a structured format like databases and unstructured format files, across HHS 2020 modules.
SI Solution Architect	Responsible for devising technical solutions to address business requirements concerning the Integration Platform. Defines the structure, characteristics, behavior, and other aspects of software, and presents them to project stakeholders. Provides specifications by which the solution is defined, developed, managed, and delivered.
Infrastructure Engineer	Responsible for infrastructure capacity planning, design, and configuration of hardware and network infrastructure.
System Security Engineer	Responsible for the security architecture of the SI Platform as per the System Design Plan (SDP) and System Security Plan (SSP), according to Centers for Medicare and Medicaid Services (CMS) requirements.
Software Developer	Participates in the system design of the SI platform in accordance with the ASRs, system attributes like performance, security. The system design will be the baseline for the detail design and development of each work stream of the SI platform.
Database Administrator	Responsible for database setup and design administrative tasks such as troubleshooting, performance tuning, security audit, backup, and data recovery.
Build and Package Engineer	Responsible for building and packaging the source code as executables, binaries, and scripts for the target environments based on the system design defined in SDD. Also responsible for build automation CI/CD setup and management, build and deployment (manual), build artifacts, and version control.

Role	Responsibility
Configuration Manager	Manages the configuration management database. Performs overall management of all configuration items and their relationships. Directs other teams in configuration management practices.
Application Support Engineer	Responsible for operational activities in production and pre-production environments based on the design articulated in the SDD. Monitors the application, network, infrastructure, managing the applications logs, etc. The application support engineer will perform testing and maintenance of infrastructure across environments.
Project Manager	Plans and coordinates the testing activities within the overall project schedule, tracks testing activities, mitigates risks and participates in testing readiness reviews (review gates) to ensure that the appropriate artifacts are generated for certification purposes.
QA Testers	Performs test based on the system designs defined in SDD. QA testers take system designs defined in this document to articulate the system test plan and validates the system in accordance with requirements, system test plan, and system design goals like performance, scalability, and security.
Certification Manager	Leads the Contractor's Certification Team. Provides guidance and direction on meeting CMS Certification requirements.
Independent Verification and Validation (IV&V)	Conducts IV&V assessments. Identifies potential improvements or identifies problems before they occur. Reviews the overall process design, development, and validation processes.
Architecture Review Board (ARB)	The ARB is a governance body that reviews and approved the designs to ensure its alignment with enterprise architecture. Technical changes to the approved designs that affect the project will also need approval with the ARB.
Change Control Board (CCB)	Provides the change management forum for establishing baselines and approving, or not approving, subsequent changes to those baselines in accordance with CMS policies, plans, guidance, processes, and procedures.
SMEs from NM HSD	Provides business knowledge while designing the SI Platform.

Role	Responsibility
Other Stakeholders	The Department of IT (DoIT) has oversight authority on the overall program initiatives and through the Technical Architecture Review Committee (TARC) acts as an approving authority to the system design.

1.5 Relationship to Other Plans

The following table lists the plans associated with the SDD of the SI platform. The table provides a brief description of each plan and its relationship with the SDD. This plan references other deliverables to provide additional details for certain sections. It also contains lists other deliverables and their relevance to the plan.

Table 1-2: SDD Relationship to Other Plans

Deliverable Name	Relationship with SDD
Project Management Plan	• The Project Management Plan describes the approach to how the overall HHS 2020 enterprise solution is executed, monitored, and controlled. • The SDD supplements the Project Management Plan with details about high-level hardware and software architecture strategies.
Configuration and Continuous Integration Service Management Plan	• The Configuration and Continuous Integration Service Management Plan (CCIS) Plan is the over-arching plan that describes how the MMISR project team will integrate modules and services into the Integration Platform (IP) through a repeatable, iterative approach. • The architecturally significant requirements of the IP are driven by the needs of CCIS to design a modular, continuously evolving system.
Requirements Management Plan	• The Requirements Management Plan (RMP) describes the approach to managing and maintaining the HHS 2020 project requirements life cycle. • The SDD describes the high-level architecturally significant requirements.
Requirements Traceability Matrix	• The Requirements Traceability Matrix (RTM) will define the process of tracking requirements throughout the MMISR project life cycle. • The SDD will provide the bi-directional traceability between architecturally significant requirements and the high-level design.

Deliverable Name	Relationship with SDD
Development Plan	- The Development Plan includes the approach for design, development, configuration, and unit testing of the systems involved in the overall HHS 2020 enterprise solution. - The SDD adheres to the design principles, design patterns and the design approach defined in Development Plan.
Test Management Plan	- The Test Management Plan (TMP) describes the overall testing approach, numerous testing functions that will be conducted during the life cycle of systems, and components in the HHS 2020 enterprise solution. - The designs proposed in the SDD will factor in the testing strategies explained in the TMP.
Quality Management Plan	- The Quality Management Plan (QMP) describes the approach to monitor, evaluate, and provide oversight of quality in all the work and deliverables of the HHS 2020 enterprise solution. - The SDD adheres to the principles of software quality assurance put forth by the QMP.
Risk Management Plan	- The Risk Management Plan demonstrates how to manage the complex environment of risks proficiently, including documentation, analysis, tools, escalation, and remediation. - Risk mitigation strategies are factored in the design of the SI Platform.
System Security Plan (SSP)	- The System Security Plan provides the security architecture approach to establish security and privacy controls for systems associated with the HHS 2020 enterprise architecture. - The System Security Design sections of the SDD will follow the guidelines prescribed by the SSP.
Installation Plan	- The Installation Plan provides the detailed steps necessary for the Installation and Configuration (I&C) of both hardware and software in setting up and managing all of the environments for the SI Platform. - The SDD is the prerequisite for the Installation Plan and serves as the blueprint on which the SDD is designed.

Deliverable Name	Relationship with SDD
Capacity Plan	• The Capacity Plan will describe their approach to measure the S Contractor's current hardware and software capacity and model future capacity based on the business needs identified by the State. • The design rationale of relying on Commercial Off-The-Shelf (COTS) products like Oracle Fusion Middleware is to accommodate the future capacity needs of the SI Platform.

1.6 Updates to System Design Document

The SI Contractor will take ownership of maintaining the SDD throughout the life cycle of the HHS 2020 enterprise solution and will coordinate with NM HSD's PMO and ITD staff to make any required changes or updates when new module contractors come on-board.

1.7 Current MMIS System

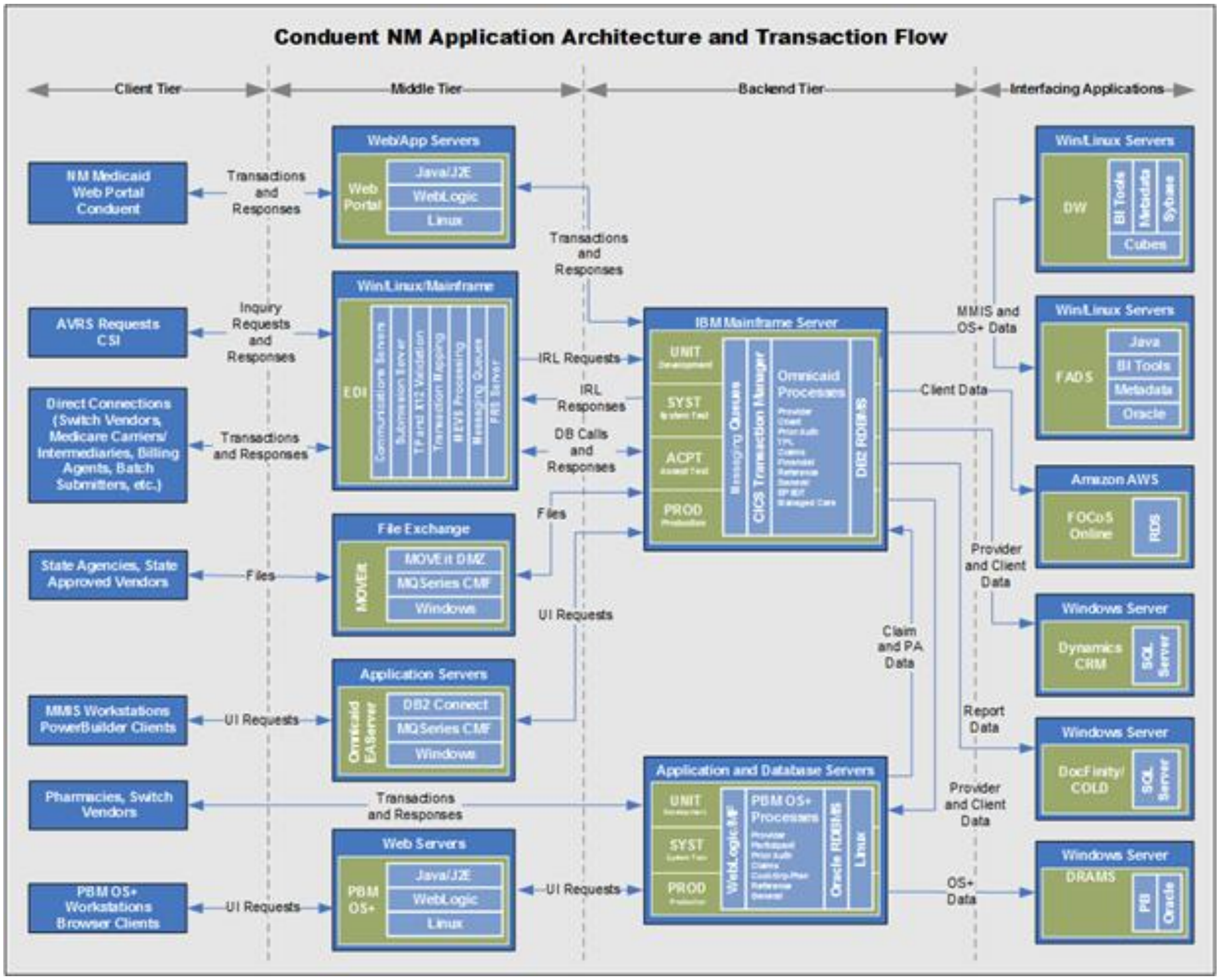
The following subsection describes the technical architecture of the current MMIS system, Omnicaid, depicted in the Figure 1 below and based on the description from the NM MMISR Concept of Operations (ConOps) document.

Omnicaid utilizes a three-tiered client-server architecture:

1. Client workstations run on Windows 7 Enterprise with PowerBuilder run-time, responsible for edit and front-end business logic.
2. Sybase Enterprise Application Server middle tier provides an abstraction layer to back-end CICS transactions and DB2 database on the mainframe.
3. Mainframe back end runs on IBM CICS and DB2 relational database management system.

The goals, objectives, and rationale for building the new system are explained in Section 4 of the ConOps Document.

Figure 1-1: Conduent NM Application Architecture



2 General Overview, Design Guidelines, and Approach

The following subsections outline the principles and strategies used as guidelines for designing the SI Platform.

2.1 General Overview

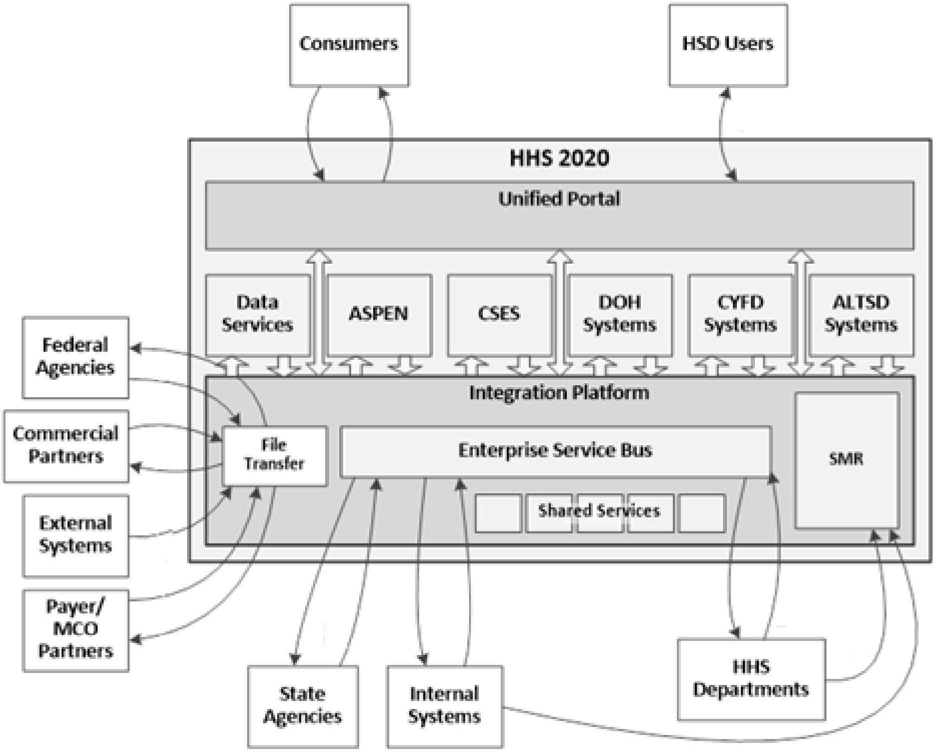
The SI in collaboration with NM HSD is tasked with developing, testing and implementing an SI Platform. This is at the heart of the MMISR and HHS 2020 vision for creating a technology platform that helps different business modules “talk to each other” through business services. It also provides a rich common purpose and efficient shared services. The SI Platform is also expected to collate enterprise data and make it available for reporting and insights. The platform should be secure and provide services to help make end-user access to State enterprise applications easy while being secure.

The design goals include extensibility, interoperability, service, and SOA driven solutions, as well as configurable and Commercial Off-The-Shelf (COTS) driven solutions. In addition to business services, these design goals include data and infrastructure driven technology services. SI has proposed and is implementing an Oracle Fusion Middleware (OFMW) based ESB and SOA platforms to achieve the design goals. The MarkLogic based Not Only SQL (NoSQL) driven data platform is also part of the SI Platform. It provides the extensible data platform needed to integrate all myriad data sources of the NM HHS 2020 program.

The SDD describes design goals and considerations, provides a high-level overview of the system architecture, and the data design associated with the system, as well as the human-machine interface and operational scenarios. The high-level system design is further decomposed into low-level detailed design specifications for each system component, including hardware, internal communications, software, system integrity controls, and external interfaces. The high-level system design serves as the primary input to the Preliminary Design Review (PDR). The low-level detailed design serves as input to the Detailed Design Review (DDR).

The following Context Diagram, referenced from the ConOps document that shows the types of external entities (organizations, people, and systems) that will interface with MMISR.

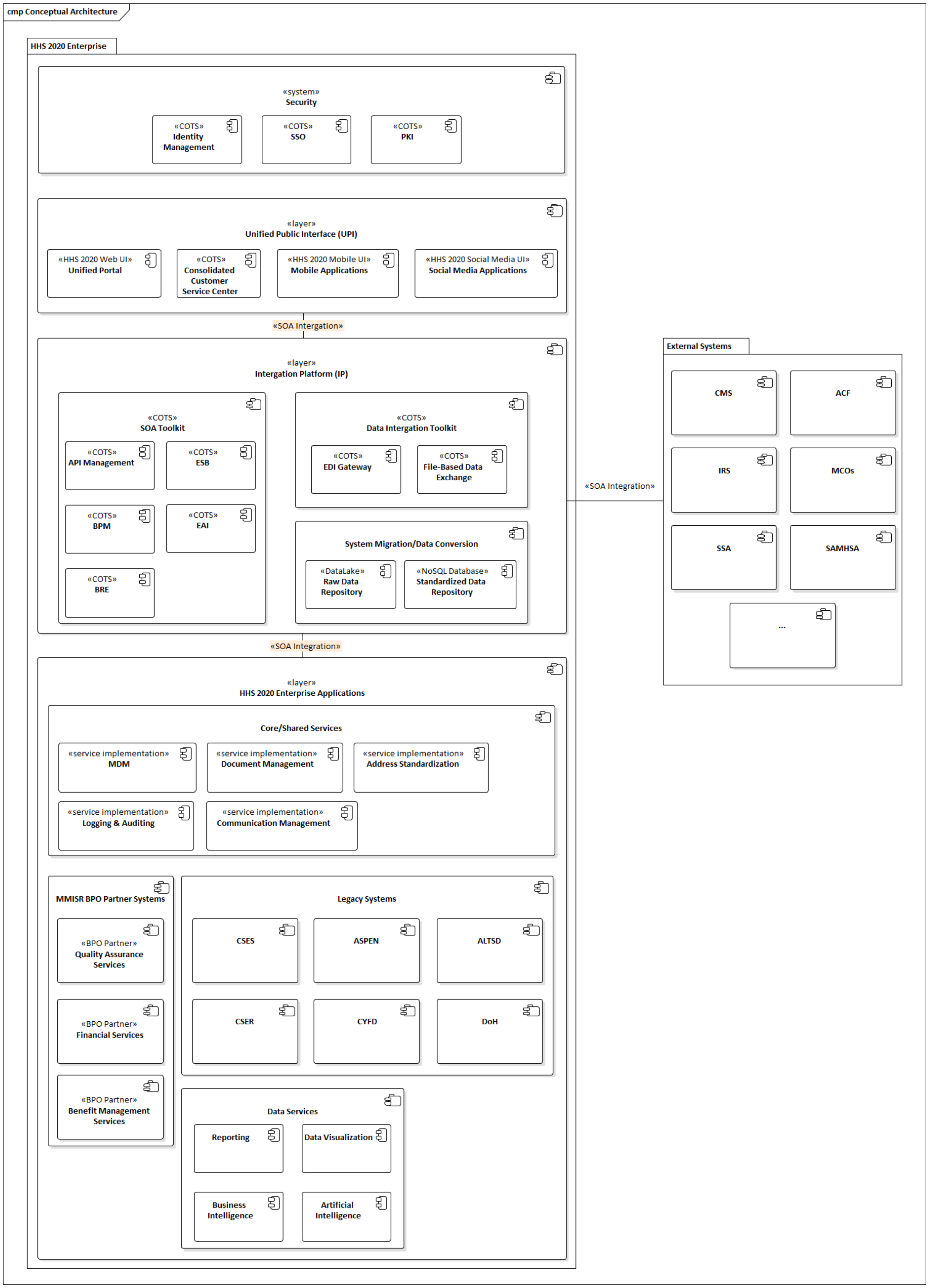
Figure 2-1: MMISR – Context Diagram



Please refer the ConOps document for more details about Context diagram for inventory of interfaces grouped by the major actor groupings in figure above.

The following figure is the conceptual architecture from the NM HHS 2020 Reference Architecture document.

Figure 2-2: NM HHS 2020 Conceptual Architecture



2.2 Assumptions, Constraints, and Risks

The following list of assumptions, constraints, and risks are associated with the SI Platform design document. This list will be updated when new module contractors come on board and identify additional assumptions or constraints that apply to their programs.

2.2.1 Assumptions

The following are the assumptions identified while writing this document. These assumptions were determined while architecting the SI Platform:

- The System Migration Repository (SMR) will be used for migration purposes only.
- The new modules are expected to adhere to the design goals, principles, and standards published by the SI Platform and exceptions will be handled on a case-by-case basis.
- All new modules are assumed to consume and publish canonical data defined and published by the SI Contractor models for message exchange.
- NM HSD assets like Active Directories are assumed to be in place and ready for integration to design and implement security solutions like Identity and Access Management.
- The existing licenses of NM HSD will be used for Linux / Windows servers. For any new-build, or the renewal of an existing license, NM HSD will purchase the licenses.

2.2.2 Constraints

The following is a list of constraints identified while writing this document:

- The overall architecture and system design captured in this document assume a “big bang” approach of MMISR module implementation rollout. This means that all modules (legacy and new), post integration, and user acceptance testing are moved into the production phase at the same time to replace the legacy MMIS system Omnicaid.
- The initial hardware and infrastructure set up and configuration will be done without VMWare vRealize Suite being acquired.
- If any integrating module is not able to publish and consume endpoints based on canonical data models published by the SI Contractor, it will be an additional effort to create virtualized proxy endpoints in the ESB layer and perform translation from native to canonical model.
- Enterprise Shared Services like Address Standardization, Validation & Verification, as well as others like Document Services or Communication Services will be built on enterprise software tools such as SAP or ImageNow. These enterprise software tools are foundational to the shared service design but have not yet been fully explored. A high-level understanding of their use is now being considered, along with the overall architecture and design of their shared services.

2.2.3 Risks

The following are the project risks associated with system design identified while writing this document. The SI Contractor documents project risks in SharePoint according to the Risk Management Plan. The latest list can be found in the Risk Management Plan and include:

- The integration and system testing of HHS 2020 enterprise solution will depend on the schedule of all integrating contractor systems. Any delay in the integration will impact the overall project schedule.
- Inconsistency in individual module contractor's development practices or tool usage can lead to difficulties in effectively managing the testing process.
- It is possible that the legacy interfaces may not be adequately documented, in which case as new modules take over the interfaces there may be issues with the message interaction and the data in the messages.
- DoIT approval of any new hardware and software acquisition can potentially alter the system design and architectural approaches.

Though the Integration Platform recommends security standards, protocols, and practices, some of the interface partners may not able to support the standards and instead support only a predecessor/deprecated version of them. For example, some interface partners will only be supporting Secure Sockets Layer (SSL), while the preferred standard for the Integration Platform is the successor of SSL, TLS (Transport Layer Security). Deprecated versions of protocols will lead to security risks like Man-In-the-Middle-Attack (MITM).

2.3 Alignment with Federal Enterprise Architecture

The SI Platform architecture is consistent with HSD's HHS 2020 enterprise architecture and includes guidelines and best practices regarding an efficient and sustainable approach to implementing the State's 2020 vision. The SI Platform's system design as driven by the prescribed federal Enterprise Architecture (EA).

CMS mandates MITA Application and Technical Architecture framework as the overarching architectural framework. This framework defines the relationship between end users, services and infrastructure and guides all State Medicaid Agencies and the business modules on how to connect services and infrastructures to improve services for end users.

3 Design Considerations

The following are the design considerations factored in the production of this design document. It explains the design goals, factors influencing design decisions and the overall design approach.

3.1 Goals and Guidelines

The MITA framework and CMS seven conditions and standards primarily drive the design of the SI Platform. Some of these key drivers are mentioned below as the goals and guidelines:

- **Modular Approach** – The MMISR architecture enables a highly available, horizontally scalable solution to easily expand the infrastructure needs, if the system load increases in the future. This modular approach is intended to create a framework aligned with MITA version 3.0, which supports NM HSD's goal of operating Medicaid functions at a MITA maturity level 4 in all business and technical areas.
- **Compliance with federal standards and State guidelines** – The MMISR solution will comply with the CMS Seven Conditions and Standards, promote the use of industry standards for information exchange and interoperability, and provide a seamless business services environment for users. The system also complies with MITA 3.0 requirements and standards and well as the CMS EA. The NM HSD supplied Enterprise Reference Architecture and the Technical Management Strategy documents the design guidelines and best practices to help in Architecture Governance. The Architecture Review Board (ARB) and Data Management Group act as governance bodies to review and evaluate design choices presented by SI as part of the continuous evolution of the SI Platform component design.
- **Tools-Driven** – The system design will maximize the use of readily available COTS tools while developing solutions and information exchange across workflows. The use of COTS products is preferred to minimize custom implementations.
- **Interoperable, Sustainable, and Adaptable** – The system design is built to maximize flexibility, reusability, and interoperability across workflows and platforms. The system makes use of SOA principles to deliver the solution, thus enabling continual enterprise evolution to meet evolving business needs.
- **Reusability** – The MMISR system will promote reuse of IT assets within and outside HHS organizations and systems.

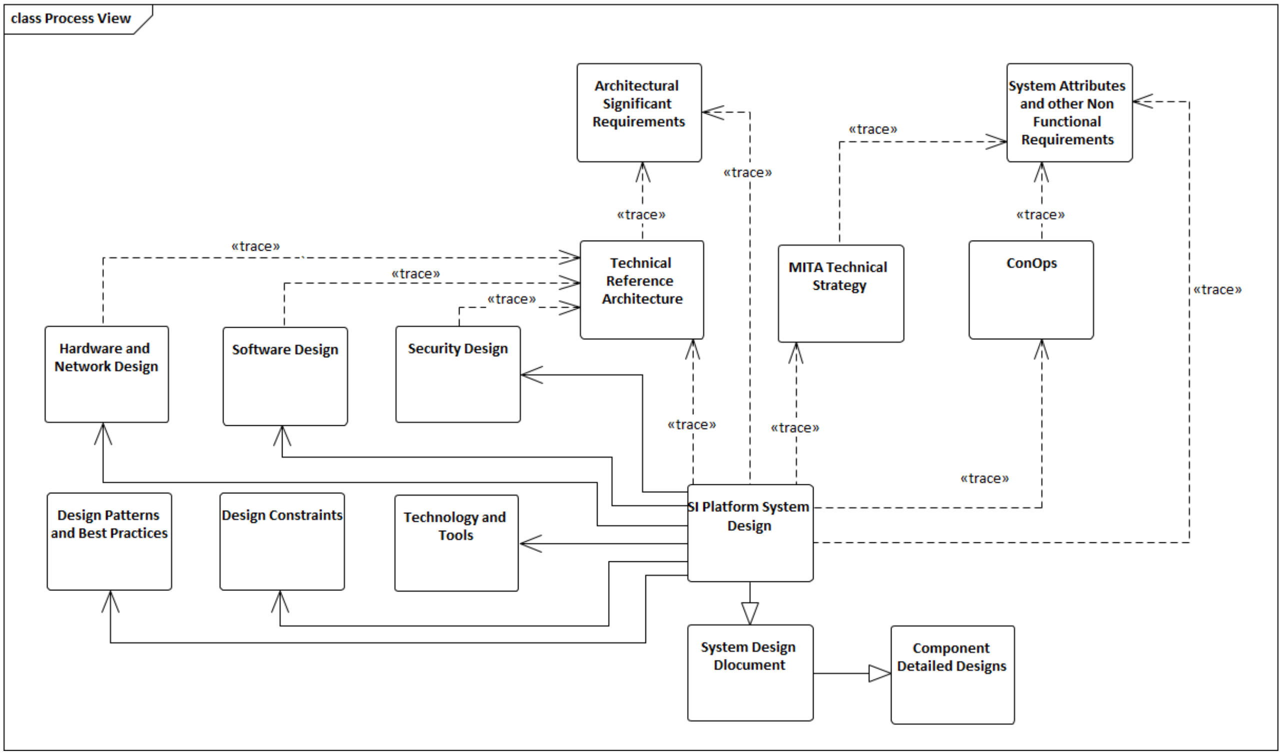
3.2 Development Methods and Contingencies

The SI Contractor's design approach takes into consideration all the requirements, standards, and factors influencing the design and supports an iterative evolution of this complex SI Platform. A component driven design of SI Platform that is driven by functional and non-functional and best practices is at the heart of providing inter-module communication platform.

This component design at a high level is captured in this SDD and acts as the overall blueprint for an iterative evolution of the individual components. These components are refined for gradual evolutionary changes involving configuration and/or customization, which are driven by newly discovered business requirements. The healthy eco-system of tools like Jama, Jira, and other Software Development Life Cycle (SDLC) aiding tools help in this evolution. The individual work stream and component design artifacts follow SDD as well as NM HSD/SMA supplied Enterprise Reference Architecture/Technical Management Strategy documents.

Figure 4 below explains the overall design approach.

Figure 3-1: Design Approach


class Process View
Architectural Significant Requirements
System Attributes and other Non Functional Requirements
«trace»
«trace»
«trace»
«trace»
Technical Reference Architecture
MITA Technical Strategy
ConOps
«trace»
«trace»
«trace»
«trace»
Hardware and Network Design
Software Design
Security Design
«trace»
«trace»
«trace»
SI Platform System Design
Design Patterns and Best Practices
Design Constraints
Technology and Tools
System Design Dlocument
Component Detailed Designs

3.3 Architectural Strategies

The following are the design decisions and/or strategies that affect the overall organization of the components of the SI Platform. These strategies provide insight into the key abstractions and mechanisms used in the system architecture.

The SI Platform encompasses the core infrastructure that will enable migration from the existing MMIS, communications across the MMISR solution as well as HHS 2020 participating modules, secure access to data and processes, functionality to support MMIS operations, data transfer, and data integrity.

The following are the software architecture guiding principles and architecture decisions that will be adhered to while designing and implementing the SI Platform.

Adherence to Standards:

- Build a standards-based SI Platform to integrate disparate systems and support different integration models (batch, real-time, etc.).
- The architecture follows standards like Medicaid Information Technology Architecture (MITA), National Information Exchange Model (NIEM), Health Insurance Portability and Accountability Act (HIPAA) and Health Level Seven International (HL7).
- Message Standardization - The solution will implement standard canonical message exchange data models between the legacy and new MMISR modules using NIEM, National Human Services Interoperability Architecture (NHSIA) and Federal Health Information Model (FHIM) models and message standards.

COTS Products:

- SI Platform prefers to use the COTS software and configure in place of custom code wherever possible unless required to support the required solution. The complete list of software, tools (including COTS) and libraries required to build SI platform can be found in Appendix G.
- The decision to implement a COTS solution for implementing SI platform enhances the ability to provide reliable, proven solution.

Security:

- Integrated security and privacy – The solution will include centralized security and access control mechanisms across the SI Platform. The solution will comply with (Minimum Acceptable Risk Standards for Exchanges) MARS-E specified security controls.
- Federal and State Security Compliance – The Architecture will meet federal and State security compliance requirements.

Performance:

- Augment traditional integration solutions with real-time integrations to reduce latency, improve decision-making, and responsiveness.
- Adaptable, extensible, and scalable – The SI Platform will implement an SOA and Event-Driven integrations in a modular and loosely coupled fashion to accommodate future expansion in response to evolving MMISR business requirements.
- Performance Scalability – The system design will meet performance and scalability needs.

Interoperability:

- The Architecture will follow SOA design principles to ensure seamless functionality between the SI Platform and other entities.
- The architecture will support reusability and interoperability of discrete services across multiple systems.

Governance:

- Build a centralized governance mechanism to analyze, monitor the usage of the platform and maintain a record of resource levels and consumption within the solution.

4 System Architecture and Architecture Design

The following sections outline the SI Platform’s architecture design and the architectural considerations that went into achieving the non-functional design goals.

4.1 Logical View

The software architecture logical view provides a high-level description of the components and support functions that characterize SI Platform and its operations. The logical view of architecture shows, at a high level, all business capabilities, functionalities, and the internal/external systems participating in the business. The logical view also focuses on the types of users who would be using the system, and the access channels to the overall system built.

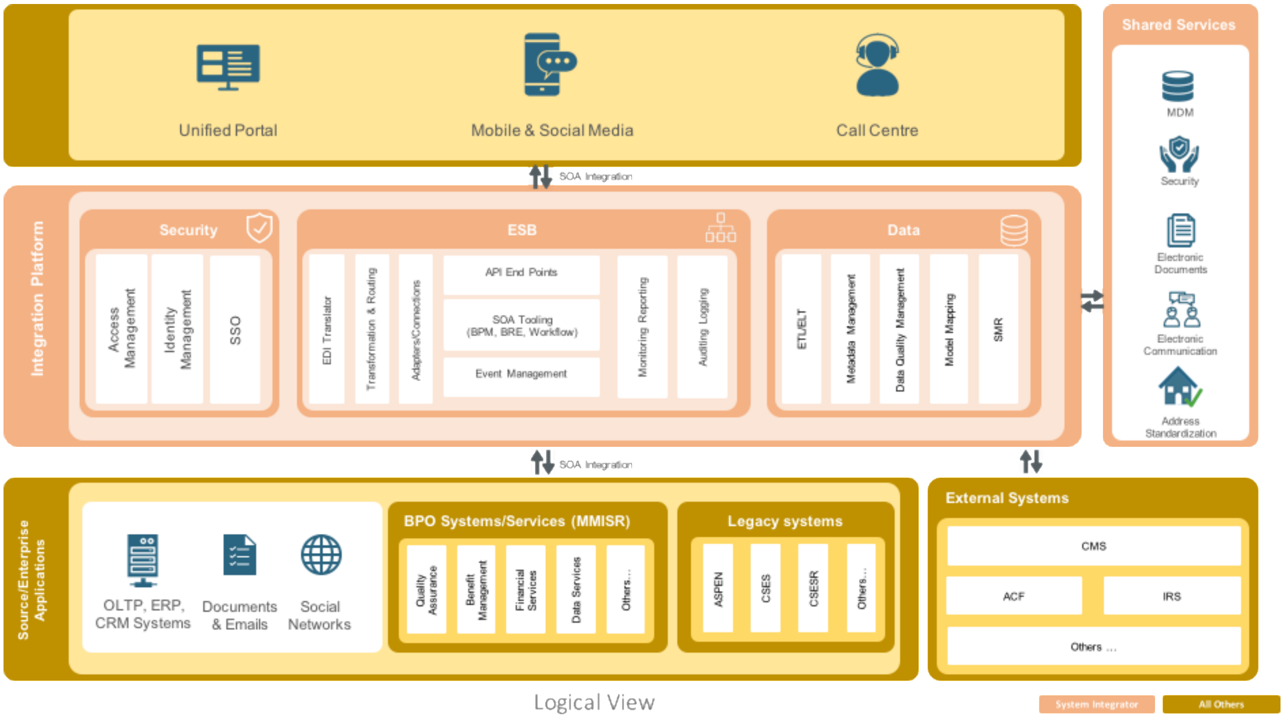
The SI Platform's logical view consists of the following components:

- ESB, which is the bus on which all modular interactions happen.
- Data platform on which integrated data is made visible and supplied to all interested parties.
- Security implementation, which includes identity access management and federated single sign-on features.
- Enterprise shared services built on common infrastructure leveraging enterprise assets to offer common business functions like document management, customer communications, and common business utility functions such as Address services and access to the Master Data Management (MDM), which consist of a mastered view of enterprise data.

These SI Platform components interact with end users via the collective user interfaces of in a device-neutral way by facilitating an SOA integration with ESB and other modules/systems. The legacy modules like ASPEN and newly on-boarded modules like financial services and others will also be attached to the ESB through SOA integration and can talk to each other. The external systems, which include other State agencies, federal agency systems as well as third party vendor systems, will connect to the SI Platform through different synchronous or asynchronous communication mechanisms and can transact to offer and consume services.

A detailed breakdown of all the logical and physical layers of the SI Platform solution is captured in software architecture, which deals with detailed design.

Figure 4-1: Logical View



4.2 Hardware Architecture

This subsection describes the overall system hardware and organization, indicating whether the processing system is distributed or centralized. It Identifies the type, number, and location of all hardware components including the presentation, application, and data servers, any peripheral devices, and resource estimates for processor capacity, memory, online storage, and auxiliary storage.

4.2.1 Virtual Hardware Architecture

The SI Platform runs on VMware virtual infrastructure on top of DELL VxRack Flex 1000 Hyper-Converged Infrastructure (HCI) hardware. The following subsections outline the details for the hardware and the VMware virtual infrastructure. Two racks are utilized to mount 15 Dell PowerEdge R640 1U1IN nodes for the compute cluster and 3 nodes for the management cluster. The management cluster hosts the Virtual Machines (VM) that manage the infrastructure, and the compute cluster hosts all the systems of the SI Platform. The compute cluster is broken down into three workload clusters of five nodes each. Other VMware software will be installed in the management cluster as needed, such as NSX and vRealize.

The Disaster Recovery environment has no physical hardware and will be configured and hosted in the Oracle Government Cloud, where all the is replicated using third-party software.

The VxRack Flex 1000 System for the SI Platform is modular and allows extreme scalability and flexibility for mixed workloads and tier 2 applications. The system is configured for deployments of large numbers of VMs. This allows the SI Platform to be scalable vertically and horizontally. Horizontal scaling is achieved by adding, moving, and removing nodes on-the-fly.

The figures and table below describe the nodes and network switches placed in these racks.

4.2.1.1 *VxRack layout of the SI Platform*

Compute – VxRack Flex 1000 compute enclosures are high-density, two-socket, 1 RU rack-mount enclosures that are built for production-level applications. There are 15 compute nodes with all flash storage.

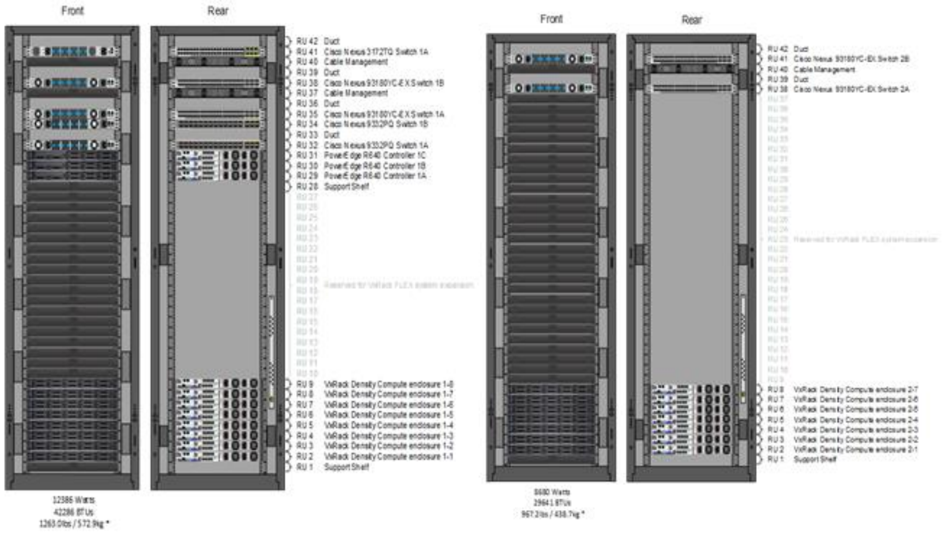
Network – The Cisco Nexus switches in the network layer provide 10 Gbe Internet Protocol (IP) connectivity between the VxRack Flex 1000 System and the NM HSD network (East-West and North-South Traffic). VxRack Flex 1000 includes Cisco Nexus 93180YC-EX Top of Rack (ToR) switches, as well as the Cisco Nexus 9332 inter-rack switches.

The compute layer connects to the Ethernet component of the network layer.

Storage – VxRack Flex 1000 System uses the VMware Virtual Storage Area Network (VSAN) and VxFlex OS software-defined storage technology. VSAN is used for the management cluster of the Integration Platform that allows creation of a VSAN from the local data stores on clustered ESXi instances. It then becomes a shared data-store spanning all the disks from the local data stores on clustered ESXi instances, hence it becomes a shared-data spanning all the hosts within the 3-node cluster.

For the compute cluster, the VxFlexOS storage will be configured across all the three clusters with 15 nodes as one protection domain based on the Dell best practices.

Figure 4-2: Rack elevation of the VxRack Hardware


Front
Rear
RU 42 Duct
RU 41 Cisco Nexus 3172TQ Switch 1A
RU 40 Cable Management
RU 39 Duct
RU 38 Cisco Nexus 93180YC-EX Switch 1B
RU 37 Cable Management
RU 36 Duct
RU 35 Cisco Nexus 93180YC-EX Switch 1A
RU 34 Cisco Nexus 9332PQ Switch 1B
RU 33 Duct
RU 32 Cisco Nexus 9332PQ Switch 1A
RU 31 PowerEdge R640 Controller 1C
RU 30 PowerEdge R640 Controller 1B
RU 29 PowerEdge R640 Controller 1A
RU 28 Support Shelf
RU 9 VxRack Density Compute enclosure 1-8
RU 8 VxRack Density Compute enclosure 1-7
RU 7 VxRack Density Compute enclosure 1-6
RU 6 VxRack Density Compute enclosure 1-5
RU 5 VxRack Density Compute enclosure 1-4
RU 4 VxRack Density Compute enclosure 1-3
RU 3 VxRack Density Compute enclosure 1-2
RU 2 VxRack Density Compute enclosure 1-1
RU 1 Support Shelf
12386 Watts
42286 BTUs
1263.0lbs / 572.9kg *
Front
Rear
RU 42 Duct
RU 41 Cisco Nexus 93180YC-EX Switch 2B
RU 40 Cable Management
RU 39 Duct
RU 38 Cisco Nexus 93180YC-EX Switch 2A
RU 8 VxRack Density Compute enclosure 2-7
RU 7 VxRack Density Compute enclosure 2-6
RU 6 VxRack Density Compute enclosure 2-5
RU 5 VxRack Density Compute enclosure 2-4
RU 4 VxRack Density Compute enclosure 2-3
RU 3 VxRack Density Compute enclosure 2-2
RU 2 VxRack Density Compute enclosure 2-1
RU 1 Support Shelf
8680 Watts
29641 BTUs
967.2lbs / 438.7kg *

4.2.1.2 *VxRack Hardware Specifications*

The SI Platform VxRack Flex 1000 System includes compute, network, storage, virtualization, and management components. All the physical servers have identical Central Processing Unit (CPU) and memory configuration in the management and compute clusters respectively.

Table 4-1: VxRack Components

Resource	Components	Hardware Specifications	Quantity
Compute	VxRack 1 RU Node Compute enclosures	VxRack Nodes 1U1N-232/576G Total – 420 Cores & 8.6TB RAM	15
Network	Top of Rack (ToR) Switches	Cisco Nexus 93180YC-EX TOR Switches	4
Inter-rack Switch	Cisco Nexus 9332 Aggregation Switches	2	
Storage	Dell EMC FlexOS (ScaleIO) HCI	FLEX-SYS-6412 - 10x3.84TB SAS = 38.4 TB	15 = 262.08 TB
VxRack Management Cluster		Cisco Nexus 3172T Switch	1
		Uplink 10G to NM HSD network	2
		VxRack Management Servers - Single Socket Intel 6130 (16 cores)192GB RAM 5x1.92TB SSD	3
		VxRack Manager	1

4.2.1.3 Network Cable Layout

The diagram below describes how the data network path is configured for the SI Platform VxRack Flex 1000 in NM HSD. The VxRack Flex 1000 nodes are connected to the Top of Rack (TOR) switches, which handles all the East-West Traffic within the same subnet. For traffic across subnets, the East-West traffic flows to the Palo Alto Firewall. The Top of Rack switches is connected to the aggregate switches which handles all North-South bound traffic with the next network hop being the NM HSD core switch and to the Firewall.

Figure 4-3: Network Cable Layout

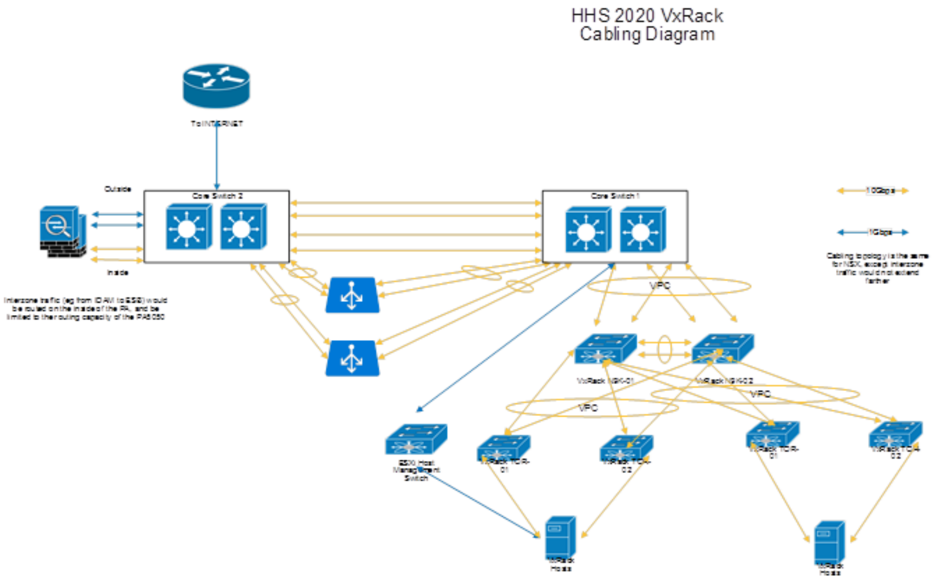
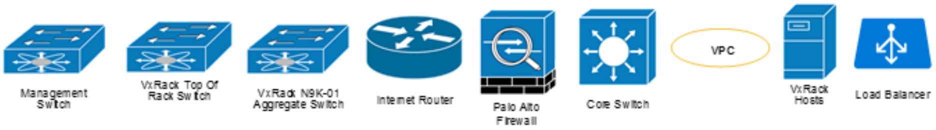

HHS 2020 VxRack
Cabling Diagram
To INTERNET
Outside
Core Switch 2
Core Switch 1
Inside
Interzone traffic (eg from IDAM to ESB) would be routed on the inside of the PA, and be limited to the routing capacity of the PA5050
10Gbps
1Gbps
Cabling topology is the same for NSX, except interzone traffic would not extend farther
VPC
VxRack N9K-01
VxRack N9K-02
VPC
VPC
ESXi Host Management Switch
VxRack TOR-01
VxRack TOR-02
VxRack TOR-01
VxRack TOR-02
VxRack Hosts
VxRack Hosts

Figure 4-4: Network Icons



Top of Rack Switch – This provides network connectivity across subnets and keeps all the internal traffic within the VxRack, referred to East-West Traffic. At NM HSD, traffic within the same subnet traverse through the Top of Rack switch only and traffic across subnet (North-South) traverses through the Palo Alto Firewall.

Load Balancer – This provides redundancy and load balancing between clustered servers, so for the SI Platform minimal downtime.

Aggregate Switches – This provides network connectivity for network traffic flowing out of the VxRack infrastructure referred to as North-South traffic.

VxRack – Dell Hyper-Converged System with a complete set of compute, storage, and network infrastructure ready to be deployed in the environment.

Virtual Port Channel (VPC) – VPC is configured on switches for increased bandwidth and redundancy. The Top of rack switches are configured to the Aggregation switches with VPC and, the Aggregation switches are connected to the core switches to provide extra bandwidth and redundancy.

Management Switch – This provides connectivity to the infrastructure hosts as a redundant path to login interface is unavailable.

Core Switch – The core switch provides network connectivity for systems to the VxRack for internal and external traffic, as NM HSD core switches are Layer 2 communications.

Firewall – This provides protection to any external traffic flowing in and out of the NM HSD network and all layer 3 network traffic flow through the firewall at NM HSD.

Internet Router –Gateway router that allows internet traffic in and out of the datacenter.

4.2.1.4 Physical Servers

Below is the physical server specification for the Platform at NM HSD. The DR for the SI Platform at NM HSD is hosted in the cloud (Oracle Government Cloud - OGC or Amazon Web Services - AWS).

Table 4-2: Physical Servers

Server Name	Environment	Cluster	Operating System	CPU Cores	RAM - Gb	Disk - TB
esxvxhst01	Prod	VX-Prod	ESXi 6.5	28	576	38.4
esxvxhst02	Prod		ESXi 6.5	28	576	38.4
esxvxhst03	Prod		ESXi 6.5	28	576	38.4
esxvxhst04	Prod		ESXi 6.5	28	576	38.4
esxvxhst05	Prod		ESXi 6.5	28	576	38.4
esxvxhst06	Non-Prod1	VX-NON-PROD1	ESXi 6.5	28	576	38.4
esxvxhst07	Non-Prod1		ESXi 6.5	28	576	38.4
esxvxhst08	Non-Prod1		ESXi 6.5	28	576	38.4
esxvxhst09	Non-Prod1		ESXi 6.5	28	576	38.4
esxvxhst10	Non-Prod1		ESXi 6.5	28	576	38.4
esxvxhst11	Non-Prod2	VX-NON-PROD2	ESXi 6.5	28	576	38.4
esxvxhst12	Non-Prod2		ESXi 6.5	28	576	38.4
esxvxhst13	Non-Prod2		ESXi 6.5	28	576	38.4
esxvxhst14	Non-Prod2		ESXi 6.5	28	576	38.4
esxvxhst15	Non-Prod2		ESXi 6.5	28	576	38.4
esxvxmgt01	VXMA	Management Cluster	ESXi 6.5	1	191	7

Server Name	Environment	Cluster	Operating System	CPU Cores	RAM - Gb	Disk - TB
esxvxmgt02	VXMA	Management Cluster	ESXi 6.5	1	191	7
esxvxmgt03	VXMA	Management Cluster	ESXi 6.5	1	191	7

4.2.2 Security Hardware Architecture

SI Platform's hardware architecture includes the firewalls, routers, load balancers, and the configuration supporting the security and privacy of the system.

4.2.2.1 Palo Alto

The Palo Alto firewall provides protection for external (North-South) traffic and for internal (East-West) traffic across subnets. The Palo Alto is a next-generation firewall that offers traditional stateful firewall capabilities along with deep packet inspection (DPI). DPI will check for malicious code and drop any suspicious traffic. Additionally, the Palo Alto firewall has a built-in Intrusion Protection System that will be used to monitor the network, detect known virus signatures, and drop those packets. Virtual Private Network (VPN) will be deployed to enhance confidentiality and integrity over remote connections. The VPN connection will leverage Internet Protocol Security and Internet Key Exchange version 2 (IKE v2) with the American Encryption Standard (AES) 256 for remote connections.

4.2.2.2 F5 Load Balancer

The F5 load balancer will also be utilized in an High Availability (HA) mode which will ensure the web traffic is managed between the different components of the application. F5 also has built-in security features which can detect security attacks related DDOS.

4.2.3 Performance Hardware Architecture

Dell VxRack is a scale-out, high performance and extremely resilient infrastructure system with unmatched performance. The hardware is built with high resiliency and there is no single point of failure within the HCI system. The hardware can be scaled vertically and horizontally by adding new nodes and removing old nodes.

The HCI maintains redundant copies of data; the data is recovered when hardware components (disks) fail. Data loss prevention is achieved by distributing data across a larger pool of storage media. When a storage device inside a ScaleIO in the storage pool fails, ScaleIO will swiftly re-protect any affected data. ScaleIO's balanced layout of data means that every storage device in a storage pool will have equivalent levels of primary data, mirrored data, and free space. The wide-striping, mirrored mesh layout of ScaleIO volumes, allows the data to be striped across every storage device in the storage pool.

4.2.4 Virtual Architecture Overview

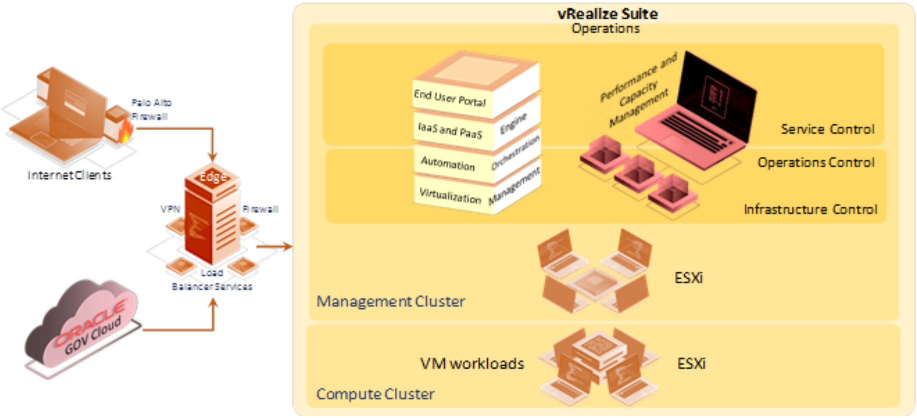
This subsection gives an architectural design overview of virtualization products being implemented for the SI Platform on the VxRack Flex 1000 Dell Hyper-Converged Infrastructure (HCI) for multiple virtual environments at NM HSD. It also offers an overview of how those products are deployed into compute and management.

The SI Platform virtual environment consists of vSphere and vRealize Suite solutions. The figure below provides the design of the environment. Dell VxFlexOS Storage and Layer 2 Networking are provided by Dell and Cisco technologies. Oracle GOV Cloud is used to provide Disaster Recovery services between Santa Fe and OGC datacenters. Third-party software will be used for replicating data to the cloud environment, mirroring the Production environment. vCenter is the critical software of VMware, which controls all other components, once the vCenter is installed and reachable via the browser the following object folders are created to begin with:

1. Datacenter
2. Clusters
3. Hosts

The datacenter is the top-level folder that holds the clusters, and the clusters hold the ESXi hosts upon which the virtual machines are installed and configured.

Figure 4-5: VMWare Architecture



The management cluster is used to run the components required to support the SI Platform at NM HSD datacenter such as management, monitoring, and infrastructure services. A management cluster provides resource isolation, which helps services operate at their best performance level. A separate cluster is configured to have physical isolation between management and production hardware.

The compute cluster supports virtualized network devices that provide interconnectivity between environments. These network devices provide protected capacity by which internal platform traffic connects via gateways to external networks (North-South Traffic). Networking services and network traffic management occur in this cluster on a distributed virtual switch (DVS). The compute cluster also supports the delivery of all other external traffic. Multiple compute workloads are segregated using port groups to separate the environment and spread the load for different workload types for the SI Platform.

The information below describes the VMware components that are being installed for the SI Platform.

Figure 4-6: Overview of the Components of the SI Platform

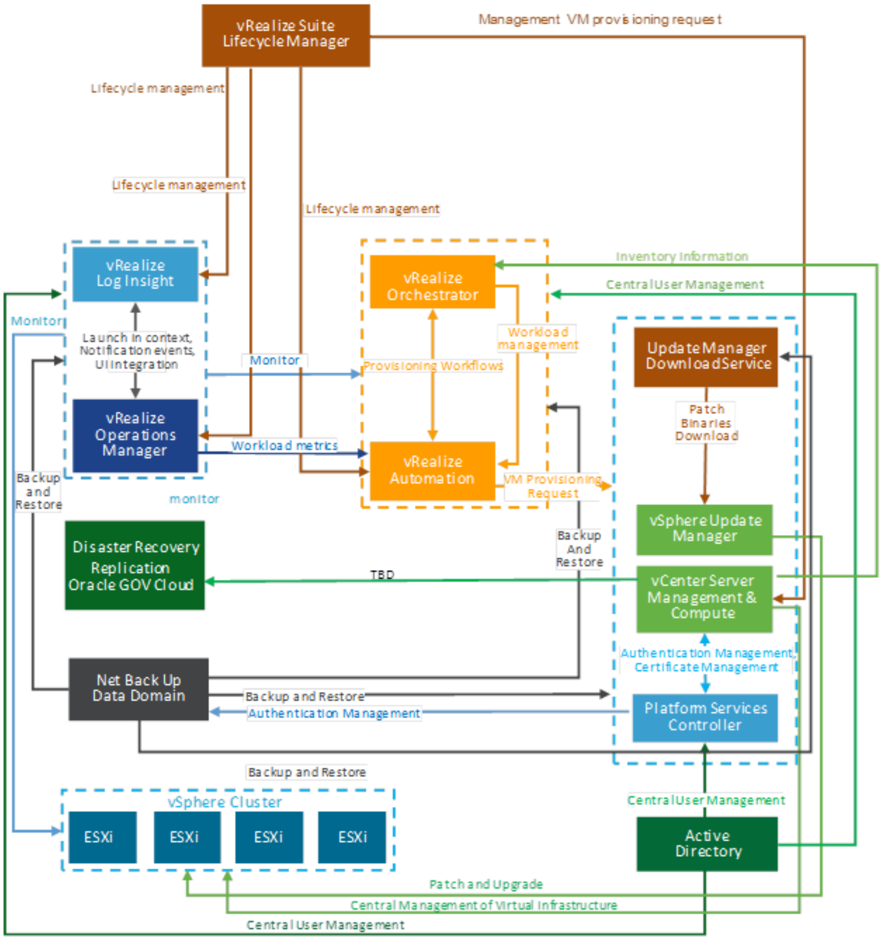


Table 4-3: VMware Components Descriptions

VMware Components	Function Description
ESXi hypervisor	Provides bare-metal virtualization of servers so you can consolidate applications using less hardware. ESXi includes vSAN & Flex OS for hyper-converged storage.
vCenter Server	Provides a centralized platform for managing vSphere environments and provides a set of common infrastructure services that encompasses single sign-on (SSO), licensing, and a certificate authority (CA).
SSO (Single Sign ON)	Single Sign ON is used in vCenter so a given user can log in to multiple application using the same credentials while connected to the network.

VMware Components	Function Description
Platform Service Controller (PSC)	Platform services come with two versions (embedded and external) it is used for Authentication, licensing, certificate services that will be used at NM HSD.
vRealize Automation	Provides a self-service, policy-enabled IT and application services catalog for deploying and provisioning physical infrastructure, hypervisors, and virtual machines.
vRealize Operations	Provides a set of components for the automation of operations including infrastructure health, configurations and compliance, application discovery, and monitoring of hardware and software
vRealize Operations Manager	Provides comprehensive visibility and insights into the performance, capacity, and health of your infrastructure.
vRealize Infrastructure Navigator	Provides automated discovery of application services, visualizes relationships, and maps dependencies of applications on virtualized compute, storage, and network resources.
vRealize Log Insight	Provides analytics capabilities to unstructured data and log management, which gives operational intelligence and deep, enterprise-wide visibility across all tiers of the IT infrastructure and applications.
vRealize Orchestrator	Provides the capability to create workflows that automate activities, such as provisioning VM, performing scheduled maintenance, and starting backups.

The VMware products also have dependencies external components as shown in the table below.

Table 4-4: External Components

Components	Function Description
Identity source – Active Directory	Identity sources (Active Directory, or Local OS) or similar is required to implement and operate the vRealize Suite infrastructure.
DNS	DNS must be configured for connectivity between vCenter Server, Active Directory, ESXi hosts, and the VMs.

Components	Function Description
Time synchronization	Accurate timekeeping and time synchronization is critical for a healthy infrastructure. All components (including ESXi hosts, vCenter Server, the SAN, physical network infrastructure, and VM guest operating systems) must have accurate timekeeping.
Database	Embedded PostgreSQL DB
Third Party Replication Software	Allows synchronization with systems in the Oracle Government Cloud.
Backup	Veritas NetBackup Solution will be used for backing up data.

4.2.4.1 Virtualization Components Version

The following table lists the product versions used in this design. It is a best practice to use the newest version available on the vendor website or use the NM HSD approved versions.

Table 4-5: VMware Products and Versions

VMware Product	Version Number	Build Number
Platform Services Controller Appliance	6.5	9451637
vCenter Server Appliance	6.5	9451637
vSphere Automation	6.5	XXXXXX
VMware ESXi	6.5	10175896
vSphere Update Manager	6.5	40982
vROPS	7.X	XXXXXX
vSAN	6.5	XXXXXX
vRealize Suite	6.5	XXXXXX
vRealize Log Insight	6.5	XXXXXX

4.2.4.2 Management Cluster Servers

The number of VMware components in the management cluster increases as capabilities are added. This section addresses the VMware management components that are being used. Third-party add-ons must be sized separately.

4.2.4.3 Management Cluster Virtual Machine

This subsection provides end-to-end sizing of an entire VMware environment including DELL/EMC Vision software for systems management. The number of virtual CPUs, memory size, storage size, and network bandwidth are given for each VM, and the VMs are grouped by each major component or appliance at a high-level.

The SI Platform virtual infrastructure is load balanced and fault-tolerant using vSphere HA. When there is a failure, the vCenter automatically restarts the VM on another server within the cluster, but there is minimal downtime while the services come back online.

The table below lists each management cluster VM for vSphere, with its recommended size of virtual CPUs, RAM, storage, and networking.

Table 4-6: Management Cluster VMs for vSphere

VM Description	CPU (vCPUs)	Memory (GB)	Storage (GB)	Network bandwidth
vCenter Server with embedded PSC(1) Management Cluster	8	24	50	1 Gbe
vCenter Server(2) Edge and Compute Cluster with external PSC	8	24	50	1 Gbe
vRealize Orchestrator Appliance	2	3	12	1 Gbe

The table below lists each management cluster VM for vRealize Automation with its size in terms of virtual CPUs, RAM, storage, and networking.

Table 4-7: Management Cluster VMs for vRealize Automation

VM Description	CPU (vCPUs)	Memory (GB)	Storage (GB)	Network Bandwidth
vRealize Automation Appliance	4	16	30	1 Gbe
Infrastructure Web Server	2	4	40	1 Gbe
Infrastructure Manager Server	2	4	40	1 Gbe
vSphere Proxy Agent	2	4	40	1 Gbe
vRealize Application Services	8	16	50	1 Gbe

This table below lists each management cluster VM for vRealize Operations Manager with its size in terms of virtual CPUs, RAM, storage, and networking.

Table 4-8: Management Cluster VMs for vRealize Operations Manager

VM Description	CPU (vCPUs)	Memory (GB)	Storage (GB)	Network Bandwidth
vRealize Operations Manager – Master	4	16	500	1 Gbe
vRealize Operations Manager – Data	4	16	500	1 Gbe
vRealize Configuration Manager Database (MS SQL)	4	16	1000	1 Gbe
vRealize Hyperic Server	8	12	16	1 Gbe
vRealize Hyperic Server - PostgreSQL DB	8	12	75	1 Gbe
vRealize Infrastructure Navigator	2	4	24	1 Gbe

The table below lists each of the remaining management cluster VMs.

Table 4-9: Other Management Cluster VMs

VM Description	CPU (vCPUs)	Memory (GB)	Storage (GB)	Network bandwidth	High Availability
vRealize Log Insight	8	16	100 GB	1 Gbe	Cluster of 3 nodes

Hyper-Converged Storage Using vSAN

The total capacity for the management cluster VMs is 20.96 TB. In NM HSD there are three ESXi hosts configured with vSAN. Managed by two virtual machines in Active/passive mode and a third one is the witness VM.

4.2.4.4 Server Configuration

The management cluster has three hosts for high availability and to provide a quorum that re-runs in a vSAN hyper-converged cluster. The following VMs are the initially required management servers.

Table 4-10: Server Configuration

Function	VM Name	Total Cores	RAM	HA Protected
Flex OS	itdsfavxflxm01.nmhsd.lcl	4	16 GB	true
Jump Host	itdsfavxjmp01.nmhsd.lcl	2	4 GB	true
Management vCenter	itdsfavxmvc01.nmhsd.lcl	4	16 GB	true

Function	VM Name	Total Cores	RAM	HA Protected
Open Manage Enterprise	itdsfavxome01.nmhsd.lcl	4	8 GB	true
Platform Service Controller	itdsfavxpsc01.nmhsd.lcl	2	4 GB	true
Platform Service Controller	itdsfavxpsc02.nmhsd.lcl	2	4 GB	true
Secure Remote Support	itdsfavxsrs01.nmhsd.lcl	1	4 GB	true
Compute vCenter	itdsfavxvc01.nmhsd.lcl	16	32 GB	true
Vision	itdsfavxvis01.nmhsd.lcl	2	8 GB	true
Vision	itdsfavxvis02.nmhsd.lcl	4	12 GB	true
Scale IO	SIO-10.63.193.132-GW	2	3 GB	true

4.2.4.5 Compute Cluster Servers

For the SI Platform, as part of this design, vSphere clusters will be created to aggregate hosts. This design is based on the following assumptions:

- Host failures for VMware HA are expressed in a number of allowable host failures, meaning that the expected load should be able to run on surviving hosts. HA policies can also be applied to a percentage spare capacity basis.
- Host with minimum workload will be used for maintenance to offload running VMs from other hosts that need to undergo maintenance. When not being used for maintenance, such host will provide demands for resources.
- Hosts will be appropriately placed in the designated clusters per their workload and purpose.

4.2.4.6 ESX Cluster Details

The table below provides the total number of ESXi clusters for the SI Platform.

Table 4-11: ESX Cluster Details

Cluster	Santa Fe DC
Production	5 Hosts
Non-Prod 1	5 Hosts
Non-Prod 2	5 Hosts
VxRack Management	3 Hosts

4.2.4.7 ESXi Design Specifications

The following table provides the ESXi design specifications.

Table 4-12: ESXi Design Specifications

Attribute	Specification
Host type and version	ESXi 6.5 UP2
Number of CPUs	2-CPU 14 Core 2.2 GHz E5-2650v4 Xeon Processors
Number of cores	2*14 Cores
Total number of cores	28
Processor speed	2.32 GHz Processors
Memory	576 GB Memory
Number of NIC ports	3
Host Naming convention	FUNCTION/LOCATION/ /NUMBER
Boot source	Local
Partitioning	Default
Syslog	Log Insight
Authentication	LDAP/AD + Root
Time Synchronization	External Time source or DC
Alert Monitoring	Virtual Center and VROPS

4.2.4.8 Cluster Resource Distribution

The following table provided the cluster resource distribution.

Table 4-13: Cluster Resource Distribution

Cluster	Resource Pool	Priority	Share Value	Reservation	Hosts	HA	DRS	Storage
MGMT	2 RP	Normal	Normal	No	3	Yes	Yes	Storage Pool
PROD	7 RP	High	Normal	20%	5	yes	Yes	Storage Pool

Cluster	Resource Pool	Priority	Share Value	Reservation	Hosts	HA	DRS	Storage
NON-PROD-2 UAT/PRODSUPPORT /PRODPATCH	3 RP	Low	Normal	No	5	Yes	Yes	Storage Pool
NON-PROD-2 DEV/QA/SIT	3 RP	Low	Normal	No	5	Yes	Yes	Storage Pool

Note: This configuration may change over time.

4.2.4.9 VMware Licensing

The tables below provide the VMware licensing. It is recommended that NM HSD purchase the vRealize Enterprise for the SI Platform.

Table 4-14: VxRack & vSphere SW Licensing

VxRack & VSphere SW Licensing	License Quantity
EMC ScaleIO Entitlement	576
VMware vSphere Enterprise Plus	30
VMware vCenter - Management Control (AMP) and Workload Cluster	2
Vision for VxRack FLEX	18
VxRack Manager – Management Control (AMP)	1
Microsoft Windows Server Std. 2012 (for mgmt. components)	3
VMware vSAN Enterprise Plus Edition (per CPU socket) – Management Controllers (AMP)	3

The VMware vSphere Enterprise Plus license edition will be used for the virtual infrastructure for the SI Platform,

This edition provides the following licensed features:

- 24vCPU/VM
- vMotion
- Hot add virtual hardware support
- Storage vMotion
- High Availability
- Data Protection and Replication
- vShield Endpoint

- Distributed Resource Scheduler
- Power Management
- Storage APIs for Array Integration, Multi-pathing
- Distributed Switch
- Storage DRS and Profile-Driven Storage I/O
- Host Profiles and Auto Deploy

Additional licenses to be procured follow.

The table below lists the standard and optional components that are provided for the vRealize Suite License.

Table 4-15: VMware vRealize Suite Licensing

License	Component	vRealize Enterprise	Version
Base	vSphere		TBD
	vRealize Automation	Enterprise	TBD
	vRealize Operations		TBD
	vRealize Operations Manager		TBD
	vRealize Infrastructure Nav		TBD
	vRealize Hyperi		TBD
	vSphere Replication		TBD
	vSphere Data		TBD
	vRealize Log Insight		TBD
Other licenses	vRealize Orchestrator		TBD

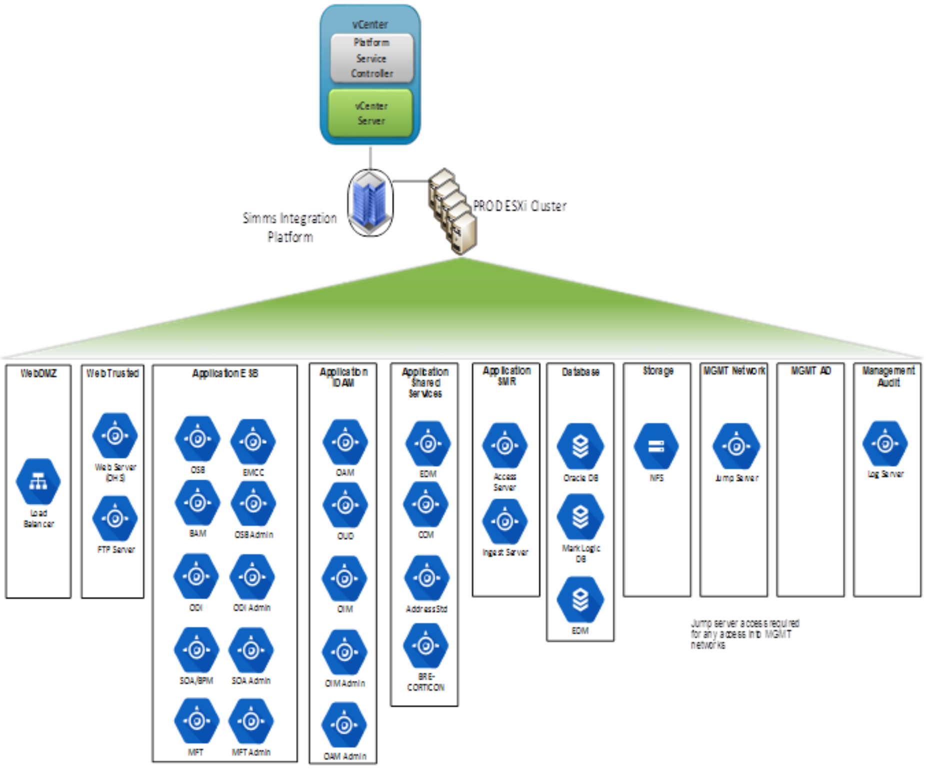
4.2.4.10 VMWare Design

The SI Platform vCenter Server Appliance for the management cluster is configured with an embedded PSC, and the vCenter Server Appliance for the compute cluster is configured with two external PSCs, in Active/Standby mode, both will be deployed in the management cluster. The vCenter will use the built-in vPostgres DB with an SSO domain configuration integrated with NM HSD existing Active Directory for authentication and authorization.

By utilizing this logical design and creating separate virtual environments, the SI Platform can meet the requirements unique to each virtual environment. Maintaining separation between the environments will allow each environment to be upgraded separately. This is a key NM HSD requirement for the SI Platform.

The following figure shows the logical design of the SI Platform virtual environment with the segmentation of each of the functional components and their respective network zones. Servers are grouped and segmented based on the modules for the SI Platform.

Figure 4-7: Virtualization Logical Design



The table below explains the function and network segments for the SI Platform.

Table 4-16: Function and Network for the SI Platform

Server Name	Zone	Server Tag	Function Description
WebDMZ/WebTrust			
Server 1	Web DMZ	Load Balancer	Provide Failover and load balancing feature for multiple application nodes through a Virtual IP.
Server 2	Web Trusted	WEB(OHS)	Handle external User HTTP traffic.
Server 3	Web Trusted	FTP Server	Support send and receive files from and to Data sources.
Application - ESB			
Server 4	ESB	OSB	Provide proxy and virtualization to the ESB Services.
Server 5	ESB	BAM	Monitor and Report Business SLAs.
Server 6	ESB	ODI	Transform data to different formats.
Server 7	ESB	SOA/BPM	Provide SOAP/Rest based web services along with workflow.
Server 8	ESB	MFT	Provides an Interface for File Transfers between different work streams.
Server 9	ESB	EMCC	Enterprise Monitoring and Operations tool.
Server 10	ESB	SOA Admin	WebLogic admin server that manages SO/BPM processes.
Server 11	ESB	OSB Admin	WebLogic admin server that manages OSB processes.
Server 12	ESB	ODI Admin	WebLogic admin server that manages ODI processes.
Server 13	ESB	MFT Admin	WebLogic admin server that manages MFT processes
Application - IdAM			
Server 14	IdAM	OAM	Provide User Authentication and Authorization Services.
Server 15	IdAM	OUD	LDAP for Individuals.
Server 16	IdAM	OIM	User Self Service and role management services.
Server 17	IdAM	OIM Admin	Administration server managing the IdAM servers.
Server 18	IdAM	OAM Admin	Administration server managing the IdAM servers.

Server Name	Zone	Server Tag	Function Description
Application - Shared Services			
Server 19	Shared Services	EDM	Centralized Enterprise Document Management system for archiving electronic documents, and to merge all documents under one record when client merges occur.
Server 20	Shared Services	CCM	CCM server provides notifications to internal and external consumers, providers, payers, and stakeholders.
Server 21	Shared Services	Address Std.	Address Standardization service for changing addresses to adhere to USPS standards.
Server 22	Shared Services	BRE-Oracle	Business Process Rules Engine.
Application - SMR			
Server 23	SMR	Access Server	Connect and extract the data from Legacy data sources.
Server 24	SMR	Ingest Server	Bulk data file ingestion to MarkLogic.
Database			
Server 25	Database	ESB/IdAM DB	Store and process custom and runtime data.
Server 26	Database	EDM DB	Store and process custom and runtime data.
Server 27	Database	CCM DB	Store and process custom and runtime data.
Server 28	Database	MarkLogic DB	NoSQL database used for SMR and MDM.
Storage			
Server 29	Storage	NFS	Provide Shared file system for Highly available components.
Management			
Server 30	MGMT-NW	Jump Server	Jump server access required for any access into the management network.
Management AD			

Server Name	Zone	Server Tag	Function Description
	MGMT-AD		
Management Audit			
Server 31	MGMT-Audit	Log Server	Capture logs of ingress and egress traffic between various segments.

4.3 Software Architecture

The software architecture of the SI Platform includes the Enterprise Service Bus (ESB) for loosely coupled and reusable, vendor-, product-, and technology-agnostic, easily discoverable and interoperable services integration with rest of the modules of HHS 2020 enterprise solution and external systems and the Enterprise Shared Systems by imposing security and integrity. It also encompasses the components responsible for migrating data from the legacy systems to the enterprise data repository (SMR) and managing the Master Data (MDM). The SI Platform ESB architecture is consistent with NM HSD enterprise architecture and includes best practices regarding an efficient and sustainable approach to implementing the HHS 2020 vision.

The SI Platform leverages many of the COTS products like Oracle Fusion Middleware as the ESB/SOA platform solution and MarkLogic as the SMR and the Master Data Management (MDM) solution. Oracle Fusion suite of products meets the NM HSD Medicaid enterprise need for a highly flexible, scalable SOA framework that can loosely couple disparate systems and applications into enterprise services. MarkLogic provides a NoSQL database into which large volumes of data of any schema can be ingested, normalized, de-duplicated, harmonized, and mastered into a single source of truth, which can be accessed by stakeholders for data migration, data cleansing, and analytics. The combination of these technologies facilitates the system integration, collection, and management of data NM HSD Medicaid operations.

The SI Platform is architected to achieve (a) the data integration, and (b) the service integration. The data integration is achieved in the SMR and the MDM solutions.

The SMR will have the following logical components:

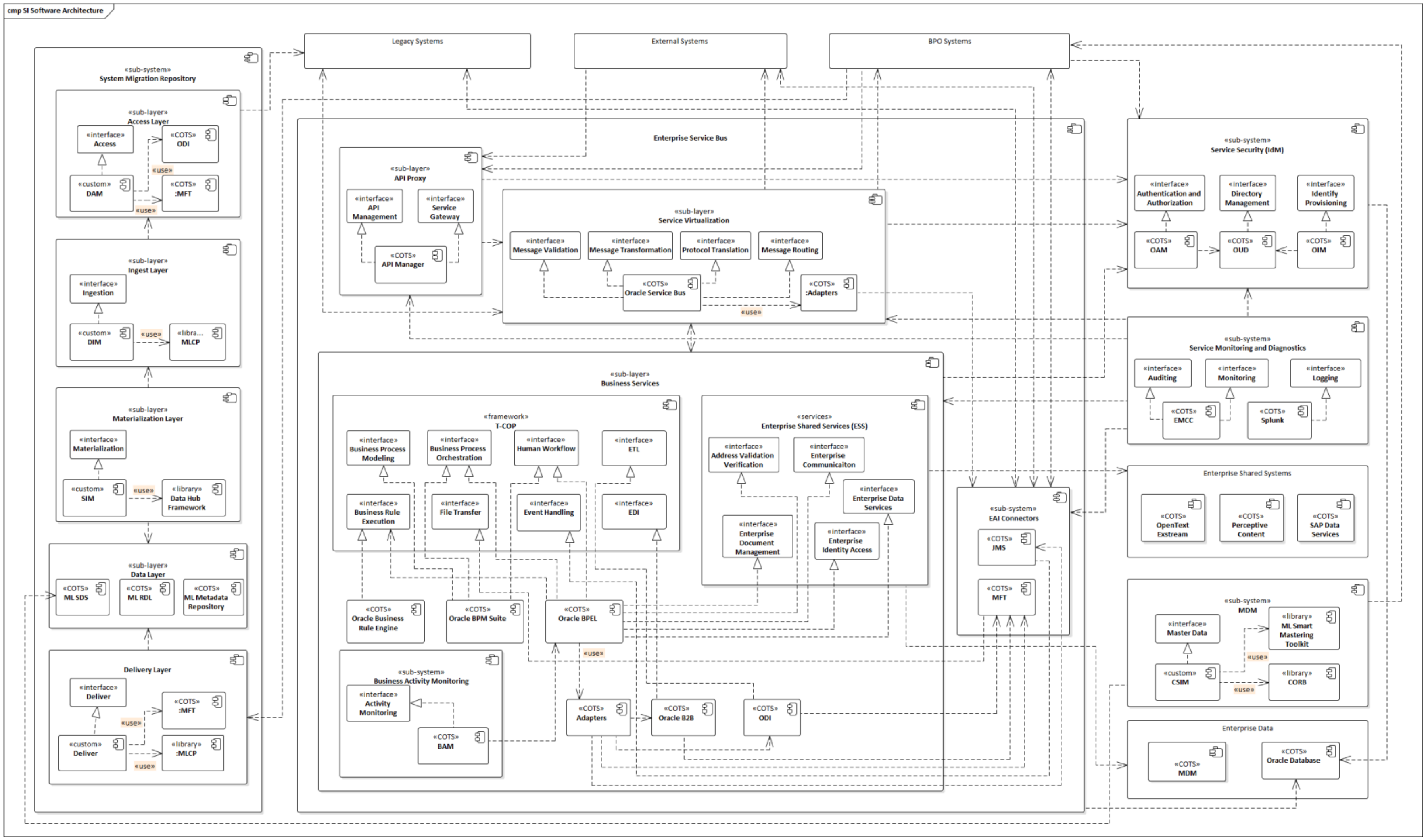
- The SMR core framework that is a repeatable, fault-tolerant, high-available, NoSQL database-based data standardization, and transformation process.
- The Raw Data Lake (RDL) that stores the legacy system data in their raw formats.
- The Standardized Data Store (SDS), which stores the enterprise MMISR data, standardized to the common data models.
- The Metadata Repository, where the metadata from all of the data sources and the SMR is stored, cataloged, and version controlled.
- The standardized reference code set repository, where the enterprise code sets are defined, and enhanced with the industry Medicaid.

The MDM solution provides an integrated view of various data elements of the enterprise. The major MDM data elements are Client and Organization. The MDM solution also provides the data as a service, for any integrating system to make use of the integrated data that is shared across the enterprise. The service integration is achieved by the ESB framework, which incorporates various logical components, as explained in Subsection 6.2. Major components of the ESB framework are:

- ESB Core framework component that implements exception and error management, and audit logging.
- Message routing
- Service versioning
- Message transformation
- Protocol transformation
- Event handling
- Rules engine integration
- Message validation
- Adapters and Connectors
- API Discovery
- API creation and publication
- API subscription
- Service monitoring

The figure below shows the high-level architectural building blocks (components) of the SI Platform's architecture.

Figure 4-8: SI Software Architecture



Note: The interfaces like Message Validation, Message Transformation, Human Workflow, Authentication & Authorization, etc., are depicted in the figure above are for representational purposes only. These are the capabilities of the tools/components, which are internal and will not be exposed as services to other components.

Table 4-17: **Architectural Components – Enterprise Service Bus**

Name	Enterprise Service Bus
Description	The Enterprise Service Bus (ESB) exposes and coordinates the functionalities of the loosely coupled modules as enterprise services, and performs functions, such as routing messages, transforming message formats, transforming message structures, and translating transport protocols. SI Contractor will leverage Oracle Fusion suite of tools and implement a standards-based service integration platform for HHS 2020 enterprise solution.
Components	• API Proxy • Service Virtualization • Business Services ○ T-Cop ○ Enterprise Shared Services ○ Business Activity Monitoring
Platform Tools	• Oracle Service Bus • Oracle BPEL • Oracle BPM Suite • Oracle Business Rules • Oracle Business Activity Monitoring (BAM) • Oracle B2B • Oracle MFT • Oracle Data Integrator (ODI) • Oracle Database • JMS Queues/Topics
Capabilities	• Validation • Transformations • Protocol Translations • Technology for module integration • Web service Security • Business Process Orchestration • Business Process Management • Business Rules Validation • Web services (SOAP/Rest) • Business Activity Monitoring

Table 4-18: SI Architectural Components – Enterprise Shared Services

Name	Enterprise Shared Services
Description	The purpose of Enterprise shared services is to create enterprise-wide services around the functionality provided by the shared services systems like address standardization, document management, communications management, and master data management. The Enterprise shared service are set of APIs through the ESB that can be consumed by other HHS 2020 enterprise modules in a consistent, centralized, and mediated fashion.
Components	EDAS – Enterprise Data as Service is a composite service that enables access to the MDM as well as all other data services EDS – Enterprise Documentation Service is a composite service that enables access to Hyland-Perceptive Content (formerly ImageNow) as, the Enterprise Shared System suite. ECS – Enterprise Communication Service is a composite service that enables access to OpenText Exstream, Enterprise Shared System suite. EIAS – Enterprise Identity and Access Service is a composite service that enables access to Service Security sub-system (Oracle IdM). EAVS – Enterprise Address Validation Service is a composite service that enables access to SAP Data Services for address standardization and validation shared service.
Platform Tools	• Oracle BPEL • BPM Suite • Oracle Business Rules • Oracle Database
Capabilities	• ESB composite services that enable access to Enterprise shared systems.

Table 4-19: SI Architectural Components – Service Security (Oracle IdM)

Name	Service Security (IdAM)
Description	Service Security is the enterprise Identity and Access Management (IdAM) framework that uses the Oracle IdM suite of products. This includes the technology to support the identity management and business rules to ensure appropriate constituent access to resources, across the HHS 2020 enterprise modules. It serves as a centralized security solution for providing authentication, coarse-grain authorization, and session management of BPO systems. All HHS 2020 enterprise modules that participate in the ESB platform for integration through Web Services require both authentication and verification of the identity and authorization, and determination of permission to access data or services provided by IdAM Service Security framework.
Platform Tools	• Oracle Access Manager (OAM) • Oracle Identity Manager (OIM) • Oracle HTTP Server (OHS) • Oracle WebGate • Oracle Unified Directory Server (OUD)
Capabilities	User Authentication, User Authorization, Identity Store & Management, User Lifecycle Management.

Table 4-20: SI Architectural Components – Service Monitoring and Diagnostics

Name	Service Monitoring and Diagnostics
Description	Service Monitoring and Diagnostics component of the SI Platform is monitoring for the whole enterprise. It provides logging, auditing, monitoring and reports services for all the endpoints and workflows published on the SI Platform. This component is implemented using Oracle BAM, Oracle EMCC, Oracle B2B Console, Oracle Enterprise Manager, and Splunk.
Platform Tools	• Oracle Enterprise Manager • Oracle Enterprise Cloud Control • Oracle B2B Console • Splunk
Capabilities	Logging, Auditing, Monitoring, and Reports.

Table 4-21: SI Architectural Components – SMR

Name	System Migration Repository (SMR)
Description	The SMR is responsible for enabling data migration from source systems into new modules through SI Platform. It acts as the source system for the initial load of enterprise data into MDM stores. It helps in standardizing and unifying disparate source data models into a canonical form and helps in the creation of a common message model across all HHS 2020 module interactions. The SMR will be built on MarkLogic NoSQL Database.
Components	DAM – Data Access Module DIM – Data Ingestion Module SIM – Source (specific) Integration Module Deliver – Delivers the data to modules through a suite of APIs and Services
Platform Tools	• MarkLogic NoSQL database • ML CORB • ML Data Movement SDK • ML Content Pump • ML REST APIs
Capabilities	Data Integration and Data Modeling.

Table 4-22: SI Architectural Components – MDM

Name	Master Data Management (MDM)
Description	Single source of truth for identified data domains or MDM entities across participating source data providers. Uses MarkLogic based smart mastering toolkit, algorithms, and APIs to match, merge and de-duplicate same data from different sources. It helps in Data cleansing activities across the enterprise.
Components	Ingestor – Helps MDM stores ingest from sources in a bulk initial way as well as in incremental fashion. CSIM – Cross-Source Integration Module - Helps match the same data from different sources into one entity based on scoring and fuzzy logic. Deliver – Delivers the data to modules through a suite of APIs and Services.
Platform Tools	• MarkLogic NoSQL database • MarkLogic Smart Mastering toolkit • MarkLogic CORB

Name	Master Data Management (MDM)
	• MarkLogic Data Movement SDK • MarkLogic Content Pump • MarkLogic REST APIs
Capabilities	Master Data Management.

The complete list of software, tools, and libraries necessary to build SI Platform components can be found in Appendix G.

4.3.1 Security Software Architecture

The SI Platform uses an in-depth defense strategy to ensure that the confidentiality, integrity, and availability of the platform is secured and maintained. It is comprised of six functional segments that include the infrastructure, operating system, data, services, and access. The security architecture is comprised of software and hardware technologies in each of these functional areas that implement the required security safeguards per the MARS-E v2.0 and best practices.

Infrastructure security or platform-level security is achieved by using the Dell VX Rack hardware with VMWare virtualization technology that separate the user functionalities from the security configurations. This virtual environment is established in a secure datacenter that has no direct public access. The VPN that is configured into this network is also configured on a 2-way authenticated system configuration to control the access into these servers.

At the operating system level, the vSphere is the compute component of the VxRack that provides the virtual machines. Each of these machines will carry a gold image of the Red Hat operating system. This image will be patched and configured to meet the United States Government Configuration Baseline Standards (USGCB). The Red Hat operating system has an Evaluation Assurance Level of 4+ for the operating system protection profile; thus, it meets most security requirements such as process isolation.

The SI Platform ensures that all interfaces are encrypted to protect all data being exchanged with external and internal entities. Communication and messages exchanged, whether with other system modules or system users, are encrypted via SSL. SSL uses a checksum to protect the integrity of the data in transit. The data security is achieved by leveraging the appropriate MarkLogic and Oracle functionalities that support data encryption at rest. For data at transit, the SI Platform utilizes options available at various Oracle Fusion Middleware and SOA components, at each of the transport and service layers.

An API proxy server surrounded by two stateful firewalls, on either side, creating a Demilitarized Zone or DMZ, segments the access area. API Proxy layer acts as the first line of defense in the DMZ layer by adding comprehensive enterprise-grade security, including transport-level security, message-level security, Security Assertion Markup Language (SAML), and identification, and authentication.

The security software components and their architecture are described in Subsection 4.6

4.3.2 Performance Software Architecture

The SI Platform solution is designed for high availability to meet the challenges of a large user and transaction base anticipated in the MMISR solution with real-time response, but ongoing performance monitoring enables us to maintain a well-tuned system.

The SI Platform is installed and configured on a VMWare infrastructure that enables easy server scale-out. The SI Platform's tiered architecture also allows for adding more hardware resources and increasing the processing capacity of one tier without impacting another tier. This ability contributes to making the application highly scalable. Application components of the SI Platform are built as multi-node clusters to enable seamless session failover. The SI Platform leverages the capabilities of the WebLogic Automatic Service Migration that is configured for the JMS Service. In addition, the SI Platform utilizes the scalability features of the Oracle and MarkLogic NoSQL databases to achieve its performance goals.

The performance software and infrastructure software components/tools are further explained in Subsection 6.4.

4.4 Information Architecture

This subsection describes the information that will be stored in the system (e.g., beneficiary information and claim data), identifying any of the information is Personally Identifiable Information (PII), Individually Identifiable Information (IIF), or Personal Health Information (PHI).

The SI Platform does not own the transactional data. It, however, maintains the request/response and audit information of the data that passes through it.

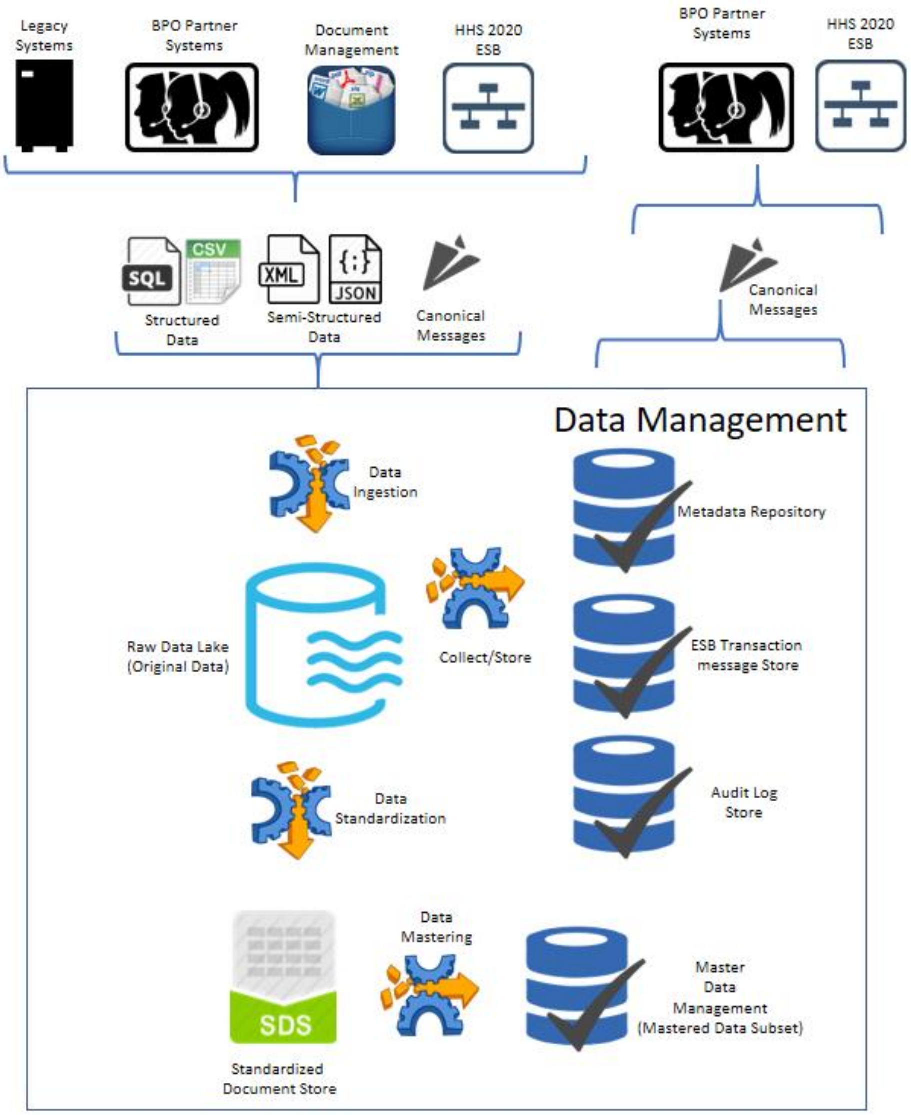
The data that is stored by the SI Platform is the enterprise MMISR data that it stores in the SMR subsystem. It will also store the mastered enterprise MDM data.

The SI Platform will follow the HHS 2020 enterprise data architecture described in the NM HSD Concept of Operations (ConOps) document.

The following are the data stores of the SI Platform:

- Raw Data Lake (RDL)
- Standardized Document Store (SDS)
- Master Data Management (MDM)
- Metadata Repository
- ESB transaction Message store
- Audit log store

Figure 4-9: HHS 2020 Enterprise Data Architecture



The following subsections describe the nature of the data that is stored in each of the above-mentioned data stores.

4.4.1 Data

The SI Platform receives its data from the following data sources:

- NM HSD Legacy Systems
- HHS 2020 BPO Modules

The NM HSD Legacy systems currently own the enterprise MMIS and all the related data. Following are the legacy systems identified so far:

- Omnicaid – The MMIS system that maintains the current and historical information of claim, client, provider, prior authorization, care providers, pharmacy, and drug-related subsystems.
- Automated System Program and Eligibility Network (ASPEN) – It mainly contains client eligibility and client and provider enrollment related subsystems.
- Child Support Enforcement System (CSES) – CSES maintains the child support enforcement-related information. It also has client and employer details.
- Department of Health (DOH) – <TBD>
- Behavioral Health Enforcement Division (BHSD) – <TBD>
- Aging and Long-Term Services Department (ALTYSD) – <TBD>

Following are the HHS 2020 BPO Modules that will provide data to the SI Platform post-production:

- Data Services (DS)
- Benefits Management Services (BMS)
- Financial Services (FS)
- Quality Assurance Services (QA)

HHS 2020 ESB Subsystem is a part of the SI Platform and receives data from the HHS 2020 messaging systems. It produces the audit, request, response, and transaction data.

The data procured from the sources mentioned above and systems will contain the information that is classified and categorized as follows,

The data attribute will be tiered for its sensitivity as:

- **High** – High sensitivity categorization refers to the data affecting the organization seriously. Information released to the public or unauthorized persons could cause grave damage, loss of life, or major monetary damage, and require legal action.
- **Medium** – This information, when released to the public or unauthorized persons, could cause significant embarrassment and/or damage in money, property, or personnel to the organization or require legal action.
- **Low** – This information, when released to the public or unauthorized persons, could cause minor embarrassment and/or damage and only require administrative action for correction.
- **None** – This information is already available to the public or availability to the public will not cause damage.

The data attribute will be marked for its criticality as:

- **Essential** – This refers to the data that is considered critical to the business. When essential data is not available, or its integrity is questionable, the business will not be able to function. The impact may be a loss of revenue and potential liability.
- **Required** – This refers to the data that is important to the business, but operations would continue if the data were not available for a pre-determined period.

- **Deferrable** – This refers to the data in the absence of which operations would continue for extended periods.

The data attribute will be grouped as one of the following information types:

- **Protected Health Information (PHI)** – PHI refers to any health information that is individually identifiable and created or received by a provider of health care, a health plan operator, or health-clearing house.
- **Personally Identifiable Information (PII)** – Any information about an individual maintained by an agency, including (1) any information that can be used to distinguish or trace an individual's identity, such as name, social security number, date and place of birth, mother's maiden name, or biometric records; and (2) any other information that is linked or linkable to an individual, such as medical, educational, financial, and employment information. Note that SSA data is considered PII.
- **Federal Tax Information (FTI)** – FTI consists of federal tax returns and returns information (and information derived from it) that is in the agency's possession or control subject to Internal Revenue Code (IRC) 6103(p)(4) safeguarding requirements including Internal Revenue Service (IRS) oversight. FTI is categorized as Sensitive but Unclassified information and may contain PII. FTI includes any information created by the recipient that is derived from federal return or return information received from the IRS or obtained through a secondary source. FTI may not be masked to change the character of information to circumvent IRC 6103 confidentiality requirements.
- **Payment Card Industry (PCI)** – PCI is comprised of payments and related fees provided via Interactive Voice Response (IVR), or via the Internet by using an individual's debit and/or credit card.
- **Federal Parent Locator Service (FPLS)** – FPL information consists of the National Directory of New Hires (NDNH) data, Debtor File data, and the Federal Case Registry (FCR) data. These are components of an automated national information system, which locates employment, income, asset, and home address information on parents in child support cases for State CS agencies. The Debtor File contains PII including names, SSN, and other private data.
- **Substance Use Disorder Patient Records** – A federal statute called Confidentiality of Alcohol and Drug Abuse Patient Records regulates confidentiality for people seeking treatment for substance use disorders from federally assisted programs. This law requires that federally assisted substance use programs have a patient's consent before releasing information to others. Therefore, this confidential data must be classified separately.
- **Primary Key** – The data that uniquely recognizes a record and is source-specific in nature. It is not important for the business, nor grouped in any of the above categories.
- **Uncategorized** – The data that is not categorized as any of the above.

The attribute will be characterized as:

- Confidential
- Internal
- Public

This categorization follows the instructions provided in the following table.

Table 4-23: HHS 2020 Data Classification

HHS 2020 Data Classification	Data characterization (Public, Internal, or Confidential)
Business data containing FTI, ePHI, PII, and FPLS.	Confidential
Business data that does not contain FTI, ePHI, or PII, but should not be public.	Internal
Business data that does not contain FTI, ePHI, or PII, and could be public.	Public
Content on the HHS 2020 Enterprise Public-Facing Website.	Public
Administrative data for system access (passwords, etc.).	Confidential
Technical data, system security architecture, and settings.	Confidential
Admin data, not security related (org charts, cell phone numbers, etc.).	Internal

4.4.2 Manual/Electronic Inputs

This subsection identifies all of the input patterns, including the electronic, or file-based accesses, and/or any manual input points for the SI Platform.

The following are the access patterns of SMR:

- File-Based Access (Batch) – To access the file-based data sources.

This access pattern is used to access the legacy source Omnicaid, which provides data-extracts as delimited Comma Separated Values (CSV) files:

- ETL Based Access (Batch) – To support the RDBMS data sources.

Extract, Transform, Load (ETL)/ Extract, Load, and Transform (ELT) access patterns will be used to access the legacy sources of ASPEN and CSES. Both of these legacy sources provide ODBC/JDBC access to their databases. Oracle Data Integrator (ODI), an ELT tool, will be used to directly connect to these databases and extract the required information:

- SOAP/– To support the Web-API enabled integrating systems. XML is default message format for SOAP and REST based web services.

These access patterns will be used to access the real-time transactional data procured by the HHS 2020 ESB system, which transfers data produced by the HHS 2020 BPO modules.

Apart from storing the data in a NoSQL MarkLogic database, and/RDBMS Oracle databases, the SI Platform also receives and manages the inactive, delimited CSV files obtained via the Batch and ETL means.

Detailed information on the design and implementation of these access patterns is provided in Subsection 5.1.3.1.

4.4.3 Records Management

This subsection describes the nature and management of records in the SI Platform. It deals with the structured storage repositories like System Migration Repository and the MDM, and unstructured records like log files and other system-generated files.

4.4.3.1 State Regulations

The following table identifies various records that may require retention by HSD. They are direct references from the New Mexico Administrative Code (NMAC) and identify the record, description, and period for retention. These cover project contracts, files, security, and other possible records that may apply to MMISR. The "As Needed" notation indicates multiple retention requirements listed in the code. This list supersedes the previous schedule for NM HSD, which was repealed in 2015.

Table 4-24: State Regulations

Item	Type	Retention Period in years
1	**1.21.2.110 Logs** 1. Category: Administration – general management. 2. Description: Logs used to monitor or control. 3. Retention: Retain until no longer needed for reference. [1.21.2.110 NMAC - N, 10/01/2015]	As needed
2	**1.21.2.101 Authorization** 1. Category: Administration – general management. 2. Description: Records related to authorization of personnel or entities to perform specific duties and not identified in other classifications. 3. Retention: Destroy ten years from date file closed. [1.21.2.101 NMAC - N, 10/01/2015]	10
3	**1.21.2.156 Access and Control** 1. Category: Administration – information technology 2. Description: Records related to security and access to information technology. 3. Retention: Destroy three years from date file closed. [1.21.2.156 NMAC - N, 10/01/2015]	3

Item	Type	Retention Period in years
4	**1.21.2.157 Systems and Networks** 1. Category: Administration – information technology. 2. Description: Records related to development and maintenance of voice and data networks, infrastructure, and computer applications. 3. Retention: Destroy when superseded or obsolete. [1.21.2.157 NMAC - N, 10/01/2015]	As needed
5	**1.21.2.413 Federal Compliance and Reporting** 1. Category: Governance and compliance – audit, oversight, and compliance. 2. Description: Records related to oversight and federal compliance reporting. 3. Retention: Destroy three years from date file closed. [1.21.2.413 NMAC - N, 10/01/2015]	3
6	**1.21.2.834 Programs – Medical and Hospital** 1. Category: Public health and social services – hospital and medical. 2. Description: Records related to clinical and health programs. 3. Retention: Destroy five years from date file closed. [1.21.2.834 NMAC - N, 10/01/2015]	5

Table 4-25: Federal Regulations

Item	Type	Retention Period in Years
1	HIPAA – Administrative Simplification rules require a covered entity, such as physician billing Medicare.	Six years from the date of its creation or the date when it last was in effect, whichever is later.
2	Providers submitting cost reports to be retained in their original or legally reproduced from.	A minimum five (5) years after the closure of the cost report.
3	Medicare managed care program providers retain patient records.	10 years

Item	Type	Retention Period in Years
4	Electronic Code of Federal Regulations (ECFR) – Title 36 Parks, Forests, and Public Property – 1200-1299 - National Archives and Records Administration.	

4.4.4 Log Aggregation and Monitoring

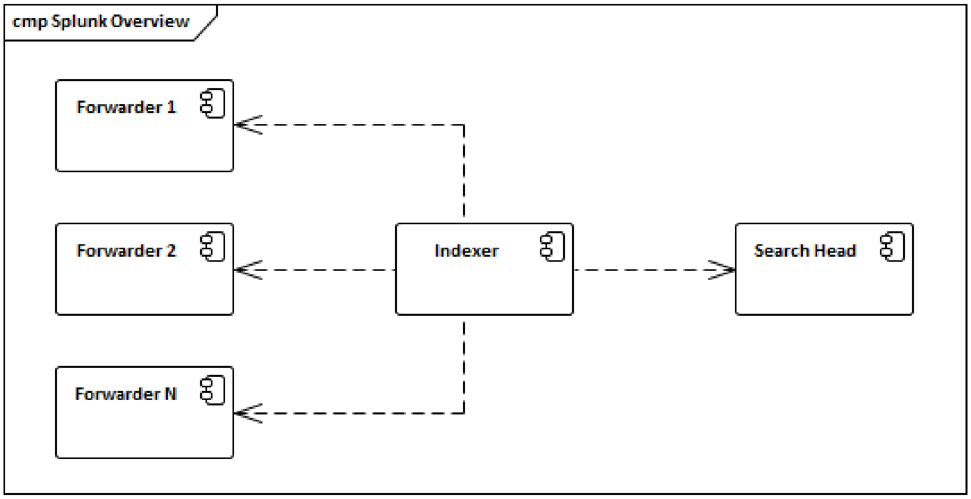
In the SI Platform, Splunk will be used as a log aggregation and Security Information and Event Management (SIEM). The log feeds from the Splunk Forwarders installed in each component will be aggregated for monitoring and analysis purposes.

The following are the capabilities of the Splunk as an enterprise SIEM tool.

- **Real-Time Visibility**
    - Live dashboards
    - Event correlation
    - Monitoring and alerting
    - Transactional insights
    - Identify performance issues
    - SLA tracking
- **Search/Navigate**
    - Data drill-down
    - String Search - Needle in a haystack
    - Root cause analysis
    - Troubleshooting
    - Incident investigations
- **Historical Analytics:**
    - Baseline and thresholds analysis
    - Trending
    - Operational insights
    - Historical patterns
    - Compliance reports

The following are the high-level components of Splunk.

Figure 4-10: Splunk Overview



Forwarder – Splunk forwarder is a component, which works as an agent for Splunk to feed the data to the Splunk Indexer. It collects the log data from different sources like ESB, SMR, IdAM, etc., and forwards collected data to the indexer for indexing. The SI Platform will use the Splunk's Universal Forwarder to feed the raw log events as is to the Indexer.

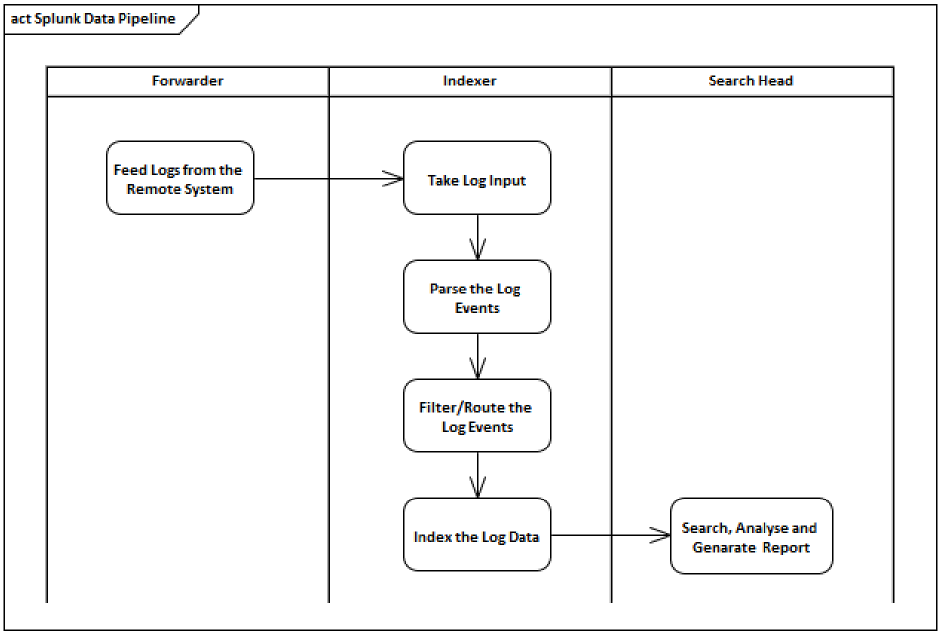
Indexer – The Indexer is the Splunk Enterprise component that creates and manages indexes. The primary functions of an Indexer include:

- Indexing incoming data.
- Searching the indexed data.

Search Head – Splunk search head is a GUI for Splunk where we can search, analyze, and report.

The following are the activities of data pipeline of Splunk.

Figure 4-11: Splunk Data Pipeline



4.4.5 Master Files

Data obtained from the data sources are ingested into the RDL subsystem, which is built on MarkLogic NoSQL database. The RDL stores data in Extensible Markup Language (EML) format, but in the native data model.

Data from RDL is then standardized into the HHS 2020 enterprise common data model, which is canonical in nature and stored in the SDS subsystem. The SDS is built on the MarkLogic NoSQL database, and stores data as XML documents.

The MDM subsystem utilizes the standardized data to master the information across the sources. The MDM will contain a subset of the information available in the SDS.

4.5 Internal Communications Architecture

This subsection describes the SI Platform's communications network, denoting the communications architecture(s) being implemented.

4.5.1 Palo Alto Firewall Architecture

The SI Platform for the MMISR project at NM HSD will be connected to the existing Palo Alto firewall at the datacenter. The figure below shows its connectivity and how the traffic flows, in and out of the VxRack infrastructure, and into the internet.

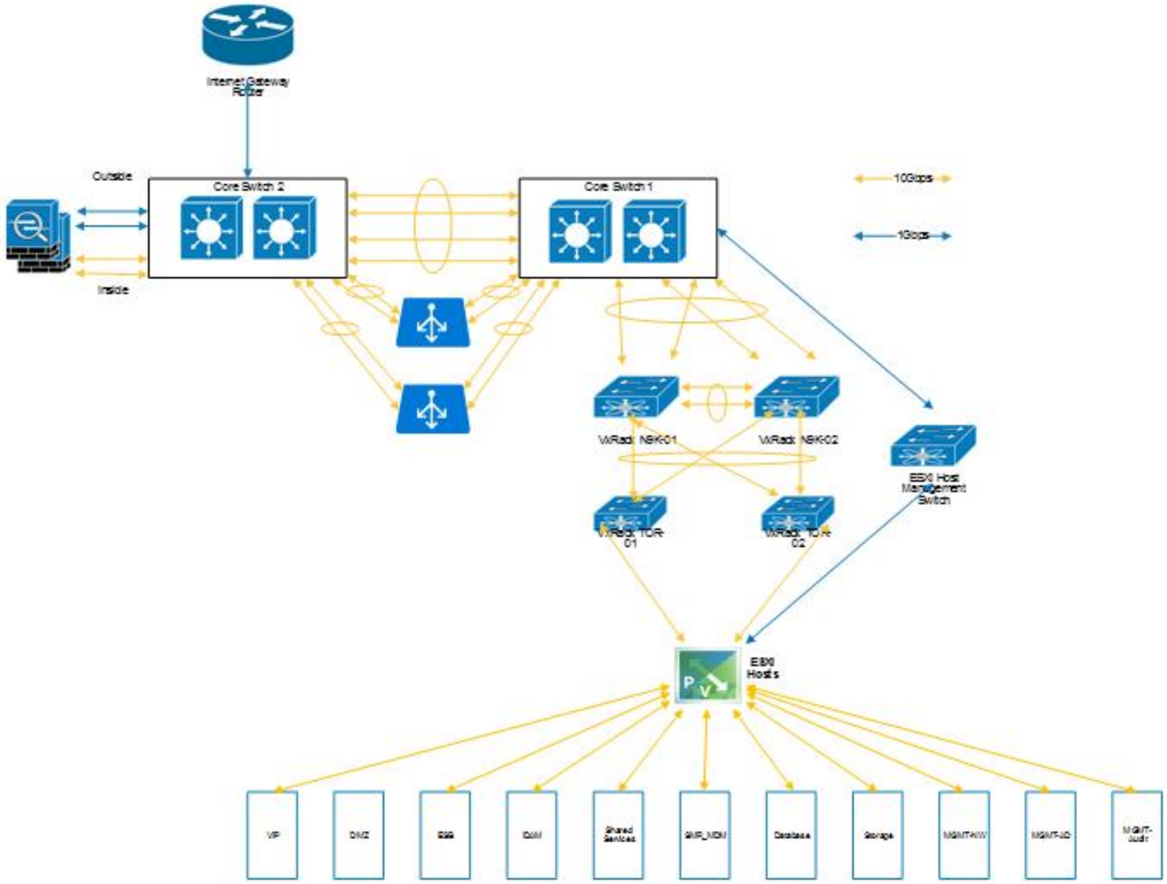
For the SI Platform virtual infrastructure, applications are segregated by VLANs. Port groups are created for each of the VLAN networks that house these systems. These port groups use the physical network card configured inside the Distributed switch for traffic that flows in and out of the VxRack.

VxRack comes built with Top of Rack, Management Switch, and Aggregated switches as network components.

The VxRack top of Rack switches handles traffic to flow within the rack (East-West) for systems built in the same subnet.

Traffic across the subnets will flow through the Top of Rack switches and traverse through the Aggregation switches and core network switches before being filtered at the Palo Alto firewall. Therefore, a packet leaves a given system and passes through the top of rack switches, aggregation switches, and to the core switches, it gets filtered and monitored by the firewall and directed to a different network from the original server. It again passes through the core switches, aggregation switches, and Top of Rack switches before ending up on the server the packet is destined for.

Figure 4-12: Zone Layout for Palo Alto Firewall



4.6 Security Architecture

The SI Platform uses an in-depth defense strategy to ensure that the confidentiality, integrity, and availability of the platform is secured and maintained. The SI Platform is comprised of six functional segments that include the infrastructure, operating system/network, platform, data, services, and access. The security architecture is comprised of software and hardware technologies in each of these functional areas that implement the required security safeguards per the MARS-E v2.0 and best practices. The security measures for each of these areas are described below.

4.6.1 Infrastructure/Network

The Palo Alto firewall will provide protection for external (North-South) traffic, and in addition, provide protection for internal (East-West) traffic across subnets. The Palo Alto is a next-generation firewall that provides traditional stateful firewall capabilities along with deep packet inspection (DPI). DPI will check for malicious code and drop any suspicious traffic. Additionally, the Palo Alto firewall has a built-in Intrusion Protection System that will be used to monitor the network, detect known virus signatures, and drop those packets. Virtual Private Network (VPN) will be deployed to enhance confidentiality and integrity over remote connections. The VPN connection will leverage Internet Protocol Security and Internet Key Exchange version 2 (IKE v2) with the American Encryption Standard (AES) 256 for remote connections.

The F5 load balancer will also be utilized in an HA mode which will ensure the web traffic is managed between the different components of the application. F5 also has built-in security features, which can detect security attacks related DDOS.

4.6.2 Operating System

The vSphere consists of the ESXi cluster hosts that support server hardware virtualization and virtual machines (VM). A gold image of the Red Hat Enterprise Linux (RHEL) v7.5 operating system will be created and configured to meet the United States Government Configuration Baseline Standards (USGCB). The Red Hat operating system has an Evaluation Assurance Level of 4+ for the operating system protection profile; thus, it meets most security requirements such as process isolation.

4.6.3 Platform

The SI Platform will have an Oracle Fusion Middleware (OFMW) based Enterprise Service Bus (ESB) and Service Oriented Architecture (SOA). The ESB is a message-oriented middleware (MOM) solution that will act as the backbone of SI Platform, enabling modules of disparate functionalities and technology platforms to communicate with each other. The ESB will provide and consume services between modules via one of three connection methods:

- Web services (via Hypertext Transfer Protocol Secure (HTTPS)-port 8443).
- Database adaptor (via Open Database Connectivity (ODBC)-port 1433 and Java Database Connectivity (JDBC)-port 1433).
- Managed File Transfer (MFT) (via File Transfer Protocol/Secure (FTP/S)-port 990 or 21)
- Extract Transform (ETL).

Communication between any two modules in MMISR will occur through Web Services or database connections, via Application Programming Interface (API) deployed on ESB. It can also occur via file transfer or ETL. The Service bus utilizes the Oracle Web Services Manager (OWSM). The OWSM is a component of the OFMW and will be configured to manage security policies and propagate identities across web services.

OWSM policies will require that the systems that interact with ESB have proper authentication and can only access the authorized resources. It ensures that the messages entering or leaving the ESB platform are encrypted to prevent data leakages and maintain the integrity of the messages. Data traversing between nodes within ESB will be encrypted via Transport Layer Security (TLS) v1.2.

4.6.4 Data

The SI Platform ensures all interfaces are encrypted to protect all data being exchanged with external and internal entities. Communication and messages exchanged, whether with other system modules or system users, are encrypted via SSL. SSL uses a checksum to protect the integrity of the data in transit.

Sensitive data stored in the database is secured using Oracle Advanced Security feature called Transparent Data Encryption (TDE). TDE offers the ability to encrypt at the file level. TDE performs real-time input/output so that encryption and decryption of the file appear seamless. The encryption uses a symmetric key called the database encryption key or (DEK) and employs AES encryption mechanisms.

4.6.5 Services

The SI Platform will use the following mechanisms to secure the web services deployed on the servers:

- Transport Layer Security
- Server Authentication
- User Authentication
- Transport Encoding
- Authorization

4.6.5.1 Transport Layer Security

Transport Layer Security is achieved by encryption using TLS 1.2 protocol. Transport Layer Encryption provides numerous benefits beyond traffic confidentiality, including integrity protection, replay defenses, and server authentication. ESB uses SOAP over HTTPS protocol for successful web services communication.

4.6.5.2 Server Authentication

TLS 1.2 and above is used to authenticate the service provider to the service consumer. It verifies the server certificate that is issued by a trusted provider, is not expired, is not revoked, and matches the domain name of the service. SSL certificates from a trusted provider will be used.

4.6.5.3 User Authentication

User authentication verifies the identity of the user or the system trying to connect to the service. This authentication is usually a function of the container of the web service and includes:

- Internal web services – Username/Password in SOAP header for SOAP web services, HTTP basic for REST web services are implemented to achieve user authentication. The authentication mechanism and credentials are integrated with IdAM framework.

- External web services – Client Certificate authentication is implemented to achieve user authentication.

4.6.5.4 Transport Encoding

The SOAP encoding styles transport data between software objects into XML format and back again. Hence, the same encoding style between the client and the server is enforced.

4.6.5.5 Authorization

Web services must authorize web service clients in the same way that web applications authorize users. A web service needs to ensure that a web service client is authorized to perform a certain action (coarse-grained); on the requested data (fine-grained). This is achieved by separating normal users from administrative users thereby access to administration and management functions within the web service application are limited to web service administrators.

4.6.6 Access

The access area is segmented by an API proxy server surrounded by two stateful firewalls, on either side, creating a Demilitarized Zone (DMZ). API Proxy layer acts as the first line of defense in the DMZ layer by adding comprehensive enterprise-grade security, including transport-level security, message-level security, SAML, and identification, and authentication.

Identity and Access Management will manage authentication into the MMISR network. The SI Platform serves Authentication, Multifactor Authentication, Federation, coarse-grained authorization, Provisioning, De-Provisioning, Delegated Administration, and Password Management services. Fine-grained authorization for protected resources is managed by the role-based access policies defined within each of the applications. Identities from outside users will be exported from the existing Active Directory in the ASPEN legacy system.

In MMISR, SAML will be used as the Federation protocol, which is an XML-based protocol for exchanging security information between disparate entities. Some of the existing HSD applications will be integrated into MMISR systems using SAML based Identity Federation.

4.6.6.1 Audit

Across the architecture of the SI Platform, audit trails will be logged and stored. The audit information will capture information that establishes what type of event occurred, when the event occurred, where the event occurred, the source of the event, the outcome, and the identity of any individuals or objects associated with the event. Specifically, the SI Platform will provide the following auditing capabilities:

- Configuration changes – The ESB platform provides an administrative console to view and access the history of configuration changes to the ESB application.
- User profile change and User access activity – The ESB platform keeps track of the users who log into the application and stores their session details, such as the message exchanges and services consumed.
- Message flows – The ESB platform creates persistent files of the messages that flow across the pipeline.

Storage for Oracle auditing is held temporarily on each source database and later pushed to Splunk. The Oracle Enterprise Manager monitors space on each source computer and will alert the DBA to shortages on tablespace allocations.

Audit trails also support the log/audit requirements of regulations such as the Health Insurance Portability and Accountability (HIPAA) §164.308(a)(1)(ii)(D):Security Management Process to implement policies and procedures to prevent, detect, contain, and correct security violations including implementing procedures to regularly review records of information system activity.

4.6.7 Performance

Performance of the SI Platform is designed based on the expected transaction throughput, average response times, error rates, and availability requirements. The SI Platform environments will be configured to scale easily to provide increased throughput and optimal performance during runtime. The SI Platform will be built out to support the transaction volumes projected during the RFP phase. More detailed service-level agreements (SLAs) for business transaction volumes, transactions response times, and availability metrics will be defined during the non-functional requirements phase.

An iterative performance testing will be performed during the development cycle of each module within the SI Platform, and both system and application components will be tuned to achieve the expected performance metrics. More details on performance monitoring and tuning are given in Subsection 6.4

5 System Design

This section outlines the system design considerations made to achieve the functional design goals of the SI Platform.

5.1 Database Design

Currently, NM HSD Medicaid data is exchanged across multiple systems in various formats. It is difficult and time consuming for agency staff to gather all related data from various data sources. The SMR will provide the current view of a consistent and consolidated enterprise data in a centralized repository. The following subsections provide more details regarding nature and data structures of the RDL and SDS.

The following are the architecturally significant requirements addressed in this subsection:

- A copy of the legacy system’s data in a complete raw form will be obtained.
- Complete raw legacy data set will be placed into a data lake repository.
- Convert the raw data into a canonical format, standardize reference values and enhance data quality through de-duplication.
- Place canonical, standardized, and de-duplicated data into the NoSQL database of the System Migration Repository.
- Make NoSQL database contents available for ingestion by new BPO Partner systems to enable their operations within HHS 2020 Enterprise post-go-live.
- Establish master data management for Client, Provider, Employer, and other data.

5.1.1 Data Objects and Resultant Data Structures

The SMR will have the following three data store components, with each having its own data objects and structures as explained in the following subsections.

- Raw Data Lake (RDL)
- Standardizes Document Store (SDS)
- Metadata Repository

An Operational Data Store (ODS) will be built on top of the SMR.

5.1.1.1 The RDL

Since the RDL stores the data as loaded, it follows the native data format from the legacy source system. The RDL supports the following data formats:

- Structured data from relational databases, including rows and columns.
- Semi-structured data, such as comma-separated values (CSV) logs, extensible Markup Language (XML), and JavaScript Object Notation (JSON).
- Binary data, such as Portable Document Format (PDF) files, images, audio, and video
- Document or Object-based NoSQL data.

For example, the native structure of the Omnicaid data is preserved along with the original integrity, type, length, and structural constraints.

When a row of tabular data is loaded into the MarkLogic, it is represented as an XML document.

5.1.1.2 The Standardized Document Store (SDS)

The SDS stores data in a standardized format that is defined by the Physical Data Models (PDM).

The figure below provides a very high-level view of how the data domains interact with each other at the conceptual level. This figure does not show the details of the domains. However, the details of each data domain along with the business entities and the data elements contained in those data domains are addressed in the following deliverables:

- SIDM1: Conceptual Data Model (CDM)
- SIDM2: Logical Data Model (LDM)
- SIDM3: Physical Data Model (PDM)
- SIDM4: Information Governance Catalog

An Enterprise Data Model (EDM) is an integrated view of the data produced and consumed across an entire organization. It follows the standard message exchange models and guidelines like NIEM and FIHR. An EDM defines single integrated data that is unbiased by any systems or applications. It is independent of how data is physically sourced, stored, processed, or accessed. The model unites, formalizes, and represents the information important to an organization, as well as the rules governing them.

A high-level definition of data domains and business entities is provided below:

- A data domain is a high-level functional area (or subsystem) of the given source system through which the source system coordinates with the related business workflows and Business Entities.
- Business entities are recognizable concepts – such as a person, places, things, or events – that represent the given Data Domain.

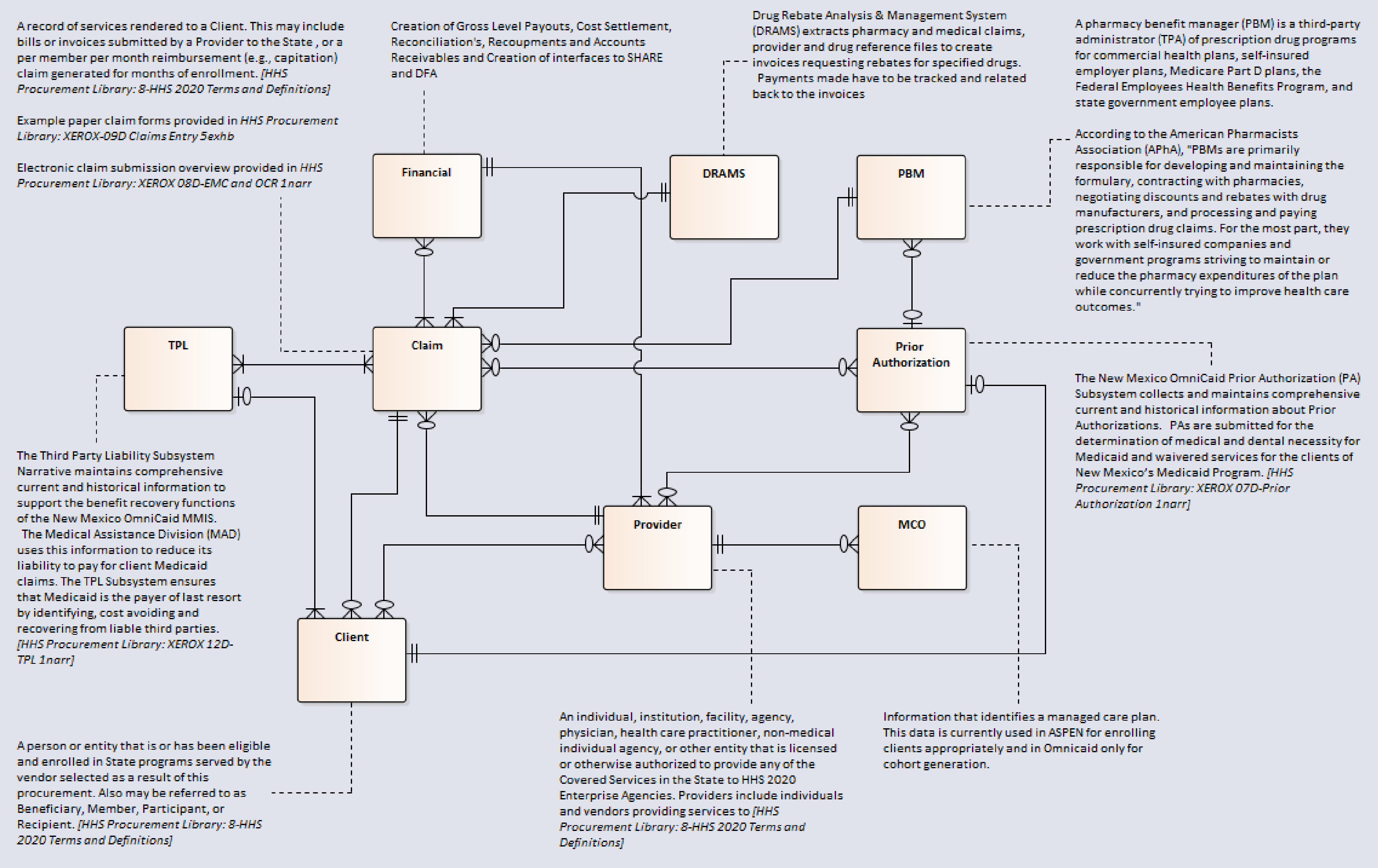
For example, “Provider” is a data domain and the Provider License or Provider Demographics are business entities.

The MMISR data is modeled into the following nine data domains:

1. **Client** – A Client is a person or entity that is or has been eligible and enrolled in the State's MMIS program.
2. **Claim** – A Claim is a bill for services submitted by a Provider to the State or depending on the claim type, a line item of service on a bill, or all services for one Member within a bill.
3. **Drug Rebate Analysis and Management System (DRAMS**) – DRAMS extracts pharmacy and medical claims, provider, and drug reference files to create invoices requesting rebates for specified drugs. Payments made have to be tracked and related back to the invoices.
4. **Financial** – Financial captures Gross Level Payouts, Cost Settlement, Reconciliations, Recoupments and Accounts Receivables and Creation of interfaces to SHARE and DFA.
5. **Managed Care Organizations (MCO)** – MCO shares member affiliations with PCPs, Health Homes, Disability status, Long Term Care status, Care Coordination and member assessments and reporting on MCO activities such as pilot projects, value-based purchasing agreements, etc.
6. **Pharmacy Benefits Management (PBM)** – PBM is a third-party administrator (TPA) of prescription drug programs for commercial health plans, self-insured employer plans, Medicare Part D plans, the Federal Employees Health Benefits Program, and state government employee plans. Payment based on what the PBM has adjudicated.
7. **Prior Authorization (PA)** – PA subsystem collects and maintains comprehensive current and historical information about Prior Authorizations. PAs are submitted for the determination of the medical and dental necessity for Medicaid and waivered services for the clients of New Mexico's Medicaid Program.
8. **Provider** – The provider is an individual, institution, facility, agency, physician, health care practitioner, non-medical individual agency, or other entity that is licensed or otherwise authorized to provide any of the Covered Services in the State to HHS 2020 Enterprise Agencies. Providers include individuals and vendors providing services under a Managed Care contract agreement to.
9. **Third Party Liability (TPL**) – The TPL subsystem maintains comprehensive current and historical information to support the benefit recovery functions of the New Mexico Omnicaid MMIS. The Medical Assistance Division (MAD) uses this information to reduce its liability to pay for client Medicaid claims. The TPL Subsystem ensures that Medicaid is the payer of last resort by identifying, cost avoiding and recovering from liable third parties.

The figure below provides a high-level Entity Relationship Diagram (ERD) across the above-listed data domains.

Figure 5-1: **High-Level Entity Relationship Diagram across Data Domains**



A record of services rendered to a Client. This may include bills or invoices submitted by a Provider to the State , or a per member per month reimbursement (e.g., capitation) claim generated for months of enrollment. *[HHS Procurement Library: 8-HHS 2020 Terms and Definitions]*

Example paper claim forms provided in *HHS Procurement Library: XEROX-09D Claims Entry 5exhb*

Electronic claim submission overview provided in *HHS Procurement Library: XEROX 08D-EMC and OCR 1narr*

Creation of Gross Level Payouts, Cost Settlement, Reconciliation's, Recoupments and Accounts Receivables and Creation of interfaces to SHARE and DFA

Drug Rebate Analysis & Management System (DRAMS) extracts pharmacy and medical claims, provider and drug reference files to create invoices requesting rebates for specified drugs. Payments made have to be tracked and related back to the invoices

A pharmacy benefit manager (PBM) is a third-party administrator (TPA) of prescription drug programs for commercial health plans, self-insured employer plans, Medicare Part D plans, the Federal Employees Health Benefits Program, and state government employee plans.

According to the American Pharmacists Association (APhA), "PBMs are primarily responsible for developing and maintaining the formulary, contracting with pharmacies, negotiating discounts and rebates with drug manufacturers, and processing and paying prescription drug claims. For the most part, they work with self-insured companies and government programs striving to maintain or reduce the pharmacy expenditures of the plan while concurrently trying to improve health care outcomes."

The New Mexico OmniCaid Prior Authorization (PA) Subsystem collects and maintains comprehensive current and historical information about Prior Authorizations. PAs are submitted for the determination of medical and dental necessity for Medicaid and waivered services for the clients of New Mexico's Medicaid Program. *[HHS Procurement Library: XEROX 07D-Prior Authorization 1narr]*

The Third Party Liability Subsystem Narrative maintains comprehensive current and historical information to support the benefit recovery functions of the New Mexico OmniCaid MMIS. The Medical Assistance Division (MAD) uses this information to reduce its liability to pay for client Medicaid claims. The TPL Subsystem ensures that Medicaid is the payer of last resort by identifying, cost avoiding and recovering from liable third parties. *[HHS Procurement Library: XEROX 12D-TPL 1narr]*

A person or entity that is or has been eligible and enrolled in State programs served by the vendor selected as a result of this procurement. Also may be referred to as Beneficiary, Member, Participant, or Recipient. *[HHS Procurement Library: 8-HHS 2020 Terms and Definitions]*

An individual, institution, facility, agency, physician, health care practitioner, non-medical individual agency, or other entity that is licensed or otherwise authorized to provide any of the Covered Services in the State to HHS 2020 Enterprise Agencies. Providers include individuals and vendors providing services to *[HHS Procurement Library: 8-HHS 2020 Terms and Definitions]*

Information that identifies a managed care plan. This data is currently used in ASPEN for enrolling clients appropriately and in Omnicaid only for cohort generation.

See Appendix J for all CD.

5.1.1.3 Metadata Repository

Metadata is data about data and provides context for source and target datasets. The Metadata repository containing all source schemas and the associated source record names uses a format similar to the following sample format.

Along with the versioned source schemas, the metadata repository also stores SMR's Physical Data Models (PDM). The PDM represents identified data domains and business entities, data elements contained in those entities, and the relationship between each of them. It also contains the physical properties of the data format, such as the schema definitions. Every PDM is depicted as an XSD, which also functions as metadata and a specification that the data, stored as an XML, will conform to. More information on the PDM as metadata can be found in deliverable SIDM4: Information Governance Catalog.

5.1.1.4 The ODS

The ODS will consolidate the data updates from the legacy data sources into the materialized data available in the SDS. ODS acts as a single point of source for the enterprise MMISR data. The ODS data structures follow the data structures defined in SDS.

Along with the data structure, for each document, the ODS captures the following metadata items:

- Last update time – In the MMDDYYYY HH:mm:ss format.
- Last updated by – Userid for a real-time update, or the incremental load process name for the bulk updates.
- Is new – A Boolean that indicates - true: if the document was created during the incremental load, or false: if the document was updated.
- Source metadata version – The version of the source metadata the document complies with
- Target metadata version – The version of the SDS metadata the materialized document complies with.
- Source name – The name of the legacy data source that sent the document to the ODS.
- URIs of the SDS and/or RDL documents – Used in creating the document in the ODS.

5.1.2 File and Database Structures

This subsection identifies and describes the management of the legacy source data extract files. The following information is included for every file:

- Record structures, record keys or indexes, and data elements referenced within the records.
- Record length (fixed or maximum variable length) and blocking factors.
- Access method (e.g., index sequential, virtual sequential and random access).
- An estimate of the file size or volume of data within the file, including overhead resulting from file access methods.

- Definition of the update frequency of the file (If the file is part of an online transaction-based system, provide the estimated number of transactions per unit of time, and the statistical mean, mode, and distribution of those transactions).
- Backup and recovery specifications.

This subsection will be expanded after receiving the appropriate data capacity, and security requirements.

5.1.2.1 Database Management System Files

The following are the databases of the SI Platform, and the file formats supported in each of those:

- The RDL – Since the RDL has been built on the MarkLogic NoSQL database and will contain data as XML documents.
- The SDS – The SDS is built on the MarkLogic NoSQL database and will contain data as XML documents.
- The MDM – The MDM is built on the MarkLogic NoSQL database and will contain data as XML documents.
- The ODS – The ODS is built on the MarkLogic NoSQL database and will contain data as XML documents.
- ESB Transaction Message Store – The ESB transactions are stored as JSON, XML, or binary files in the relational Oracle databases.
- Audit Log Store – The log files are stored on the Logging server as text files.

5.1.2.2 Non-Database Management System Files

The SMR, after ingesting the accessed source data into MarkLogic will have the source data in "inactive" CSV files. These files are then securely archived for data provenance purposes.

5.1.2.3 Database Backup

The backup represents the system baseline prior to any human and/or software interaction with the system or system data outside of the normal operating policies, processes, and procedures. If needed, a backup can be used to restore the system.

A database backup will be taken incrementally while stepping through the process of preparing, moving, and manipulating data. This is done to allow the project team to revert to any point in the data migration process if they run into issues.

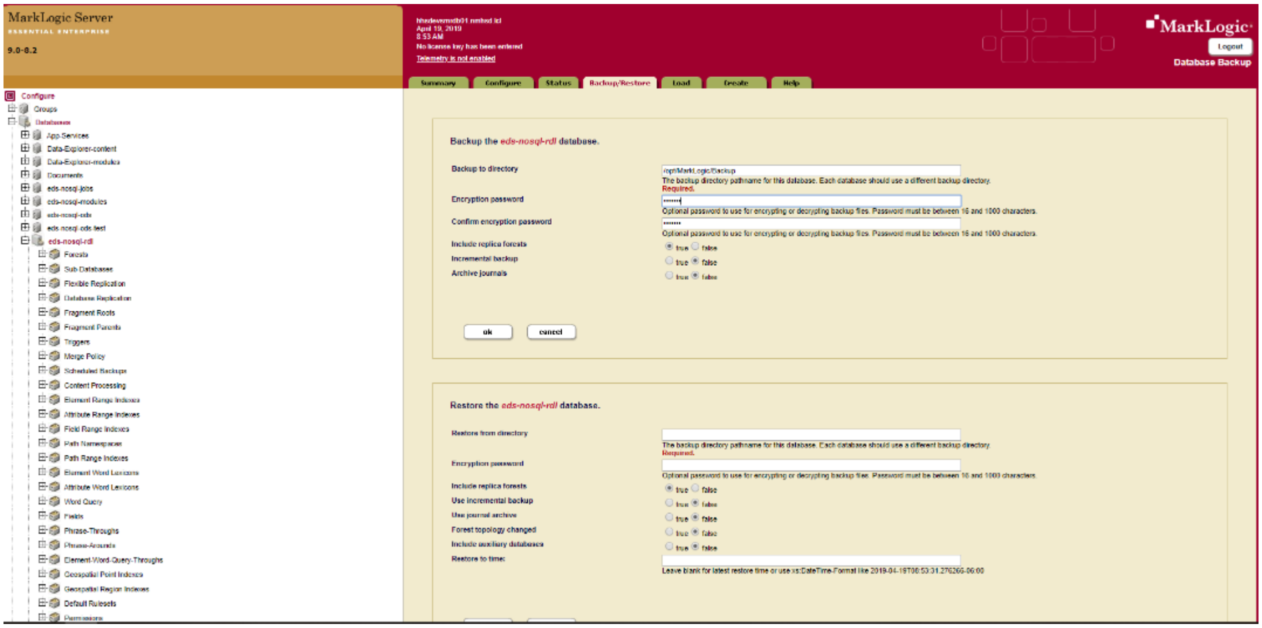
The SI Platform will support the following types of backup:

- Full back up of the database immediately.
- Scheduled backups.

The backup/restore operations, with MarkLogic journal archiving enabled, in the SMR, Metadata Repository, and MDM, provide a point-in-time recovery option that enables the restoration of database changes to a specific point in time between full backups, with the input of a wall clock time.

The SI Contractor DBA will schedule, using the MarkLogic admin console, the backup procedures. The following is a screenshot to help with the operation.

Figure 5-2: MarkLogic Backup/Restore Configuration



Backup the *eds-nosql-rdl* database.

Backup to directory: /opt/MarkLogic/Backup
The backup directory pathname for this database. Each database should use a different backup directory.
Required.

Encryption password: ••••••••
Optional password to use for encrypting or decrypting backup files. Password must be between 16 and 1000 characters.

Confirm encryption password: ••••••••
Optional password to use for encrypting or decrypting backup files. Password must be between 16 and 1000 characters.

Include replica forests: ● true ○ false

Incremental backup: ○ true ● false

Archive journals: ○ true ● false

Restore the *eds-nosql-rdl* database.

Restore from directory:
The backup directory pathname for this database. Each database should use a different backup directory.
Required.

Encryption password:
Optional password to use for encrypting or decrypting backup files. Password must be between 16 and 1000 characters.

Include replica forests: ● true ○ false

Use incremental backup: ○ true ● false

Use journal archive: ○ true ● false

Forest topology changed: ○ true ● false

Include auxiliary databases: ○ true ● false

Restore to time:
Leave blank for latest restore time or use xs:DateTime-Format like 2019-04-19T08:53:31.276266-06:00

On any failure, the objective is to ensure that the SMR is available for integration, and ultimately to end-users, within an acceptable time boundary, while ensuring that there is no loss of data. The plan includes the following for an ongoing backup configuration:

- Full data backup (RDL) weekly.
- One-time full data backup (SDS, Metadata Repository, ODS, and MDM).
- Incremental backup (SDS, Metadata Repository, ODS, and MDM) daily.

5.2 Data Conversion

This subsection describes the objectives of data conversion and references the data conversion deliverables. This subsection also explains the details of the ETL/ELT processes that integrate the data and describes the components of the SMR subsystem.

The SMR is a data repository which integrates different source module data identified as data providers and exposes the transformed and standardized (but still source-specific) data to the new MMISR modules. This source-specific data in the Common Data Model (CDM) format also includes the MDM component of the SI Platform. The SMR consumes copies of entire databases, files, and other types of extracted data to measure and improve data quality. The data will then be made available in an approved format (schema) to systems inside the HHS 2020 Enterprise.

The following picture explains the SMR as an independent, repeatable process that integrates data from heterogeneous and disparate systems such as Omnicaid, ASPEN, CSES, and DOH (DDSD), and populating the MMISR modules.

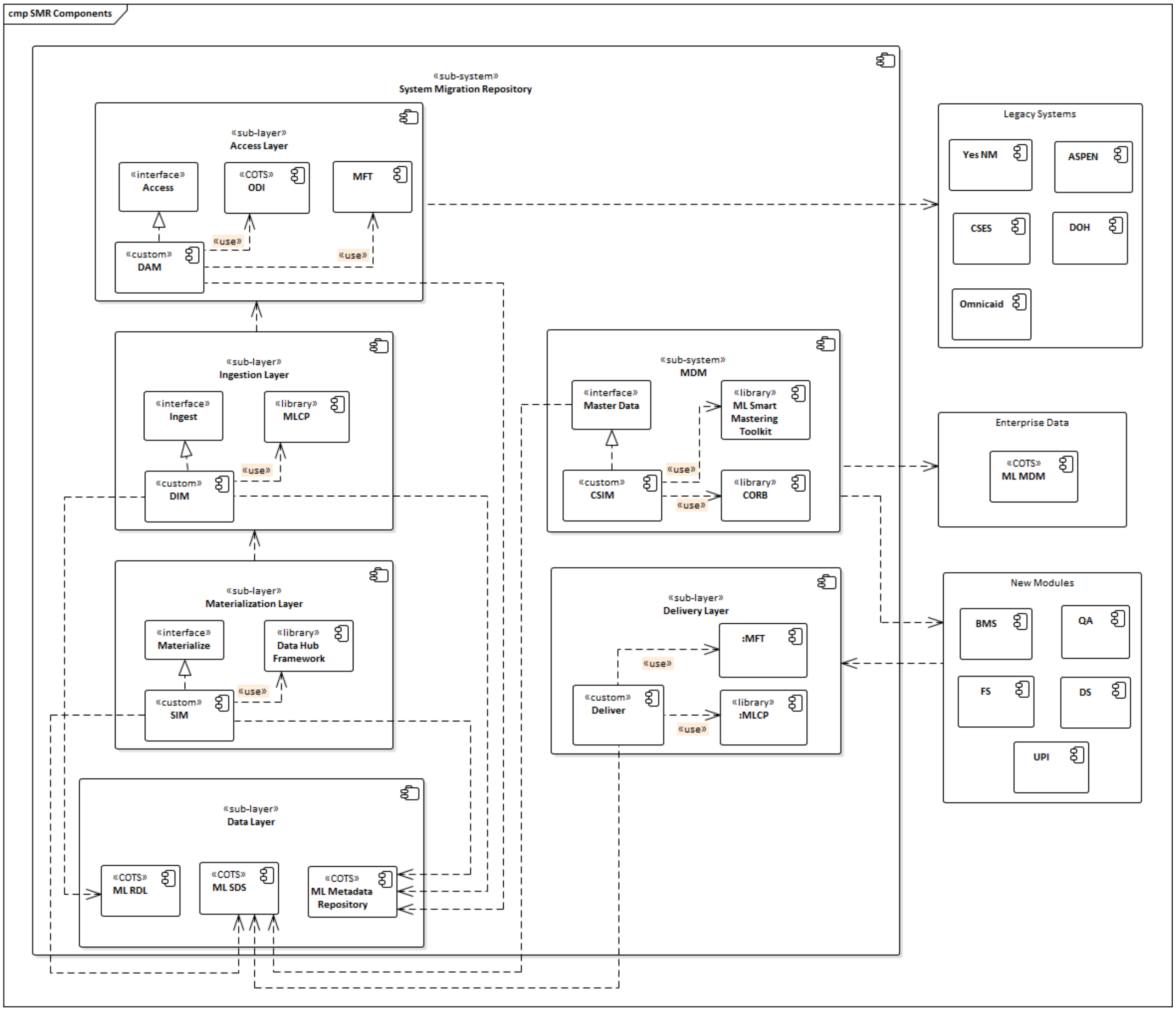
5.2.1 Alignment with PADU Approach and MITA Technical Strategy

The SMR/MDM solution is developed using the COTS NoSQL MarkLogic database. Although the MarkLogic ecosystem has the required libraries like MLCP and smart mastering to facilitate SMR and MDM solution, they require some additional custom development efforts on top of the libraries to facilitate factors like end-to-end automation, cohesiveness, and robustness. Due to these additional custom development efforts, the alignment with PADU for this framework is deemed as “Acceptable” level. As prescribed in the Reference architecture library, this missing functionality will be an architectural component implemented as a shared, reusable framework applicable to all data sources.

5.2.2 Logical Component view of SMR

The below figure shows the logical component view of SMR. This figure shows the SMR components as grouped under different physical layers of the SI Platform. It also depicts the data givers and data takers for the SMR.

Figure 5-3: Logical Component View of SMR

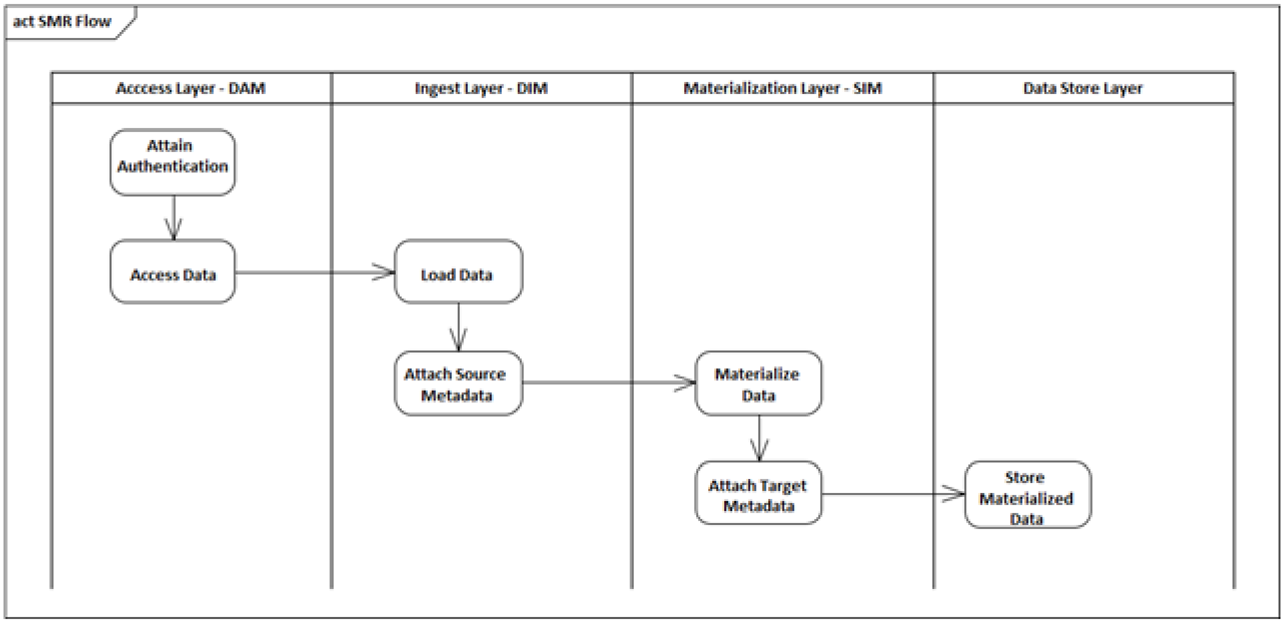


5.2.3 Components of the SMR

The SMR is implemented with the four custom components, namely the Data Access Module (DAM), the Data Ingest Module (DIM), the Source-specific Integration Module (SIM), and the Deliver module, that are independently developed, deployed, and maintained, and can be used repeatedly for new sets of data. The SMR components are reusable for any new data source identified in the future for data migration.

The SMR is built on the platform of MarkLogic, a multi-model NoSQL database. The figure below shows the SMR's logical and physical components and the information flow.

Figure 5-4: SMR Information Flow



5.2.3.1 Data Access Module (DAM)

The DAM component accesses the data at the source using a standard set of processes and technologies based on the type of the data source being accessed like RDBMS systems, NoSQL middle-tier databases, Web-API enabled integrating systems, and file-based data sources. The accessed data is then stored as XML, or delimited CSV files within the DAM. The DAM implements one-time bulk data loads as well as continuous incremental loads.

5.2.3.1.1 Bulk Data Access

The DAM supports the following bulk-load access patterns:

- File-Based Access (Batch) – To access the file-based data sources.
- ETL Based Access (Batch) – To support RDBMS data sources.
- REST-Based Access (Real-time) – To support the Web-API enabled integrating systems.

The DAM is implemented using Oracle Data Integrator (ODI), Oracle Managed File Transfer (MFT) and custom-built JDBC-based Java programs.

The table below provides details on the supported access patterns and their implementation in SMR.

Figure 5-5: Supported Access Patterns in SMR

Pattern	Description
File-Based Access (Batch)	Step 1: The data source places a file on a mutually accessible FTP server. The following are the supported file types: • XML • CSV Step 2: The presence of the file in the FTP server triggers the managed file transfer (MFT) service to initiate the DIM process. Oracle MFT will be used in this step for file transfer. Omnicaid will be accessed in this fashion.
ETL Based Access (Batch)	Step 1: The ETL process queries the database to extract data from tables. Oracle Data Integrator (ODI) tool will be used to connect to the data sources such as ASPEN and CSES. Step 2: ETL loads the data to the RDL collection as JSON records.
REST-Based Access (Real-time)	Step 1: The ESB exposes a web service the connecting system invokes to send the data in the message. The following is the supported format: • XML in the HHS 2020 common messaging format. Step 2: The ESB invokes SMR’s REST web service to load the data into the RDL.

5.2.3.1.2 Incremental Data Access

Before loading each incremental load, the data changed/added since the last load needs to be determined. Then, the appropriate integration hub component loads the data into the RDL. The incremental load use case illustrates the incremental load.

Following are the strategies used in SMR to determine the delta/incremental subset:

- The source system (Omnicaid) extracts the new/changed data using time stamps on CREATE and LAST_UPDATED fields.
- The ETL service extracts the new/changed data subset from the source system (ASPEN and CSES) using time stamps on CREATE and LAST_UPDATED fields.
- If the source system does not have timestamps in the database, the ETL service reads all of the data (in the file or web service payload) and uses process intensive ETL functions to extract the differential data by comparing the latest set with the previous set of data.

5.2.3.2 Data Ingest Module (DIM)

The DIM component transports raw format data from the DAM into the MarkLogic NoSQL-based RDL. The DIM supports real-time ingest of data into MarkLogic as well as batch ingests of bulk-data.

The DIM is implemented using various frameworks supported by MarkLogic based on the ingest pattern to be supported.

- MarkLogic RESTful Web Services are used for real-time data ingest.
- CORB with MarkLogic Content Pump (MLCP) is used for batch ingest of bulk data.
- Built-in XQuery functions for Change Data Capture (CDC) of incremental data.
- Java Client API/The Data Movement SDK also will be used for batch ingest of data.

5.2.3.2.1 Process Flow

The figure below describes the DIM process flow.

SMR.java is the starting point for the DIM and SIM activities of SMR. It takes the process parameter that takes the following values:

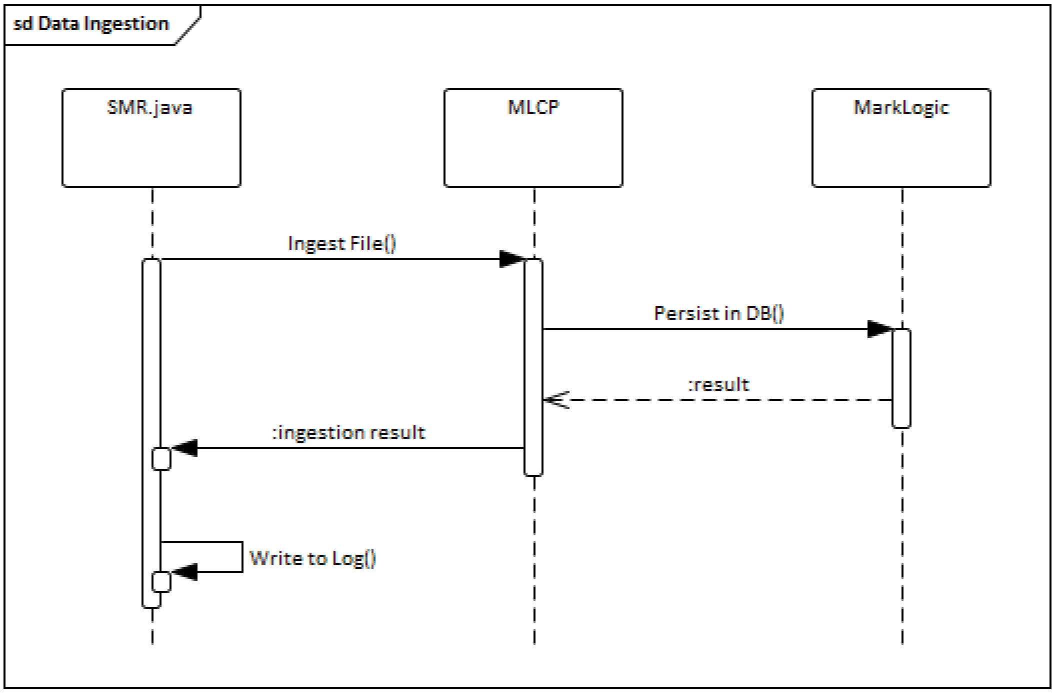
- Ingest
- Materialize

The DIM process is invoked by passing the “Ingest” value for this parameter. The ingest process invokes the MarkLogic Content Pump program that transforms the input CSV file into a series of XML documents before storing them in the RDL.

The SMR program for ingesting also takes the following additional parameters:

- CSV file location.
- The number of threads to be executed.
- Preferred Collection name.
- Additional metadata elements to be captured.

Figure 5-6: Data Ingestion Module Process Flow



5.2.3.2.2 The RDL

The RDL in SMR is considered the authoritative source of HHS 2020 legacy data and is used for designing the schema, consolidating reference data values into data standards, identifying relational database constraints (i.e., primary keys and foreign keys), creating metadata definitions, determining records of authority, is database modeling, and XML schema modeling.

The RDL will:

- Ingest any data in its native format.
- Capture metadata from the source.
- Store data as loaded.
- Permanently store a copy of the source data.

Instead of forcing data integration first, as in a traditional data warehouse, the SMR parses (materializes) the data later. That is, the RDL stores the data as ingested, and the SDS contains the materialized data—the specific data views and structures that support the business need for real-time analytics, discovery and ideation, data models, and data marts. The SMR can build a single view of the client over time, adding demographic data, then eligibility data, and then, claims and encounters data.

An SMR implements the RDL using commodity cluster computing techniques for massively scalable, low-cost data storage.

5.2.3.2.3 Architecture of Data Ingestion

The RDL ingests any data in the native format. The RDL stores the data in collections of related data according to the source, the table it was sourced from, and the time of ingestion, adding updates to the data as new records in the appropriate collection. The RDL ingests raw data from multiple sources, each of which may have a different native schema.

The RDL ingests any data in any format, including:

- Structured data from relational databases (rows and columns).
- Semi-structured data, such as comma-separated values (CSV), logs, extensible Markup Language (XML), and JavaScript Object Notation (JSON).
- Binary data, such as Portable Document Format (PDF) files, images, audio, and video.

The RDL can load the data set as a whole as well as interpret and load individual records if possible. The DIM attempts to match the data source/content to an RDL source schema. The DIM interprets the data and loads the data to the RDL in XML records in the appropriate collection. The XML notation preserves the schema of the source by using the same column names and table names.

For example, the ETL service can interpret the claims records in a table from Omnicaid and load each claim as a record in the Omnicaid and Claim collections. The RDL may store both the data set and the XML records in situations that require retention of an exact copy of the transmitted data or when the data format is unavailable during ingest.

The implementation defines the content, format, metadata, and destination collection for the data integration process. The SMR determines which collection to load the data based on the content and source of the data.

5.2.3.2.4 Fallout and Exception Management

The following are some of the exceptions that DIM handles:

- MarkLogic unavailability – This happens due to several reasons such as network, storage, and/or other issues internal to the MarkLogic database.

In the case of file-based source data, the MarkLogic Content Pump (MLCP) tool will ingest the files into the RDL. The MLCP tool recovers from the MarkLogic originated errors by invoking a fail-over procedure. It has in-built host switchover and retries mechanisms to leverage multiple hosts in the cluster:

- Incorrect or incomplete data set – Since ingestion follows the original data format, the XML, when formed in MarkLogic may be incomplete.

All application-specific errors generated as part of MLCP execution are logged in MarkLogic log files. The MarkLogic log files are then be aggregated, monitored, and analyzed via the Splunk log-monitoring tool.

In the case of API based ingestion, the Java Client API and Data movement SDK or REST APIs will be invoked from the respective ODI ELT program.

5.2.3.2.5 Logging and Audit

Process logs help investigate and troubleshoot potential failures and errors in data ingestion. In the event of an error, the admin can utilize log files to identify the point of failure. Log data will be both comprehensive and easy to analyze. The logs demonstrate the compliance for safe handling of data by documenting critical information about files, jobs, and data as well as user activity.

The following table lists the DIM logging data elements.

Table 5-1: DIM Logging Data Elements

Job and file data elements	User activity data elements
Filename Source location Target location Transfer initiating system (source name) Initiating procedure name Initiation time Completion time Size of the file Transfer status	Usernames Log-in time Session length Transfers initiated Folders accessed Files read/updated/deleted

The Splunk forwarders will be installed at SMR components to feed the logs generated to the Splunk enterprise for centralized log monitoring and analysis. The details of log aggregation and monitoring are explained in Subsection 4.5.4

5.2.3.3 Source-specific Integration Module (SIM)

The SIM component refers to the materialization of the heterogeneous data into a Common Data Model (CDM). Materialization will standardize and transform the data enriching it per industry-accepted standards.

The following are the objectives of the SIM:

- Standardize dates and other fields.
- Enrich data with additional metadata.
- Extract important data into indexes for faster searching.
- Leverage semantic triples/relationships.
- De-normalizing multiple data sources into one document.
- Historical data management.

The SIM is developed by creating Entity Templates in MarkLogic, where a template resembles the target entity and contains the XPath values of each element that forms the entity keeping the data lineage and provenance intact using MarkLogic's robust URI and Collection patterns.

5.2.3.3.1 The SDS

The SDS in the SMR is the source of HHS 2020 MMISR data and is materialized from the RDL using the data standards, identifying the source integrity, and applying the SMR metadata definitions. The data in the SDS will reference back to the RDL documents from where this document has been materialized. SDS will store data in the document NoSQL format. The data structure will follow the SMR data models explained in Subsection 5.1.1.2.

The SDS will:

- Integrate the source-specific data from its native format.
- Attaches the source and SMR metadata.
- Standardize the date fields to the ISO defined format.

An SMR implements the SDS using commodity cluster computing techniques for massively scalable, low-cost data storage.

5.2.3.3.2 Indexing

The table below explains multiple types of indexes supported by and used in the SDS. These indexes are implemented using MarkLogic libraries. Some of the indexes from this list help speed up searches that are based on key elements in the source XML. Other indexes are used as applicable.

Table 5-2: SDS Indexing

Index Type	Description
Word index	Word indexing enables simple single word searches. The SDS word indexing also supports an inverted index, which inverts the relationship between the word and document.

Index Type	Description
Phrase index	Phrase indexing enables two-word phrase searches. When a document is materialized, the SDS creates all possible two-word indexes.
Relationship index	Each XML element is used as an index for searching.
Value index	Value indexing maps strings with a value using a hashing technique. Instead of storing a long string, the SDS hashes element text to a succinct integer. Regardless of the element name and text string length, the string is only a small entry value in the index.
Range index	Range indexing enables searches based on inequalities, for example, Policy amount > 100.00.
Lexicons	Lexicons allow quick access to the document and collection Uniform Resource Identifiers (URI) in the database. The SDS can create a quick reference for lists of unique words or values. Type of lexicons include: • Word lexicon • Value lexicon • Value co-occurrences lexicon • Geospatial lexicon • Range lexicon • URI lexicon • Collection lexicon
Triple index	All triples are indexed to enable quick reference.

The SDS also uses metadata indexing that enables quick search of collections, directories, and security rules. The supported metadata indexes are:

- Collection indexes – Each document is tagged as belonging to any number of collections. Collection names are indexed and can be used for a search.
- Directory indexes – Similar to collections index; however, directory indexes are hierarchical and non-overlapping.
- Security indexes – User role-based document restriction is enabled using the security index.
- Properties indexes – Document properties are also used for searching.

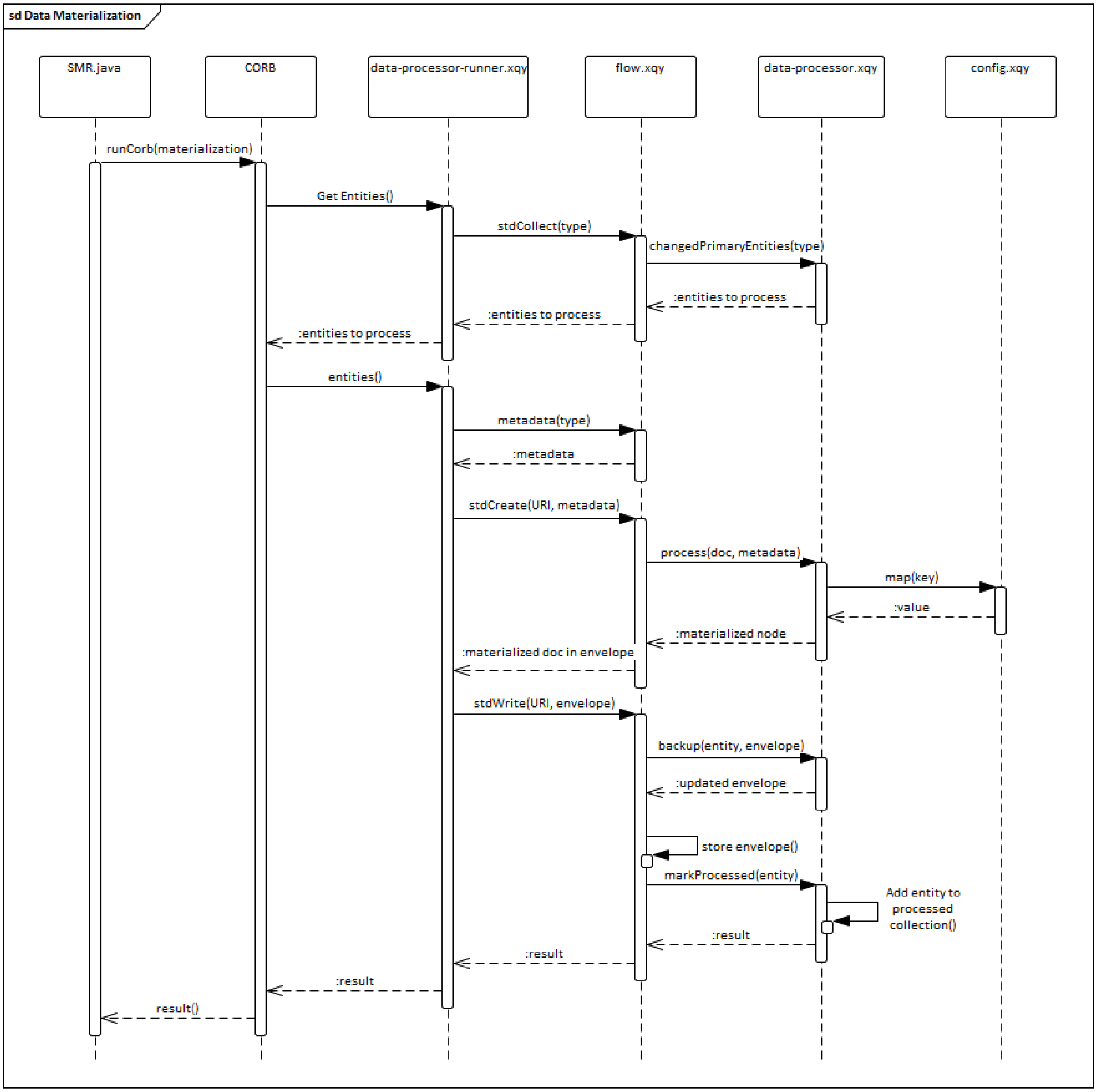
5.2.3.3.3 Process Flow

The figure below shows the process flow of the SIM process that generates the standardized documents in SDS.

The SIM program is executed in the following steps:

- The SMR.java invokes the CORB function of materialization.
- The CORB function then invokes the XML Query (XQuery) program named data-processor-runner.
- This program invokes the XQuery function to create collections.
- The program captures metadata.
- The program then invokes a function to create a standardized document:
    - The standardization function invokes function to retrieve the mapping of source elements to the standardized data elements.
    - Backup if the document already exists, and versions the document accordingly.
- The created standardized document contains appropriate indexes as mentioned in the mapping file.

Figure 5-7: SIM Process Flow



5.2.3.4 Update the ODS

The Update ODS component refers to the creation or updating of the materialized document into the ODS database. This component moves the document from the SDS into the ODS if the document is not already available in the ODS. If the document already exists in the ODS, it will be replaced with the copy from the SDS as the SDS data is considered the latest.

This component is developed by creating Entity Templates in MarkLogic, where a template resembles the target entity and contains the XPath values of each element that forms the entity keeping the data lineage and provenance intact using MarkLogic's robust URI and Collection patterns.

5.2.3.4.1 The ODS

The ODS is the source of HHS 2020 MMISR operational data and is materialized from the RDL using the data standards, identifying the source integrity, and applying the SMR metadata definitions. The data in the ODS will reference back to the RDL, and/or SDS documents from where this document was materialized. ODS will store data in the document NoSQL format. The data structure will follow the SMR data models explained in Subsection 5.1.1.2, and will leverage the indexes mentioned in Subsection 5.2.3.3.2.

The ODS will:

- Integrate the source-specific data from its native format.
- Attaches the source and SMR metadata.
- Standardize the date fields to the ISO defined format.

The ODS is also implemented, like SDS, using commodity cluster computing techniques for massively scalable, low-cost data storage, and the MarkLogic NoSQL database.

5.2.3.5 Cross-Source Integration Module (CSIM) or Data Mastering

MDM defines and manages the critical data of an organization to provide a single point of authoritative reference to it. The data that is mastered may include reference data- the set of permissible values, and the analytical data that supports decision-making. This subsection describes the process for data migration from SMR into MDM as well as the tools utilized. As part of this step, multiple source entity records are matched and merged. This subsection describes process flow with detailed steps.

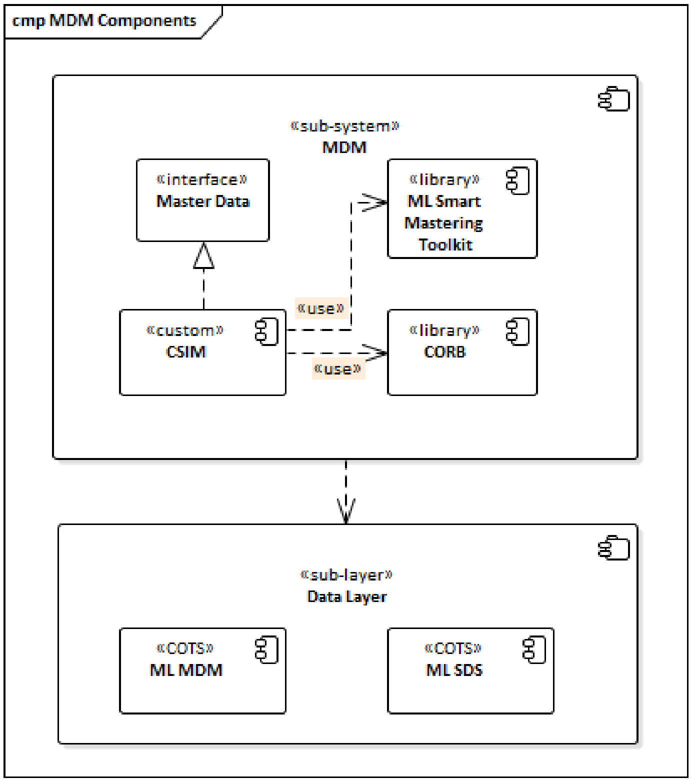
The following are some examples of the steps employed in this phase:

- Determine potential matches for each entity.
- Merge/unmerge records based on a threshold.
- Manage potential matches.
- Define fallout management approach.

The CSIM is the process where multiple source entity records are matched and merged. During this process, data across all sources are matched and merged, data is survived, and a preferred record is identified to assist in the creation of an entity profile and subsequently a consolidated view of a member.

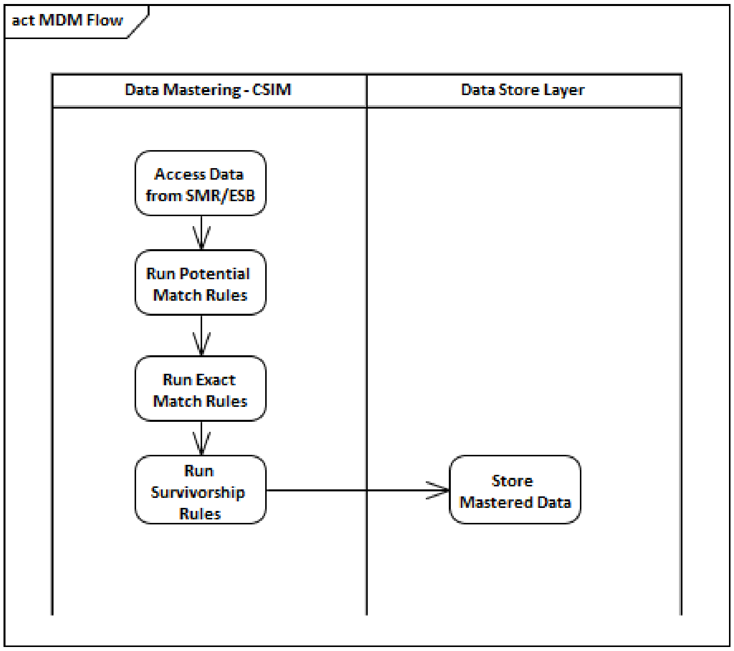
The following figure shows the logical components of MDM.

Figure 5-8: MDM Components



The overall process flow is described in the figure below.

Figure 5-9: MDM Process Flow



Step 1: Determine Potential Matches for Each Entity

The first step in the de-duplication of entity records within MDM is to identify potential matches. Based on the tool used, this could be broken down into a two-step process. The steps are:

- **Candidate List Builder** – In step 1 (Candidate List Builder), a few Specific Criteria would have to be identified that use a direct search on the entire set of SIM results across sources in the database. This process starts with one document in the SIM results and using the criteria finds all the other SIM documents that fulfill at least one of the criteria among them. This is done to reduce the number of compare operations that require lifting the documents from the disk. The candidate list builder rules are meant to cast a wider net across all the source records. These rules should follow a few key best practices:
    - The rules should use match on one or more of the critical data elements. For example, fuzzy name, DOB, or SSN match can be a candidate list builder rule for a person entity.
    - The rules should be loose enough that no SIM entity that can match the incoming entity is left out. For example, a fuzzy name match alone can be used as a candidate list builder rule. However, SSN match should not be used as a candidate list builder rule alone unless SSN is the only matching criteria for a Client entity.
    - The rules should be tight enough to not bring such a large result set, that the performance is impacted. For example, Address State or Zip Code alone may be too wide a criterion. It should be used with perhaps a DOB in conjunction.
- **Identify Potential Match** – In step 2, a set of rules will be identified that use a few fields in a match combination on the candidate list of SIM documents that have been identified in step 1. If there is more than one rule identified for this step, there would be an order in which these rules are applied on the list. Every document returned in the candidate list is compared to determine one or many potential merge groups. These potential merge groups then can be passed on for forming an entity using the survivorship rules. All these rules are typically associated with a match score and the combined match score can be compared with the merge/unmerge threshold to determine whether the potential matches are sent to automatic merge or a queue for data stewards to review.

Step 2: Merge/Unmerge

Once the identification of the potential matches is done, and a match score is calculated for a group of source records, the match score is compared with the merge thresholds. Comparing the match scores with the merge thresholds may result in three scenarios that are automatically handled by the CSIM module:

- **Automatic Merge** – When the match score for a match group is above the “Automatic Merge Threshold”, then all the source records in the match group are combined together and a new CSIM/MDM document is created. This new document is assigned a new unique identifier that becomes the enterprise entity identifier for the entity. To derive values to survive from the source records, a set of survivorship rules are applied. If a conflict exists where even after all the survivorship rules are applied, then the matched pair is considered fallout and would be handled through the fallout management process.

- **Discard Match Group** – If the match score for a match group is below the "Non-match Threshold," then all the source records in the match group are discarded as a match of each other, and no further processing would be done on those candidate records.
- **Automatic Unmerge** – If because of an update of a source record, and after running the match rules, the match scores of the source records are determined to be below the "Non-match Threshold," then an existing entity can be required to be unmerged, and it is broken down into the source records and the matching rules are applied to each of the source records once again to recreate new match groups.

Step 3: Manage potential matches

A potential match is defined as a set of source records that belong to a match group once the match rules have been applied to them. If the match score for a match group is between the "Automatic Merge Threshold" and the "Non-match Threshold," then the match group is marked for review by the data stewards.

A data steward can potentially mark the match group as ready for merge, then the survivorship rules would be applied, and a merged record is created.

If a data steward determines that the match group is not ready for merge, then the match group is discarded, and no further processing is done on them.

While the incremental Change data capture process is in progress, the source records and match groups that are part of the data stewardship review queue can be handled in two different ways:

- **Continuous Updates** – The way the data stewardship review queue is constantly updated as new records are ingested or updates are made on the existing source records, as every new record, and every update can change the potential match groups. This could mean that while the data steward is reviewing a match, the data may have updated in effect nullifying the review process altogether.
- **Locked Updates** – This way the source records and match groups in the data stewardship review queue are locked and no further updates would be accepted into the MDM until the process is resolved. This could mean that the MDM would not remain current with all the updates from the source system and all updates to source record are queued up until the issue is resolved.

Fallout Management

Within the context of MDM, the fallout can be defined as the records that cannot be processed by the rules defined within the MDM. One of the essential features of MDM is to be able to identify the records that get match scores that meet a threshold but are lower than the threshold for merging and marked for human review. Based on these identifications, the data stewards are expected to perform data improvement tasks at the source systems.

- **Not enough data/No minimum data set** – If a source record arrives at MDM, which does not contain a set of valid values defined in the "Minimum Data Set," it becomes impossible for MDM to apply the matching rules. This can lead to issues with data quality within the MDM. Thus, every source record is checked before ingesting into MDM for the availability of valid values for "Minimum Data Set" elements. If the record does not pass this check, it is considered fallout.

- **Conflicting data values** – Once a match group has been identified for an automatic match, the survivorship rules are applied on the data to combine the source record values into a master entity record. If there are not enough rules identified to mitigate a conflict that exists in the data of the records within a match group, then the entire match group is marked as fallout

Once fallout has been identified, they are handled in one of the following ways:

- If fallouts are related to data issues and source data needs to be corrected, the fallout records are added to a fallout queue for the data stewards and data administrators to review and update the data in the source system themselves.
- If fallouts are related to systemic issues and policy around the data or rules within the MDM needs changing, a change request would need to be raised and a new MDM rule would be added, or an existing MDM rule would be updated. This would mean that once the change is applied, all the records within MDM would need to be re-evaluated using the new rules. Depending on the magnitude of the change, it can be applied in two different ways:
    - As a trickle-down update, where the new rules are applied only when MDM touches an existing entity.
    - As a total refresh update, where the system halts processing of new records, and the entire MDM data is re-evaluated with the new set of rules and re-synched with the source systems.

Tools for Data Migration from SMR into MDM

The MDM is developed using the Smart Mastering libraries provided by MarkLogic and is contained within the MarkLogic database. The data in MDM and the SMR are stored in separate instances of the MarkLogic database. Hence, data migration is required to move the data from SMR into the MDM. The tools required for the data migration process are as follows.

Smart Mastering Framework

MarkLogic provided Smart Mastering Framework include a set of RESTful API extensions. The MDM core API (MAC) consists of REST API endpoints to invoke mastering and merging. These APIs are configuration driven to take parameters as input for the API. The API suite also includes services to retrieve history for documents or individual properties within the merged documents. Some of the key APIs provided by this suite are:

- Match – This API identifies the list of documents matching the given document.
- Merge – This API saves or provides a preview of a merge document, combining two or more other documents. The delete option on the same API will unmerge a previously merged document, restoring the original documents.
- Match-And-Merge – This API provides the convenience of calling both the match and merge in a single API.
- History - Smart Mastering Core tracks the match and merge the history of a document, as well as its provenance.
- Notifications – Notifications API identify the potential matches but did not score high enough to automatically merge. These are then presented to the data stewards for manual merge.

In addition to these APIs, there are other utilities and miscellaneous APIs focusing on nonfunctional aspects of the smart mastering library like statistics, dictionary, etc.

MDM XQuery Libraries

The XQuery libraries are structured to separate the API from the implementation. The APIs acts as a facade to external consumers and does not undergo drastic changes. However, the XQuery APIs continue to evolve to make the MDM operations efficient and robust. These XQuery libraries persist inside the MarkLogic's modules database. The modules database is an auxiliary database that is used to store executable XQuery, JavaScript, and REST code. These libraries are loaded inside MarkLogic with execute permissions. The loading operations, along with setting security privileges, are handled via Gradle scripts.

Some of the key XQuery libraries used by the Smart Mastering Core are:

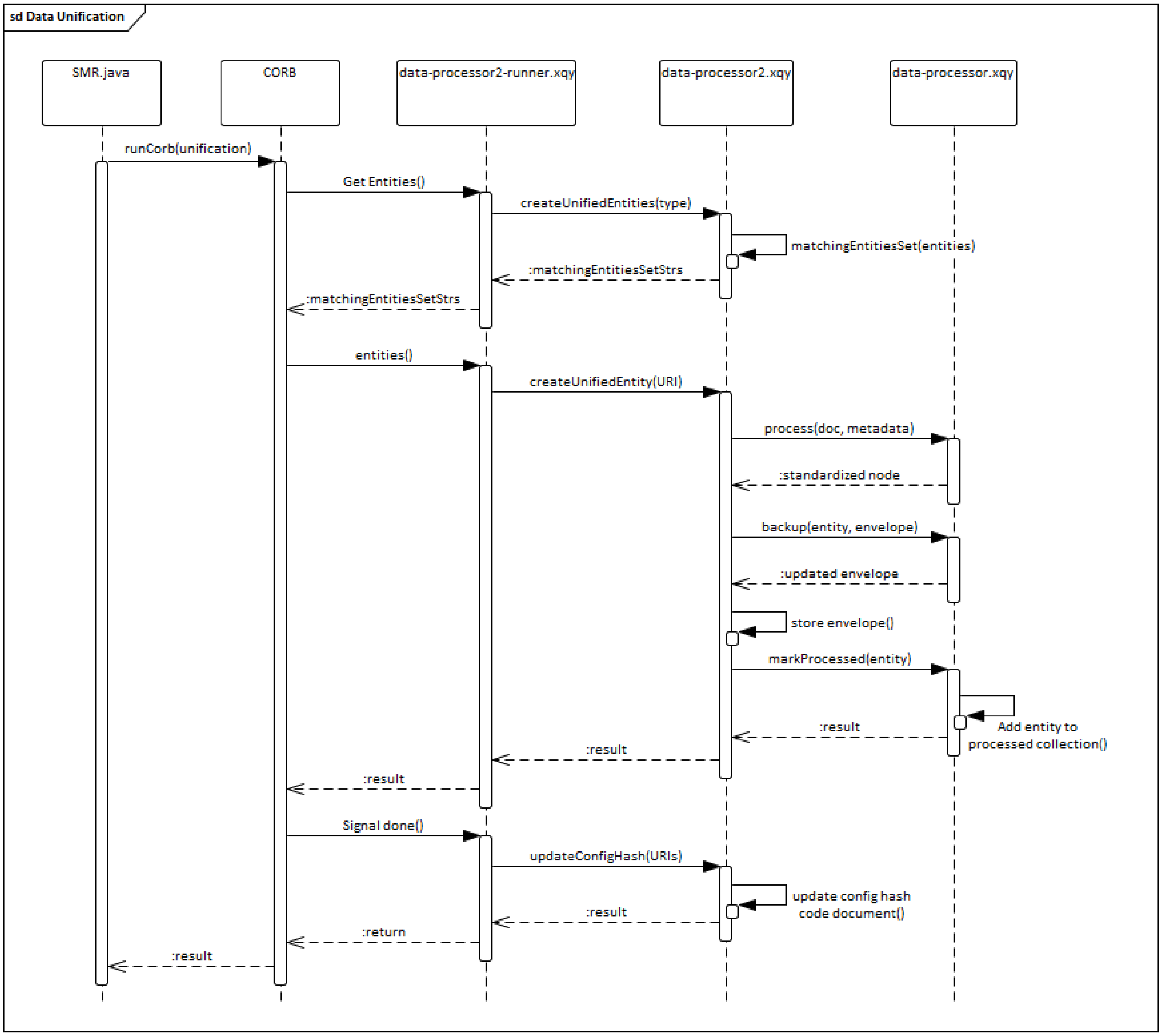
- Matcher.xqy – This module provides functions to store, retrieve, delete, and list match options; find potential matches for a document; and to store, retrieve, delete, and list match blocks.
- Merging.xqy – This module provides functions to build (preview), save, or remove merged documents and to store, retrieve, delete, and list merge options.
- Process-records.xqy – This module provides two functions to run through both matching and merging for a particular document.
- Match-And-Merge-Trigger.xqy – This module implements a trigger to process matching and merging any time a new document is inserted into the content collection.

The MDM function in the SI Platform is achieved by executing the following custom XQuery programs. These programs are executed by invoking a configurable Java program called SMR with "Unification" as the parameter. The program follows the following steps:

1. Java invokes a CORB function that integrates the different parts of the MDM.
2. The CORB program determines the potential matches.
3. For each entity retrieved, the program attaches the metadata and creates the entity if the document does not exist already.
4. If the document exists, it performs the unmerge and then merge based on the new matches.

The sequence of method calls is provided in the following figure.

Figure 5-10: CSIM Sequence Diagram



5.2.3.6 Deliver

This module exposes the data in the SMR for processing and consumption in the form of web services, data files, and secured direct access.

MarkLogic provides the following means of data transfer to external systems.

5.2.3.6.1 Bulk Data Migration

In MarkLogic, data can be exported as files for consumption using MLCP and output can be written to the native file system or to a database.

The content could also go through a server-side transformation process during export so that the format is suitable to be consumed by MMISR modules. MLCP also supports redaction of the content during export.

5.2.3.6.2 SQL Client based Migration

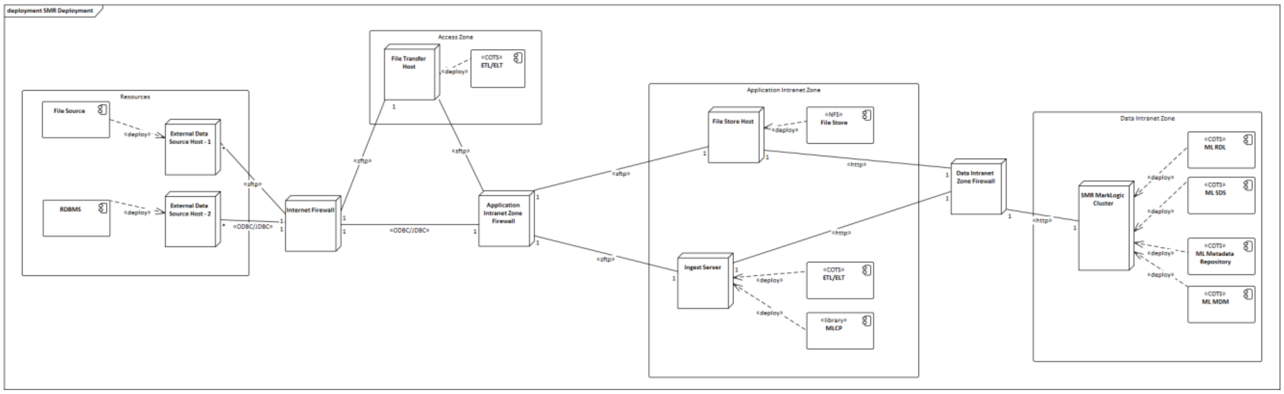
The SQL clients use the following means to access the data stored in the SMR:

- Template Driven Extraction (TDE): TDE enables the SMR to define a relational lens over the document data to query parts of the data using SQL.
- Optic API: The MarkLogic Optic API makes it possible to perform relational operations on indexed values and documents. The Optic API is not a single API, but a set of APIs exposed within the XQuery, JavaScript, and Java languages.

5.2.4 SMR Deployment View

The figure below provides the logical deployment view of SMR.

Figure 5-11: SMR – Deployment View



The components of SMR will be developed using COTS products of Oracle Fusion Middleware and MarkLogic NoSQL database along with the associated libraries. These components are spread in the following three architectural zones:

- Access zone
- Application intranet zone
- Data intranet zone

The DMZ will have the File Transfer component to poll the data from the external data sources providing data dumps as files.

The Application intranet zone will have the ETL/ELT components that access the ODBC/JDBC data sources, and the libraries are ingesting data into the SMR data stores.

The Data intranet zone will contain SMR's various data stores and repositories.

5.3 Data Security Design

The importance of securing the data stored in the SI Platform increases due to the nature of the data itself. As discussed above, the SMR in particular, and MDM stores data that are classified as PII, PHI, FTI, PCI, and so on. To secure the data stored, SI Platform design suggests the following three layers of security be implemented:

- Environment or Network-Level Security
- Encryption at Rest
- Database Role-based Access Security
- Document Element-Level Security

5.3.1 Environment Security

The SMR and MDM are deployed on the servers available in the secured Data-Zone as described in the Infrastructure Design. The Data-Zone is not available to the external users to access.

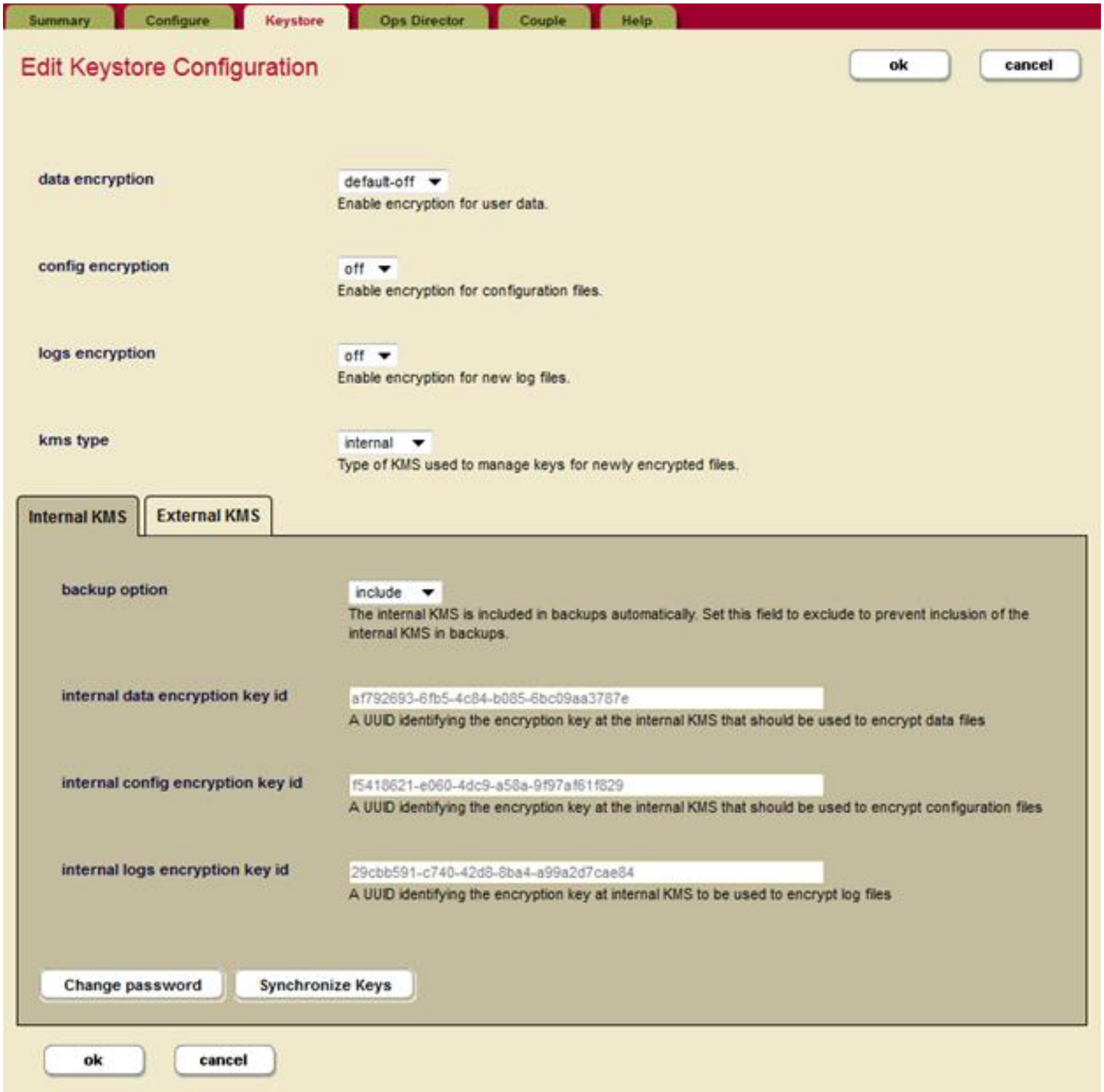
Any access to the MarkLogic cluster is further enabled based on the TLS/SSL authentication. SSL will be enabled on MarkLogic cluster.

5.3.2 Encryption at Rest

In the case of MarkLogic, the cluster data will be encrypted with a Key Management Service (KMS). The KMS manages the keystore that stores the encryption keys used to encrypt data in a secure location. This keystore can be either the MarkLogic embedded Public Key Cryptography Standards (PKCS) #11 secured wallet or an external third party KMS that conforms to the Key Management Interoperability Protocol (KMIP)-standard interface.

The figure below shows the option to configure a MarkLogic cluster with an internal KMS.

Figure 5-12: Data Encryption in MarkLogic Database



Edit Keystore Configuration

data encryption: default-off
Enable encryption for user data.

config encryption: off
Enable encryption for configuration files.

logs encryption: off
Enable encryption for new log files.

kms type: internal
Type of KMS used to manage keys for newly encrypted files.

backup option: include
The internal KMS is included in backups automatically. Set this field to exclude to prevent inclusion of the internal KMS in backups.

internal data encryption key id: af792693-6fb5-4c84-b085-6bc09aa3787e
A UUID identifying the encryption key at the internal KMS that should be used to encrypt data files

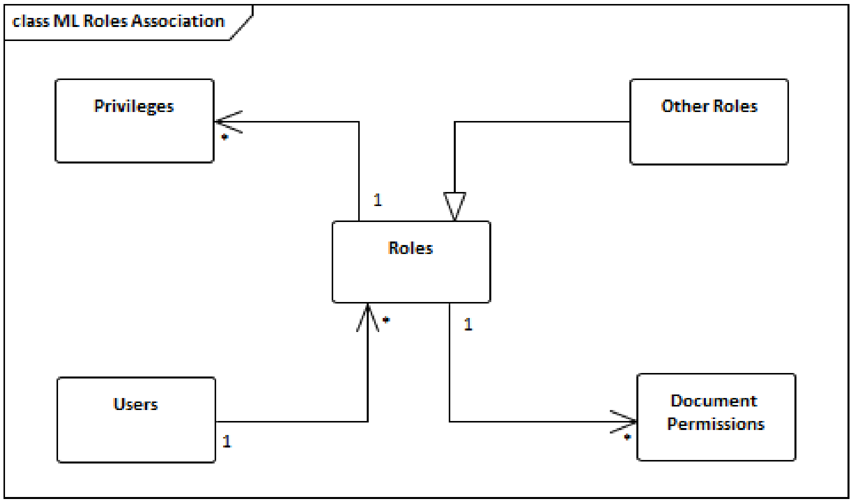
internal config encryption key id: f5418621-e060-4dc9-a58a-9f97af61f829
A UUID identifying the encryption key at the internal KMS that should be used to encrypt configuration files

internal logs encryption key id: 29cbb591-c740-42d8-8ba4-a99a2d7cae84
A UUID identifying the encryption key at internal KMS to be used to encrypt log files

5.3.3 Role-Based Data Access Security

The MarkLogic security model is based on the Principle of Least Privilege. The principle of least privilege is that users are given only those privileges that are actually required to efficiently perform their jobs. The privileges are in turn derived from the roles. Roles are the central point of authorization in the MarkLogic Server security model. Privileges, users, other roles, and document permissions all relate directly to roles. The following conceptual diagram shows how each of these entities points into one or more roles.

Figure 5-13: Role-Based Access and Permissions



As suggested in the MarkLogic Security Guide the following two types of privileges will be used by the SI Platform: Uniform Resource Identifier (URI) privileges and execute privileges:

- URI privileges are used to assign document creation permissions with URIs.
- Execute privileges are used to protect the execution of functions in XQuery code.

The following are the roles that have been identified to be configured for the SMR and MDM data stores.

Table 5-3: SMR and MDM Data Stores Roles

Role	Description
public	Users assigned this role can view the information that may be broadly distributed without causing damage to the organization, its employees, and stakeholders. The [PR Office/Marketing Dept/Information Security Management Department, etc.] must pre-approve the use of this classification. These users may also be outside the organization.
restricted-confidential	This role enables users to view highly sensitive or valuable information, both proprietary and personal. The users may not be outside of the organization or equipped with explicit permissions of a Director-level senior manager.
restricted-internal	This role enables the users to view the information whose unauthorized disclosure, particularly outside the organization, would be inappropriate and inconvenient. Disclosure to anyone outside of [NMHSD] requires management authorization.

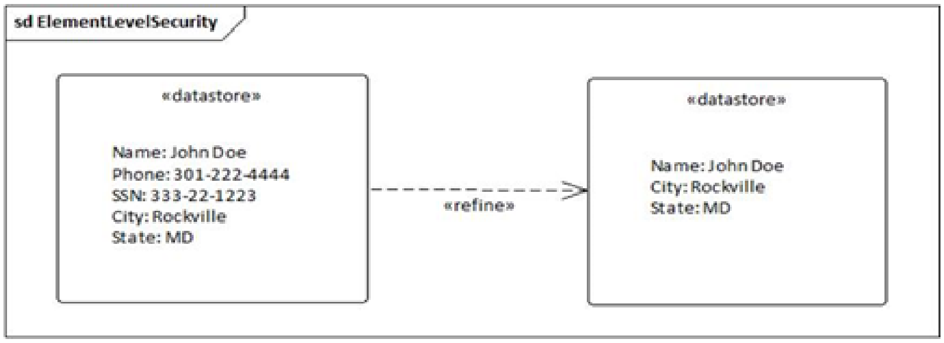
Role	Description
phi-confidential	PHI is individually identifiable health information that relates to the past, present, or future physical or mental health or condition of an individual. This is a provision of health care to the individual by a covered entity (for example, hospital or doctor), past, present, or future payment for the provision of health care to the individual. The HIPAA Security Rule specifies safeguards that covered entities and their business associates must implement to protect PHI confidentiality, integrity, and availability. PHI is not available or disclosed to unauthorized persons or processes without the explicit permission of a Director-level senior manager.
pii-confidential	This role enables the users to view the information that permits the identity of an individual to whom the information applies to be reasonably inferred by either direct or indirect means. The PII is defined as information: (i) that directly identifies an individual (e.g., name, address, social security number or other identifying number or code, telephone number, email address, etc.) or (ii) by which an agency intends to identify specific individuals in conjunction with other data elements, i.e., indirect identification. (These data elements may include a combination of gender, race, birth date, geographic indicator, and other descriptors). Additionally, information permitting the physical or online contacting of a specific individual is the same as personally identifiable information. This information can be maintained in either paper, electronic or other media. PII is not available or disclosed to unauthorized persons or processes without the explicit permission of a Director-level senior manager.
system	External Systems with appropriate authorization to view ALL the data. These are generally used in real-time information exchange between the consumer and provider systems.

Role	Description
admin	This role enables users to: • View, create, update, and delete the data. • Modify the application and database configuration. • Deploy the database applications. • Start, resume, and stop the database. • View, and configure databases using admin consoles.
operator	This role enables users to: • View the performance monitoring application. • View, and configure databases using admin consoles. • Start, resume, and stop the database.
tester	Allows access to the databases for testing purposes.
developer	Allows access to the databases for development purposes.
dba	This role gives users: • Full access to create, manage and drop databases/schemas. • Full access to manage tablespaces, user privileges, configuring user accesses.

5.3.4 Document Element-Level Security

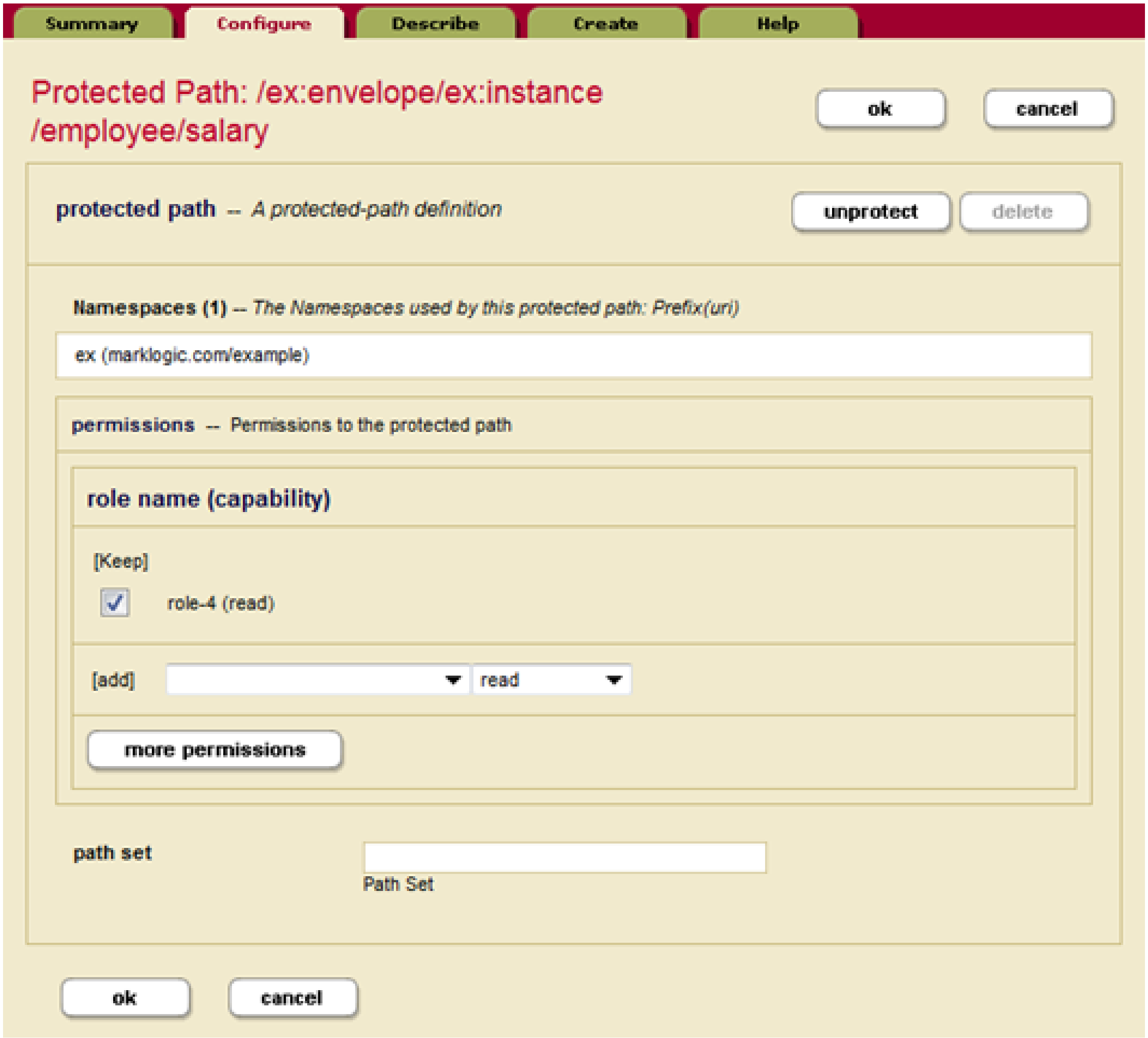
The SMR and MDM components will also enable element-level security applied at the document level. The element level security protects a part of the document from being visible to unauthorized users. Elements of a document will be protected from being viewed or from being updated by a user unless the user has the appropriate role-based authorization. The following figure describes the element security applied to an XML document when viewed by a user with lesser privilege.

Figure 5-14: Element-Level Security Example



Element level security works by specifying an 'indexable' path to an element (or JSON property) and configuring permissions on that path – creating a protected path. A path to an element in a document that has been configured with permissions is called a protected path. Permissions will be defined on an element the same way it is defined on a document. Document path can be configured to be protected via programs or through the admin configuration, as shown in the figure below.

Figure 5-15: Enable Element-Level Security in MarkLogic



Protected Path: /ex:envelope/ex:instance /employee/salary

protected path -- *A protected-path definition*

Namespaces (1) -- *The Namespaces used by this protected path: Prefix(uri)*

ex (marklogic.com/example)

permissions -- Permissions to the protected path

role name (capability)

[Keep]

☑ role-4 (read)

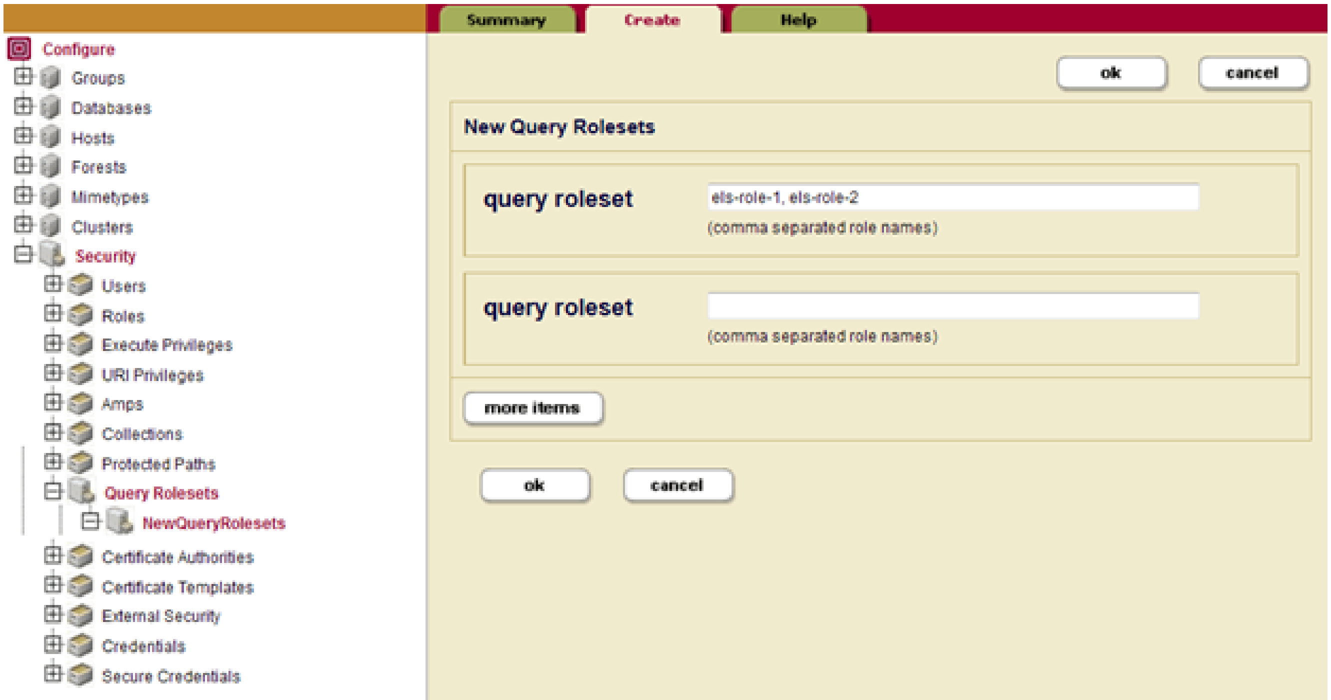
[add] read

path set

Path Set

In addition to these protected paths, the query rolesets need to be appropriately defined for the roles for element-level security to work. Query rolesets are used by the MarkLogic database to figure out the search results, based on the role(s) of the user running the query, in addition to the term being searched. Similar to protected paths, query rolesets can be configured programmatically or through the admin interface, as shown in the figure below.

Figure 5-16: Query Rolesets



New Query Rolesets

query roleset els-role-1, els-role-2
(comma separated role names)

query roleset
(comma separated role names)

The redaction feature is a read transformation applied on top of XML and JSON documents. Redaction addresses privacy concerns by making it possible to remove or mask information when importing, exporting, or copying data into and outside of MarkLogic. This prevents leakage of sensitive information to unauthorized users.

Figure 5-17: Element Redaction

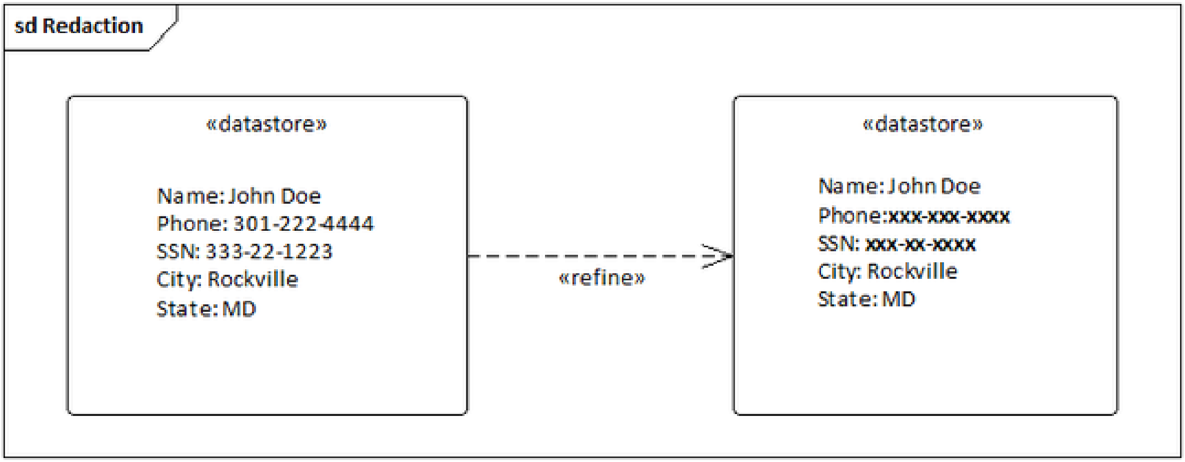
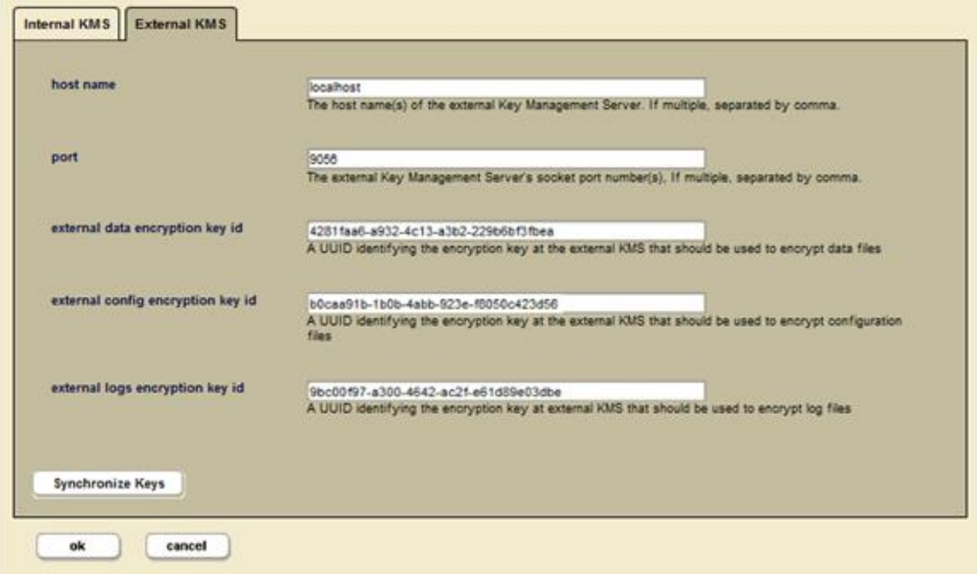


Table 5-4: Redaction Type Variations

Redaction Type	Variations	Description
Masking	Full	The original value is completely obscured. For example, 123-45-6789 becomes ###-##-####.
Masking	Partial	A portion of the original value is retained. For example, 123-45-6789 becomes ###-##-6789.
Masking	Deterministic	The same input always results in the same redacted output. For example, the value '12345' becomes '11111' everywhere it appears in content selected for redaction.
Masking	Random	Each input results in a random redacted value. For example, the value '12345' might be masked as '1a2f578' in one place and '30da61b' in another.
Masking	Dictionary-based	A form of random or deterministic masking in which the replacement value is drawn from a user-defined redaction dictionary.
Concealment	N/A	The original value (and potentially the containing XML element or JSON property) is entirely removed. For example, if you conceal the value of /a/b, then <a><b>12345</b></a> might become </a>.

MarkLogic uses rule-based redaction when exporting the document to determine the redaction logic. A redaction rule tells MarkLogic how to locate the content within a document that should be redacted and how to modify that portion. A rule expresses the business logic, independent of the documents to be redacted.

Figure 5-18: External KMS Configuration



host name: localhost
The host name(s) of the external Key Management Server. If multiple, separated by comma.

port: 9056
The external Key Management Server's socket port number(s). If multiple, separated by comma.

external data encryption key id: 4281faa6-a932-4c13-a3b2-229b6bf3fbea
A UUID identifying the encryption key at the external KMS that should be used to encrypt data files

external config encryption key id: b0caa91b-1b0b-4abb-923e-f8050c423d56
A UUID identifying the encryption key at the external KMS that should be used to encrypt configuration files

external logs encryption key id: 9bc00f97-a300-4642-ac2f-e61d89e03dbe
A UUID identifying the encryption key at external KMS that should be used to encrypt log files

5.3.5 Logging and Audit

Process logs help investigate and troubleshoot potential failures and errors in data ingestion. In the event of an error, the administrator can utilize log files to identify the point of failure. Log data will be both comprehensive and easy to analyze. The logs demonstrate the compliance for safe handling of data by documenting critical information about files, jobs, and data as well as user activity.

The following table lists the data elements that will be logged.

Table 5-5: Data Elements Log

Job and file data elements	User activity data elements
Filename Source location Target location Transfer initiating system (source name) Initiating procedure name Initiation time Completion time Size of the file Transfer status	Usernames Log-in time Session length Transfers initiated Folders accessed Files read/updated/deleted

5.4 User Machine-Readable Interface

The following are the systems and user roles associated with the SI Platform including both the internal and external users of the SMR, and MDM systems, as well as the databases.

5.4.1 Inputs

The following are the different input data formats the SI Platform supports:

- Data extracts via comma separated delimited files – All legacy systems provide data input as CSV files.
- XML data in canonical data formats – All ESB transactional data will be received in this structure.
- Binary – the EDS APIs receive transactional data as binary documents.

5.4.2 Outputs

The SI Platform supports the following output data formats:

- Data extracts as XML documents.
- Bulk data extracts as compressed files.
- Data in XML canonical structures for real-time messaging.

5.4.3 User Roles and Privileges

The following are the system roles and privileges of the SI Platform.

Table 5-6: User Roles and Privileges

User Type	User Group/Role	Privileges
Application Administrators	Admin	• View the application configuration, including the encrypted value of some encrypted attributes. • Modify the application configuration. • Deploy Enterprise Applications and Web Service modules. • Start, resume, and stop the servers. • Install the software patches.
Application Support Operations	Operator	• View the server configuration, except for encrypted attributes. • Start, resume, and stop the servers.
Testing Team (SIT, QAT, UAT)	AppTester	• Access applications for testing purposes that are running in Administration mode.
Application Deployers	Deployer	• View the server configuration, including some encrypted attributes related to deployment activities. • Deploy the application code and software updates.
SI Users	Users	• This group contains all users who have been authenticated.
Database Administrators	DBA	• Full access to create, manage, and drop databases/schemas. • Manage tablespaces, user privileges, access.

User Type	User Group/Role	Privileges
System Administrator	root	- Set up and maintain the system accounts, user profiles, file systems, software, third-party libraries. - Verify the peripherals are functional - Handle hardware failures, monitor system performance - Manage system backup and recovery - Create and configure the Virtual Machines. - Implement security policies for the users of the system.
Network Administrators	administrator	- Create and manage subnets, IP addresses, routing, and ACL rules. - Manage the firewall connectivity between the environments, subnets, WAN, VPN, and LAN. - Create VIPs, service pools on the load balancer.

5.5 User Interface Design

The SI Platform uses out-of-the-box user interface components provided by the COTS products such as Oracle Fusion Middleware Administrative Consoles, MarkLogic Administration Console, and MarkLogic Data Explorer.

5.5.1 Section 508 Compliance

The administrative and management consoles that come as part of Oracle Fusion Middleware like API manager, BPEL Process Manager, etc. are Section 508 compliant. The details can be found in the product documentation here and, process to enable accessibility mode can be found here.

6 Detailed Design

This section elaborates on the components of the SI Platform, the various patterns each component supports, and the tools used to implement these components.

6.1 Hardware Detailed Design

6.1.1 ESXi Host Design

The ESXi host resources are distributed to run virtual as machines and are aggregated to build clusters of highly available pools of compute resources. For the SI Platform, NM HSD will be using a Dell hyper-converged infrastructure for the MMISR project.

6.1.1.1 ESXi Host Configuration

The following subsection describes the ESXi host configurations in the environment.

6.1.1.1.1 Hardware and System Resources

For the SI Platform:

- The ESXi 6.5 update 2B will be installed on the physical nodes. The physical nodes have 2 CPU that are 64-bit x86 CPUs with 576 Gb RAM, 1 GB, 10 GB, or 40 GB Ethernet adapters and VxFlex OS with one protection domain.
- The bios version on the servers will be in compliant with the approved vendor hardware revision level. All populated sockets will be enabled to recognize the cores; Hardware assist will be enabled at the BIOS level to support the infrastructure.
- The ESXI Hosts for the SI Platform will be booted off the disk with the designated memory assigned to the host, along with the boot a scratch partition that will be created on the boot device.
- The SI Platform hardware configurations for the HCI environment is configured on the VXRack Flex 1000. The configuration and assembly process for each system is standardized, with all components installed in the same manner on each host.

The following table lists ESXi host physical configuration made for the SI Platform.

Table 6-1: ESXi Host Physical Configuration

<table>
<tr><th>Compute Server</th><th>Quantity</th><th>CPU</th><th colspan="2">RAM</th><th>BIOS Version</th><th>Firmware Version</th></tr>
<tr><td>PowerEdge R640 – 2-Intel(R) Xeon(R) Gold 6132 CPU @ 2.60GHz/Model 85</td><td>15</td><td>Total - 420 Cores</td><td colspan="2">8.6 TB RAM</td><td>1.4.9</td><td>3.21.23.22</td></tr>
<tr><td>Storage Node Type</td><td>Quantity</td><td colspan="2">SSD Capacity Per Node</td><td colspan="2">Total Usable Capacity</td><td>Flex OS Version</td></tr>
<tr><td>FLEX-SYS-6412</td><td>15</td><td colspan="2">10x3.84TB SAS</td><td colspan="2">262.08TB</td><td>R2_6.0.0</td></tr>
</table>

6.1.1.1.2 ESXi Host and User Access

The ESXi hosts are administered through vCenter Server using the jump server installed in Dell VxRACK Flex 1000 infrastructure. Direct access to the host console is available, to do troubleshooting via Secure Shell (SSH) Access - Remote command line console access. Only the root user is allowed to log in to an ESXi host by default. The SI Platform's ESXi 6.5 hosts will be joined to the Active Directory domain to allow administrative access using AD Groups and maintain a log of who has logged into the host.

6.1.1.1.3 ESXi Scratch Configuration

The ESXi 6.5 is installed on to a local disk for each host, and a 4 GB VFAT scratch partition is created and is used to store persistent log files in order to provide VM-support outputs used by VMware to troubleshoot issues on the ESXi host.

6.1.1.1.4 Virtual Machine Swap Configuration

For the ESXi Hosts, the swap file location will be stored in the same location as the virtual machine's configuration file. This is the default and recommended configuration. This will reduce the complexity of the configuration. Host Profiles do not have a configuration setting for Host Swap File Location in vSphere 6.5.

6.1.1.1.5 ESXi Host Design Decisions

The following table lists ESXi host logical design decisions made for this architecture design.

Table 6-2: ESXi Host Design Specifications

Decision ID	Design Decision	Design Justification	Design Implication
D1	SSH and ESXi Shell services will be enabled on the hosts for only root	Enabling access decreases delays for troubleshooting efforts.	Enabling access decreases the security of ESXi hosts and infrastructure.
D2	Lockdown Mode will be enabled on the hosts.	Enabling Lockdown Mode increases the security of ESXi Hosts.	With this setting on the only root will be able to login as mentioned above rest of the users under any authentication group will not be able to login to the ESXI hosts

D3	Timeout values will be set for SSH, Shell, and DCUI in the Advanced ESXi host User Vars settings. Services will stay running but the users will be logged out after 1O minutes of inactivity.
	ESXiShell = 0
	ESXiShellInteractive = 600
	DCUI = 600
		These timeout settings will balance standard troubleshooting practices with security needs and allow administrators to troubleshoot without interruption.	Additional settings must be configured in the ESXi host profiles. Exact setting names are documented in the install guide.

Decision ID	Design Decision	Design Justification	Design Implication
D4	All ESXi hosts will be added to the nmhsd.lcl domain and user access will be with Active Directory accounts.	Using Active Directory membership provides greater flexibility in granting access to ESXi hosts. A single group can be used for administrative access and managed through existing policies. Having users log in with a unique user account allows greater visibility for auditing.	Additional settings must be configured in the ESXi host profiles using an AD Service account.
D5	The default ESX Admins group will be changed to The AD GROUP (TBD) ESXi host administrators belong to the above group will be accessed through the Jump Host.	Additional changes must be made to the advanced settings for the host in the ESXi host profiles.	Additional changes must be made to the advanced settings for the host in the ESXi host profiles.

6.1.1.1.6 vCenter Server Design

For the SI Platform, there will be two vCenter servers with embedded PostgreSQL database and embedded PSC for Management Cluster and external PSC for Compute Cluster with other add-ons such as Log in-sight, vRealize, vROPS, and Automation. The vCenter will be deployed within a single SSO domain and will be configured to use Active Directory (integrated Microsoft Windows authentication) identity source configured with the default nmhsd.lcl domain to give access to admins and users. Each vCenter is a single instance in the Management and Compute cluster. Backup for each of the vCenter will be on a separately dedicated data store using appliance level backups.

Figure 6-1: Compute Cluster

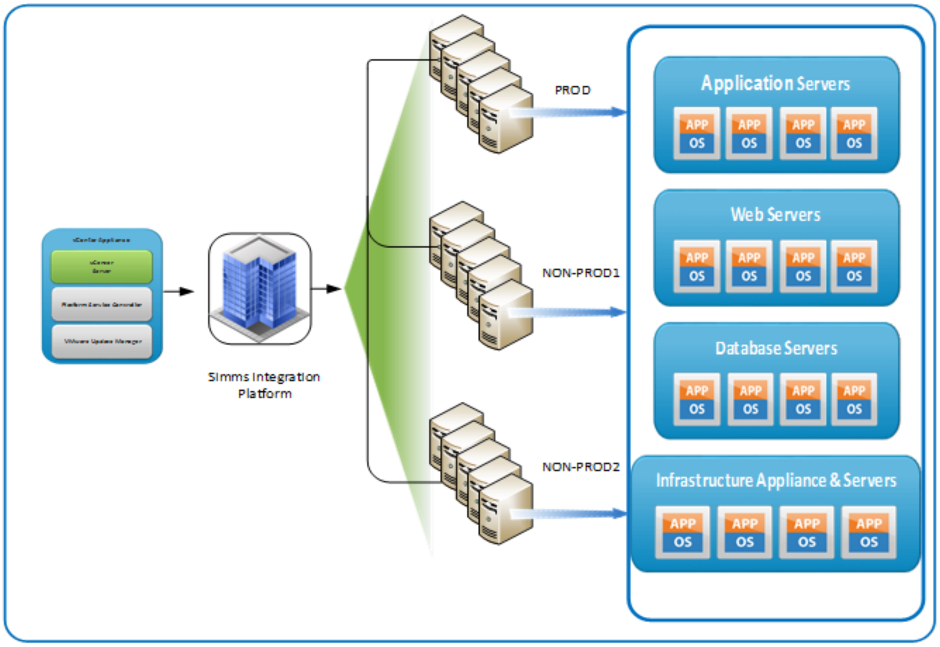


Figure 6-2: Management Cluster

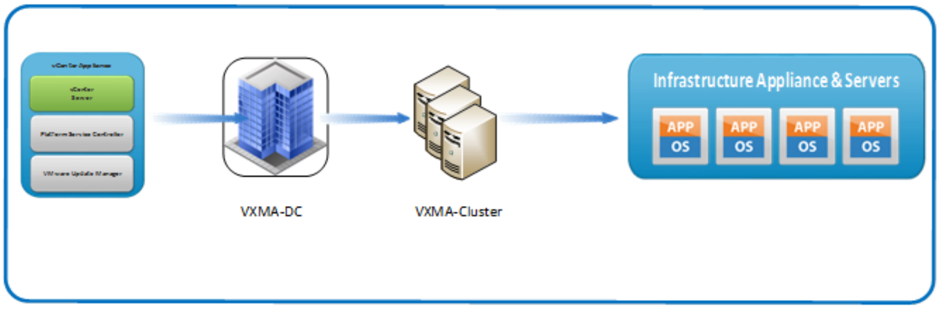


Table 6-3: vCenter Server Appliance Specifications

Attribute	Specification
vCenter Server version	vCenter Server 6.X
Physical or virtual system	Virtual
Number of CPUs	4
Memory	16 GB
Number of NIC and ports	1/1
Storage disk size	290 GB

Attribute	Specification
Operating system and SP level	VMware Appliance

6.1.1.1.7 Server Decisions

The following table lists decisions made regarding the vCenter Server implementations.

Table 6-4: vCenter Server and Platform Service Controller Design Decisions

Decision ID	Design Decision	Design Justification	Design Implication
D6	Two vCenter server will be deployed for the SI Platform	Segregate Management and Compute workload of this requirement.	Logical design is simplified.
D7	Backups of vCenter and PSC appliances will be performed with full appliance backups.	In vSphere 6.5, to back up and restore a virtual machine that contains a vCenter Server Appliance you must do a full appliance backup.	Requires additional capacity on Veritas NetBackup.
D8	All vCenter Server instances required within the solution will be deployed using the VMware provided virtual appliance.	Allows for a more rapid deployment method, enabling faster scalability changes and a reduction in Microsoft licensing costs.	None
D9	vCenter Server Appliances the Small size vCenter appliance with PSC Appliance for Management and Compute Cluster.	Most NM HSD vSphere environments will use no more than 100 ESXi hosts, which calls for a Small sized vCenter Appliance with PSC deployment.	None
D10	All vCenter Server instances utilize the embedded PostgreSQL database on the vCenter Server Appliance.	Reduces both management overhead and Microsoft or Oracle licensing costs.	None
D11	vCenter appliances will use static Internet Protocol (IP) addresses and DNS entries. All components are referenced by FQDN.	Static IP address, DNS entries, and FQDN references are a best practice.	None

Decision ID	Design Decision	Design Justification	Design Implication
D12	vCenter Single Sign-On will be configured to use Active Directory (with integrated Microsoft Windows authentication).	vCenter Single Sign-On will be connected to Active Directory to allow users to login and be assigned permissions with their Active Directory credentials.	Reliant on Active Directory for the majority of users to login.
D13	vSphere Environment will have multiple workload clusters supporting the VM the environments that will be managed under this vCenter.	Three clusters will be configured to accommodate PROD and NON-PROD virtual environment.	
D14	HA redundancy will be configured. Cluster redundancy will be configured for the equivalent of N+1 HA protection. The cluster design will use 1 host failure capacity tolerated.	HA is required for all vSphere workloads.	None
D15	DRS will be enabled for all clusters and will be set to Manual.	DRS is required for all vSphere workloads. Disabling DRS in a cluster that has resource pools will delete all the resource pools. Instead, set DRS to manual when needed.	Resource Pools are used in the environment for vRA integration.
D16	Cluster size will be up to five hosts.	This is an SI Platform standard based on the workload and can be changed at any time based on the workload demand.	None
D17	Storage and Networking Boundaries will be maintained within a cluster.	This is a vSphere architecture requirement. Only storage presented everywhere is the template data stores.	No data store will be across multiple clusters.

Table 6-5: Workload Cluster Design Specifications

Attribute	Specifications
Number of hosts required to support workloads with no over- commitment	5

Attribute	Specifications
Number of hosts in cluster with HA allowance	5
Capacity for host failures per cluster	1 Host reserved
Number of "usable" hosts per cluster	4 or 5 usable hosts

6.1.1.1.8 vCenter Server Feature Design

NM HSD vSphere Environments will have the following features enabled or disabled.

Table 6-6: vCenter Server Feature Usage Summary

Feature	Enabled/Disabled
vSphere HA	Enabled
vSphere HA Admission Control	Enabled
vSphere HA VM Monitoring	Disabled
vSphere FT	Enabled
vSphere DRS	Enabled
vSphere EVC	Enabled
vSphere Resource Pools	Enabled

6.1.1.1.9 vSphere HA & Admission Control

The following table lists the vSphere HA & Admission decisions made for this architecture design.

Table 6-7: vSphere HA Design Decisions

Decision ID	Design Decision	Design Justification	Design Implication
D18	vSphere HA will be enabled on all the management clusters to protect against ESXi host failure.	This will provide High Availability of management infrastructure workloads.	There must be sufficient resources on the remaining host to satisfy the server requirements in the event of a host outage.
D19	vSphere HA will be enabled on all workload to protect against ESXi failure.	This will provide High Availability of migrated VM workloads. N+1 redundancy.	There must be sufficient resources on the cluster remaining host to satisfy the server requirements in the event of a host outage.

Decision ID	Design Decision	Design Justification	Design Implication
D20	Isolation response will be disabled (default).	In the unlikely event that the hosts become isolated within HA, the virtual machines will continue to run.	If the network issue set causing isolation also affects the virtual machines, then they also become unavailable and will not be started in non-isolated hosts.

Table 6-8: vSphere HA Admission Control Design Decisions

Decision ID	Design Decision	Design Justification	Design Implication
D21	All clusters will have admission control set define failover capacity by the static number of hosts with the reserved capacity of 1 host.	This meets the SI Platform design requirement of N+1 redundancy.	There must be sufficient resources on the remaining host to satisfy the server requirements in the event of a host outage.
D22	All clusters will have a slot size policy set to cover all powered on virtual machines.	This enables management of cluster capacities according to the declared service levels and avoids the power-up of any more virtual machines when maximum capacity is reached.	There must be sufficient resources on the remaining host to satisfy the server requirements in the event of a host outage.

Table 6-9: vSphere HA Monitor Virtual Machines Design Decision

Decision ID	Design Decision	Design Justification	Design Implication
D23	Virtual Machine Monitoring is disabled and is not in use.	The SI Platform does not want to risk any vSphere Infrastructure Services potentially restarting VMs without their knowledge.	None

6.1.1.1.10 VMware HA Cluster Configuration

For this design, NM HSD has made the following decisions, as described in the table below.

Table 6-10: VMware HA Cluster Confirmation Specifications

Attribute	Specification
Enable host monitoring	Enable
Admission control	Allow VMs to be powered on even if they violate availability constraints.
Admission control policy	Cluster tolerates 1 host failure.

Attribute	Specification
Default VM restart priority	High (critical VMs) Medium (majority of VMs) Disabled (non-critical VMs)
Host isolation response	VMs remain on
Enable VM monitoring	Disable
VM monitoring sensitivity	Medium

- Each element in the SI Platform VMware infrastructure is built with the high-availability.
- All the servers will have the NIC teaming enabled and all the storage controllers will have multi-pathing configured for high-availability.
- All of the ESXi host will be clustered to provide dynamic resource scheduling and high-availability to all the VMs in the infrastructure.

6.1.1.1.11 vSphere Fault Tolerance Decisions

vSphere Fault Tolerance protects the virtual machine against host failures by providing recovery with zero downtime. This is achieved by creating an identical copy of the virtual machine on the secondary ESXi host, mirroring all activity.

The following section describes vSphere Fault Tolerance design choices for the ESXi hosts in the environment. At NM HSD, FT is configured and enabled.

Table 6-11: vSphere FT Design Decisions

Decision ID	Design Decision	Design Justification	Design Implication
D24	vSphere FT is configured for production environment at this time.	Production only exists for redundancy.	Only critical servers with single point of failure will be configured for fault tolerance.

6.1.1.1.12 VMWare vSphere Distributed Resource Scheduler

For the SI Platform infrastructure, Load Balancing functionality is achieved using vSphere DRS and vMotion to migrate workloads from heavily loaded hosts to less utilized hosts in the cluster and set to manual. The following table lists the vSphere DRS design decisions made for this architecture design.

Table 6-12: vSphere DRS Design Decisions

Decision ID	Design Decision	Design Justification	Design Implication
D25	vSphere DRS will be enabled on all clusters and set to the default manual setting.	This provides the best tradeoff between load balancing and excessive vSphere vMotion events and activity.	None

Each cluster can be configured for VMware Distributed Resource Scheduling (DRS) to automatically balance the load of the VMs within a cluster. This will be set at the default setting of moderate for all clusters. A decision on this will be taken once we reach the stage of Production build out.

- Migration threshold will be set to Default (Moderate).
- DRS policies will be created as per the application requirement (separate VMs or keep the VMs together).
- Affinity rules to be set on the VMs, as required by the application.

6.1.1.1.13 VMWare vSphere Enhanced vMotion Compatibility (EVC)

EVC allows vMotion migrations from one host to another host successfully by checking the CPU feature set are similar within the same cluster. If the CPU feature set are not similar, it will prevent migrations from failing because of incompatible CPUs.

By setting EVC on the cluster, hosts with newer CPUs can be added at a later date, without disruption. EVC can also be used to perform a rolling upgrade of all hardware with zero downtime.

The following table lists the vSphere EVC design decision made for this architecture design.

Table 6-13: EVC Design Decisions

Decision ID	Design Decision	Design Justification	Design Implication
D26	EVC mode will be enabled on clusters and will be set to match the highest level available for the different hardware available in each cluster.	This permits newer hosts to be added at a later date. Clusters must contain hosts with CPUs from the same vendor for EVC to be enabled.	vSphere vMotion perform migrations between the new and older CPU families.

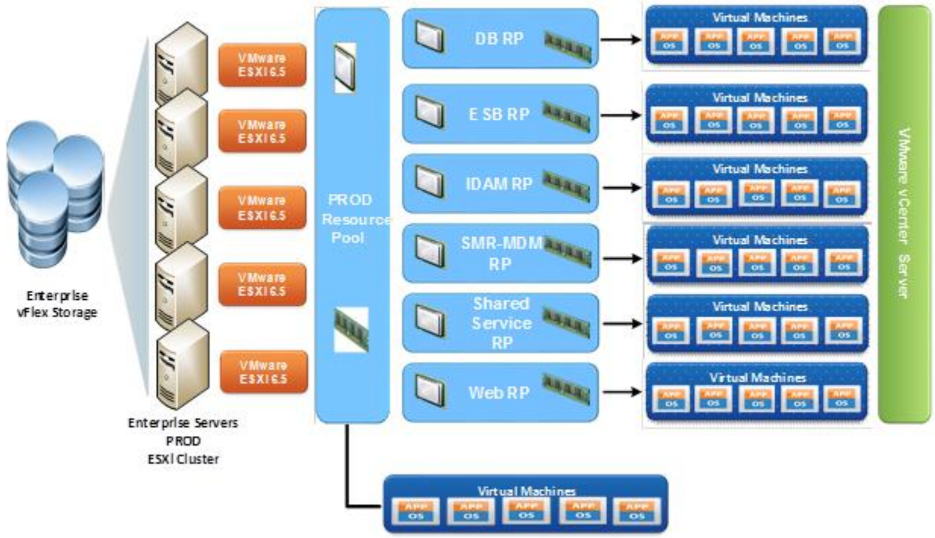
6.1.1.2 Resource Pools

The following section describes resource pool design choices in the environment.

6.1.1.2.1 Resource Pools Feature Details

A resource pool is a logical abstraction allowing for flexible management of CPU and Memory resources. Resource pools are being grouped into hierarchies (Parent & Child) and are used to partition available CPU and Memory. Resource pool with limit and reservation settings will be used for virtual machine workloads that require dedicated and isolated resources, such as Oracle Database servers.

Table 6-14: Resource Pool Logical Diagram



The SI Platform will have resource pools configured, based on the functions and virtual environments.

6.1.1.2.2 Resource Pools Design Decisions

The following table lists the resource pool design decisions made for this architecture design.

Table 6-15: Resource Pools Design Decisions

Decision ID	Design Decision	Design Justification	Design Implication
D27	Resource Pools will be configured based on each of the clusters for PROD & Non-PROD. Resource Pools are tied to each of the functions and environments (DEV, SIT, QAT,	Allows for resources to be isolated to prevent unfair resource utilization between environments.	None

Decision ID	Design Decision	Design Justification	Design Implication
	UAT, PROD SUPPORT, PRODPATCH, and PROD) to help enforce limits on consumed resources.		

6.1.1.2.3 Resource Pools

Below are the resource pools for all the environments based on the workload.

Table 6-16: Resource Pools Allocations

Cluster	Resource Pool	Priority	Parent RP	Server Group RP	CPU	Memory
PROD	1 Parent RP + 6 Child RP	High	Production RP		518 vCPU	3276 GB
				DB RP	160 vCPU	1216 GB
				ESB RP	124 vCPU	752 GB
				IdAM RP	60 vCPU	360 GB
				SMR-MDM RP	16 vCPU	96 GB
				Shared Service RP	96 vCPU	512 GB
				Web RP	22 vCPU	100 GB
NON-PROD-1			UAT		134 vCPU	896 GB
			PRODSUPPORT		348 vCPU	2064 GB
			PRODPATCH		348 vCPU	2064 GB
NON-PROD-2			DEV		122 vCPU	832 GB

Cluster	Resource Pool	Priority	Parent RP	Server Group RP	CPU	Memory
			QA		70 vCPU	520 GB
			SIT		62 vCPU	424 GB

This Configuration will be changed over time.

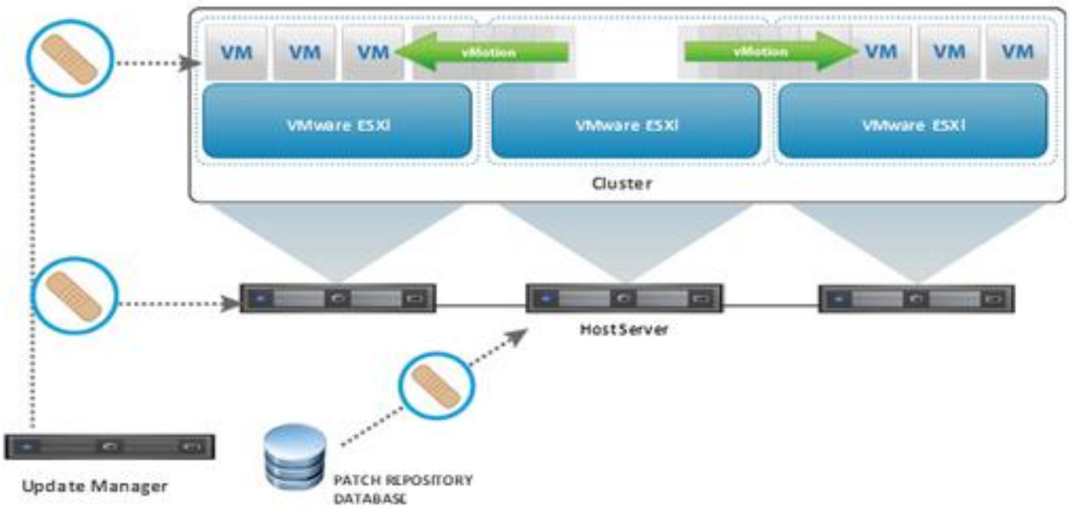
6.1.1.3 vSphere Update Manager Design

This section describes the design for VMware vSphere Update Manager (VUM). The vSphere update manager is embedded in the vCenter appliance and is used to manage the vSphere patch and version upgrade for ESXi hosts, virtual machines, and virtual appliances. There are two instances of vCenter server for the SI Platform, and each instance has a separate vSphee Update Manager. A separate patch repository data store will be created to download third-party patches and VMware patches using the windows jump host server because there is no direct connection to the Internet from the ESXi hosts and vCenter.

The remediation of the VxRack and VMware infrastructure is done using the Dell Release Certification Matrix (RCM) for all major releases, including firmware and software. Minor VMware security patches can be installed with VUM.

Note: Virtual Machine operating systems will be updated with RHEL satellite server.

Figure 6-3: VMWare Update Manager



6.1.1.3.1 vSphere Update Manager Design Decisions

The following table lists the vCenter Update Manager design decisions made for this architecture design.

Table 6-17: vSphere Update Manager Design Decisions

Decision ID	Design Decision	Design Justification	Design Implication
D28	vSphere Update Manager will be installed as part of the vCenter appliance.	Part of vCenter design and will be enabled for distribution.	Manual intervention required for remediation effort due to RCM dependency.
D29	vSphere Update Manager will use a separate patch repository.	Update Manager does not have direct access to internet and will have a manual download and stored on patch repository.	Additional data stores required.
D30	Default baselines for critical and non-critical patches will be configured for management, Edge, and Payload clusters.	No customized baselines required for the environment.	Added through RCM.
D31	Hosts, VMs, and VAs remediated on a monthly basis as per NM HSD business guidelines.	The schedule has to be aligned to the business policies at NM HSD.	None

6.1.1.4 Virtualization Network Layer Design

The following section describes the physical and virtual network layer design for the SI Platform.

6.1.1.4.1 Physical Network Layout

VxRack Hardware for the SI Platform consists of four top of the rack switches, two aggregate switches and a management switches installed in the VXRack and will be connected to the existing network at the datacenter.

All components (Hosts) of the VxRACK will be connected to the Top of Rack switch. The Top of rack are connected to the Cisco Aggregation switch within the VxRACK cabinet. The Aggregation switch are connected to the core switch of NM HSD network infrastructure, which acts as Layer 2 communication. Traffic will traverse through the core switch to the Palo-Alto firewall and terminates at Layer 3. The firewall address configured for the Vlan will be used as the default gateway.

Please refer the Physical Diagram exists in Section 4.2.1.3, Figure 4.6 (Network Cable Layout).

6.1.1.4.2 Virtual Network Layout

Network segments for the virtual infrastructure which are routable IPs are routed through a Distributed Virtual Switch (DVS) via port groups configured for each Vlans. All distributed switches are configured with redundant physical NIC's considering for any Hardware failures on each host.

NIC teaming are used to increase the network bandwidth available in the network path and provides redundancy to avoid single points of failure for networks and is done by assigning multiple physical NICs to a virtual switch.

Management Cluster – There are three Distributed Virtual Switches configured for management cluster, and their purpose is as follows:

- Dvswitch-0 – This switch is the primary virtual switch for all traffic including Fault Tolerance, Vmotion, and public network. Within this switch, there are 7 port groups created for each VLAN for network segmentation.
- Dvswitch-1 and Dvswitch-2 – These two Distributed virtual switches are created for redundant Disk path and will never be modified or changed.

The break out-of-the-port groups is shown above, and additional port groups will be created as the need arises to accommodate more networks. Please refer to the Diagram that maps to the VLANs is in 4.3.4.10 Figure 6 4 (Virtualization Logical Design).

Table 6-18: Virtual Networks for Management Cluster

Vlan ID	Port Group	DvSwitch
1	default	DvSwitch-0
400	sys-mgmt-400	DvSwitch-0
401	flexmgr-install-401	DvSwitch-0
402	sys-esx-mgmt-402	DvSwitch-0
403	sys-esx-vmotion-403	DvSwitch-0
404	sys-vsan-404	DvSwitch-0
405	sys-sio-mgmt-405	DvSwitch-0
406	sys-sio-data1-406	DvSwitch-1
407	sys-sio-data2-407	DvSwitch-2

Compute Cluster – There are three Distributed Virtual Switches configured for the compute cluster, and their purpose is as follows:

- Dvswitch-0: This switch is the primary virtual switch for all traffic including Fault Tolerance, Vmotion, and public network. Within this switch, there are 106 port groups created for each VLAN for network segmentation.
- Dvswitch-1 and Dvswitch-2: These two Distributed virtual switches are created for redundant Disk path and will never be modified or changed.

The break out of the port groups is shown below, and additional port groups will be created as the need arises to accommodate more networks.

Table 6-19: Virtual Networks for Compute Cluster

Vlan ID	Switch A Name	DvSwitch
1	default	DvSwitch-0
401	flexmgr-install-401	DvSwitch-0
402	sys-esx-mgmt-402	DvSwitch-0
403	sys-esx-vmotion-403	DvSwitch-0
404	sys-vsan-404	DvSwitch-0
405	sys-sio-mgmt-405	DvSwitch-0
406	sys-sio-data1-406	DvSwitch-1
407	sys-sio-data2-407	DvSwitch-2
600	Prod-VIP	DvSwitch-0
601	Prod-DMZ	DvSwitch-0
602	Prod-ESB	DvSwitch-0
603	Prod-IDAM	DvSwitch-0
604	Prod-SS	DvSwitch-0
605	Prod-SMR-MDM	DvSwitch-0
606	Prod-DB	DvSwitch-0
607	Prod-Storage	DvSwitch-0
608	Prod-MGMT-NW	DvSwitch-0
609	Prod-MGMT-AD	DvSwitch-0
610	Prod-MGMT-Audit	DvSwitch-0

Vlan ID	Switch A Name	DvSwitch
611	ProdPatch-VIP	DvSwitch-0
612	ProdPatch-DMZ	DvSwitch-0
613	ProdPatch-ESB	DvSwitch-0
614	ProdPatch-IDAM	DvSwitch-0
615	ProdPatch-SS	DvSwitch-0
616	ProdPatch-SMR-MDM	DvSwitch-0
617	ProdPatch-DB	DvSwitch-0
618	ProdPatch-Storage	DvSwitch-0
619	ProdPatch-MGMT-NW	DvSwitch-0
620	ProdPatch-MGMT-AD	DvSwitch-0
621	ProdPatch-MGMT-Audit	DvSwitch-0
622	UAT-VIP	DvSwitch-0
623	UAT-DMZ	DvSwitch-0
624	UAT-ESB	DvSwitch-0
625	UAT-IDAM	DvSwitch-0
626	UAT-SS	DvSwitch-0
627	UAT-SMR-MDM	DvSwitch-0
628	UAT-DB	DvSwitch-0
629	UAT-Storage	DvSwitch-0
630	UAT-MGMT-NW	DvSwitch-0
631	UAT-MGMT-AD	DvSwitch-0
632	UAT-MGMT-Audit	DvSwitch-0
633	SIT-VIP	DvSwitch-0
634	SIT-DMZ	DvSwitch-0
635	SIT-ESB	DvSwitch-0
636	SIT-IDAM	DvSwitch-0
637	SIT-SS	DvSwitch-0
638	SIT-SMR-MDM	DvSwitch-0

Vlan ID	Switch A Name	DvSwitch
639	SIT-DB	DvSwitch-0
640	SIT-Storage	DvSwitch-0
641	SIT-MGMT-NW	DvSwitch-0
642	SIT-MGMT-AD	DvSwitch-0
643	SIT-MGMT-Audit	DvSwitch-0
644	QAT-VIP	DvSwitch-0
645	QAT-DMZ	DvSwitch-0
646	QAT-ESB	DvSwitch-0
647	QAT-IDAM	DvSwitch-0
648	QAT-SS	DvSwitch-0
649	QAT-SMR-MDM	DvSwitch-0
650	QAT-DB	DvSwitch-0
651	QAT-Storage	DvSwitch-0
652	QAT-MGMT-NW	DvSwitch-0
653	QAT-MGMT-AD	DvSwitch-0
654	QAT-MGMT-Audit	DvSwitch-0
655	DEV-VIP	DvSwitch-0
656	DEV-DMZ	DvSwitch-0
657	DEV-ESB	DvSwitch-0
658	DEV-IDAM	DvSwitch-0
659	DEV-SS	DvSwitch-0
660	DEV-SMR-MDM	DvSwitch-0
661	DEV-DB	DvSwitch-0
662	DEV-Storage	DvSwitch-0
663	DEV-MGMT-NW	DvSwitch-0
664	DEV-MGMT-AD	DvSwitch-0
665	DEV_MGMT-Audit	DvSwitch-0
666	ProdSupport-VIP	DvSwitch-0

Vlan ID	Switch A Name	DvSwitch
667	ProdSupport-DMZ	DvSwitch-0
668	ProdSupport-ESB	DvSwitch-0
669	ProdSupport-IDAM	DvSwitch-0
670	ProdSupport-SS	DvSwitch-0
671	ProdSupport-SMR-MDM	DvSwitch-0
672	ProdSupport-DB	DvSwitch-0
673	ProdSupport-Storage	DvSwitch-0
674	ProdSupport-MGMT-NW	DvSwitch-0
675	ProdSupport-MGMT-AD	DvSwitch-0
676	ProdSupport-MGMT-Audit	DvSwitch-0
677	Training-VIP	DvSwitch-0
678	Training-DMZ	DvSwitch-0
679	Training-ESB	DvSwitch-0
680	Training-IDAM	DvSwitch-0
681	Training-SS	DvSwitch-0
682	Training-SMR-MDM	DvSwitch-0
683	Training-DB	DvSwitch-0
684	Training-Storage	DvSwitch-0
685	Training-MGMT-NW	DvSwitch-0
686	Training-MGMT-AD	DvSwitch-0
687	Training-MGMT-Audit	DvSwitch-0
688	Performance-VIP	DvSwitch-0
689	Performance-DMZ	DvSwitch-0
690	Performance-ESB	DvSwitch-0
691	Performance-IDAM	DvSwitch-0
692	Performance-SS	DvSwitch-0
693	Performance-SMR-MDM	DvSwitch-0
694	Performance-DB	DvSwitch-0

Vlan ID	Switch A Name	DvSwitch
695	Performance-Storage	DvSwitch-0
696	Performance-MGMT-NW	DvSwitch-0
697	Performance-MGMT-AD	DvSwitch-0
698	Performance-MGMT-Audit	DvSwitch-0

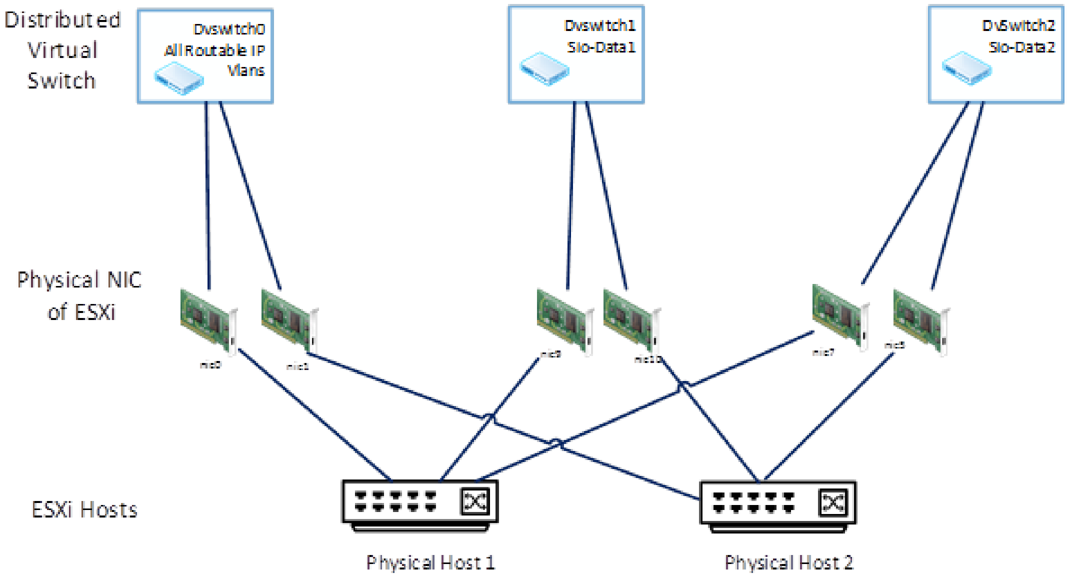
6.1.1.4.3 NIC Team Implementation Diagram

The following diagram shows the NIC teaming configuration implemented for the SI Platform environment. This may vary slightly depending on the hardware capabilities. The variation would combine the vMotion and FT port groups onto the ESXi Management DvSwitch.

The below diagram (6.5) depicts an exploded view of the Physical connections between the Network Interface Crds and the ESXi hosts, which provides a redundant path for the Distributed Switch configured for each function.

In the below diagram taking an example of the distributed switch DVswitch0 (Routable IP's) the Distributed switch (RoutableIPs) are configured on redundant Network Interface Cards from separate ESXi Hosts.

Figure 6-4: NIC Teaming Diagrams for ESXi Hosts



6.1.1.4.4 Naming Conventions

The port group names are configured common across hosts to support virtual machine migration and failover. Common port group names are required by vSphere features such as vSphere vMotion, vSphere HA, and vSphere DRS.

Port group names describe their use, such as IP Storage, Management, or vSphere FT which simplifies the creation of common port group names across hosts.

<ENV><FUNCTION>

ProdSupport-DB

6.1.1.4.5 Logical Network Design Decisions

The following table lists the logical network design decisions made for this architecture design.

Table 6-20: Logical Network Design Decisions

Decision ID	Design Decision	Design Justification	Design Implication
D32	Jumbo frames will be used in this design in all Storage Distributed vSwitches.	Performance gains.	None
D33	Network segmentation will be accomplished with VLANs and trunk ports.	VLANs are already widely in use.	Activate VLANs on switches.
D34	Distributed virtual switches will be used for all routable IP traffic with Port Groups.	Simplifies configuration efforts.	None
D35	NIC teaming will be configured for all virtual switches.	Redundancy and preventing a single point of failure is a requirement for the design.	More physical NICs required for configuration.
D36	DNS and FQDN will be used for all appliances and integrations.	Best practice.	None
D37	Networks will be named following the convention: <ENV><FUNCTION>	Standardized naming allows for easy location and configuration of networks.	None

6.1.1.5 Shared Storage Design

For the SI Platform, VxRACK Flex 1000 storage will be of two configurations. The management cluster workload will be configured with vSAN (Software Defined Storage) and the compute cluster workload will be configured with ScaleIO/VxFlexOS. All disks configurations are of the same type and size across all the ESXi hosts.

- **Management Storage** – Management workload which hosts the virtual machines for the infrastructure components of the VxRACK are on vSAN and will not be used for any other data.

- **Compute Storage** – Compute workload storage is where the data resides for all Virtual Systems and used for the purpose of the SI Platform Virtual machines only. Only one Protection Domain is configured which includes all the disks from 15 compute nodes which are meant for the compute workload. The ScaleIO/VxFlexOS will be used to manage and control the disks. This will allow admins to measure and get more clarity on the performance and capacity of the storage for the SI Platform. Separate data stores will be created for each of the environments to segregate and avoid any corruption of data, for example, data store for DEV and PROD will be created separately and so on for other environments. Each data store will have DRS enabled for performance and capacity. If the capacity needs to be increased, additional new data store will be carved out of the storage pool and assigned it to the data store cluster for the respective environment.

6.1.1.6 Storage System LUNs Size, Data Store Size, and Naming Convention

For the Compute cluster, ESXi hosts that require access to the same shared LUNs will be grouped and the LUNs assigned to the cluster.

- LUNs will be configured as a minimum of 4TB, in such a way that minimum amount of space per disk is not wasted.
- Each VMware volume will have two redundant storage paths.
- Thick provisioning will be used for optimum performance from ScaleIO/VxFlexOS.
- Thin provisioning will be used within VMware data stores.

Storage naming and sizes are different in each environment. The naming standards are outlined below, all data store are local to the ESXI hosts and managed via ScaleIO/VxFlexOS.:

Naming Convention:

ENV- the type of disk – the name of storage – number of LUN.

Table 6-21: Naming Conventions

Naming Conventions						
DEV				**QAT**		
Data store NAME	Size	Total		Data store NAME	Size	Total
DEVAPP01	5TB			QATOS01	1TB	
DEVDB01	5TB			QATDB01	5TB	

Naming Conventions						
DEVDB02	5TB			QATDB02	5TB	
DEVNFS01	1.5TB			QATAPP01	2.5TB	
DEVOS01	1TB			QATNFS01	2TB	
		17.5TB				15.5TB
UAT				**SIT**		
Data store NAME	Size	Total		Data store NAME	Size	Total
UATOS01	1TB			SITOS01	1TB	
UATDB01	5TB			SITDB01	5TB	
UATDB02	5TB			SITDB02	5TB	
UATDB03	5TB			SITNFS01	2TB	
UATDB04	5TB			SITAPP01	2TB	
UATDB05	200GB					15TB
UATAPP01	5.5TB					
UATNFS01	4TB					
		31TB				
PRODSUPPORT				**PRODPATCH**		

Naming Conventions						
Data store NAME	Size	Total		Data store NAME	Size	Total
PRDSOS01	1.7TB			PRDPOS01	1.7TB	
PRDSAPP01	9.5TB			PRDPAPP01	9.5TB	
PRDSAPP02	8.5TB			PRDPAPP02	8.5TB	
PRDSAPP03	800GB			PRDPAPP03	125GB	
PRDSDB01	7TB			PRDPDB01	7TB	
PRDSDB02	7TB			PRDPDB02	7TB	
PRDSDB03	6.7TB			PRDPDB03	6.5TB	
PRDSDB04	6.5TB			PRDPDB04	6.5TB	
PRDSNFS01	9TB			PRDPNFS01	9TB	
		56.7TB				55.9TB
PRODUCTION				**SATELLITE Server**		
Data store NAME	Size	Total		Data store NAME	***6.1.1.7 Size***	Total
PRDOS01	2.2TB			RHSATOS01	30GB	
PRDAPP01	9.7TB			RHSATAPP01	500GB	
PRDAPP02	8.5TB					
PRDAPP03	775GB					

Naming Conventions						
PRDDB01	12.6TB					
PRDDB02	12.6TB					
PRDDB03	12.2TB					
PRDDB04	12TB					
PRDDB05	5.5TB					
PRDDB06	5.5TB					
PRDDB07	5.5TB					
PRDDB08	5.5TB					
PRDDB09	5.5TB					
PRDDB10	5.5TB					
PRDDB11	5.5TB					
PRDDB12	5.5TB					
PRDNFS01	18.5TB					
		133TB				

6.1.1.8 Virtual Machine Design

Virtual Machines for the SI Platform will be hosted on the compute cluster for virtual workload and installed from a VMware template designed with RedHat 7.5 version of operating systems, it will host all the database and application that are designed for the SI Platform infrastructure. All Virtual System will have a dedicated vNIC to configure the IP address for network connectivity, the disk space will be provisioned using the data store assigned.

All database and applications will be installed on their own respective file systems created from the respective data store, there will be no data of the application or database residing on the operating systems file system other than the temporary space that the database or application used to start up, shut down or run the same, upon restart of the system these spaces will be reclaimed by the Operating system.

Please see below an example of the file system layout for the systems.

Common File System Across all Servers:

Root

Temp

Swap

Opt

Export

/export/monitoring

/export/NetBackup

/export/appl

Additional File Systems for Database Server:

Db/db00*

/export/appl/oracle

Additional File Systems for Application Server:

/export/appl/(App_Name)

CPU and Memory resources will be hardcoded to each virtual system with the benefit of DRS being enabled. Each system will get its own local swap space to accommodate memory paging of processes, swap will not be configured on shared storage until and unless absolutely necessary and the system's performance is questionable.

Each Virtual host with an operating system will be installed with VMware tools to manage the VM, for efficient memory management, graceful virtual machine shutdown, allows the proper use of keyboard and mouse on vSphere console, and apply the necessary hardware drivers to ensure proper functionality.

All update patches or Operating System bugs will be applied using the RedHat satellite Servers; this is a product of RedHat, which eases the maintenance and operational tasks.

The following table shows IP Minimum and Maximum VM system sizing and provides a broad range of different-sized workloads for each of the applications.

Table 6-22: Virtual Machine and Guest Operating System Sizing and Specifications

Item	Minimum Virtual Machine	Maximum Virtual Machine
CPU	1	16
RAM	4 GB	64 GB

Please refer the sizing document attached for each of the environment.

Table 6-23: Virtual Machine Specification for the SI Platform

Environment	# Servers	Memory (GB)	CPU	Server Storage (TB)	NFS Storage (TB)	Archival & Backup Storage (TB)	Total Storage (TB)
Production	73	3276	518	112.185	18.5	112	242.685
Prod-Support (Prod-Like)	0	0	0	0	0	0	0
Prod-Patch/Performance (Prod-Like)	58	2064	348	45.21	9	71	125.21
UAT(Non-Prod)	34	896	134	27.525	4.1	18	49.625
SIT (Non-Prod)	32	424	62	12.875	2.1	18	32.975
QAT (Non-Prod)	32	520	70	13.175	2.1	18	33.275
Development(Non-Prod)	30	832	122	15.725	2.1	8	25.825
Total	**259**	**8012**	**1254**	**226.695**	**37.9**	**245**	**509.595**

6.1.1.8.1 Virtual Machine Design Decisions

For this design, NM HSD has made the following decisions, described in this table.

Table 6-24: vSphere HA Design Decisions

Decision ID	Design Decision	Design Justification	Design Implication
D38	Virtual Machine hardware version 13 or later will be used for VM's.	New install	None
D39	VMware tools will be on all virtual machines.	Adds appropriate drivers and enhances the operating system.	None
D40	Shares and reservation limits are not used.	Although good for specific use cases these settings can	DRS is more intelligent than shares and reservations, it does not get limited to the servers

Decision ID	Design Decision	Design Justification	Design Implication
		all limit the scalability of the host.	or need a higher limit configured though the option available.
D41	One Virtual Machine disk will be used for the OS and one for each Database and Applications.	This allows simplification of backup infrastructure.	None
D42	Virtual Machine Disks are stored on shared storage with virtual machine files.	This allows for a simplified management structure for virtual machines.	None
D43	Virtual machines will store the virtual swap file in the default location. In the same data store where the VM is stored.	Using the default location reduces complexity. Host Profiles do not have a configuration option for Host Swap File location.	This reduces complexity and simplifies the configuration.

6.1.1.9 Monitoring and Logging

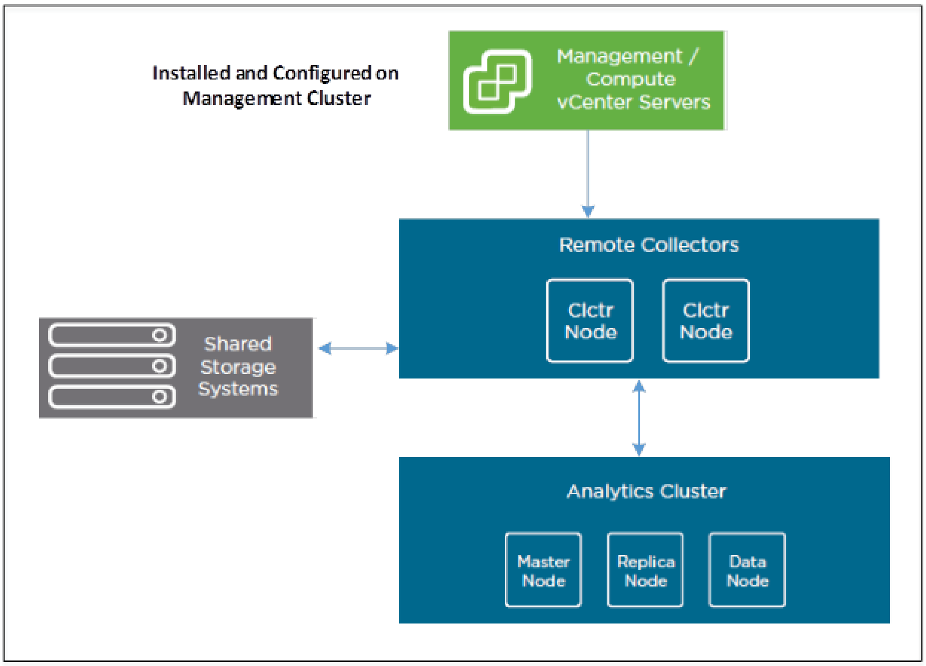
Vision & vROPS achieve the monitoring of all the components of the VxRack and the virtual machines. Vision will monitor the hardware and vROPS will monitor the virtual machines for the SI Platform. Vision integrates converged infrastructure information into VMware vRealize Operations dashboard and provides a single pane of view for both hardware and software infrastructure. vROPS consists of workflows that monitor the below line items and sends the SMTP notification (TBD) to the existing Turning Point ticketing system:

1. Uptime of server or a ping test.
2. File system usage.
3. Processes that should be running.
4. Performance stats from virtual systems will be collected for performance fine-tuning.
5. Error messages logged into the messages file.
6. Capacity.

In addition to the above, more workflows can be developed as the need arises.

The diagram below gives an overall picture of how vROPS are deployed. The design integrates solutions for compute, storage, and network. It has the capacity of communicating with vSAN. The remote collectors collect the logs from the analytics server to run an analysis of the logs collected. The analytics cluster monitors and performs diagnostics across the SI Platform using remote collectors and solution management packs.

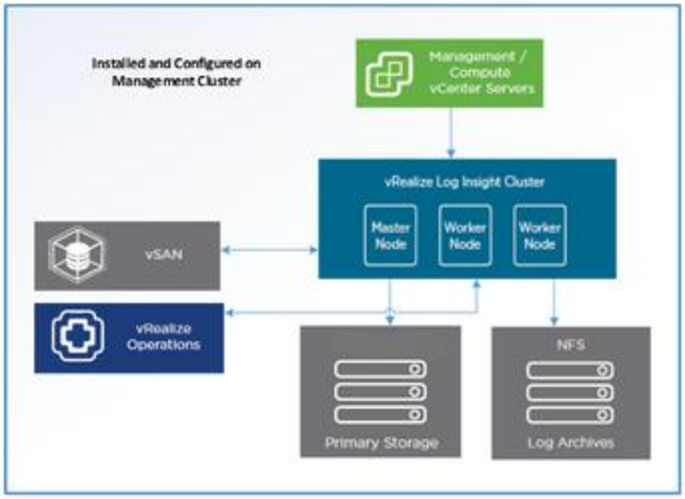
Figure 6-5 vROPS Deployment Overview



In addition to the above, log insight will be deployed which will help in logging all the events generated by the infrastructure components. The log insight has a tight integration with vROPS by providing a single pane of view from hardware and software. Log-insight will capture and will send SNMP notifications (TBD) to the existing Turning Point ticketing system, for any critical issues that arise in the infrastructure. All logs will be retained for 180 days and rotated to avoid file system usage issues.

The log insight is deployed on the management and compute VCenter's and deployed with a master node and two worker nodes model. It will collect logs from vSAN, and vROPS, and writes to the attached data store and archived to the NFS storage mount as needed to run reports.

Figure 6-6: Log-Insight Deployment Overview


Installed and Configured on Management Cluster
Management / Compute vCenter Servers
vRealize Log Insight Cluster
Master Node
Worker Node
Worker Node
vSAN
vRealize Operations
NFS
Primary Storage
Log Archives

- vRealize Log Insight cluster of three nodes, enabling continued availability and increased log ingestion rates.
- vRealize Log Insight collects and analyzes log data across the domain using the syslog protocol and the ingestion API.
- vRealize Log Insight also integrates with vRealize Operations Manager to facilitate root cause analysis.

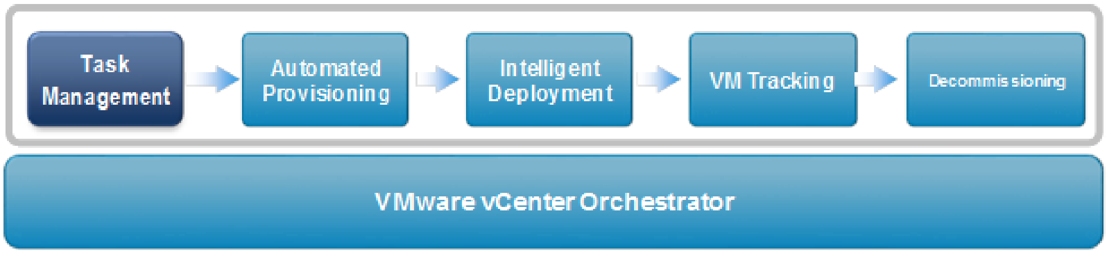
This configuration will change over time.

6.1.1.10 Automation & Orchestration

vRealize Automation and vRealize orchestrator will be installed on the management cluster of the virtual infrastructure in order to provision machines in the compute cluster. Machines are provisioned in the compute cluster and are segregated based on each environment workloads. Authentication and authorization will be through Active Directory (NMHSD.LCL) for single sign-on. vRealize suite will use embedded appliance database for maximum performance by deploying agents in the same virtual data center as the endpoint to which they are associated.

vRealize Orchestration embedded version will be deployed and comes with a few inbuilt plugins to vCenter, Active Directory, etc. Additional packages and plug-ins will be installed as needed basis for integration with the current environment in order to use with the in-built workflows.

Figure 6-7: VMware Automation & Orchestration Overview



6.1.1.11 Security

The virtualization platform is the inherently secure environment and all traffic flowing in and out of the SI Platform at NM HSD datacenter is via the Palo Alto firewall. All Virtual machines with a routable IP generating traffic for all the network segments will have to pass through the Firewall as it is the default gateway.

All Virtual machines will be configured with a Unique IP and name in the AD-DNS system.

Access to the ESXi hosts, vCenter, Virtual hosts is through AD groups defined in the Active Directory database of NM HSD, in addition, all ESXi host will have the Lockdown mode enabled which will restrict access to the ESXi hosts directly which hosts all the virtual machines.

Data from a hardware perspective will be confined to the data stores for each of the respective environments, and there will be no cross association or cross-assignment of the disk space across Virtual

Environments. Secure Baseline Configuration Guidelines will be used for securing the SI Platform's virtual environment.

6.1.1.11.1 Encryption and Security Certificates

For the SI Platform, ESXi, and vCenter Server is secured using standard X.509 version 3 certificates to encrypt session information between components using the default signed certificates that come with the VMware6.5.

The following certificates are in use, by default, for VMware virtual infrastructure:

- ESXi Certificates - Used for SSL communication to and from the ESXi host. These certificates are now provisioned by VMware CA by default and stored locally on each ESXi host.
- Machine SSL Certificates - Used for communicating to and from vCenter Servers and Platform Service Controllers. Unlike previous versions of vCenter Server, all communication goes through the reverse proxy, and therefore a single certificate can be used. These certificates are provisioned by VMware CA and stored in the VMware Endpoint Certificate Store (ECS).
- Solution User Certificates - Used by all solutions and services added to vCenter Single Sign-On for inter-component communication. These certificates are provisioned by VMware CA and stored in VMware ECS as well.
- vCenter Single Sign-On Signing Certificate - Root certificate used to sign all certificates provisioned by VMware CA. This is provisioned during the installation of the Platform Services Controller and is stored on the host file system.

6.1.1.11.2 Security Design Decisions

For this design, NM HSD has made the following decisions, as described in the table below.

Table 6-25: Security Design Decisions

Decision ID	Design Decision	Design Justification	Design Implication
D44	ESXi hosts and vCenter Server are secured by permitting access via AD groups.	Limiting user access and configuring appropriate security is a must for the virtualized environment.	None
D45	Virtual networks will be secured by using firewalls rules.	None	None

6.1.1.12 Infrastructure Backup and Restore

Backups:

All components of the VxRack which includes the vSphere, VCenter, Management workload, Virtual machine instance will be backed up using the existing NetBackup solution. For the SI Platform, NM HSD approved NetBackup software will be used to install on the system as needed and backups will be scheduled as follows.

1. Full backup will be performed every weekend.
2. Incremental backup will be performed every day of the week.

and for Management components (vSphere) it will be an appliance level backup.

Restore:

Restore will be performed in case of data corruption or an unrecoverable system crash. Restore of data on the virtual system will happen using NetBackup and for the vSphere, it will be a restore of the appliance.

6.2 Software Detailed Design

The HHS 2020 enterprise architecture adheres to service-oriented architecture (SOA) principles to distribute its Medicaid Management Information Systems (MMIS) into loosely coupled and highly reusable vendor-, product-, and technology-agnostic, easily discoverable and Interoperable services, communicated over the network through well-defined interfaces abstracting the implementation details.

The SI Platform will be implemented using Enterprise Service Bus (ESB) architecture adhering to SOA standards and principles using Oracle Fusion Middleware Stack. The detailed design and logical components of the SI Platform ESB and the Enterprise Shared Services (ESS) are explained below in detail.

6.2.1 ESB Platform Design

The SI Contractor will implement the SI Platform using the Enterprise Service Bus (ESB) based on standards and principles of Service Oriented Architecture (SOA). An ESB is critical to creating a scalable SOA backplane that provides the foundation for virtualization, policy enforcement, monitoring, event visibility, service discovery and effective governance to an enterprise.

The following are the architecturally significant requirements pertaining to ESB platform.

- **Abstraction** – Consumers will communicate with services via messages routed to the appropriate endpoints by the ESB. Services will hide their implementation details from other services

- **Loose Coupling of Services** – Loose coupling is achieved through abstracting and resolving the differences between two or more HHS 2020 modules in order to facilitate seamless integration. The SI Platform helps mediate the differences along the following lines:
    - Services are independent of one another and are stateless and idempotent where possible.
    - Service invocations should be bound at run-time, not a compilation of software.
    - The existing interfaces change as little as possible and the services support backward-compatibility with existing APIs.
    - Services should be invoked and should reply to requests with DGC-approved shared schemas.
- **Change Management** – Change management refers to the ability to change various attributes of service without impacting the operation of the system. Change Management through the ESB provides the following:
    - Add new consumers to an existing service.
    - Change service implementation.
    - Change policies associated with a service.
    - Add new versions of the service.
    - Add a new service provider without affecting the existing service consumers.
- **Discoverable** – Services will store metadata about their Contracts and Policies in a repository where the services are discoverable by other services at design, test and run-time.
- **Composability** – Supports orchestrations of coarser-grained services from finer-grained services

The SI Platform is an encapsulation of tools, technologies, and services which will help to integrate the UPI portal, the BPO modules, shared services, internal and external modules. The SI Platform enables integrating modules to communicate as if the SI Platform itself provided the services without any details of the implementation details.

For example, the Social Security Administration (SSA) can consume web services hosted by the SI Platform to exchange benefit information and SSA can provide web services used to determine constituent eligibility.

The ESB exposes and coordinates the functionalities of the loosely coupled modules as enterprise services, and performs functions, such as routing messages, transforming message formats, transforming message structures, and translating transport protocols. The SI Platform ESB architecture is consistent with HSD’s HHS 2020 enterprise architecture and includes best practices regarding an efficient and sustainable approach to implementing the State’s 2020 vision.

A module connects to the SI Platform’s ESB using one or more connections. For example:

- The portal connects to the ESB using web services.
- The ESB connects to a legacy application using a database adapter.
- The MFT application connects to the ESB using File Transfer Protocol (FTP) and web services.

6.2.1.1 Alignment with PADU Approach and MITA Technical Strategy

The ESB will be built on Oracle Fusion Middleware COTS which is the preferred technology stack as identified in the NM HHS 2020 MITA Technical Strategy document and implement a standards-based interface to the other modules. Since Oracle Fusion enables to achieve maximum configuration and minimum effort for developing WSDLs, XSLT transformations, the alignment with PADU for this framework is deemed as “Preferred” level.

Oracle Fusion meets the State’s need for a highly flexible, scalable, SOA framework that can enable loose coupling of HHS 2020 enterprise modules, workflows, business rules management and integration of disparate data formats and technologies. Oracle Fusion provides a library of connectors and Application Programming Interfaces (APIs) that can be leveraged to connect to HHS 2020 legacy modules and interfaces.

6.2.1.2 ESB Platform Layers

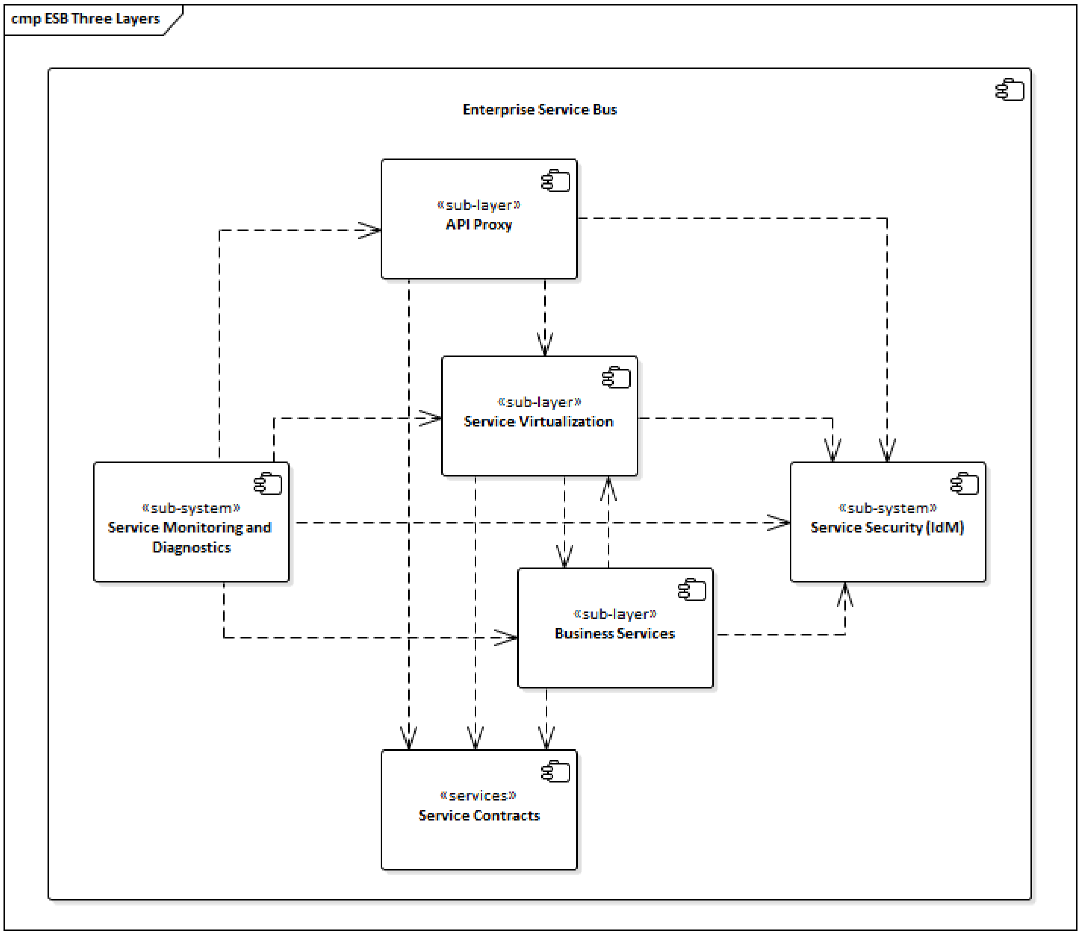
The SI Platform ESB architecture model is divided into the following three layers.

1. API Proxy
2. Service Virtualization
3. Business Service

The three layers are governed by security and include centralized monitoring, logging, and diagnostics capabilities within the SI Platform. This categorization provides a separation of IT concerns and business logic concerns and allows greater agility and flexibility in the architecture.

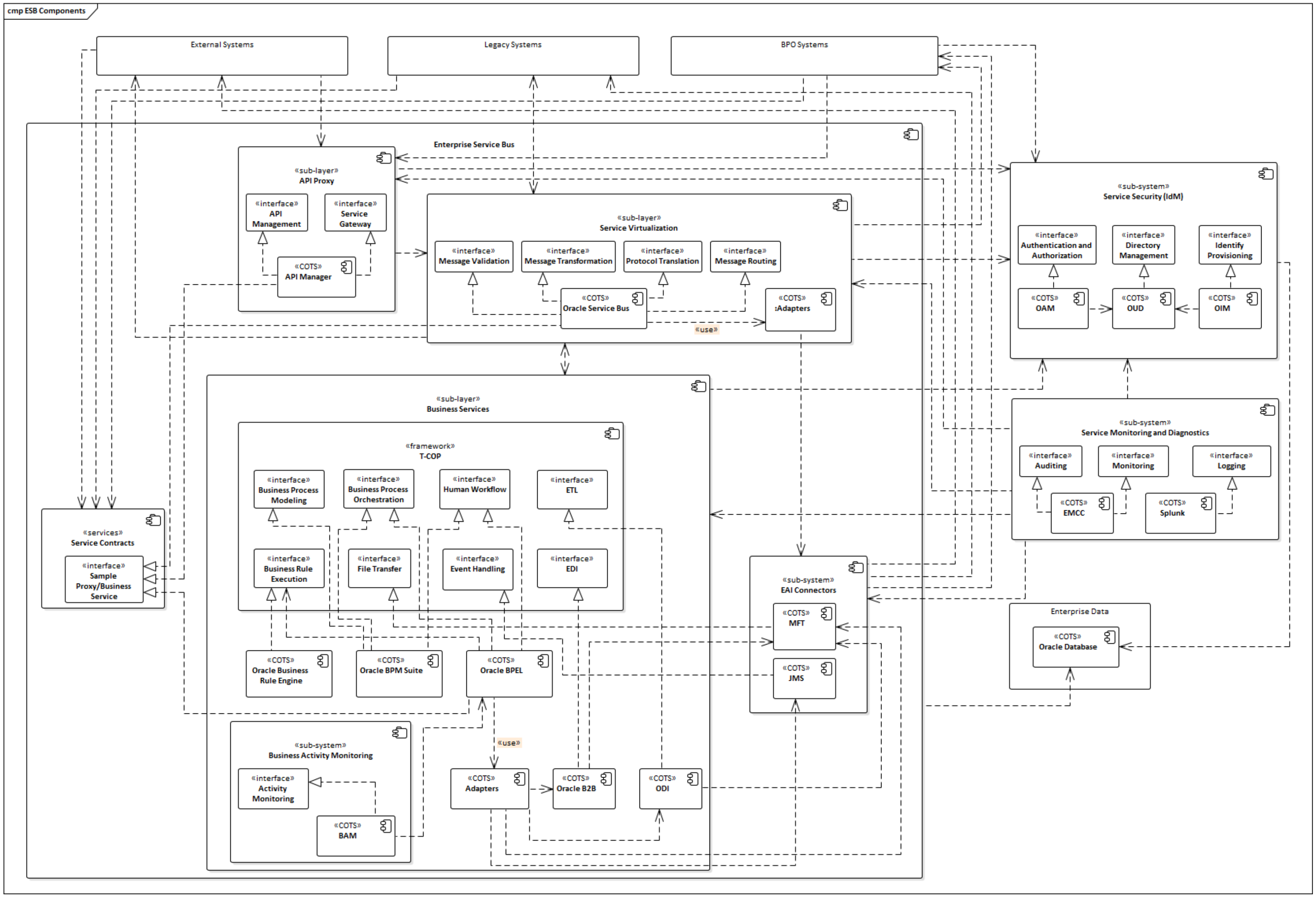
The following figure shows the UML package diagram of the three layers of the SI Platform.

Figure 6-8: ESB Platform Layered Design


cmp ESB Three Layers
Enterprise Service Bus
«sub-layer» API Proxy
«sub-layer» Service Virtualization
«sub-system» Service Monitoring and Diagnostics
«sub-system» Service Security (IdM)
«sub-layer» Business Services
«services» Service Contracts

The following figure shows the logical model of the ESB platform depicting the connections and the ESB functions that facilitate the communication between the new and existing systems which can connect directly to the ESB or to other SI Platform components, such as web services, MFT, and the ETL.

Table 6-26: Components of ESB Platform


cmp ESB Components
External Systems
Legacy Systems
BPO Systems
Enterprise Service Bus
«sub-layer» API Proxy
«interface» API Management
«interface» Service Gateway
«COTS» API Manager
«sub-layer» Service Virtualization
«interface» Message Validation
«interface» Message Transformation
«interface» Protocol Translation
«interface» Message Routing
«COTS» Oracle Service Bus
«COTS» :Adapters
«use»
«sub-system» Service Security (IdM)
«interface» Authentication and Authorization
«interface» Directory Management
«interface» Identify Provisioning
«COTS» OAM
«COTS» OUD
«COTS» OIM
«sub-layer» Business Services
«framework» T-COP
«interface» Business Process Modeling
«interface» Business Process Orchestration
«interface» Human Workflow
«interface» ETL
«interface» Business Rule Execution
«interface» File Transfer
«interface» Event Handling
«interface» EDI
«COTS» Oracle Business Rule Engine
«COTS» Oracle BPM Suite
«COTS» Oracle BPEL
«sub-system» Business Activity Monitoring
«interface» Activity Monitoring
«COTS» BAM
«use»
«COTS» Adapters
«COTS» Oracle B2B
«COTS» ODI
«services» Service Contracts
«interface» Sample Proxy/Business Service
«sub-system» Service Monitoring and Diagnostics
«interface» Auditing
«interface» Monitoring
«interface» Logging
«COTS» EMCC
«COTS» Splunk
«sub-system» EAI Connectors
«COTS» MFT
«COTS» JMS
Enterprise Data
«COTS» Oracle Database

Note: The interfaces like Message Validation, Message Transformation, Human Workflow, etc. depicted in the above figure are for representational purpose only. These are the capabilities of the tools which are internal to the tool itself and will not be exposed as services to other components. The service contracts are the actual services/proxies for the service integration which will make use of the capabilities depicted in the model above.

6.2.1.2.1 API Proxy

The API Proxy layer provides API management and API access control functionalities to the HHS 2020 internal and external modules. As the number of SOAP and REST web service APIs that platform produces and uses increases, the management and visibility of these APIs become increasingly important across the HHS 2020 modules. The API Proxy layer is responsible for providing visibility to the APIs that are available for use and control access to these APIs at runtime.

The following are the architecturally significant requirements addressed in this subsection:

- Consumers will communicate with services via messages routed to the appropriate end points by the ESB and abstract the implementation details from other services.
- Services will be discoverable through the IP.
- Enforce versioning of services and messages and the proper retirement of outdated services.

The API Proxy layer provides capabilities to publish, discover, share, monitor and manage services. Any external HHS 2020 module including UPI portal can subscribe to the catalog of APIs published by SI Platform through the API Proxy layer. The following are the functionalities that API proxy layer offers to any integrating modules:

- Acts as the first line of defense in the DMZ layer by adding comprehensive enterprise-grade security including transport-level security, message-level security, SAML, fine-grained authentication, identity management, client throttling, API aggregation.
- Stores metadata about their Contracts and Policies of the services in a repository where the services are discoverable by other services at design, test and run-time.
- Addresses IT concerns around design time and runtime visibility, governance of assets and policy enforcement.
- Tracks and monitors the usage of APIs at runtime and provides statistics about API performance and activity.
- Captures key SLA statistics, average service response times, total message count, error message count and the number of subscribers.

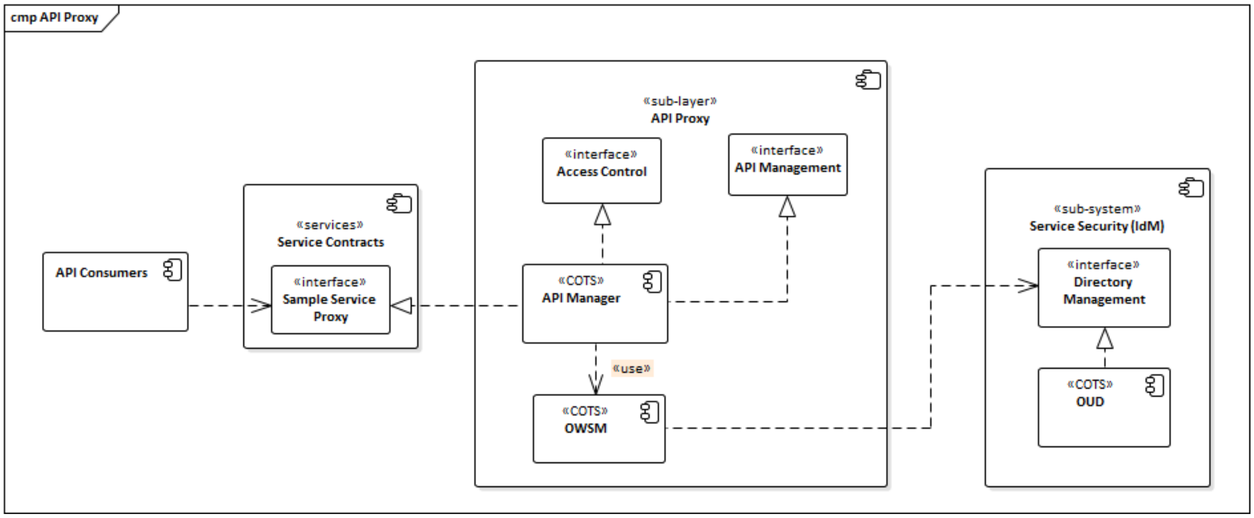
The following are the logical and physical components that represent the API Proxy layer of the SI Platform.

Table 6-27: API Proxy Layer and Physical Platform Tool

Name	API Manager
Description	API Manager is the logical component that performs API management and API access in the API Proxy layer.
Platform Tools	• Oracle API Manager 12c (12.1.3)
Capabilities	• API Management ○ Allows users to easily create APIs ○ Provides the ability to secure APIs ○ Enables easy API editing and publishing ○ Facilitates the discovery and use of APIs ○ Allows to add/edit metadata for providing the technical and non-technical information need to, understand, and use of the APIs. ○ Allows integrating components to discover and subscribe to APIs ○ API performance monitoring • API Access Control ○ Adds transport-level security, message-level security, SAML, fine-grained authentication, Identity Management, etc.

Below is the high-level component diagram of Oracle API Manager and how it interacts with other components to provide API management and access control.

Figure 6-9: API Proxy High-Level Component Diagram



6.2.1.2.2 Service Virtualization

Service virtualization layer of the SI Platform provides the abstraction of a physical service catered by the ESB through virtual endpoints or proxy. With service virtualization, the virtual endpoints interact with the service consumer (i.e., the service or application making a request). This layer will utilize the proxy pattern for the services endpoint, thereby completely decoupling the service consumer from the service provider. Decoupling enables location transparency and ability to introduce new versions of a service without impacting all clients of the service.

The following are the architecturally significant requirements addressed in this subsection.

- Abstraction via Use of Policies - Policies hide implementation details and constraints of services from outside service clients. Details of a service's implementation are hidden completely from all service clients
- Message will adhere to Shared/Canonical Schemas and validation for both schema and content as well as context-based routing will occur with assistance from a BRE.
- ESB will provide encoding (XML, JSON) and protocol (HTTP, JMS) translations to handle messages serving clients of varying technological capabilities and needs.

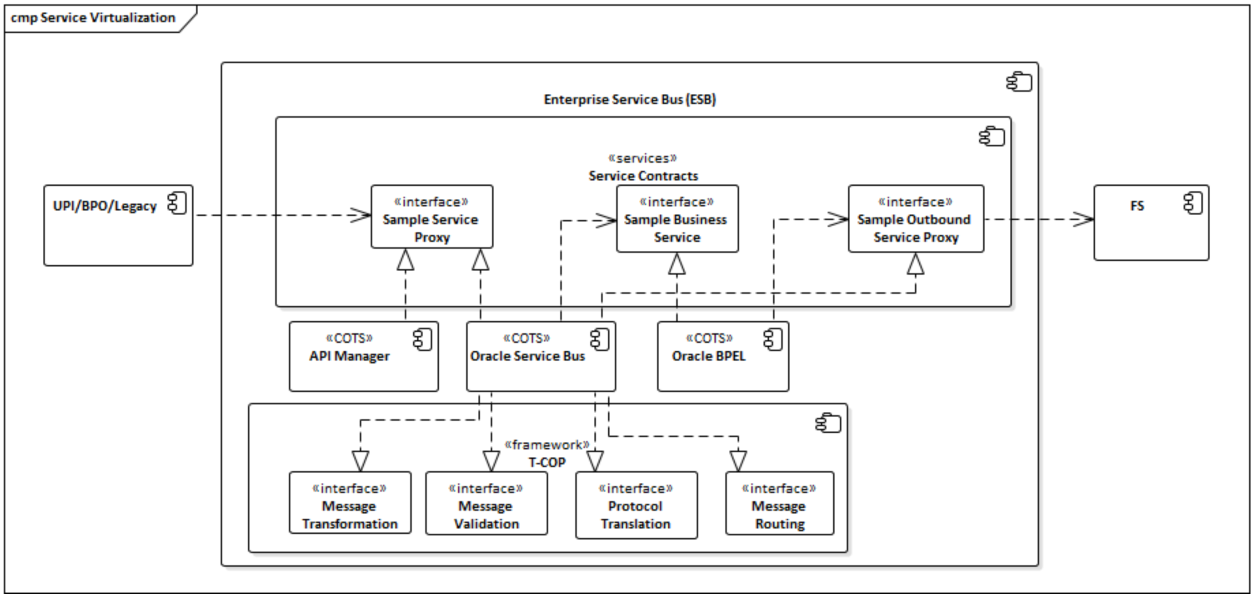
The virtualization layer provides virtual service endpoints to service consumers (i.e., the service or application making a request) and providers (i.e., the service or application that provides functionality) as shown in the below figure. That is, the web service consumers local to the network or located in the same data center makes a request directly to the Service Virtualization layer (proxy) instead of API Proxy layer which is in the DMZ. Other HHS 2020 modules which are not local network as of the ESB or located in different data center and external systems will make service request through API Proxy. The ESB sends the request to the service provider. The service provider sends a response to the virtual endpoint (provided by the ESB) for the consumer, and the ESB sends the response to the consumer.

Along with the abstraction of the actual service, the Service Virtualization layer also provides the following critical functionalities of the ESB to help integration HHS 2020 modules:

- Message Transformation
- Protocol Translation
- Message Validation
- Message Routing

The following figure shows the virtual endpoint design of ESB.

Figure 6-10: ESB Virtual Endpoint Design



The following are the logical component that represents the Service Virtualization layer and physical components that are used to implement the desired solution.

Table 6-28: SOA Service Bus Components

Name	SOA Service Bus
Components	Enterprise Service Bus (ESB)
Description	• SOA Service Bus is the service bus of the SI Platform. • It acts as a virtualization layer to enterprise system connectivity. • It takes care of message security, translations, transformations, and validation. • OSB is a service bus with rich messaging and security features used for loose coupling, location transparency, and isolation of business processes from a change of service endpoints.
Tools	Oracle Service Bus (OSB)
Capabilities	Service Virtualization, Web Services Security, Transform/Routing, Messaging, Message Validation.

Message Transformation:

Message transformation is performed to facilitate the integration between HHS 2020 modules where message formats are different between HHS 2020 message producing modules and message consuming modules.

In most cases, the messaging formats of the consumer module and the provider module differ, and the provider cannot process the message in the consumer's native format. For example, if an HHS 2020 consumer module can only send a message in JavaScript Object Notation (JSON) format and the provider can only process Extensible Markup Language (XML) formatted messages, the provider module cannot process the message as shown in the Figure below.

ESB can aid in system integration between these modules by acting as a proxy in between whereby message received from HHS 2020 module system can be converted to the message format accept by message producer module and vice versa. Transformation to and from a native format to an approved canonical format will be implemented. XML transformation will be achieved using XSLT and XQUERY in the OSB layer.

Protocol Translation:

Protocol Translation is performed to facilitate the integration between HHS 2020 systems when message producing module doesn't communicate using the protocol which is acceptable by the message receiving module.

The SI Platform's ESB component support multiple protocols, message producer can communicate with ESB which in turn converts the protocol according to the consumer as shown in the figure above.

The protocol translation is very beneficial as the consumer and provider do not need to make significant changes to their native code to support the integration because the ESB can translate protocols for them. In addition, changes to the consumer communication protocol do not require subsequent changes to the service provider; changes to the provider communication protocol do not require changes to the service consumer. The protocol translation will be implemented using REST and SOAP adapters in the OSB virtualization layer.

Message Validation:

Message Validation ensures that all the message received by the SI Platform are validated for syntax and semantics imposed by the approved message schema definition. This is primarily used for XML based messages. Any message coming into the SI Platform will be validated before it is further processed. If the message validation fails, then an error is generated and send back to the consumer with error details. The error details will have a consolidated list of validation errors. This helps maintain the message integrity within the SI Platform and avoids bad data from being sent to MMISR BPO Modules and other systems within the HHS 2020 ecosystem.

Incoming or outgoing messages will be validated against XSD using the validation action. In the case of validation failure, an error stage will be used to send the error details back to the consumer. In addition, data level validation will also be performed in the OSB layer depending on the requirement.

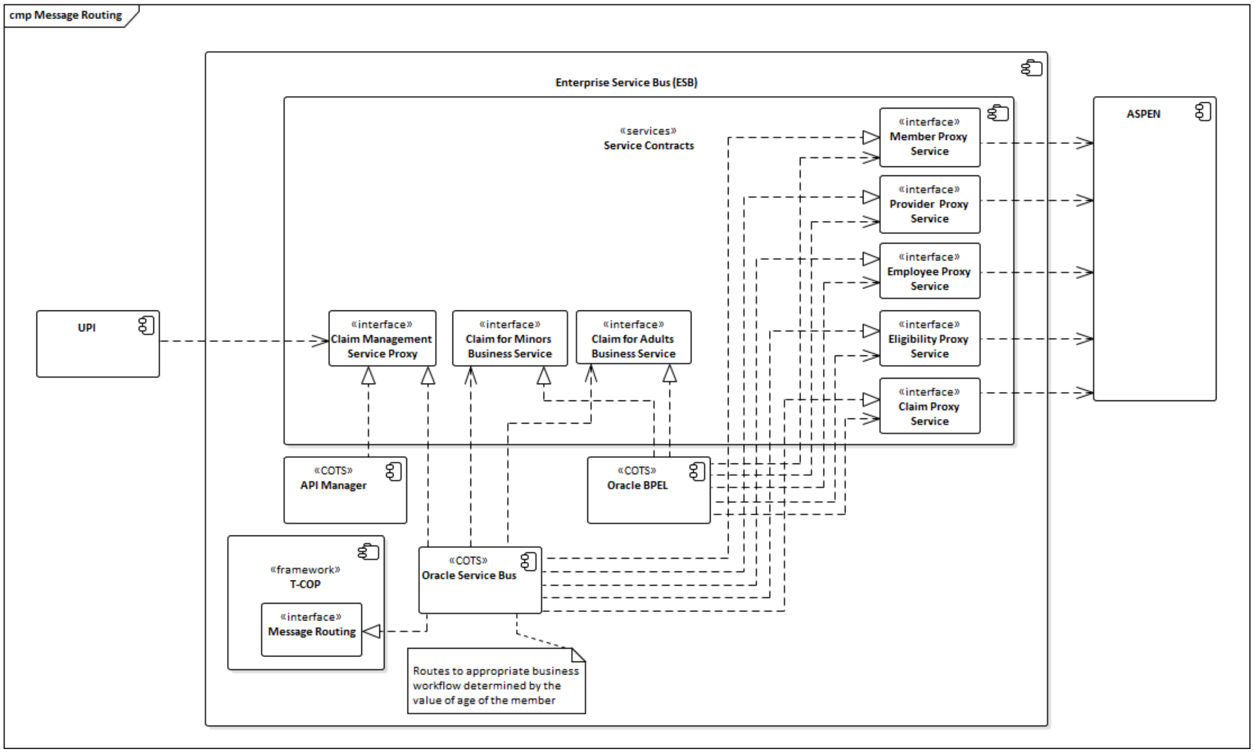
Message Routing:

Message routing and service integration are the core functionalities of the ESB platform. Message routing deals with redirecting of messages received from any system to one or more of the multiple systems and delegating the response back to the request initiator.

When the service consumer sends a request to the ESB, the consumer identifies the operation to perform and the data elements required to fulfill the operation as shown in the figure below. When the ESB has a one-to-one association between the service provider's virtual endpoint and real endpoint, the ESB routes the message to that service provider destination.

However, the ESB may have a one-to-many association between the virtual endpoint and specific provider endpoints. When the ESB has multiple destinations for the same virtual endpoint, the ESB determines where to route the request based on the multiple patterns as listed in the subsection on ESB Integration Patterns.

Figure 6-11: ESB Message Routing



6.2.1.2.3 Business Service

The Business Service layer host services that implement application logic from discrete business activities to complete business processes. It deals with concerns associated with process automation and business logic development. The business processes implemented in this layer spans across multiple HHS 2020 modules and shared services and will consist of multiple data and application services.

This layer also addresses business level visibility by integrating with business events for event-driven SOA. The following functional capabilities are addressed by this layer:

- Business process orchestration.
- Business exception management.
- Human interaction with business processes.
- Business rules modeling.

- Business Activity Monitoring interactions with other modules.

The following are the architecturally significant requirements addressed in this subsection:

- Enable the creation and evolution of composite applications through the use of Business Process Management (BPM), Workflow, Orchestration, and Business Rules Engine (BRE) tools.
- To externalize business logic controlling various long-running (process orchestrations), short running (service compositions) and system-level (ESB message validation and routing) functionalities HHS 2020 enterprise makes use of commercial BRE technology.
- Implement reusable, more adaptable, real-time reporting, analytic, and business intelligence tools to Enterprise users that leverage highly shared, cross-program information.
- Implement systemic security, auditing, logging, application performance management, and Business Activity Monitoring (BAM).
- Implement and manage common services, such as logging/auditing services, event handling, data transformation and mapping and message and event queuing and sequencing.

Any application logic delivered through the business service layer will be implemented using Oracle Fusion Middleware stack components: Oracle BPEL Process Manager, Oracle BPM Suite, Oracle Business Activity Monitoring (BAM), Oracle Business Rules Execution (BRE), adapters and Oracle Data Integrator (ODI). Below are examples of use cases that require a business process orchestration engine using the above Oracle Fusion Middleware components and will be implemented in this layer:

- Service needs to maintain State.
- Service requires complex transaction management.
- Requires multiple transactions.
- Compensation logic required on rollback.
- Short or long-lived process.
- Exception handling requires human workflow.
- Service needs to handle asynchronous callbacks reliably.

The following are the logical and physical components that represent the business service layer and implement the desired solution as per the capabilities defined above in this section.

Table 6-29: Business Service Layer

Name	Business Services
Description	• Business Services is the workflow engine of the SI Platform. • The Business Services take care of business process management, the orchestration of services, execute business logic. • These are implemented using Oracle BPEL and Oracle BPM.
Components	Traffic Cop (T-COP) - T-COP is a framework that enables the business workflow engine, business process management, orchestration of services, execution of SI platform-

Name	Business Services
	specific business logic and rules execution, EDI and bulk data processing, event handling and management of file transfers. EDAS – Enterprise Data as Service is a composite service that enables access to the MDM as well as all other data services. EDS – Enterprise Documentation Service is a composite service that enables access to Perceptive Content, the Enterprise Shared System suite. ECS – Enterprise Communication Service is a composite service that enables access to OpenText Exstream, Enterprise Shared System suite. EIAS – Enterprise Identity and Access Service is a composite service that enables access to Service Security sub-system (IdAM). EAVS – Enterprise Address Validation Service is a composite service that enables access to SAP Data Services for address standardization and validation shared service.
Tools	• Oracle BPEL • BPM Suite • Oracle Business Rules • Oracle Data Integrator (ODI) • Managed File Transfer • Oracle B2B • Oracle Business Activity Monitoring (BAM) • Oracle Database
Capabilities	Business Process Orchestration, Business Process Management, Web services (SOAP/Rest), Business Process Management Notation, Business Activity Monitoring, Auditing and Logging, Adapters/Connectors, Web service Security.

The following are the detailed capabilities of the business service layer of the ESB platform.

Business Process Modeling and Service Orchestration:

Service orchestration represents a single centralized executable business process (the orchestrator) that coordinates the interaction among different services. The orchestrator is responsible for invoking and combining the services. Orchestration employs a centralized approach for service composition.

Business process orchestration involves the coordination and central management of process events or integrating modules. Business process orchestration involves multiple modules of HHS 2020 enterprise compiled into a sequence of steps that constitutes a business workflow. The business processes are modeled using BPMN/BPEL. The business processes are identified during the requirement gathering for SI Platform workflow, BPO module integration, and Legacy module integration.

The business services will be implemented using the BPEL. The following diagram demonstrates the orchestration of the above service integration in the SOA composite layer and BPEL.

SOA Composite:

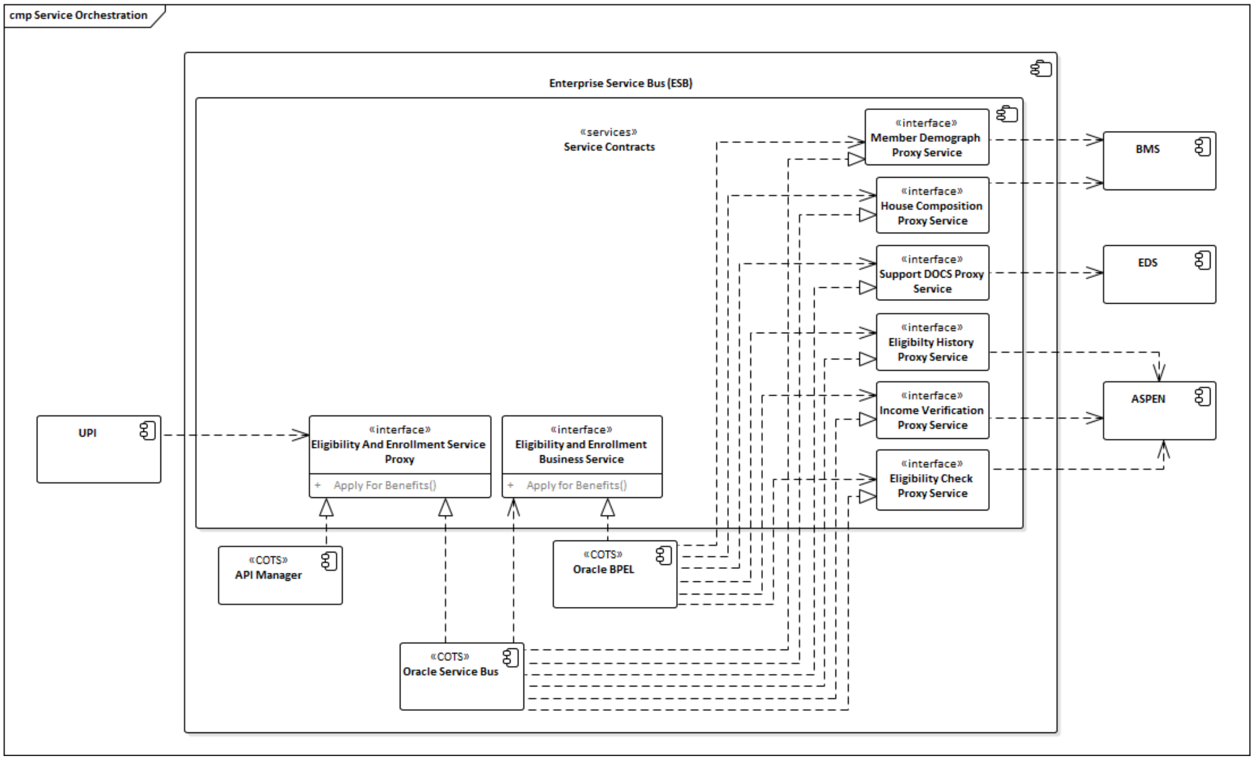
An SOA composite is an assembly of services, service components, and references designed and deployed together in a single application. Wiring between the service, service component, and reference enable message communication. The composite processes the information described in the messages. The service represents the entry point to the composite application, whereas the reference represents a target service to call. Components represent the business logic layer.

BPEL Process (Component of an SOA Composite):

BPEL process contains the logic, can invoke several services, and combine their responses into a single service response. It can also perform transformations using XSLT/XQUERY to translate the request message of the target as required before invoking the target service. Similarly, it can transform the response message received from the invoked target service to the desired format (such as to a canonical format). The business rules (typically exposed as web service) are consumed from the BPEL process, which passes the input to the rule and accepts the rule response inside the same BPEL.

For example, the figure below illustrates how the ESB orchestrates services when a member applies for benefits.

Figure 6-12: ESB Service Orchestration

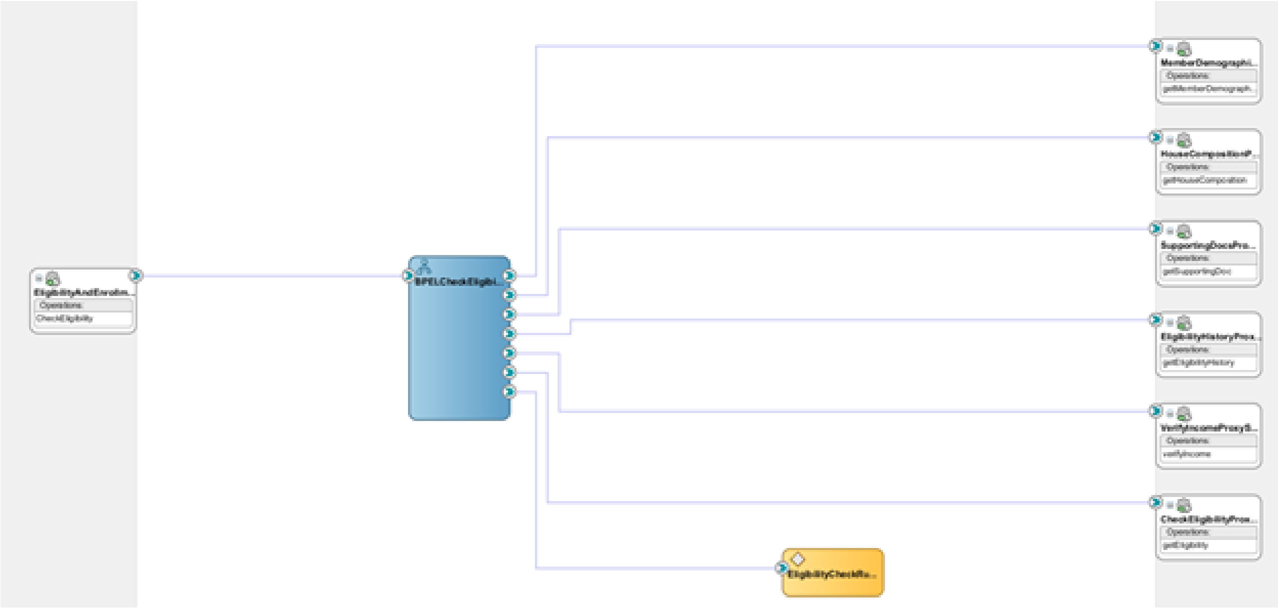

cmp Service Orchestration
Enterprise Service Bus (ESB)
«services»
Service Contracts
«interface»
Member Demograph Proxy Service
«interface»
House Composition Proxy Service
«interface»
Support DOCS Proxy Service
«interface»
Eligibilty History Proxy Service
«interface»
Income Verification Proxy Service
«interface»
Eligibility Check Proxy Service
BMS
EDS
ASPEN
UPI
«interface»
Eligibility And Enrollment Service Proxy
+ Apply For Benefits()
«interface»
Eligibility and Enrollment Business Service
+ Apply for Benefits()
«COTS»
API Manager
«COTS»
Oracle BPEL
«COTS»
Oracle Service Bus

The following are the sequence of steps for member enrolling for benefits. The sequence diagram is depicting these steps can be found in Subsection 9.2.2.

- The member enters all required information in a UPI system's portal and submits the application.
- The portal calls the EligibilityAndEnrollmentAPIService exposed through Oracle API Manager and call reached ESB and waits for a response.
    - The ESB, in turn, breaks the application data into smaller fragments and orchestrates the calling of each of the services (exposed by partner applications) that comprise the EligibilityAndEnrollmentRequest service. The services exposed by partner applications include:
        - Get member demographic information.
        - Get member household composition.
        - Verify member income.
        - Get member eligibility history.
        - Get member supporting documentation.
        - Determine member eligibility.
- The ESB returns a list of potential benefits for which the member is eligible and can be enrolled in the client browser (portal).

The below figure illustrates how different partner services orchestrate in the BPEL business service layer when a member applies for benefits.

Figure 6-13: BPEL Composite Design for Service Orchestration



Human Workflow:

A business process may require a human decision to proceed further. The human workflow component helps in managing processes that require human input as one of the steps for its completion. The following are the features that human workflow service provides.

- Task routing to users, groups, or application roles.
- Deadlines, escalations, notifications, and other features required for ensuring the timely performance of a task.
- Task forms for presentation of tasks to end users through a variety of mechanisms, including a workspace and portals.
- Organization, filtering, prioritization, dispatching rules, and other features required for end users to productively perform their tasks.

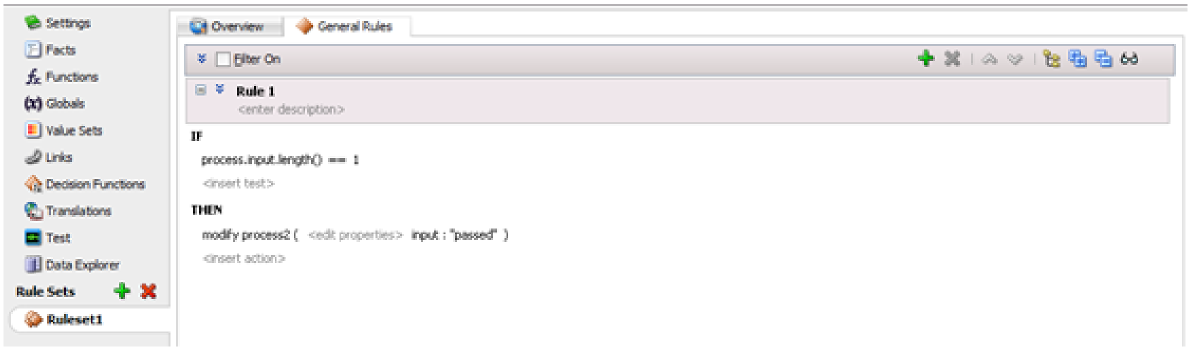
Rule Execution:

The business rule will be integrated with the BPEL service as a part of the business process orchestration. An example of the business rule can be validating the eligibility of a client to enroll in a benefit program. The business rules are declarative and can be configured using Oracle Business Rule Engine (Oracle BRE). By using Oracle Business Rules, a business analyst can change business rules without stopping a business process. Also, externalizing business rules enables process analysts to manage business rules directly, without involving the development process.

Following are the components of a business rule used during design time and runtime:

- *Rule Author and SDK* – These are used as design-time components in the JDeveloper either as a standalone service or from within a composite component as a part of BPEL process orchestration depending on the requirement.
- *Rule Language and Rule Engine* – These are used as runtime components that interact with decision components developed from design-time components. Rules can be implemented and executed in two ways inside a business rule engine. One method is if-then rules, and the other is rules in decision tables.
- *If-Then Rules* – A business rule has an IF part and a THEN part. The IF part tests one or more business terms. If the tests pass, one or more actions are performed in the THEN part. Following is an example and diagram of an If-Then rule created using JDeveloper.

Figure 6-14: Oracle BRE If-Then Rule Design

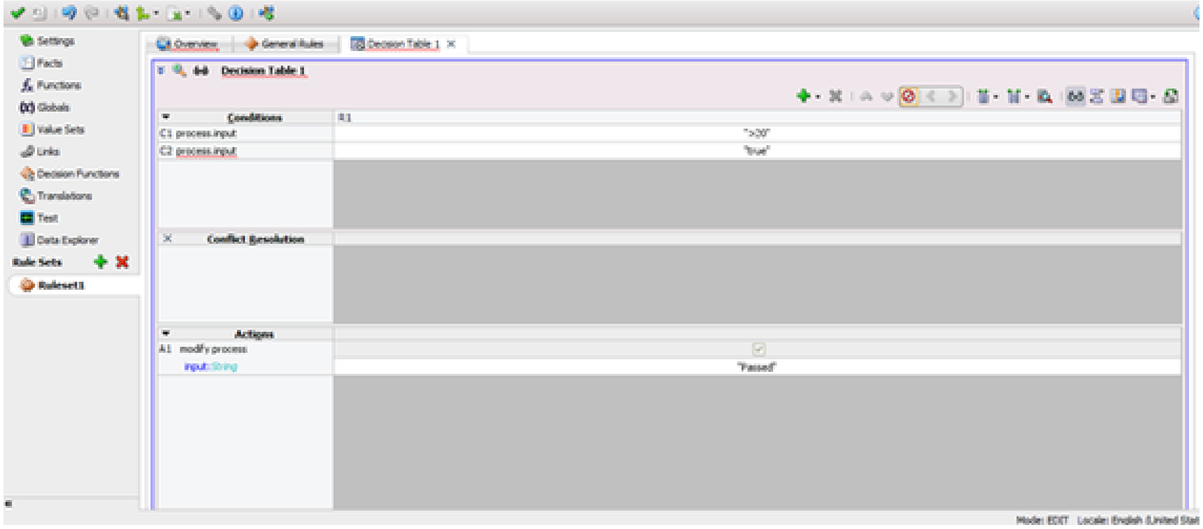


```
Rule 1
<enter description>
IF
  process.input.length() == 1
  <insert test>
THEN
  modify process2 (  <edit properties>  input : "passed"  )
  <insert action>
```

Decision Tables:

A Decision Table is an alternative business rule format that is more compact and intuitive when many rules are needed to analyze many combinations of property values. We can use a Decision Table to create a set of rules that covers all combinations or where no two combinations conflict.

Figure 6-15: Oracle BRE Decision Tables



Conditions	R1
C1 process.input	">20"
C2 process.input	"true"
Conflict Resolution	
Actions	
A1 modify process	✓
input:String	"Passed"

A Decision Table displays multiple related rules in a single spreadsheet-style view. In Rules Designer, a Decision Table presents a collection of related business rules with condition rows, rules, and actions presented in a tabular form that is easy to understand. Business users can compare cells and their values at a glance and can use Decision Table rule analysis features by clicking icons and selecting values in Rules Designer to help identify and correct conflicting or missing cases.

Users can modify the business rules at runtime using an SOA composer. It is a UI that allows the users to login and modify the rules at runtime.

Activity Monitoring:

Business Activity Monitoring provides real-time access to critical business performance indicators to improve the speed and effectiveness of business operations. It delivers real-time visibility and alerts to business users for response and analysis of their business operations.

The SI Platform will leverage Oracle Fusion Middleware Component Business Activity Monitoring (BAM) to provide real-time access to critical business process performance indicators to help business users to improve speed and effectiveness of business operations. Oracle BPEL will use the BAM sensors to feed the activity data to BAM for monitoring purposes.

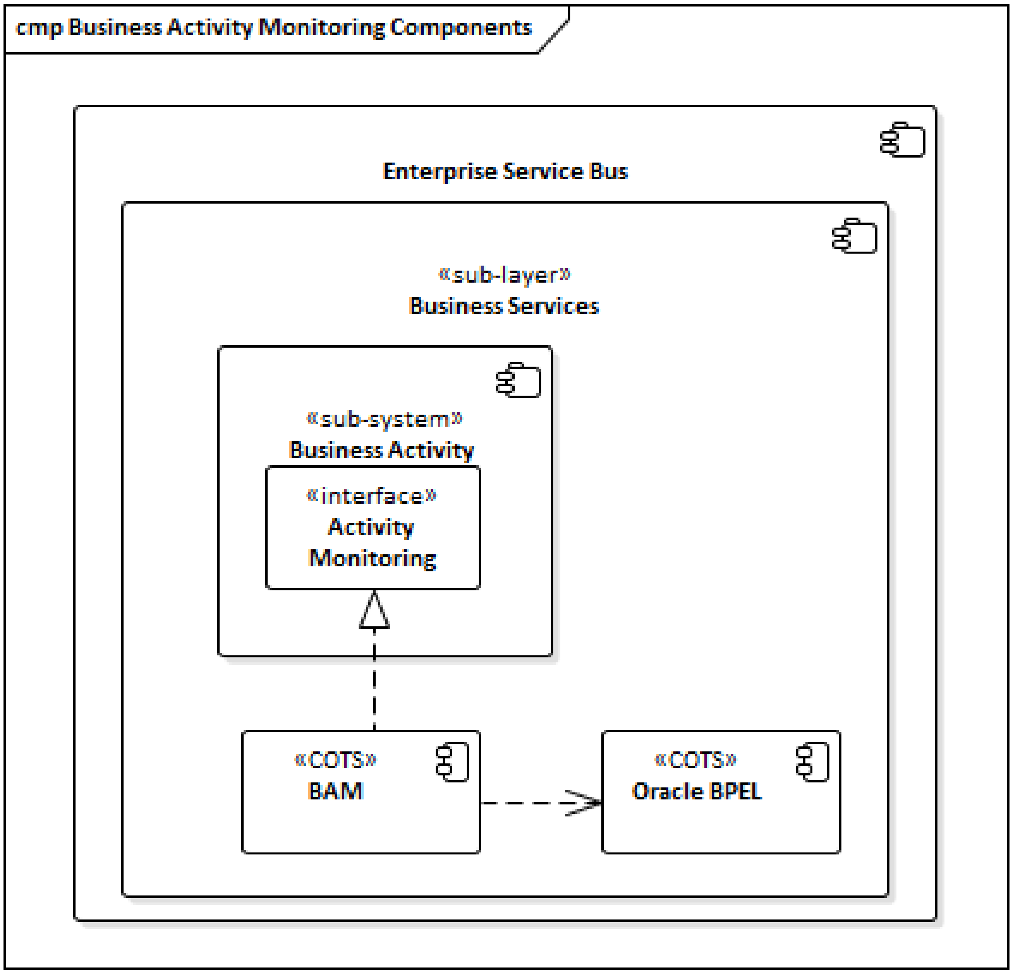
Some of the examples where the BAM will be used follow:

- Whenever a claim gets approved, denied, or suspended a real-time event will be generated for business users to look at and take action if needed.

- In a business process workflow, all events generated in a lifecycle of constituent's eligibility can be provided real-time rather than waiting for a historical report to be generated.
- Model and capture events from a UPI Portal, BPO Modules and External Trading Partners.
- Filter and correlate to identify key events from event "noise."
- Visualize data with out-of-the-box tools for rich dashboards.
- Generate user notifications and automated response to events.

The component diagram below explains the integration of the BAM and the event handling from the business services.

Figure 6-16: Business Activity Monitoring Component Model



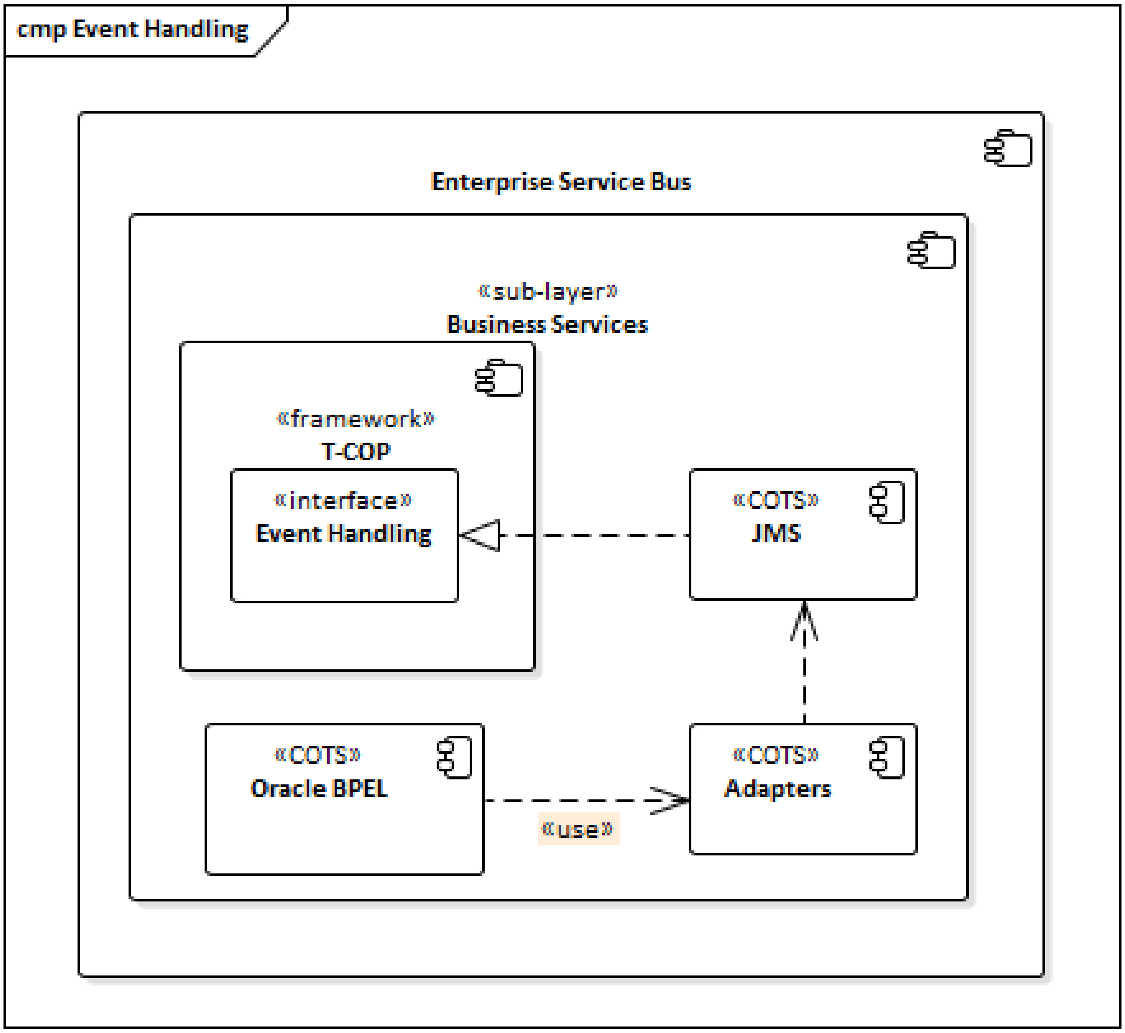
Event Handling:

Generation of events that are necessary to notify the consumer are captured and processed asynchronously using JMS queues and Topics. The events will be generated for various purposes, for example:

- Claim Approval and Claim Denial.
- Trigger another process that is waiting for the event.
- Send information to Business Activity Monitor regarding pre-defined key performance indicators.

The component diagram below explains the event handling from the business services.

Figure 6-17: Event Handling



Oracle BPEL in the business service layer. It will use the technology adapters (JMS, DB, or File) to feed the event data to event handling framework.

Technology Adapters:

Technology Adapters provide the functionality to connect to any other system using pre-existing adapters. The adapters enable connection to other systems using many of the pre-defined protocols which are part of Oracle fusion development tools. Example of connectors is REST, WS, MQ, FTP, File, JMS, Database, etc.

T-COP:

T-COP is a framework that enables the business workflow engine, business process management, orchestration of services, execution of SI Platform-specific business logic and rules, EDI, and bulk data processing, event handling, and management of file transfers. It is implemented using Oracle Fusion Middleware components Oracle BPEL and Oracle BPM which leverage other components of the platform, namely Adapters, Managed File Transfer (MFT), ODI, B2B, and BRE.

Enterprise Shared Services (ESS):

Enterprise shared services are the set of composite services implemented using Oracle BPEL and Oracle BPM that enables access to enterprise-wise shared systems like OpenText Exstream, Perceptive Content, and SAP Data Services, and Service Security (IdAM). These build around the functionality provided by the shared services tooling like address standardization, document management, communications management, and master data management. ESS is explained in greater details in subsection 6.2.7.

6.2.2 ESB Platform Security

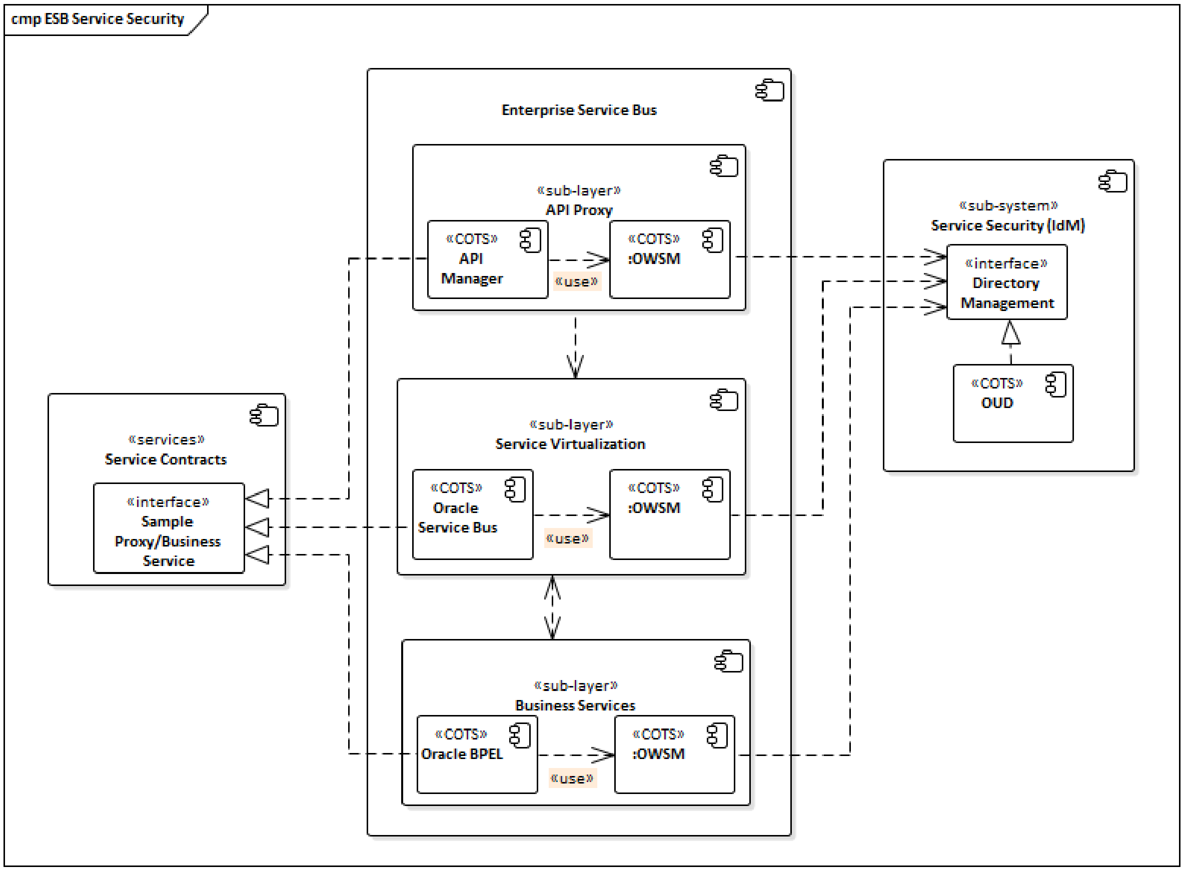
The security solution for SI Platform is designed with the “Defense in Depth” approach. The information assets are protected using multiple layers of defense, and security controls are implemented with a “data-centric” approach to ensure the overall security of the SI Platform.

The SI Platform integrates with UPI portal, BPO modules, Internal partner systems, and external partner systems. The integration is done mostly in real-time web services and batch mode. There is a need for a centralized ESB security and access controls mechanism across the platform. Oracle Fusion Middleware components Oracle Web Services Manager (OWSM), IdAM, and WebLogic will be used to implement and manage security across the ESB platform, as shown in the figure below. The security Layer ensures that the systems that interact with ESB have proper authentication and can only access the authorized resources. It ensures that the messages entering or leaving the ESB platform are encrypted at the transport layer.

Web services developed on the ESB platform will follow a basic set of authorization standards to support reliable communication among the HHS 2020 modules. Securing ESB Web Services is achieved via OWSM predefined policies. OWSM policies act as interceptors to Web service invocation and checks for a valid username/password at the message layer and authenticate against IdAM. These predefined can be attached to any REST or SOAP web services deployed on the ESB platform.

The following figure shows the components that involve securing ESB services.

Figure 6-18: ESB Service Security

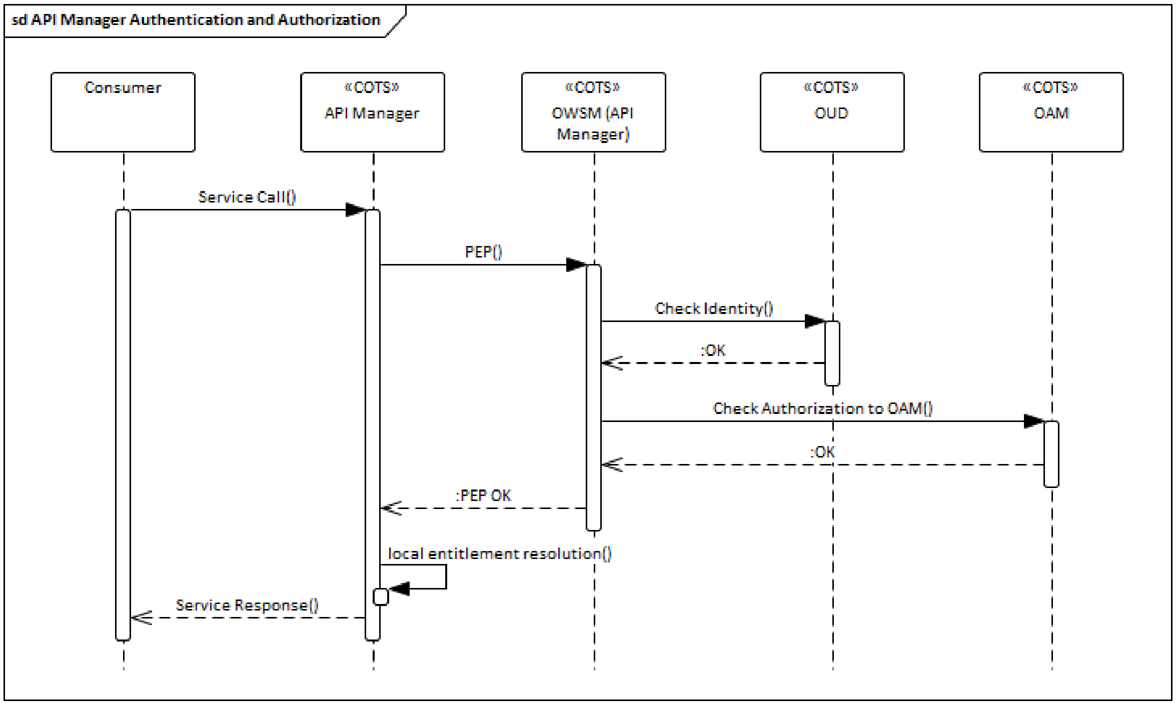


The figure below shows the sequence of API Manager authentication and authorization with IdAM component.

OWSM Predefined Policy for Authentication/Authorization:

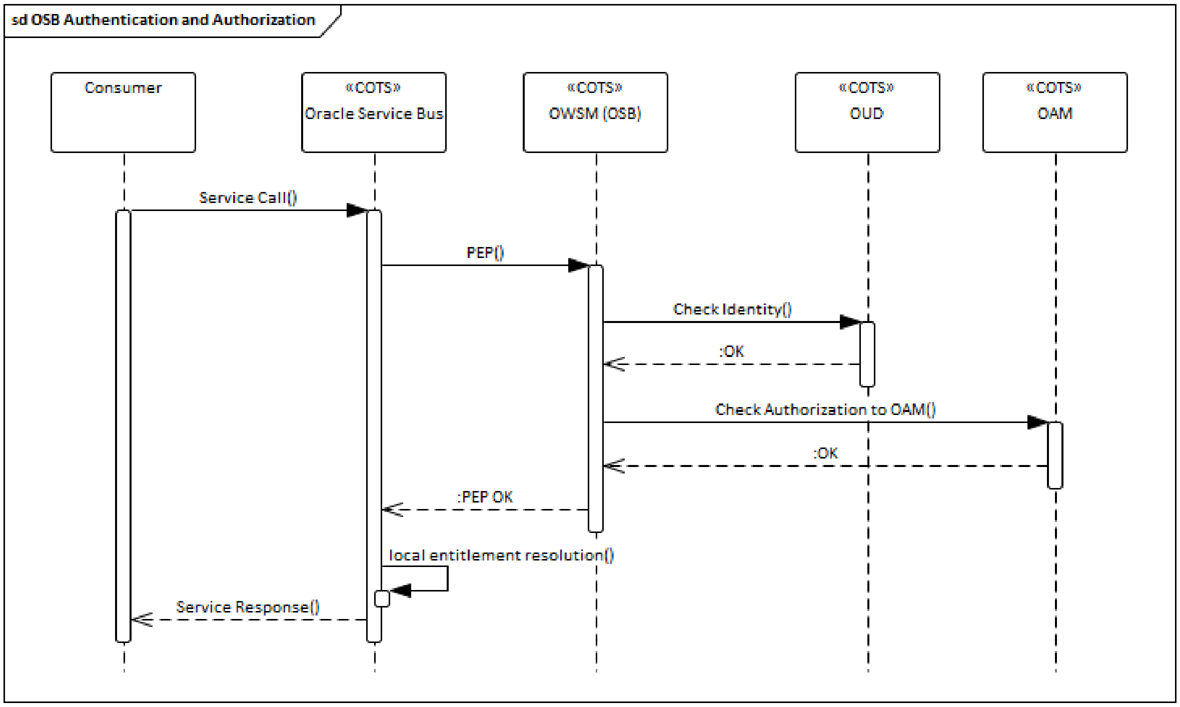
Oracle Web Services Manager (OWSM) provides a policy framework to manage and secure these Web services consistently across the SI Platform. To make secure communication between UPI portal, BPO modules, internal and external partner systems to ESB component, the SI Platform will use predefined WS-Security Username Token Profile for SOAP services and HTTP authentication for REST services.

Figure 6-19: API Manager- Service Call Authentication and Authorization Sequence


sd API Manager Authentication and Authorization
Consumer
«COTS» API Manager
«COTS» OWSM (API Manager)
«COTS» OUD
«COTS» OAM
Service Call()
PEP()
Check Identity()
:OK
Check Authorization to OAM()
:OK
:PEP OK
local entitlement resolution()
Service Response()

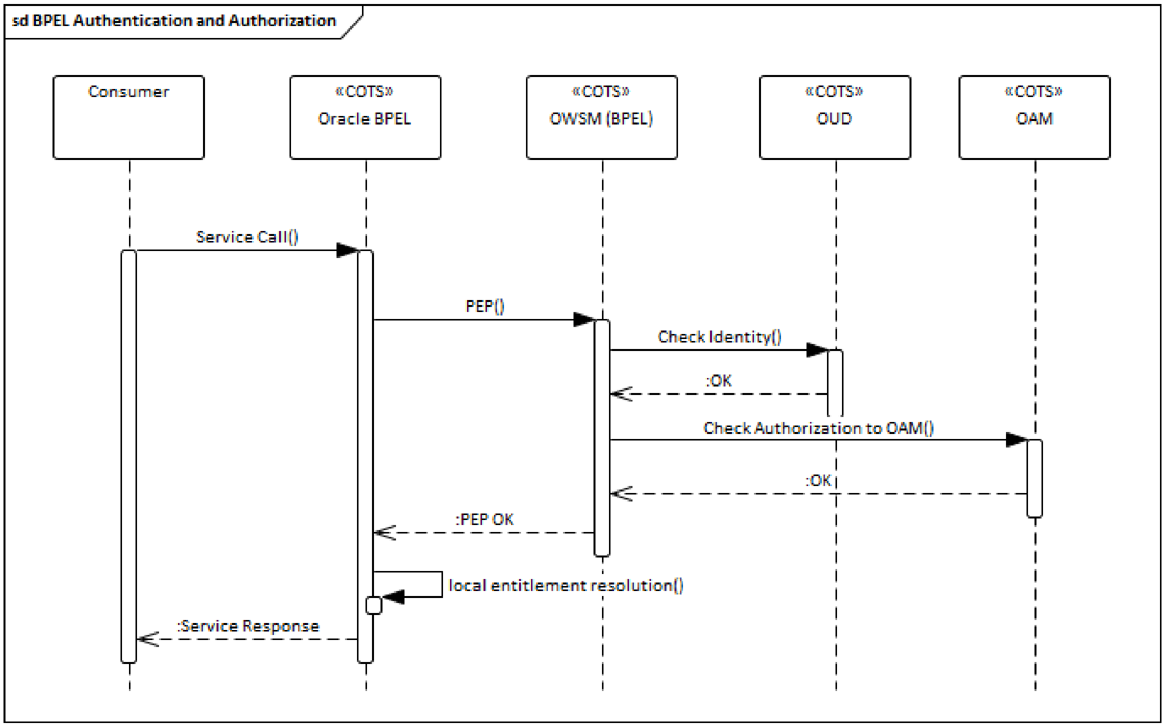
The figure below shows the sequence of Oracle Service Bus authentication and authorization with IdAM component.

Figure 6-20: Oracle Service Bus - Service Call Authentication and Authorization Sequence


sd OSB Authentication and Authorization
Consumer
«COTS» Oracle Service Bus
«COTS» OWSM (OSB)
«COTS» OUD
«COTS» OAM
Service Call()
PEP()
Check Identity()
:OK
Check Authorization to OAM()
:OK
:PEP OK
local entitlement resolution()
Service Response()

The figure below shows the sequence of Oracle BPEL authentication and authorization with IdAM component.

Figure 6-21: Oracle BPEL - Service Call Authentication and Authorization Sequence



The following are the steps that are depicted in the sequence diagrams:

1. System to system consumers, like External/Internal/BPO modules and other ESB components, make a service request to API Manager/Oracle Service Bus/Oracle BPEL components. Service Contracts are protected using was-username-token policy.

2. Service request message is intercepted by OWSM PEP agent (Policy Enforcement Point) and the message is inspected for header bearing Client Id/Password.

3. Once required header is found in incoming message, a Client Id/Password combination is extracted, and a security framework will query OUD for identity specified.

4. After OUD authenticates the client identity, OAM is referenced to validate the coarse-grain authorization for a client. For fine-grain entitlement, the client is validated locally on the corresponding domain.

5. Finally, service invocation is performed, and the result returned to consumer.

The SI Platform will use the following mechanisms to secure the web services deployed on the servers:

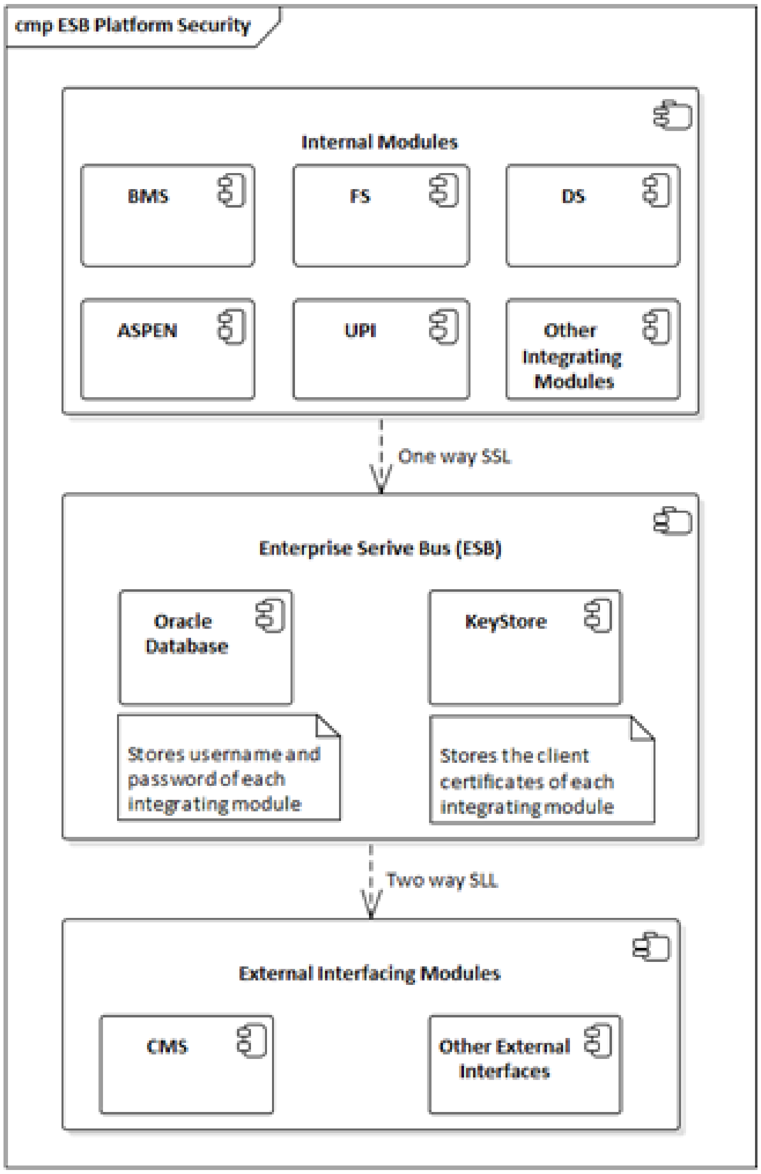
- Transport Layer Security
- User Authentication
- Transport Encoding
- Authorization

6.2.2.1 Transport Layer Security

Transport Layer Security (TLS) provides data integrity and privacy between the server and the client through secure communication. This is achieved by symmetric cryptography to encrypt the data using the TLS 1.2 protocol. Transport Layer Encryption provides numerous benefits beyond traffic confidentiality, including integrity protection, replay defenses, and server authentication. For transport layer security, the SI Platform will follow the below paradigm:

- Internal module communication will use one-way SSL.
- External module communication will use two-way SSL.

Figure 6-22: ESB Platform Security


cmp ESB Platform Security
Internal Modules
BMS
FS
DS
ASPEN
UPI
Other Integrating Modules
One way SSL
Enterprise Serive Bus (ESB)
Oracle Database
KeyStore
Stores username and password of each integrating module
Stores the client certificates of each integrating module
Two way SLL
External Interfacing Modules
CMS
Other External Interfaces

6.2.2.2 User Authentication

User authentication verifies users or systems that are attempting to connect to the service.

- Internal web services - Username/Password in SOAP header for SOAP web services and HTTP. Basic authentication for REST web services is implemented to achieve user authentication. The authentication mechanism and credentials are integrated with IdAM.
- External web services - Client Certificate authentication is implemented to achieve user authentication along with username/password in the SOAP header for SOAP web services and HTTP basic authentication for REST web services. Extra client certificate authentication is implemented for External web services, as the client is outside of the network.
- HTTP basic authentication for REST web services. Extra client certificate authentication is implemented for External web services as the client.

6.2.2.3 Authorization

Web services need to authorize web service clients the same way web applications authorize users. A web service needs to ensure that a web service client is authorized to perform a certain action (coarse-grained); on the requested data (fine-grained). This is achieved by separating normal users from administrative users, thereby access to administration and management functions within the web service application are limited to web service administrators. The details of coarse-grain and fine-grain authorization can be found in 6.3.3.4 and 6.3.3.5 respectively.

6.2.3 ESB Platform Auditing and Diagnostics

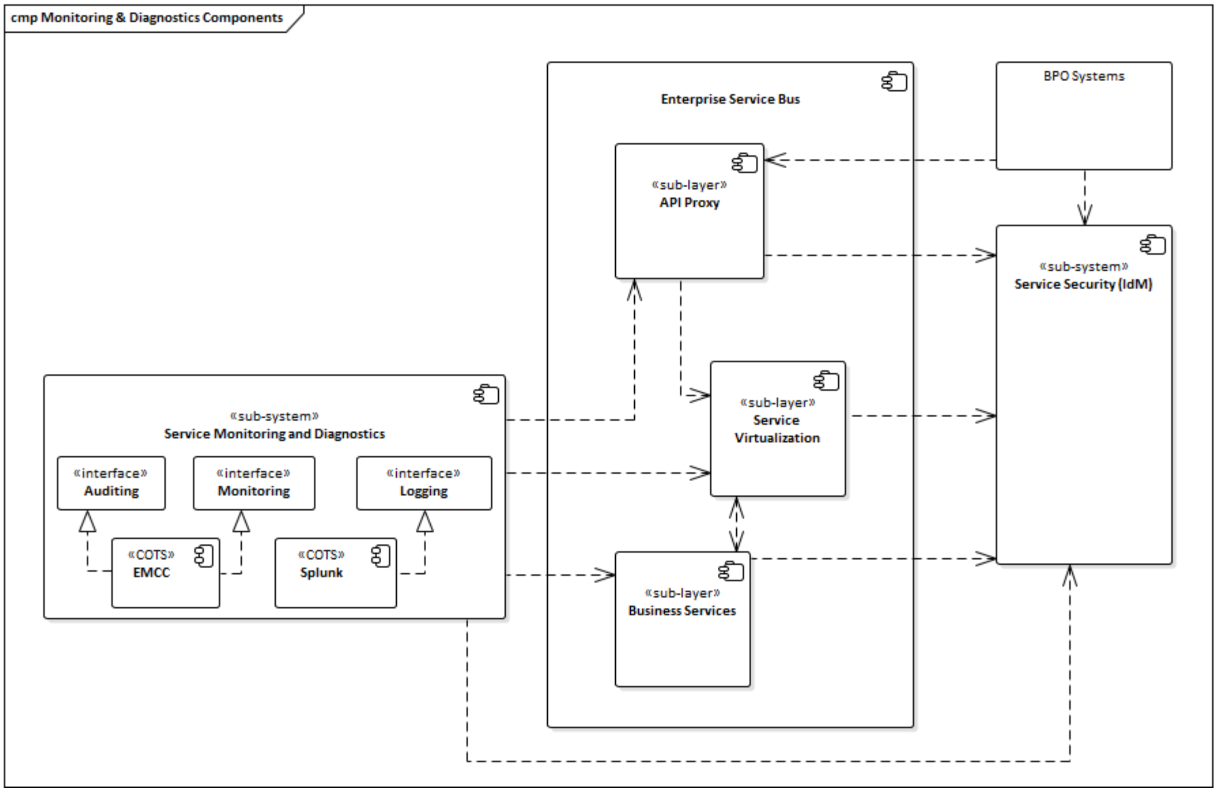
Successfully managing the SI Platform requires visibility into the physical components and their services. It also requires quick scale the SI Platform on demand. The Governance layer takes care of system monitoring, Logging, and auditing of the SI Platform and gives visibility to meet the everyday demands. The SI Platform is comprised of multiple Oracle Fusion Middleware tools which help centrally manage and monitor the ESB component. These tools offer visibility into monitoring, logging, and auditing of the entire SI Platform. Each component is configured to provide monitoring, logging, and auditing functionality certain way. Logging and Auditing can be configured to generate alerts using Splunk that can be used to augment default system monitoring capabilities.

The following are the architecturally significant requirements that are addressed in this subsection.

- Monitor usage and maintain a record of resource levels and consumption within the solution.
- Support automated and integrated service checkpoints to monitor service accuracy and completeness before proceeding to the next step or application batch process.
- Perform SOA-related business process and service management.
- Capture performance data (e.g., elapsed time, dates) to support continuous improvement.

The following figure shows the components responsible for auditing and diagnosis of ESB platform.

Figure 6-23: Auditing and Diagnosis of ESB Platform



6.2.3.1 System Monitoring

The SI Platform provides comprehensive system monitoring solution to monitor system parameters like CPU, memory, IO usage, etc. Oracle Enterprise management cloud control (EMCC) provides a comprehensive system monitoring capability for the SI Platform. Following are the capabilities of EMCC.

- Enterprise application monitoring and diagnostics.
- Middleware and database monitoring.
- Application Performance Management (APM).
- Business Transaction Management.

SI Platform will leverage EMCC as the central system monitoring solution. EMCC provides a web-based console to monitor and administer the SI Platform and its components from one location on the network. All the components and services of SI Platform, including application servers, databases, hosts, and listeners.

6.2.3.2 Logging

Logging component of the SI Platform takes care of system logs as well as application logs. Oracle Fusion Middleware product suite provides a comprehensive logging framework which generates log files containing messages that record all types of events, including startup and shutdown information, errors, warning messages, and access information on HTTP web service requests. These log files are in different formats. For example, standard output, file, JMX log broadcaster, and so on. The logging framework supports log4j style logging and will be configured to log as per the different logging levels (INFO, DEBUG, WARN, ERROR, FATAL, ALL).

The Splunk forwarders will be installed at every ESB component to feed the logs generated to the Splunk enterprise for centralized log monitoring and analysis. The details of log aggregation and monitoring is explained in Subsection 4.5.4 Log Aggregation and Monitoring.

6.2.3.3 Auditing

The SI Platform keeps track of the changes and transactions occurring in the system. The SI Platform leverages Oracle Fusion Middleware Product Suite Auditing Framework and provides the following auditing capabilities.

- Configuration changes - The SI Platform provides an administrative console to view and access the history of configuration changes to the ESB application.
- User profile change and User access activity - The SI Platform keeps track of the users who log into the application and stores their session details, such as the message exchanges and services consumed.
- Message flows - The SI Platform creates persistent files of the messages that flow across the pipeline. The auditing of the message exchanges will be configured as per the message type and as per environment-based audit levels (Production, Test, and Development). Audit information of the ESB message exchange will be viewed via the web-based Enterprise Manager Console, which is part of Oracle Fusion Middleware Product Suite.
- Operational Activities - The web-based EMCC console provides auditing information for Operational activities like starting and stopping applications, upgrades, and backups

Audit trails also support the log/audit requirements of regulations such as, the HIPAA §164.308(a)(1)(ii)(D): Security Management Process to implement policies and procedures to prevent, detect, contain, and correct security violations, including, implementing procedures to regularly review records of information system activity.

6.2.4 ESB Transaction Management and Error Handling

The subsections below describe how the SI Platform implements transaction management and Error Handling.

6.2.4.1 Transaction Management

The SI Platform business service layer will be implementing business process orchestrations for Create, Read, Update, and Delete (CRUD) operations sequentially or simultaneously based on a business workflow against HHS 2020 modules. These CRUD operations will be against HHS 2020 modules that may be legacy based, SQL databases, File Databases, JMS queues, etc.

The following is the architecturally significant requirement addressed in this subsection.

- The Open XA-compliant transaction boundaries are to be defined explicitly in BPML flows so that the Distributed Transaction Coordinator available as part of the Enterprise Application

Integration (EAI) engine takes care of all commits and rollback logic based on the collective outcome of all functionality invocations within transaction's scope.

In order to maintain the integrity of the data across the HHS 2020 modules, these heterogeneous transactions need to be executed as a single distributed transaction unit. If these are not executed as part of a single distributed transaction unit then data integrity across the systems will be compromised and will be inconsistent and thereby leading to erroneous results for constituents.

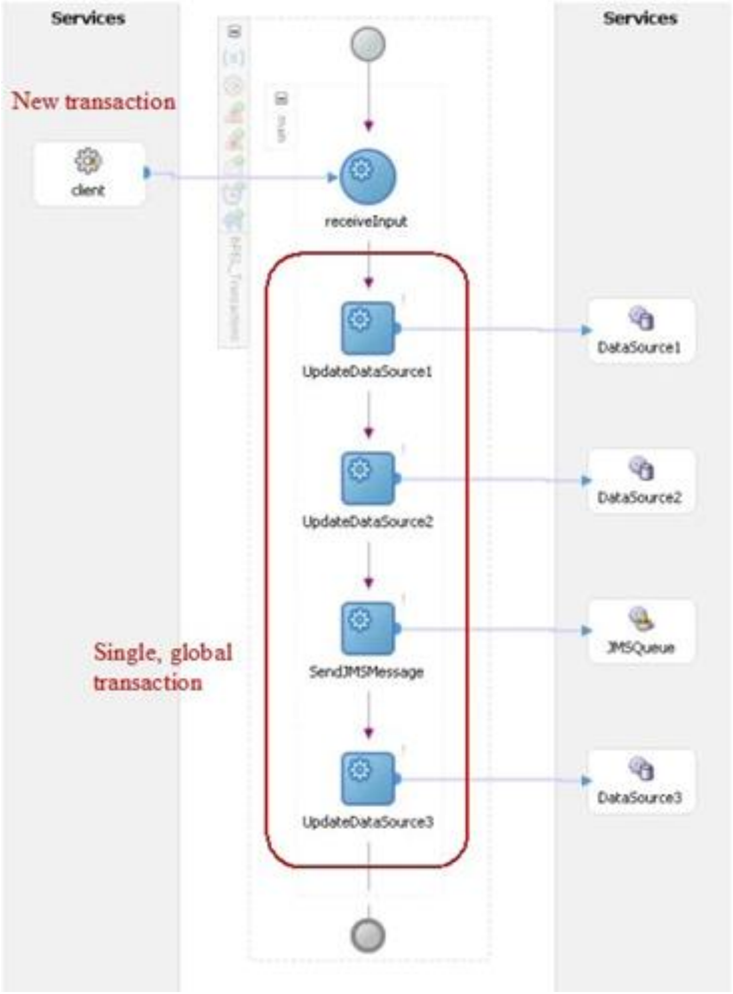
Some of the examples where heterogeneous transactions which are part of a business process orchestration need to be executed as a single distributed transaction unit are as follows:

- An update to member's employment and income information in MMISR Financial Module (FS) has to be automatically correlated with changes to benefit eligibility in ASPEN.
- Updates to both FS and DS, must take place either successfully in both modules and not at all as part of a business workflow.

The SI Platform will leverage Oracle Fusion Middleware provided XA compliant connectors and 2-Phase commit protocol mechanisms to update BPO module systems simultaneously as part of the business workflow to maintain the integrity of the data. The XA compliant connectors and 2-Phase commit protocol help a business workflow to be executed as part of the one single global distributed transaction unit thereby committing updates in all systems or rolling back the whole transaction in case there is any error. The connectors provided by Oracle Fusion Middleware: Database, JMS, etc. participate in executing a global transaction as a single unit and maintain the integrity of the data across the systems.

Below BPEL process workflow shows updates to different data sources are being executed as part of a single global transaction using XA compliant and 2-PC commit protocol mechanisms provided by Oracle Fusion Middleware Stack.

Figure 6-24: **Single Global Transaction in ESB**

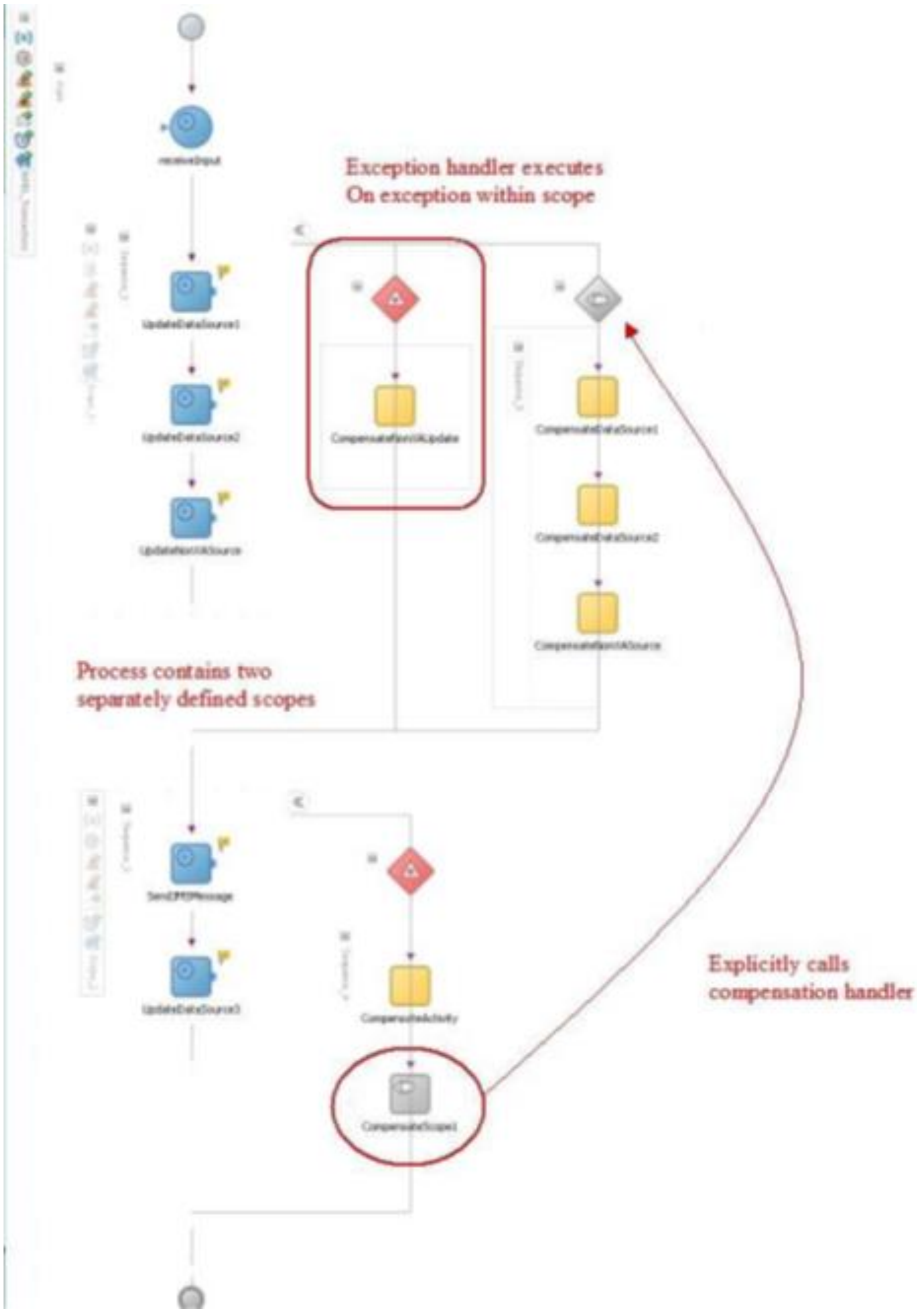


In situations where a data source within the MMSIR and HHS 2020 ecosystem cannot participate in a global distributed transaction as a single unit, Oracle Fusion Middleware offers a concept called "Compensation" which helps maintain the integrity of the data across systems. Some of the systems which cannot participate in globally distributed transactions are File based, Legacy Modules, HTTP, etc.

Compensation helps to do a transaction rollback and is applicable to idempotent transaction context. Transactions which are committed during the business process execution that require a rollback to maintain the integrity of data across the enterprise can be accomplished using the compensation. A compensation handler helps to roll back the transactions that are completed in the previous steps of a business process.

Below BPEL workflow shows how data integrity can be maintained if multiple systems need to be updated simultaneously as part of a business process which cannot participate as part of a single global transaction unit.

Figure 6-25: Transaction Involving Multiple Systems in ESB



6.2.4.2 Error and Exception Handling

ESB platform being an integration layer, exception handling is the significant part of this. SI Contractor will use the error and exception handling provided out-of-the box in Oracle Middleware and Custom Fault Handling Framework (CFF) to have a unified approach to handle error and exceptions.

The following are the architecturally significant requirements that are addressed in this subsection.

- Provide the ability to suspend the processing of erroneous transactions until the error is resolved and provide notification of the error and resolution
- Provide structured exception and error handling.
- Distinguish between errors (stop process) and exception conditions (skip transaction and continue the process).

During the course of the message processing in the ESB platform two kinds of faults may arise as described below.

Business faults: Business faults are application-specific and are unrecoverable. They are thrown when there is a problem with the information being processed. These are mainly user defined or data related faults. For example, the SSN number is not found in the database. The services need to identify, throw and handle these kinds of faults.

Runtime faults: Runtime faults are not user-defined and are thrown by the system during the erroneous condition. For example, when a web service (WS) is unavailable Simple Object Access Protocol (SOAP) fault occurs during this WS invocation, and so on.

The ESB platform provides the following out-of-the box mechanisms to track and manage errors.

- Oracle SOA Fault Management Framework – Oracle SOA Suite provides a generic fault management framework for handling faults in BPEL processes. If a fault occurs during runtime in an invoke activity in a process, the framework catches the fault and performs a user-specified action defined in a fault policy file associated with the composite or component.

Below are the different actions mentioned in the fault policy file for a fault occurs in the BPEL service. Not all the actions will be used, and depending on the requirement, the corresponding actions will be decided.

Table 6-30: BPEL Error Handling Actions

Types of Action	Function
Abort	Terminates the process flow.
Human Intervention	Causes the current activity to stop processing. Human intervention from Oracle Enterprise Manager Fusion Middleware Control is required to handle the fault.
Replay Scope	Replay the scope the Fault occurred in.
Rethrow Fault	Sends the fault to the BPEL fault handlers (catch activities in scope activities). If none are available, the fault is sent up.
Retry	Provides the following options for retrying the activity: – Retry a specified number of times.

Types of Action	Function
	– Provide a delay between retries (in seconds). – Increase the interval with an exponential back off. – Chain to a retry failure action if retry N times fails.
Java Action	Enables to execute an external Java class.
Invoke WS	Handles a rejected message by calling a web service (uses an Oracle defined schema).
Enqueue	Enqueues a rejected message to a Queue (AQ) as a JMS message with the appropriate context and payload.
File Action	Creates an error handler for messages by storing a rejected message in a file.

- Error Hospital – It will be used to recover from the recoverable faults occurred in the BPEL services. Bulk fault recoveries or bulk fault terminations can be performed in a single request.
- Resiliency (Circuit Breaker) – ESB platform will utilize this to automatically suspend upstream endpoints when a downstream service is unreachable from an SOA composite. This prevents faults from building up in the server and of having to bulk-recover faulted instances. The upstream endpoints are automatically resumed after the downstream services come back up.

Along with this, a common and custom ESB Fault Handling Framework (CFF) will be developed and deployed to the ESB platform.

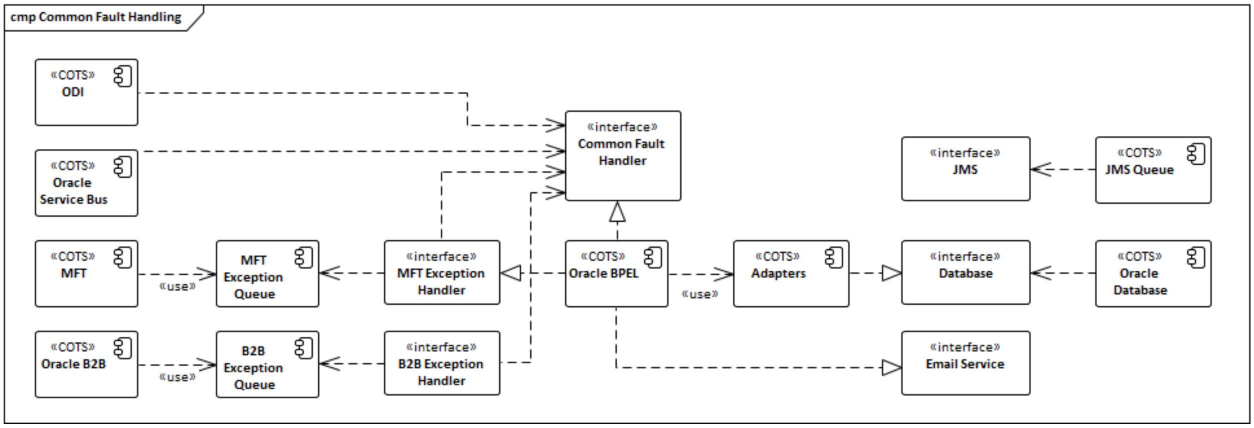
Common Fault Handling Framework (CFF):

SI contractor will implement a Common Fault Handling Framework (CFF) to handle all errors in the ESB Platform layers (OSB, BPEL, ODI, MFT, and B2B). The CFF will provide a web service based on the BPEL process that will log the errors into a central storage (Oracle DB) for all the services deployed in the ESB platform. For example, it logs the service name, message, transaction, source, and so on. This BPEL web service sends the real-time email notifications to the intended contacts (such as Technical team, Business users, Administrators and so on depending on the requirement) for all kinds of errors in the ESB services.

It also stores the error message in the JMS topic for further troubleshooting and processing. All the services in different technology stacks in the ESB platform will call this CFF service and pass on the exceptions in a common exception message format. The common exception message format (CFF Format) will be implemented by using a custom exception-handling schema.

The following component diagram provides the process flow of the CFF.

Figure 6-26: ESB Common Fault Handling Framework



1. The faulted service or a common service in the different product stack (OSB, BPEL, and ODI) invokes the CFF service. Also, the faulted service or a common service of the respective product will transform the exception message to the CFF format before invoking the CFF in the SOA domain.

2. Runtime faults are pushed to a product provided JMS Exception Queue in MFT and B2B layers. The respective custom consumers (MFT Exception Handler and B2B Exception Handler) read the exception message from the queues and invokes the CFF after transforming the exception message to the CFF format.

3. The CFF service pushes the faulted service input payload to a JMS Topic in the SOA server. The support team can later use this payload to manually process the requests.

4. The CFF service also invokes the Notification service after transforming the message to the appropriate notification service input. The SOA server provides an inbuilt email notification service. The email server details need to be configured by the administrators in the Email Drivers in the SOA server prior to receiving the email notifications.

5. The CFF service also inserts the error information into a table in the Oracle DB. Before inserting the error details to the Oracle DB, CFF will transform the error message from the CFF format to the required tabular format of the Oracle DB. This information will be utilized to report error statistics across the product stack if required.

6.2.5 ESB Integrations

ESB platform provides several integration mechanisms to interact and exchange data among disparate modules.

The following are the architecturally significant requirements that are addressed in this subsection.

- Services will interoperate via sending/receiving synchronous and asynchronous messages.

- Support common SOA and Enterprise integration patterns, including publish/subscribe, broadcast, intermediaries, splitter/aggregator, parsing and validating messages and others as recommended by the Vendor.

To connect to different BPO modules, UPI Portal, Internal and External systems within and outside of MMISR which are built on diverse platforms with varied communication mechanisms, the ESB platform implements software components commonly referred to as adapters and connectors.

An adapter is a driver that enables the ESB platform to connect to various kinds of endpoints, such as databases, files servers, message queues, etc.

A connector enables the ESB platform to interact with third-party product functionality and data, for example, connectors allow the ESB platform to interact with the application programming interfaces (API) of services provided by BPO Modules, UPI Portal, Internal and External Systems, LDAP, MQ series, etc.

The adapters and connectors provide the infrastructure to automate the integration of MMSIR systems. The table below shows the list of common technologies for which ESB provides the adapters and connectors to connect to different systems within and outside of MMISR.

Table 6-31: Technology Adapters and Connectors in ESB

Technology	Description
File/SFTP	The ESB platform supports receiving and sending files from and to data sources. For inbound data, the ESB FTP adapter service reads data from a local or remote file system, transforms the file data into an XML / JSON message, and sends the message to the ESB for further processing. For outbound data, the FTP adapter transforms the contents of an XML / JSON message to a text file and writes the text file to a local or remote file system.
SQL/database	The ESB platform supports direct access to the database of a data source. The data source provides database access with appropriate rights to the ESB. For inbound data, the ESB SQL adapter sends an XML / JSON message to the ESB when a SQL insert, update or delete operation is performed against a database. For outbound data, the SQL adapter transforms the contents of an XML / JSON message into a SQL insert, update, or delete operation on the target database.
JMS	The ESB platform supports sending and receiving XML messages using Java Message Service (JMS) queues. For inbound data, the ESB JMS adapter listens on a JMS destination (queue or a topic) and forwards incoming messages to the ESB. For outbound data, the JMS adapter writes messages from ESB to a JMS queue.
HTTPS	The ESB platform supports sending and receiving XML/ JSON messages directly via an HTTP endpoint.

Technology	Description
	For inbound data, the ESB provides a web server address that connecting applications can send messages to (using GET, POST, UPDATE, DELETE methods). The web service receives the message and forwards it to the ESB. For outbound data, the web service writes messages to an HTTP endpoint.
SMTP/email	The ESB platform supports sending emails over the Simple Mail Transfer Protocol (SMTP). The SMTP adapter subscribes to messages from the ESB, converts the messages into email messages, and sends them using the configured SMTP server. The adapter supports setting the body of the ESB messages as the subject, body, or attachment of the email. The adapter can also retrieve emails from the mail server.
Custom third-party connectors	The ESB platform supports out-of-the-box connections with many industry-leading third-party applications. Custom connectors support seamless connections with other COTS applications, such as SAP, PeopleSoft, etc. These adapters can directly connect to the business layers and data layers of third-party applications using pre-built web services.

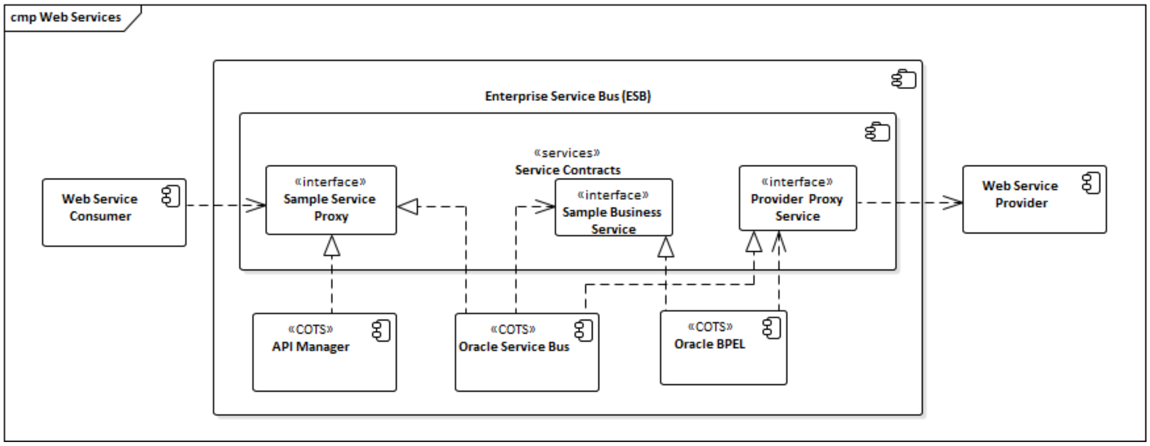
Integrating Platform uses Web Services, MFT, and ETL for integrating its services and components. The system design of each of the integration mechanism is explained below.

6.2.5.1 Web Services

A web service is a function that programs can access via the web using either Hypertext Transfer Protocol (HTTP) or HTTP over Transport Layer Security (HTTPS). Web services form a critical part of a service-oriented architecture (SOA), exposing reusable system functionality in the form of component interfaces.

The SI Platform supports both SOAP and REST web services which developed and managed by leveraging an ESB using Oracle Fusion Middleware. All web service interactions are coordinated by the ESB.

Figure 6-27: Web Service Integration in ESB



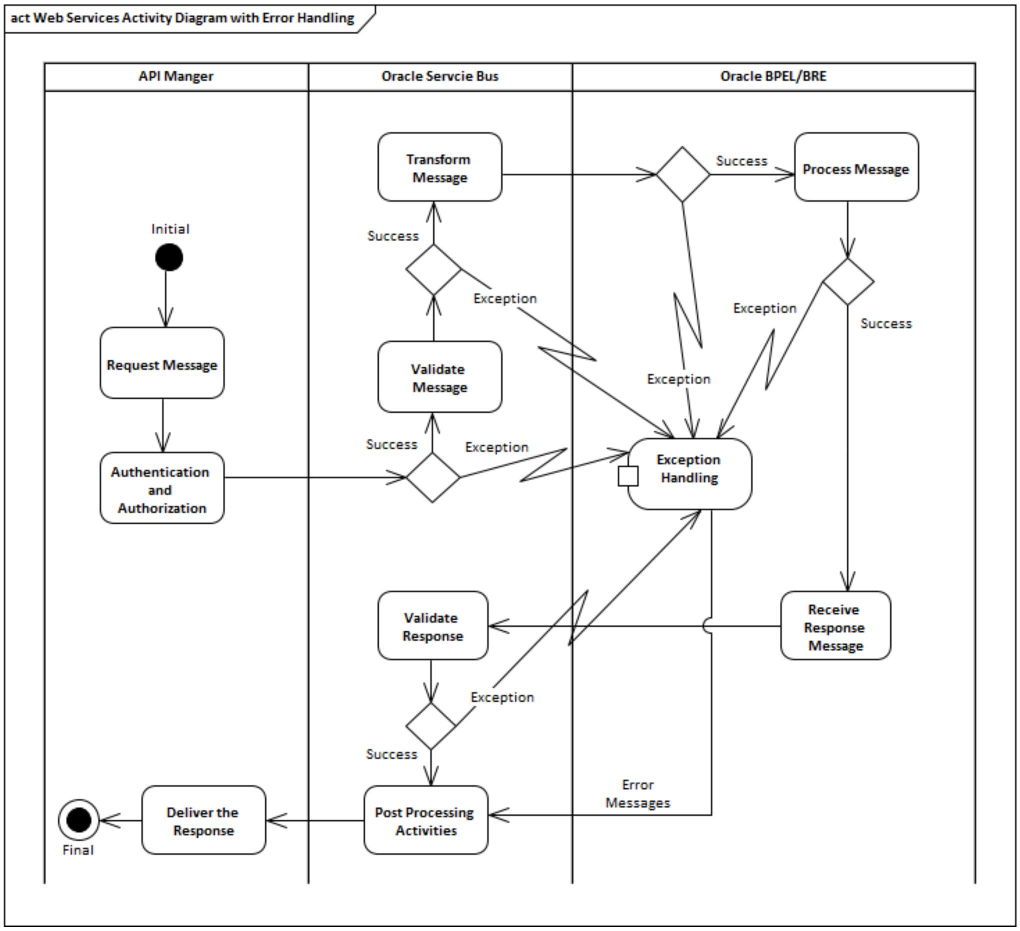
The following are the communications steps in Web Service integration that ESB platform:

1. The consumer sends the message to the ESB platform.
2. The ESB platform sends the message to the provider.
3. The provider sends a response back to the ESB platform.
4. The ESB platform sends the message back to the consumer.

Web Service Process Flow:

The following are the series of discrete steps that take place during the execution of web services, as shown in the figure below.

Figure 6-28: Web Services Activity Diagram with Error Handling



REST vs. SOAP Web Services:

Although REST web services are preferred and encouraged, the SI Platform will support both SOAP and REST web services under the appropriate conditions. All the new modules of HHS 2020 enterprise solution standardize using REST services for most web service interfaces.

REST web services usage scenarios are as follows:

- Read-caching will improve system performance, as in the case of data services.
- XML parsing will negatively impact the performance.
- Flexible data representation is required (multiple formats).
- Maintaining a state of information from one request to another is not required.
- Service-functionality can be expressed as simple CRUD operations on data resources.

SOAP-based web services usage scenarios are as follows:

- Enterprise level security requires pass-through authentication, along several intermediaries.
- Service-functionality is expressed as a series of activities to accomplish the desired outcome.
- Coordination between several services is required to accomplish the desired outcome.
- Clients require a formal service contract and the rigid specification that the WSDL provides.

6.2.5.2 Java Message Service (JMS)

The Java Message Service (JMS) API is a Java API for sending and receiving messages between the modules reliably and asynchronously. It is message-oriented middleware (MOM) standard that allows modules via ESB to create, send, receive, and read messages. It allows the communication between different modules of a distributed system to be loosely coupled, reliable, and asynchronous.

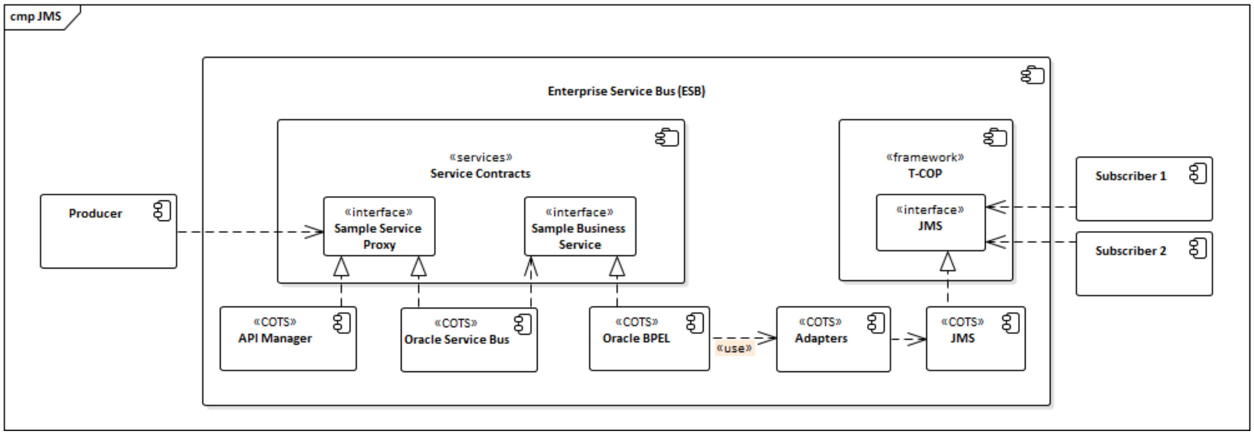
The SI Platform will implement JMS as one of the service integration mechanisms to create, send, and receive messages between different HHS 2020 BPO modules, internal systems, and external partner systems. This will make the communication between different HHS 2020 modules and the ESB component are loosely coupled, reliable, and asynchronous. The SI Platform will leverage Oracle Fusion Middleware components JMS Connectors, JMS Queues and JMS Topics in the service virtualization and business service layers to implement JMS to enable reliability and Asynchronous communication. Following are some of the benefits of implementing JMS for the integration of HHS 2020 modules via the SI Platform:

- Asynchronous: BPO modules can deliver messages to the ESB SI Platform to deliver to the target modules as they arrive; a target module does not have to request messages in order to receive them and a source module does not have to know about the receiving modules.

Reliable: The JMS API can ensure guaranteed delivery of a message by delivering it once and only once. Other levels of reliability are available via this API for modules that can afford to miss messages or to receive duplicate messages.

Integration: JMS API helps to integrate different heterogeneous HHS 2020 modules implemented in different technologies, reduce system bottlenecks, increase scalability, and respond more quickly to changes.

Figure 6-29: JMS Integration in ESB



6.2.5.3 Manage File Transfer (MFT)

Despite the advent of service-oriented architecture (SOA) and message-oriented middleware (MOM), a large portion of enterprise data exchange takes place by transferring files between systems using File Transfer Protocol (FTP) server and client technology to transfer files. However, the basic file transfers using, for example, FTP or HTTP, which were adequate when introduced, do not address rigorous enterprise requirements such as, control over sensitive information, global visibility, and integration with other enterprise systems.

In the MMISR and HHS 2020 ecosystem, batch files need to be transferred to different external partner systems and trading partners. There is a need to manage, transfer, monitor, and maintain all the file transfers which will use FTP, FTPS, and SFTP protocols to transfer files.

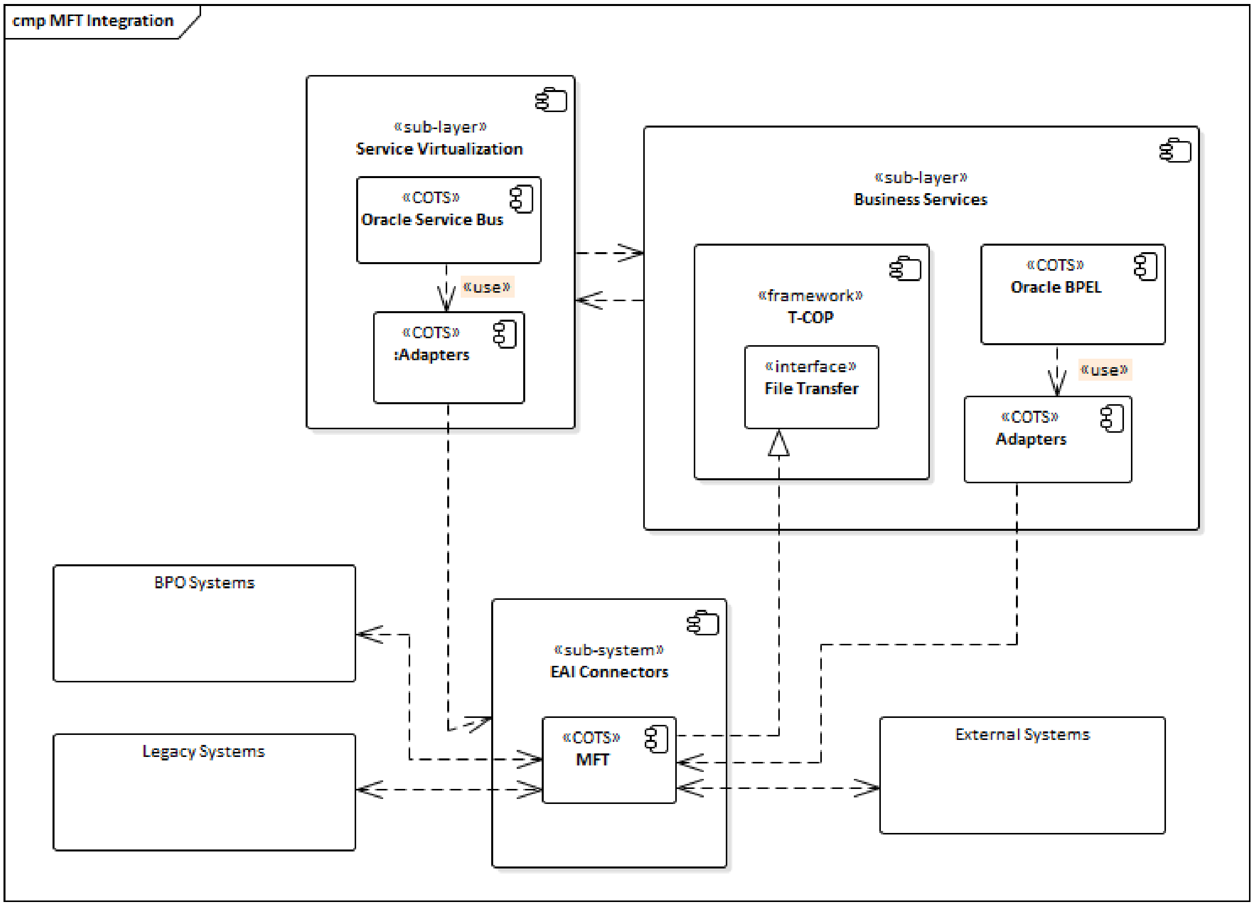
The SI Platform will leverage Oracle Fusion Middleware component “Manage File Transfer” to manage, transfer, and monitor batch files to and from the MMISR/HHS 2020 systems and external partner systems and trading partners. Oracle’s Managed file transfer (MFT) application use FTP as their foundation and add software components and communication protocols to provide services, such as the following.

- Reliably transferring data using secure protocols, including SFTP, FTPS, and HTTPS.
- Automatically transferring data, for example, transferring data based on events, schedules, file names, etc.
- Easily integrating with other technologies in the environment, such as the ESB.
- Monitoring file transfers.
- Logging processes, errors, and audit trails.

The figure below provides details on the use of MFT in the SI Platform. MFT can transfer files to and from other the SI Platform applications, such as the ESB and ETL, and between external integrating components.

The file transfers occur using industry standards, such as FTP, SSH File Transfer Protocol (SFTP), and HTTPS.

Figure 6-30: MFT Integration in ESB



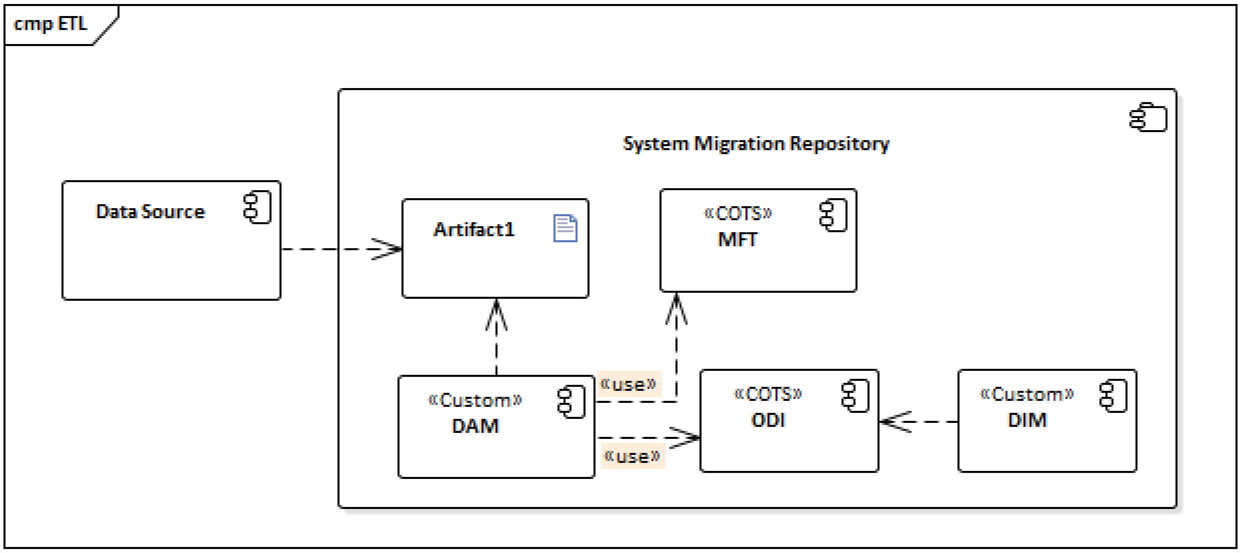
6.2.5.4 Extract Transform Load (ETL)

Extract Transform Load (ETL) is a mechanism to extract data from heterogeneous systems, transform the data, and load into different systems in different formats. MMISR and HHS 2020 systems exchange data with multiple interface partners using batch files. The SI Platform will be implemented ETL service integration mechanism to extract data from legacy and modern modules of MMISR and HHS 2020 system, transform them, and load them into different formats to be transferred to different internal and external partner systems. The data extracted can be from a file, SQL database, file database, and the data are transformed and shared with different interface partners in batch files of different formats including, XML, CSV, Excel, etc.

To implement ETL, SI Platform will be using Oracle Data Integrator (ODI) component of the Oracle Fusion Middleware Stack. ODI is a data SI Platform that covers all data integration requirements from:

- High-volume
- High-performance batches
- Event-driven
- SOA-enabled data services

Figure 6-31: ETL Process in ESB Platform



6.2.5.5 ESB Integration Patterns

The SI Platform supports and implements the established SOA/ESB and Enterprise integration design and message exchange communication patterns within ESB. The system will support the following communication and design patterns including:

Message Exchange Communication Patterns:

- Synchronous – Request/Response
- Asynchronous – One Way Fire & Forget
- Asynchronous – Delayed Response

Routing Patterns:

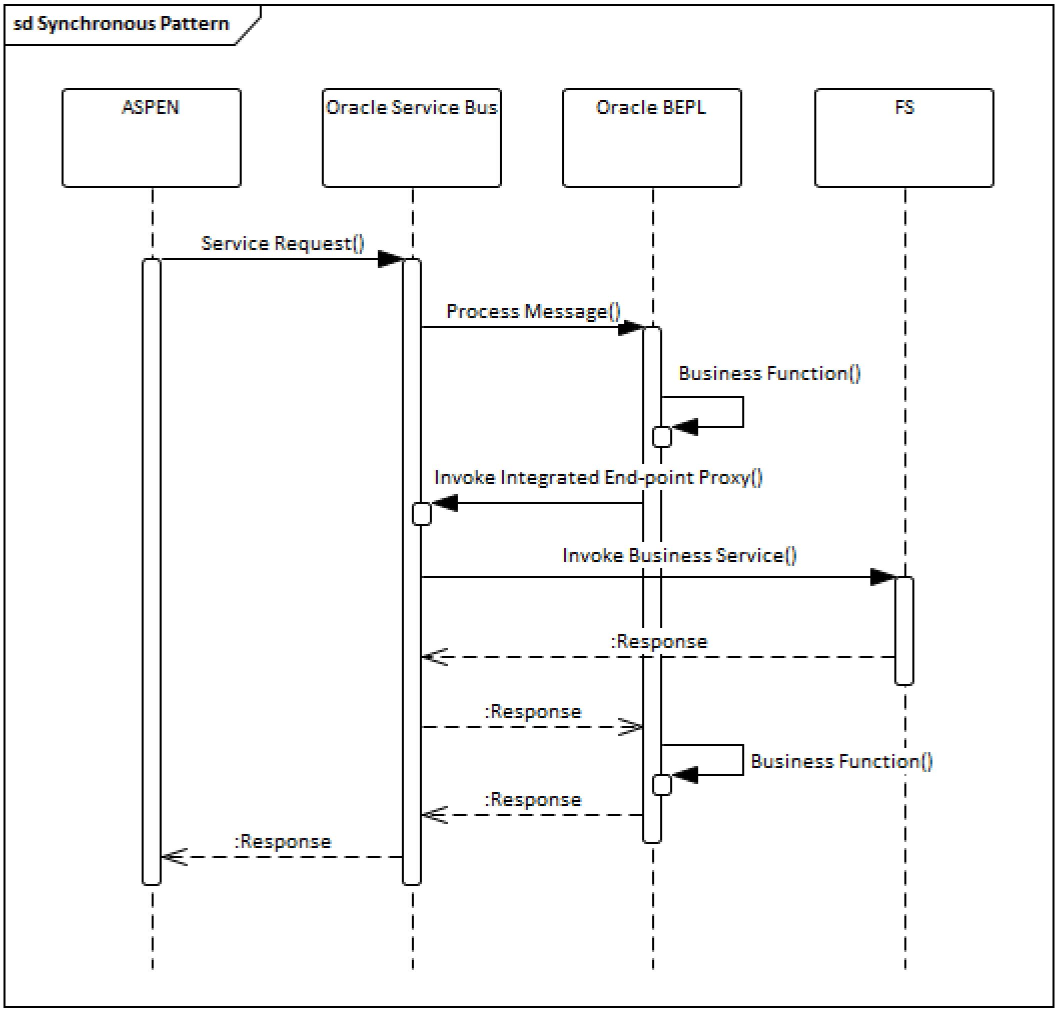
- Intermediaries
- Broadcast
- Pipeline
- Publish/Subscribe
- Splitter/Aggregator

Synchronous Request/Response:

Synchronous Request/Response communication pattern is also called blocking communication because all operations in an application that sends a request are blocked until it receives a reply. The connection between the sender and replier stays open during this period of time. This type of communication is essential when the sender application needs an immediate response to continue with further processing.

The SI Platform will implement this communication pattern whenever the source systems need an immediate response and if it is meant to run for a short duration. Examples of this pattern are when a source system is checking a claimStatus or getCaseStatus. This will be implemented mostly as a SOAP or REST Web Service.

Figure 6-32: Synchronous Communication Pattern



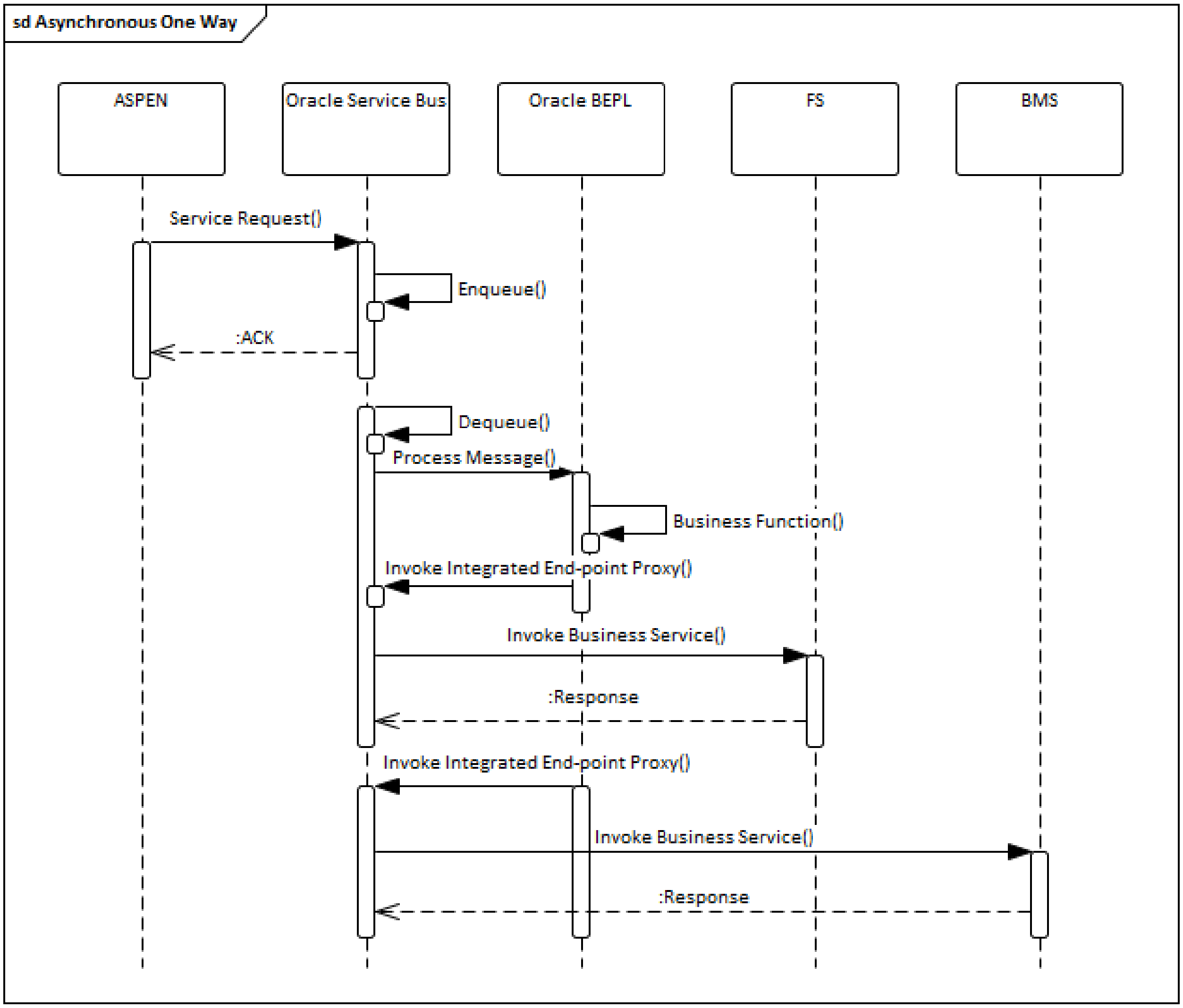
Asynchronous – One-way/Fire-Forget:

Asynchronous communication is a non-blocking communication pattern for message exchange. In this pattern, the source module that sends a request does not wait for a reply from the target module. The connection between the source module and the target module will be closed as soon as the request is sent out. This also means that the source module can execute multiple processes simultaneously. This message exchange pattern is effective when there are large volumes of data that need to be processed and when no immediate response is expected or required.

The SI Platform will implement the Asynchronous message exchange pattern, whenever the HHS 2020 source module does not expect an immediate response or immediate response is not required from the target modules. The SI Platform as a design consideration that depends on the use case and business process asynchronous message exchange pattern that will be preferred and implemented, due to its reliability, non-blocking communication, high service availability, and its ability to process large volumes of data to integrate HHS 2020 modules. Examples of this pattern are when a source module (ASPEN) wants to send out events for eligibility approved or denied asynchronously and does not require a response from target modules (FS, DS) via the SI Platform.

On the SI Platform, SOAP/REST web services will implement this message exchange pattern along with JMS Queues and Topics. Below is the high-level asynchronous message exchange pattern sequence diagram.

Figure 6-33: Asynchronous – One Way Communication Pattern



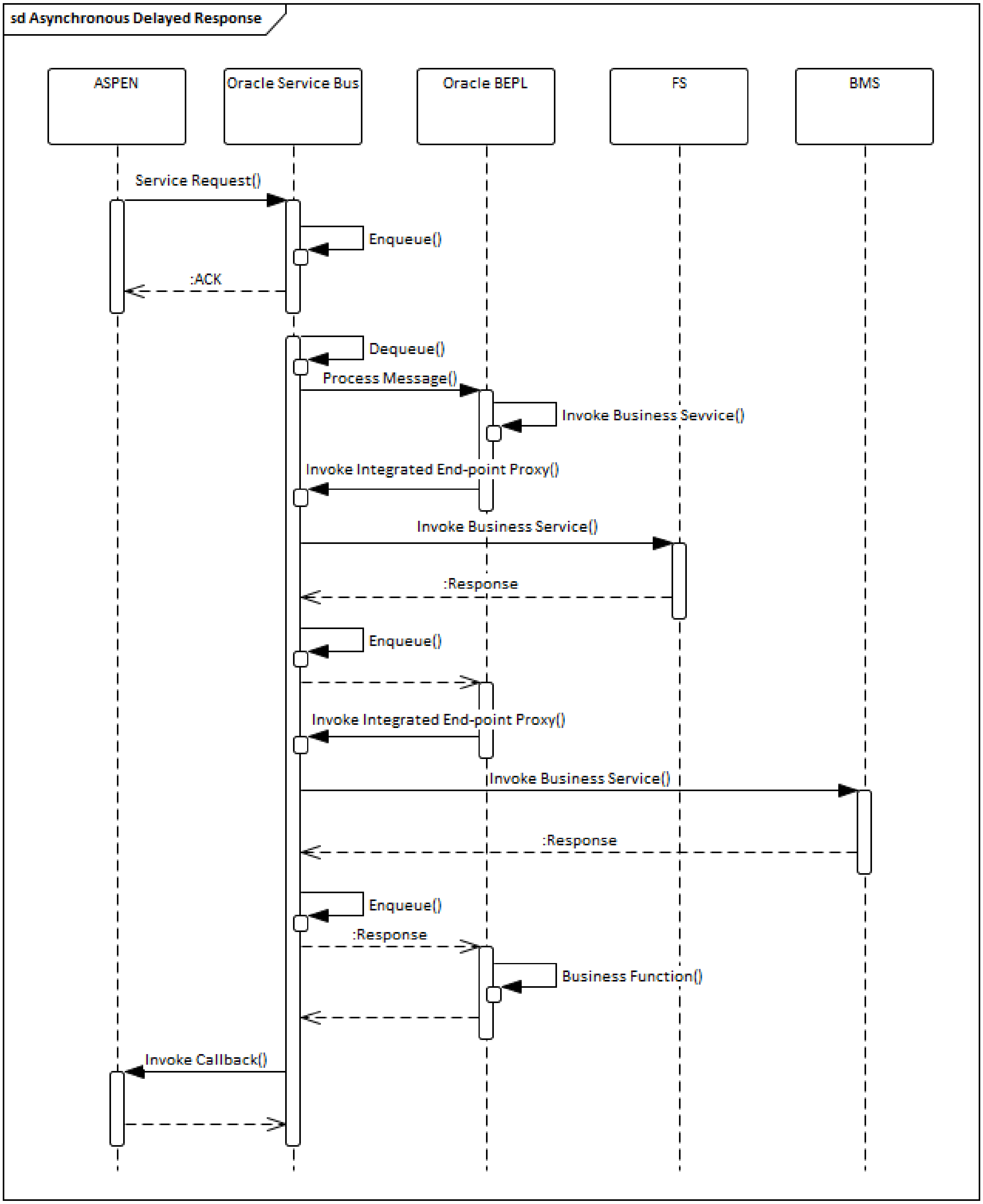
Asynchronous – Delayed Response:

The Asynchronous Delayed Response message exchange pattern is very similar to the above Asynchronous – One-way/Fire-Forget pattern except in this pattern, once the processing is completed, a callback service invokes the source system with the final result.

As a design consideration, the use case Asynchronous message exchange pattern will be preferred and implemented due to its reliability, non-blocking communication, high service availability, and its ability to process large volumes of data. Examples of this pattern are when a source module (FS) wants a process a claim or determine eligibility which does not require an immediate response but needs to process large amounts of data and aggregate data from other HHS 2020 systems via the SI Platform.

On the SI Platform, SOAP/REST web services will implement this message exchange pattern along with JMS Queues and Topics. Below is the high-level sequence diagram of this message exchange pattern.

Figure 6-34: Asynchronous – Delayed Response Communication Patterns


sd Asynchronous Delayed Response
ASPEN
Oracle Service Bus
Oracle BEPL
FS
BMS
Service Request()
Enqueue()
:ACK
Dequeue()
Process Message()
Invoke Business Sevvice()
Invoke Integrated End-point Proxy()
Invoke Business Service()
:Response
Enqueue()
Invoke Integrated End-point Proxy()
Invoke Business Service()
:Response
Enqueue()
:Response
Business Function()
Invoke Callback()

Intermediaries Pattern:

The ESB acts as service intermediary by providing virtual service endpoints to service consumers (i.e., the service or application making a request) and providers (i.e., the service or application that provides functionality) as shown in the figure. That is, the web service consumer makes a request of the virtualized destination (proxy) provided by the ESB. The ESB sends the request to the service provider. The service provider sends a response to the virtual endpoint (provided by the ESB) for the consumer, and the ESB sends the response to the consumer.

The ESB Virtual Endpoint Architecture figure shows the how intermediaries' pattern is applied in Service Virtual layer of the ESB design.

Message Content-based Routing Pattern:

Content-based routing allows the ESB to forward a message to a particular service endpoint depending on the message content. The content-based router EIP tests the name of the item in the message and sends the message to the correct services.

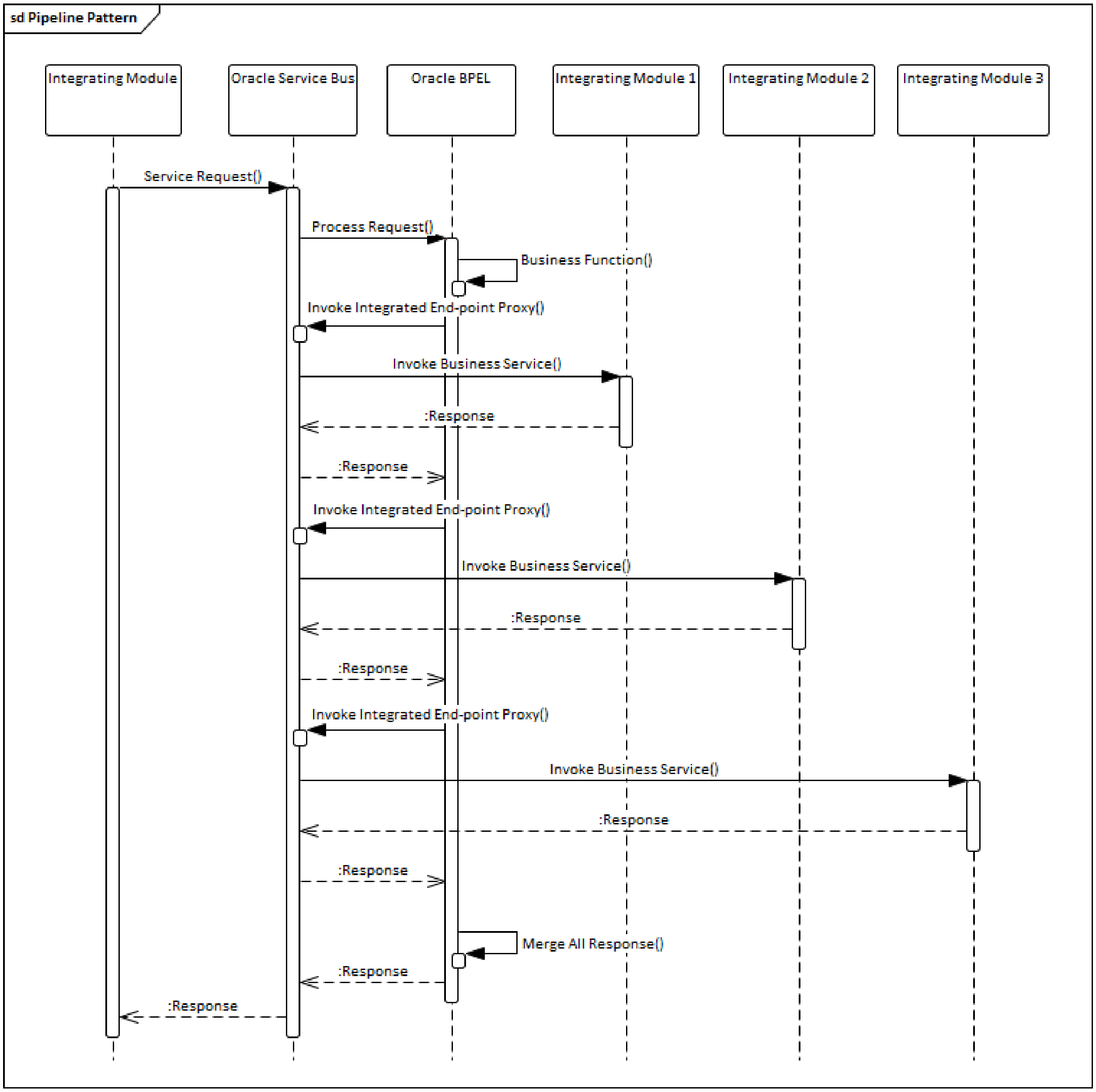
The figure above shows the consumer invoking a service that creates data entities, such as member, provider, and employee. When the service consumer invokes the create Entity service, the consumer embeds the criteria for forwarding the request in the message body or the message header. The ESB parses the message content and routes the message to the appropriate service. While using the content-based routing pattern, ESB platform tests the name of the item in the message and sends the message to the correct services.

Pipeline Pattern:

In many cases, one message routed from a provider to a consumer may require a different transformation than another message routed between the provider and consumer, while routing the same message to a different consumer may require a different protocol. Instead of creating unique services with specific features to support each instance, the ESB offers integration or composite services.

A composite service using the Pipeline is an Enterprise Integration Pattern (EIP) sequentially sends an incoming message to several services, as shown in the figure below. First, the composite service sends the message to the first consumer, and then the composite service sends the first consumer's response to the second consumer. The composite service sends the second consumer's response to the next, etc.

Figure 6-35: Pipeline Pattern



A consumer always sends the same message structure to the pipeline composite service, which is a proxy to the real service. When the service needs to change, the composite service sends the consumer message first to an XSL transformation (to adopt the consumer's message to the new service format), then it sends it to the new version of the service.

Publish-Subscribe Pattern:

The SI Platform supports the Publish-Subscribe Channel EIP, which receives messages from the publisher, and then splits and transmits them among its subscribers through the output channel.

The figures 6-34, 6-35 below shows the example of the event from the UPI makes a REST call to the SI Platform to update the address change of a member. The SI Platform's ESB component received the message transform, enriches, and adds to the appropriate topic to which integrating modules can subscribers can consume.

Figure 6-36: **Subscribe Pattern**

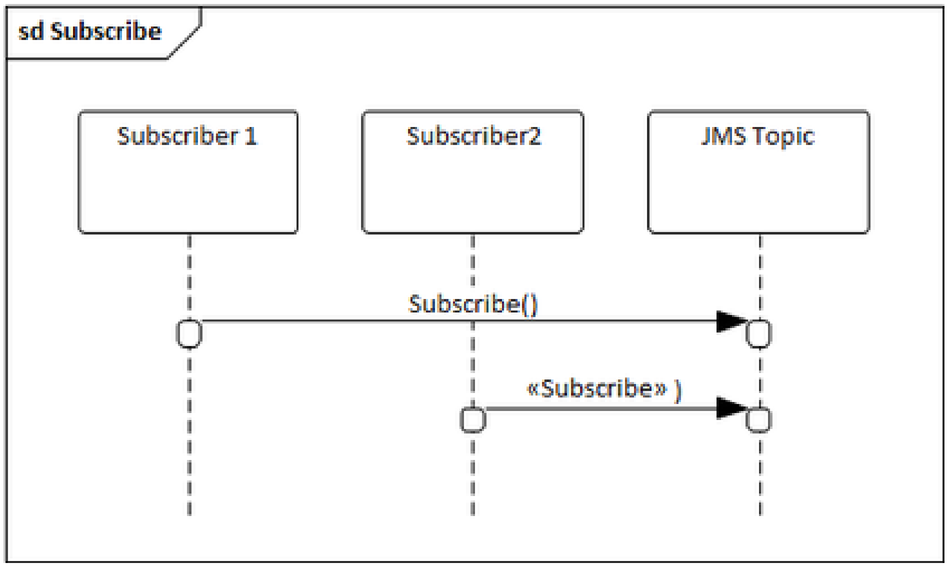
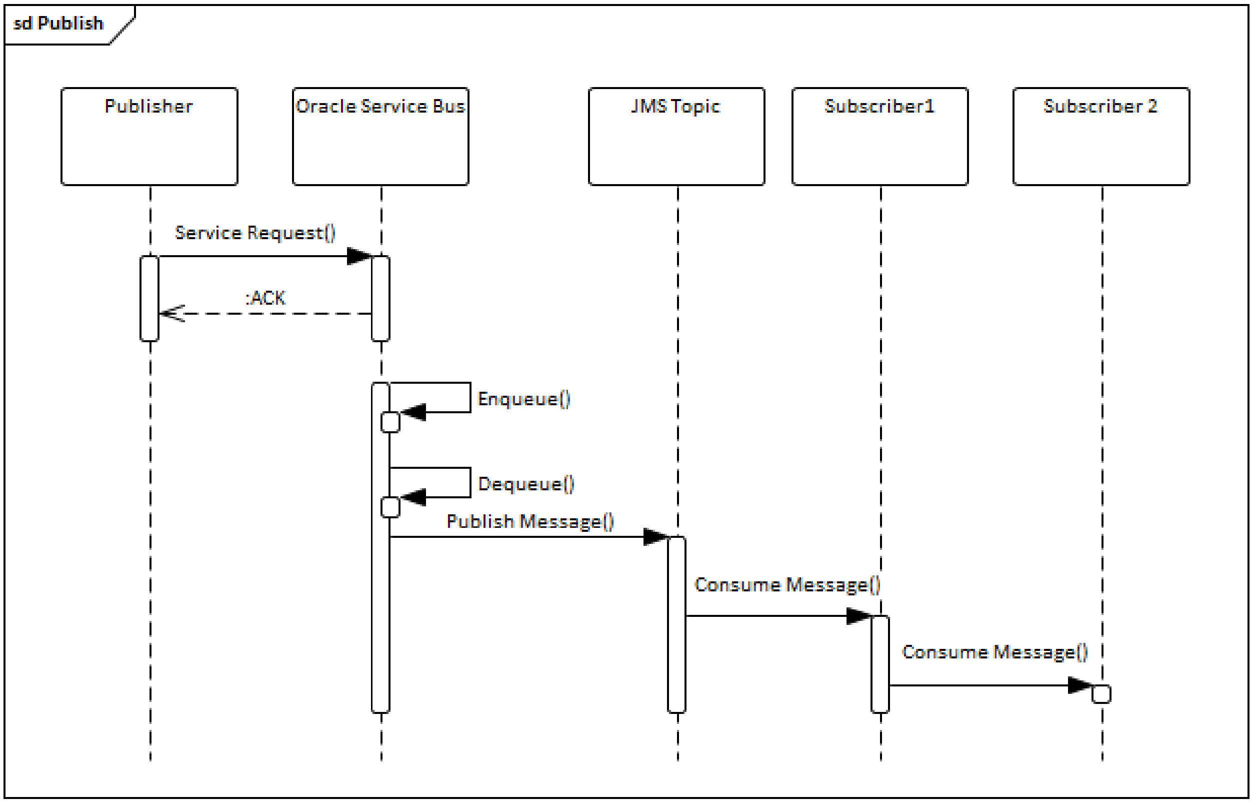


Figure 6-37: Publish Pattern

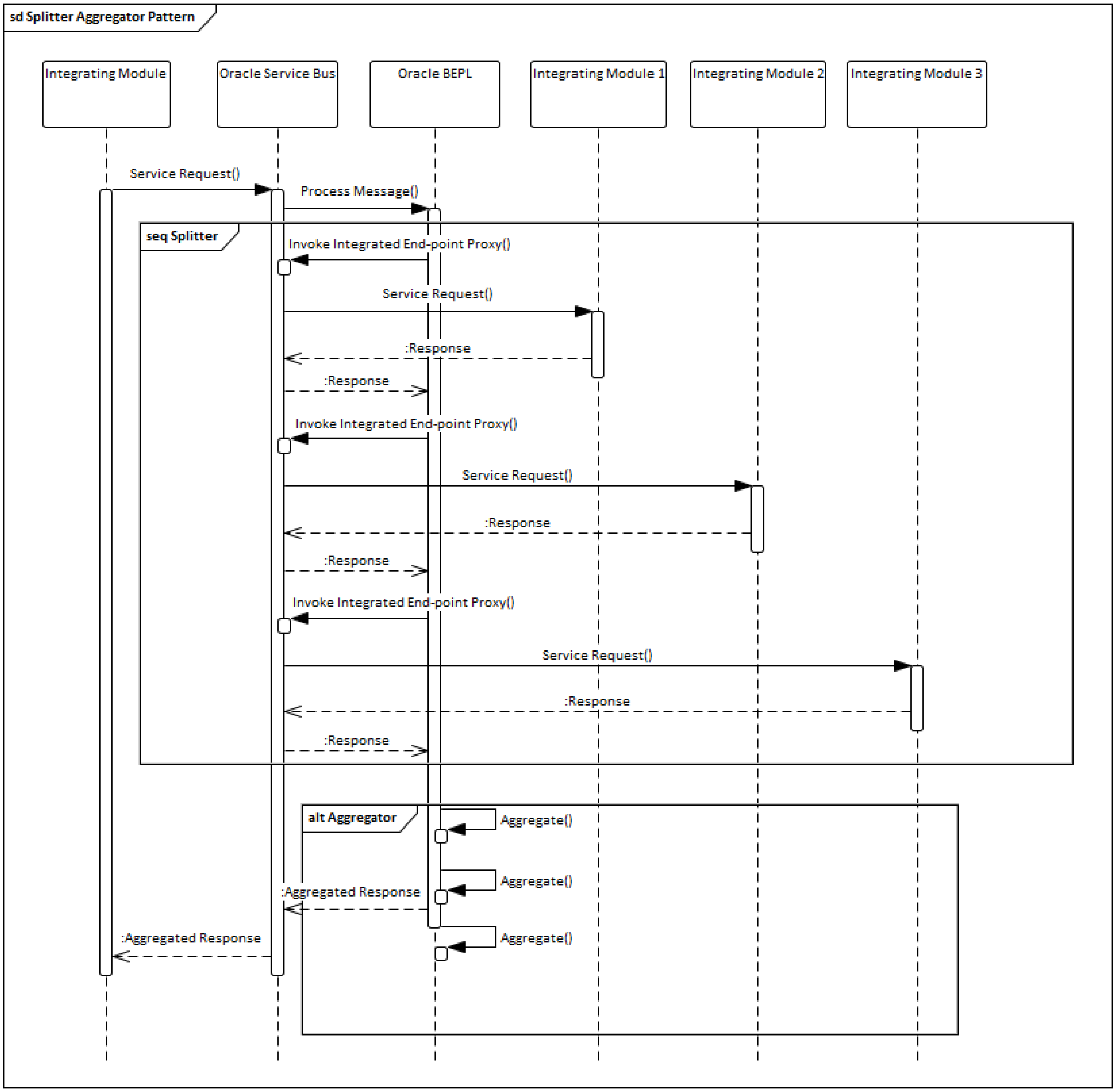


Splitter-Aggregator Pattern:

The integration pattern uses Splitter/Aggregator EIP to routes a request message to a number of service providers and then aggregates the responses into a single response message.

The figure demonstrates an implementation of Splitter-Aggregator EIP that broadcasts a message to multiple recipients the ESB. The ESB uses the Aggregate mediator to collect the responses and merge them into a single response message.

Figure 6-38: Splitter-Aggregator Pattern



6.2.6 ESB Design Considerations

In order to implement ESB best practices and streamline the service integrations, design considerations will be accounted to design and implement the ESB on this project.

6.2.6.1 Service Proliferation

With the transformation and integration of HHS2020 modules (legacy, new) into an SOA, the number of services to integrate HHS2020 modules, via ESB for each entity-based service on each client specific requirement, will grow tremendously. In these situations, teams commonly and naturally focus on their respective process areas and integrations with modules interacting within the scope of their processes, and this leads to a proliferation of services (interfaces) supporting similar functionality, yet triggered from within different functional modules and transactions of HHS 2020 systems.

Considering the example of Claim business service listed below:

- Some clients of the HHS 2020 modules want the whole Claim data with hundreds of attributes.
- Some of clients of the HHS2 020 modules want only claim status from the Claim object.
- Some clients want a mix of Claim attributes
- Some clients want in blocking synchronous mode and some want in a non-blocking asynchronous mode.

The above will result in a number of services and operation type for each client and will lead to service sprawl which in-turn cause's lifecycle burden and cost to document, test, manage, maintain, secure, audit and continuously regenerate them.

In order to avoid the ESB service sprawl issue as explained above the following will be designed and implemented for each entity based (Claim, Provider, User, etc.) service:

- **Supporting Multiple Response Formats** – The request and response of the REST web services will be either in XML or JSON. Services will receive request information as a single string parameter with string contents either as XML or as JSON.

  Responses will be rendered as a single string the same way as the corresponding request. In other words, XML request results in XML response, JSON request results in JSON response. The client has the option to request the response in XML or in JSON by providing the mime type in Accept header in the HTTP request. This allows for flexibility and can serve more clients and thereby not adding a new implementation for each client. The following is the syntax of the header.

  Accept: <MIME_type>/<MIME_subtype> [application/json, application/xml]

  Named Request Query – For Entity Get and Search requests the clients will send selection criteria in the request consisting of array of name-value pairs. The name parameter will be a predefined enumerated string from which the client will select and pass corresponding values. The service will accept entity get request parameter consisting of selection criteria.

  In the name-value pair, the client will pass the selection criteria. For example, to get a claim of a specific beneficiary by first and last name the client will pass 'First_Name' and 'Last_Name' as the name parameters and the corresponding values in the value parameter. The name parameter of the array will be an enumeration and client will select from a predefined name enumeration (First_Name, Last_Name, ClaimID, etc.) joined by logical expression AND. Following is how the selection criteria will look like for each request.

  First_Name='John' AND Last_Name='Doe'

  ClaimID='1234'

- **Response Set** – For Entity Get and Search requests the client will have the option to select the response they want from an enumerated response set. Instead of returning the complete response for every client request even though they may be interested in only a portion of it, they can choose a portion of the entity object or the full entity object.

  These selections are predefined in an array of enumerated list and need to be passed by the client when making the request. This limits the service sprawl as we do not have to implement new services for each client requirement. For an e.g. a claim an entity is associated with a provider entity. If one is looking for claims and requesting attributes of provider information to be included in the response, the provider instances would only be those related to claims that match the request criteria. The following are examples of the response strings, which a client needs to select to get the response they want for a Claim Entity.

  - Whole Claim Entity
  - Claim and Claim History
  - Claim and its Providers
  - Claim attributes alone

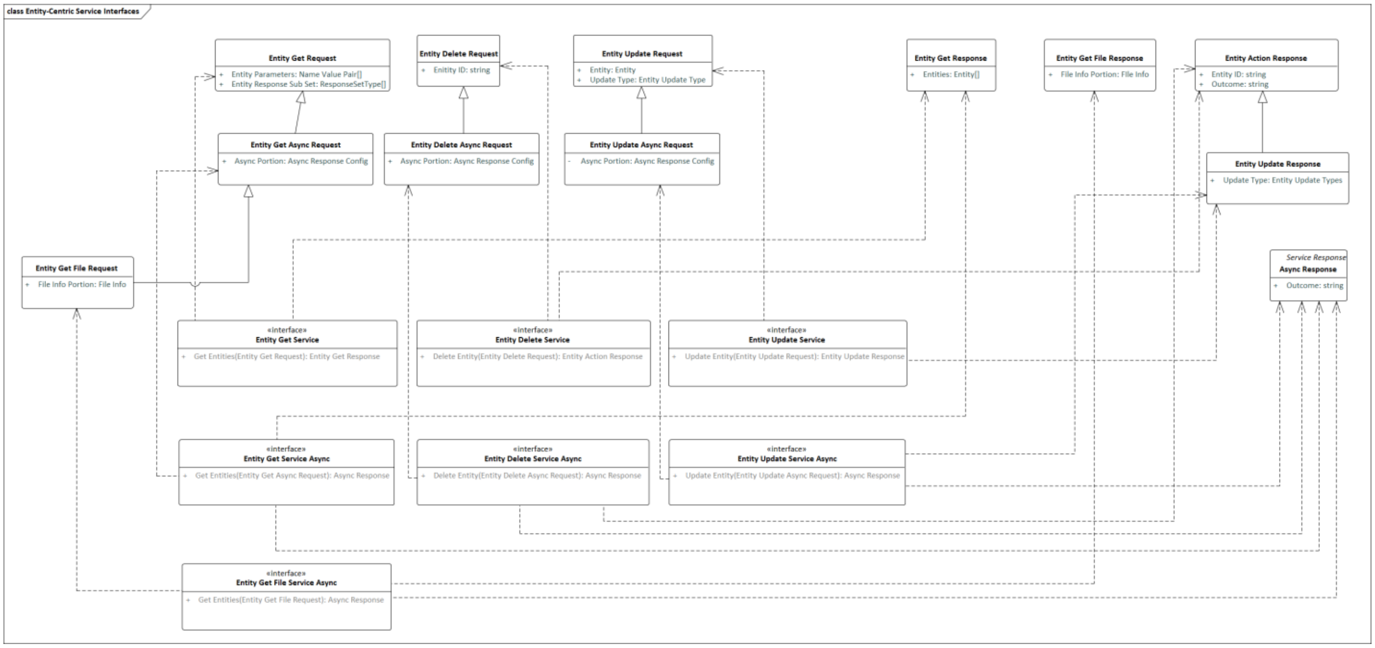
- **Limiting Entity Based Service Implementations** – Implementation of CRUD (Create, Read, Update, and Delete) operation services for each entity will be limited to one. These services will be either synchronous or asynchronous or both depending on client specific requirements.

  The following are the possibilities of services for each entity.

  - Synchronous message based get Entity
  - Synchronous message-based Update Entity
  - Synchronous message based Delete Entity
  - Asynchronous message based get Entity
  - Asynchronous message file based get Entity
  - Asynchronous message file based get Entity
  - Asynchronous message based Update Entity
  - Asynchronous message based Delete Entity

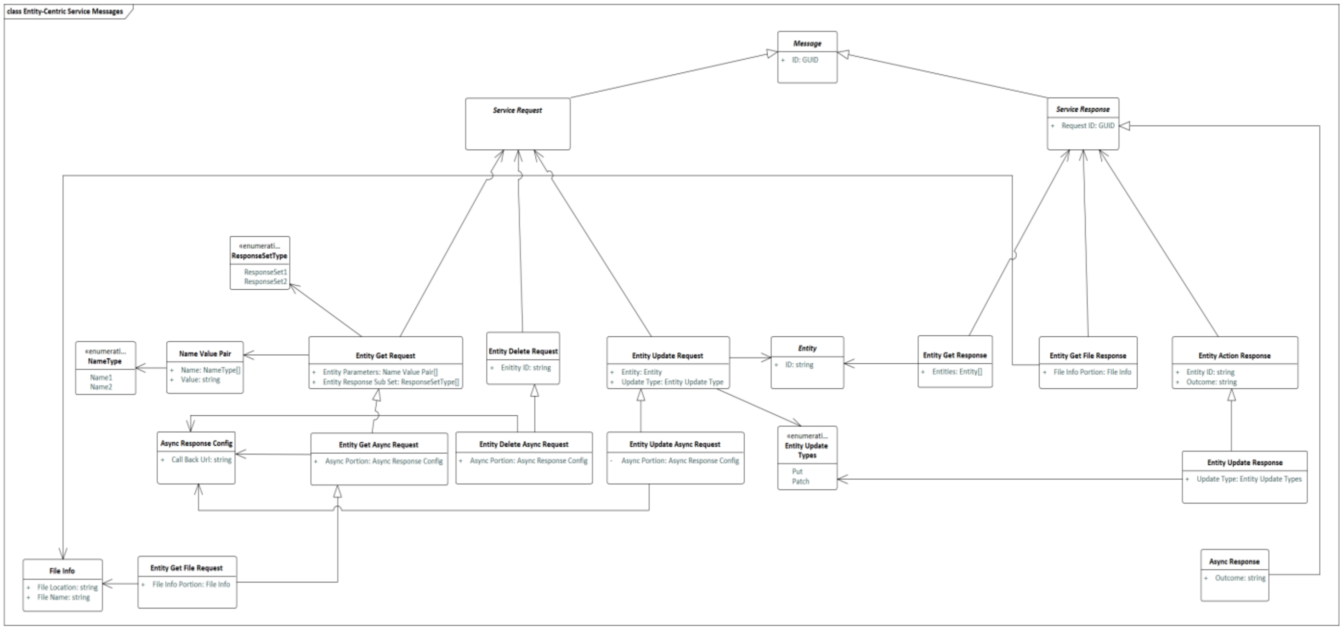
Following is the class diagram of all the seven (7) operations for each entity-based service.

Figure 6-39: Entity-Centric Service Interfaces


class Entity-Centric Service Interfaces
Entity Get Request
+ Entity Parameters: Name Value Pair[]
+ Entity Response Sub Set: ResponseSetType[]
Entity Delete Request
+ Enitity ID: string
Entity Update Request
+ Entity: Entity
+ Update Type: Entity Update Type
Entity Get Response
+ Entities: Entity[]
Entity Get File Response
+ File Info Portion: File Info
Entity Action Response
+ Entity ID: string
+ Outcome: string
Entity Get Async Request
+ Async Portion: Async Response Config
Entity Delete Async Request
+ Async Portion: Async Response Config
Entity Update Async Request
- Async Portion: Async Response Config
Entity Update Response
+ Update Type: Entity Update Types
Entity Get File Request
+ File Info Portion: File Info
Service Response
Async Response
+ Outcome: string
«interface»
Entity Get Service
+ Get Entities(Entity Get Request): Entity Get Response
«interface»
Entity Delete Service
+ Delete Entity(Entity Delete Request): Entity Action Response
«interface»
Entity Update Service
+ Update Entity(Entity Update Request): Entity Update Response
«interface»
Entity Get Service Async
+ Get Entities(Entity Get Async Request): Async Response
«interface»
Entity Delete Service Async
+ Delete Entity(Entity Delete Async Request): Async Response
«interface»
Entity Update Service Async
+ Update Entity(Entity Update Async Request): Async Response
«interface»
Entity Get File Service Async
+ Get Entities(Entity Get File Request): Async Response

Following is the message class hierarchy of the seven operations for entity-based services.

Figure 6-40: Entity-Centric Service Messages



Following are the implementation details of each entity-based operation based on the above class and message hierarchy diagrams.

Synchronous Entity Get:

- This service satisfies two purposes, Entity Search and Entity Get. The first is searching for entities that match specified criteria and the second is to retrieve a specific entity instance.
- The service will accept entity get request parameter consisting of selection criteria. The request parameter will be as per the **"**Named Request Query**"** criteria as defined above in this email. Please refer the **"**Named Request Query" definition for more details.
- The second portion of the request is an array of enumerated response strings that specify schema elements of the requested entity that needs to be brought back. For more details on this please refer *"Response Set"* definition defined above in this section.
- The service would be invoked synchronously, and the client would be blocked until an instance of Entity Get Response is returned.

Asynchronous Entity Get:

The asynchronous version of entity get service will return immediately to an instance of asynchronous response object that provides information to the caller about acceptance (or rejection) of the request. The asynchronous request is derived from the synchronous request and has the same request parameters as described above in the synchronous entity get operation, with addition of call-back URL, which the service implementation is going to call once the results have been obtained and properly encoded. The message sent to the call back URL is exactly the same return result that would have been obtained from a synchronous version of this service.

Asynchronous File-Based Entity Get:

The file-based version of the entity get service is invoked asynchronously, and rather than eventually calling the client with a message containing the requested entity instances, it generates a file containing requested instances and simply informs the caller about the availability of the file. It takes in the same request parameters as the synchronous get entity operation with the addition of file details, which the service implementation creates and places at a specific location as defined in the request.

Synchronous Entity Delete:

In this operation, each delete request will be based on a unique identifier for the entity that is to be deleted. The caller is blocked until a delete response message is sent back. Bulk deletes of the entities will not be allowed and has to be one entity at a time.

Asynchronous Entity Delete:

The asynchronous version of delete releases the client immediately and calls back with delete completion on the specified call-back URL.

Synchronous entity update:

Like deletes, updates are geared at one entity at a time. The updates will take care of both "*insert*," if the entity does not exist and "*update,"* if the entity already exists. The request message will contain the entity that is being inserted or updated. A service implementation will examine which attributes are present in the supplied entity object, thus no need for additional XPath attributes explaining what information is present in the request. This implementation will support the granularity of update of PATCH, equivalent to a database update statement and PUT, the equivalent of delete if it exists, plus insert database statements.

Asynchronous Entity Update:

The asynchronous entity update works in the same fashion as all other asynchronous versions.

6.2.6.2 Point-Point vs. Canonical Service Integration

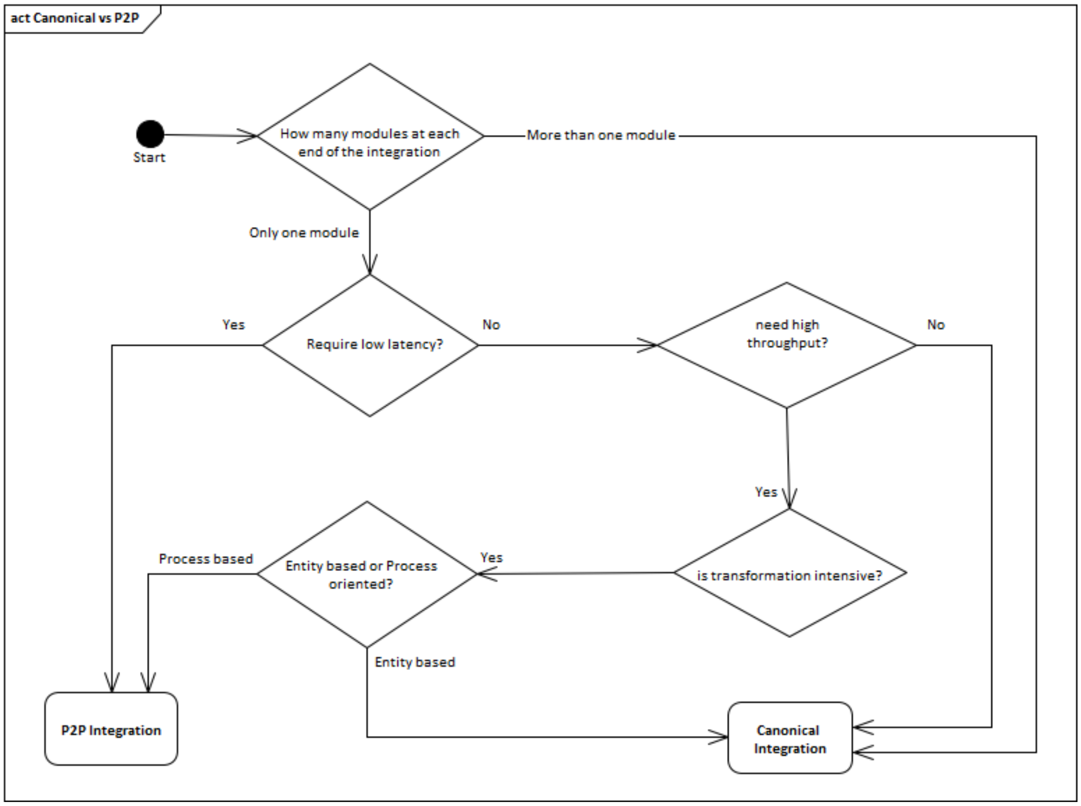
Usage of canonical data objects for service integration is a best practice, especially in integrations that involve connectivity with multiple sources and destination modules of HHS 2020 systems. Loose-Coupling through a canonical (application-independent) model is one design pattern that can be implemented in the context of an SOA to simplify the integration when there are either many instances of the same or similar participating modules. Instead of direct mappings between data models, transformations are used to map to the canonical data model. But most importantly, the impact of changes on the overall integration based on the introduction of changes to one of the participating applications is isolated. This effectively brings the cost of maintaining these more complex integration scenarios in line with the costs of a single point-to-point integration.

However, usage of a canonical data model does introduce some overhead and might introduce additional development effort. While this allows for greater reusability, the transformations both increase the message size and consume more computing resources. For this reason, choosing canonical data objects for service integration should be based on some important factors in-order not to introduce overhead, extra development, and maintenance cost across the ESB platform. The decision tree, below, illustrates an approach for deciding between a direct point-to-point service integration or canonical data object-based service integration. This approach will be used when building services on the ESB Platform.

- Examples of canonical service integration model are the cases where the API contract overlaps with the established canonical models, like Prior Authorization, Member (Client) update, Third Party Liability (TPL) information sent to Pharmacy subsystems.
- Examples of P2P integration model includes the cases where the API contract includes only a small subset of information from the canonical model and it does not merit having the entire canonical model in the message. Weekly electronic payment information submitted to Wells Fargo, Early and Periodic Screening, Diagnostic and Treatment (EPSDT) reports and notices, daily updates from CareLink NM system containing specific health information about certain subsection of clients are some applicable use cases for this model.

Specific and detailed examples of both Canonical and P2P models will be provided as part of the ESB Detailed design.

Figure 6-41: Canonical vs. P2P Integration



6.2.6.3 Synchronous Vs Asynchronous Processing

Following is the criteria of when Synchronous or Asynchronous communication pattern will be implemented. This depends on the requirements and the scenario being implemented.

Synchronous Processing:

The SI Platform will implement this communication pattern whenever the source systems need an immediate response and if it is meant to run for a short duration. Examples of this pattern are when a source system is checking a claimStatus or getCaseStatus, getClaimDetails, etc. This will be implemented mostly as a SOAP or REST Web Service.

Asynchronous Processing:

The asynchronous pattern is effective if the end user does not need immediate feedback. In this situation, the requester sends the request message and sets up a callback via a callback URL for a response. The requester does not wait for the response after sending the request message. A separate thread listens for the response message. When the response message arrives, the response thread invokes the appropriate callback, and processes the response.

The service will have a pair of operations-one for sending the request and another for receiving the response. Both the operations are independent and atomic. The providing service, after ensuring that the request is serviced invokes the response EBS. The response EBS pushes the response to the requested service waiting for the asynchronous response. If an error occurs in the providing service, the response is sent with fault information populated.

Asynchronous processing will be used for process-based services, which take a long time to complete and for which no immediate response is needed. Some of the examples of process-based services are Process Claims, ProcessEnrollments, TerminateEmployee, etc.

6.2.7 Enterprise Shared Services

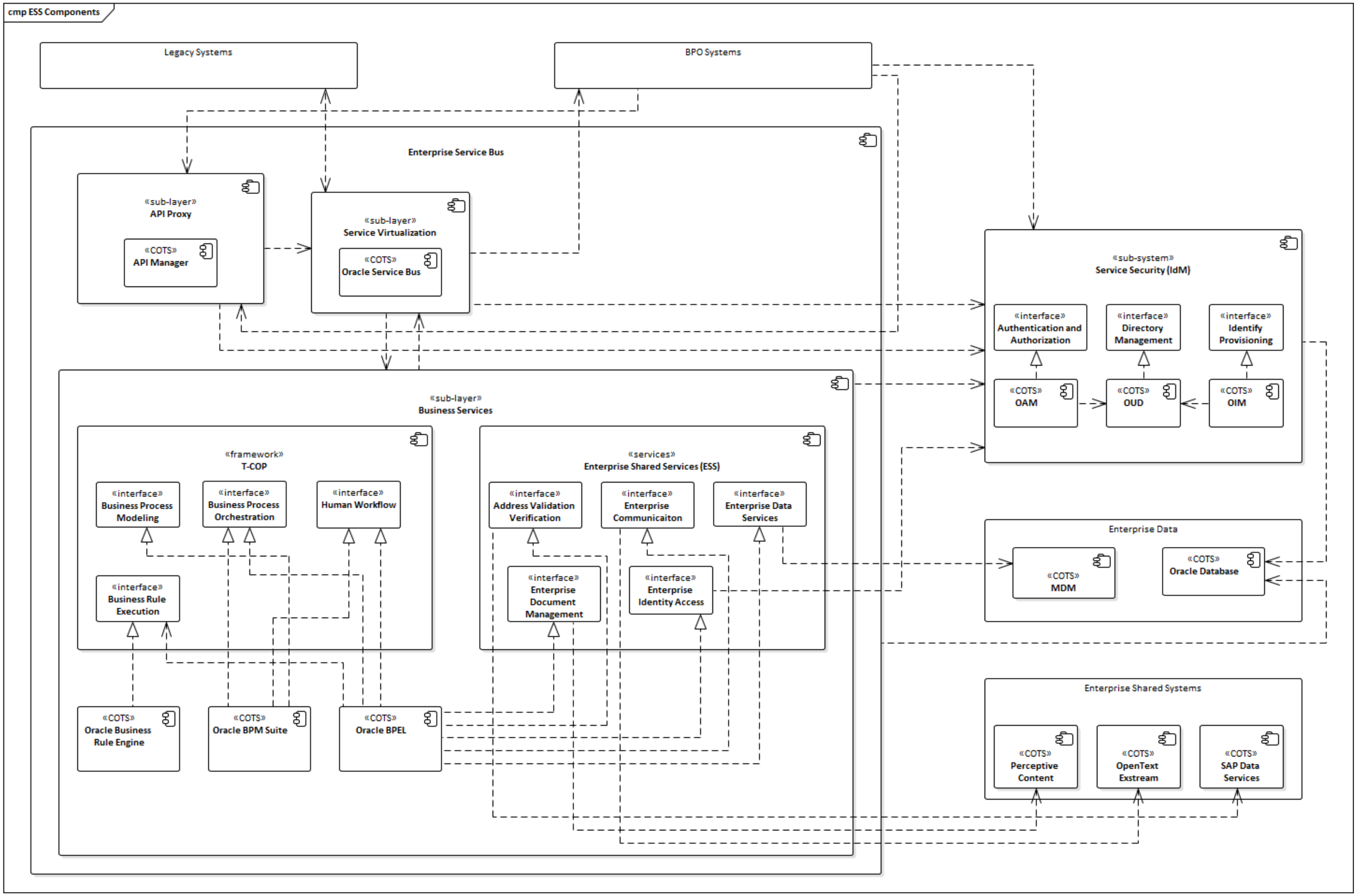
The purpose of Enterprise shared services is to create enterprise-wide services around the functionality provided by the shared services tooling like address standardization, document management, communications management, and master data management. The components and tools of these enterprise-shared services reside outside of the BPO and legacy backend systems. The ESS enables access to COTS systems in a consistent, centralized, and mediated fashion. The enterprise service publishes a set of APIs through the SI Platform that follows SOA architecture. The creation of enterprise services around shared services tooling provides an added benefit of enterprise level security, Auditing, Logging, and Monitoring apart from providing enterprise level APIs that are based on HHS 2020 enterprise data model.

The following is the architecturally significant requirements addressed in this subsection.

- Facilitate integration with access to services for data sharing between applications and entities, in accordance with service contracts and security policies.
- Shared services implement shared behavioral and non-behavioral functional requirements and are intended for broad re-use across business processes.

The diagram below shows component architecture enables the creation of Enterprise Shared Services.

Table 6-32: Enterprise Shared Services Component Diagram



Note: The interfaces like Business Process Modeling, Human Workflow, Authentication & Authorization, etc., depicted in the above figure are for representational purpose only. These are the capabilities of the tools, which are internal to the tool itself and not be exposed as services to other components. Enterprise Shared Services are composite services, which make use of the capabilities, depicted in the model above, to facilitate service integration with enterprise shared systems.

Enterprise shared services are comprised of the following set of services:

- Enterprise Document Management (EDM): EDM is a set of enterprise level APIs that expose functionalities of the document management tooling: Perceptive Content.
- Enterprise Communication Management (ECM): ECM is a set of enterprise level APIs that expose functionalities of the communication management tooling: OpenText Exstream.
- Enterprise Address Verification Service (EAVS): EAVS is a set of enterprise level APIs that expose functionalities of Address Standardization, Validation and Verification tooling: SAP Data Services.
- Enterprise Data as a Service (EDAS): EDAS is a set of enterprise level APIs that expose functionalities of the Master Data Management (MDM) implementation.
- Enterprise Identity Access Service (EIAS): EIAS is a set of enterprise level APIs that exposes Identity and access related APIs to provide a robust security framework that can be used by any HHS 2020 enterprise application for application security. The EIAS is not covered in this subsection but addressed as part of ESB security design.

As depicted in the above diagram, The API manager publishes enterprise shares services as proxy services that are delegated to corresponding service implementation in business services layer. The business services components use adapters to call tool specific APIs to communicate with the respective tools.

This subsection emphasizes how the tools and MDM implementation is integrated though ESB to publish the enterprise level APIs. A detailed design about the tools and the APIs published by enterprise services is covered in the design documents for each of the enterprise shared service work streams.

6.2.7.1 Alignment with PADU Approach and MITA Technical Strategy

The SI Contractor will use the preferred COTS tools as identified in the NM HHS 2020 MITA Technical Strategy document such as Oracle Fusion Middleware to integrate with SAP Data Services for Address standardization, Perceptive Content for document management, OpenText Exstream for communications management, master data management based on MarkLogic Smart Mastering library for enterprise data as service, and Oracle IdAM for service security (IdAM) and implement a standards-based interface to provide access to these enterprise systems to other modules.

Since Oracle Fusion enables to achieve maximum configuration and minimum effort for developing WSDLs, XSLT transformations and usage of COTS as enterprise services, the alignment with PADU for this framework is deemed as “Preferred” level.

6.2.7.2 Enterprise Document Management (EDM)

NM HSD uses Hyland-Perceptive Content (formerly ImageNow) as the content management solution. The SI Platform caters to the need of HHS 2020 enterprise by seamlessly integrating Perceptive Content and publishing enterprise level APIs that are SOAP/REST based. The integration enables various NM HSD enterprise applications to search, upload, retrieve, index, and annotate documents with access restricted by security roles. This subsection provides an overview of the integration of Perceptive Content with the SI Platform and how it will be used for enterprise applications. The tool's specific details and its integration at functional level is covered in SIPLT20 Design Document – EDM.

EDM is an enterprise application used by HSD to manage documents received through multiple channels.

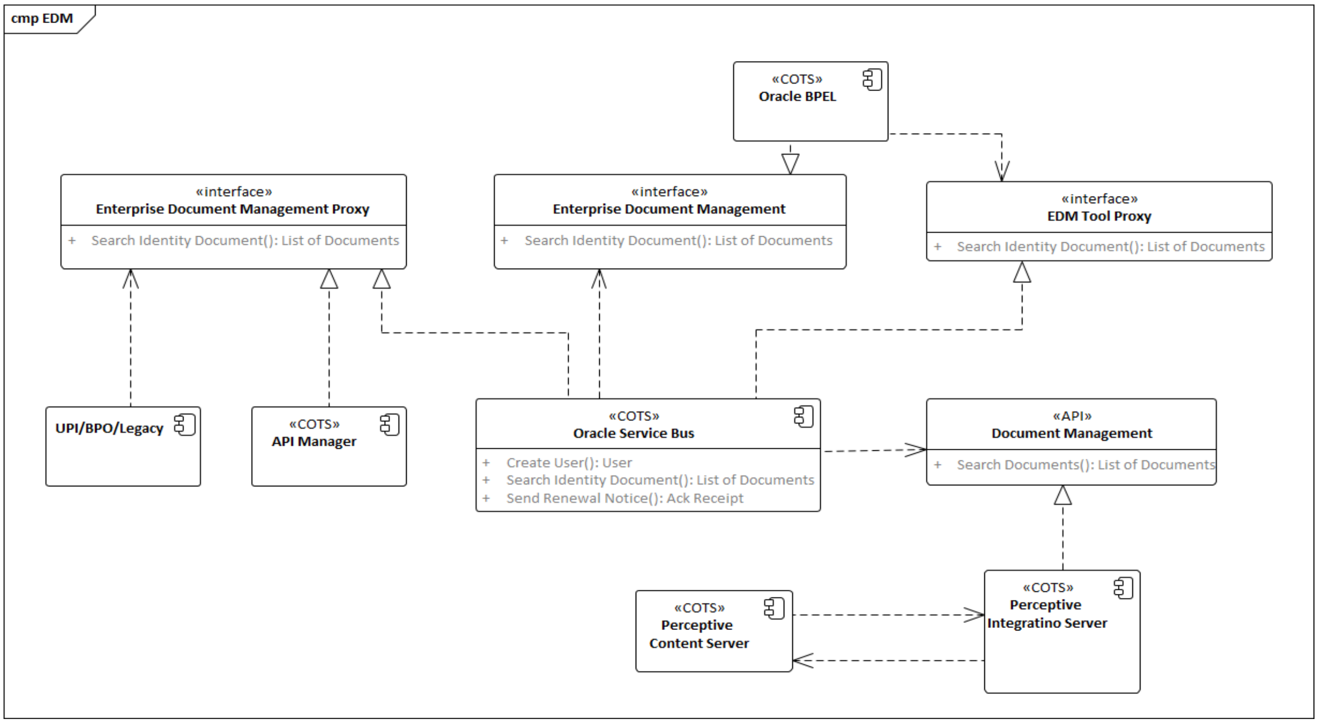
Perceptive Content Tool Capabilities:

The following are the capabilities of Perceptive Content tool for EDM.

- Uses document scanners and/or copier machines to scan paper documents.
- Captures metadata (indexed data) from scanned documents.
- Provides work queues and workflows to route documents.
- Uses the content repository for storing and archiving scanned documents and digital files.
- Provides role-based access to features, functions, and documents.
- Uses keywords in index fields to direct the search and retrieval of documents.

The diagram below provides a high-level component diagram of how EDM is setup and integrated with ESB.

Figure 6-42: **Enterprise Document Management Component Diagram**


cmp EDM
«COTS»
Oracle BPEL
«interface»
Enterprise Document Management Proxy
+ Search Identity Document(): List of Documents
«interface»
Enterprise Document Management
+ Search Identity Document(): List of Documents
«interface»
EDM Tool Proxy
+ Search Identity Document(): List of Documents
UPI/BPO/Legacy
«COTS»
API Manager
«COTS»
Oracle Service Bus
+ Create User(): User
+ Search Identity Document(): List of Documents
+ Send Renewal Notice(): Ack Receipt
«API»
Document Management
+ Search Documents(): List of Documents
«COTS»
Perceptive
Content Server
«COTS»
Perceptive
Integratino Server

The components in the diagram are explained below:

Perceptive Content Server – This is the execution engine that manages all the core functionalities related to document management, *this tier is wrapped by a rich set of web services that enable customers to integrate Perceptive content infrastructure within their larger enterprise systems. Perceptive content server act as a server to support UI client applications for perceptive content software. The server enables the user to configure a workflow for a document.*

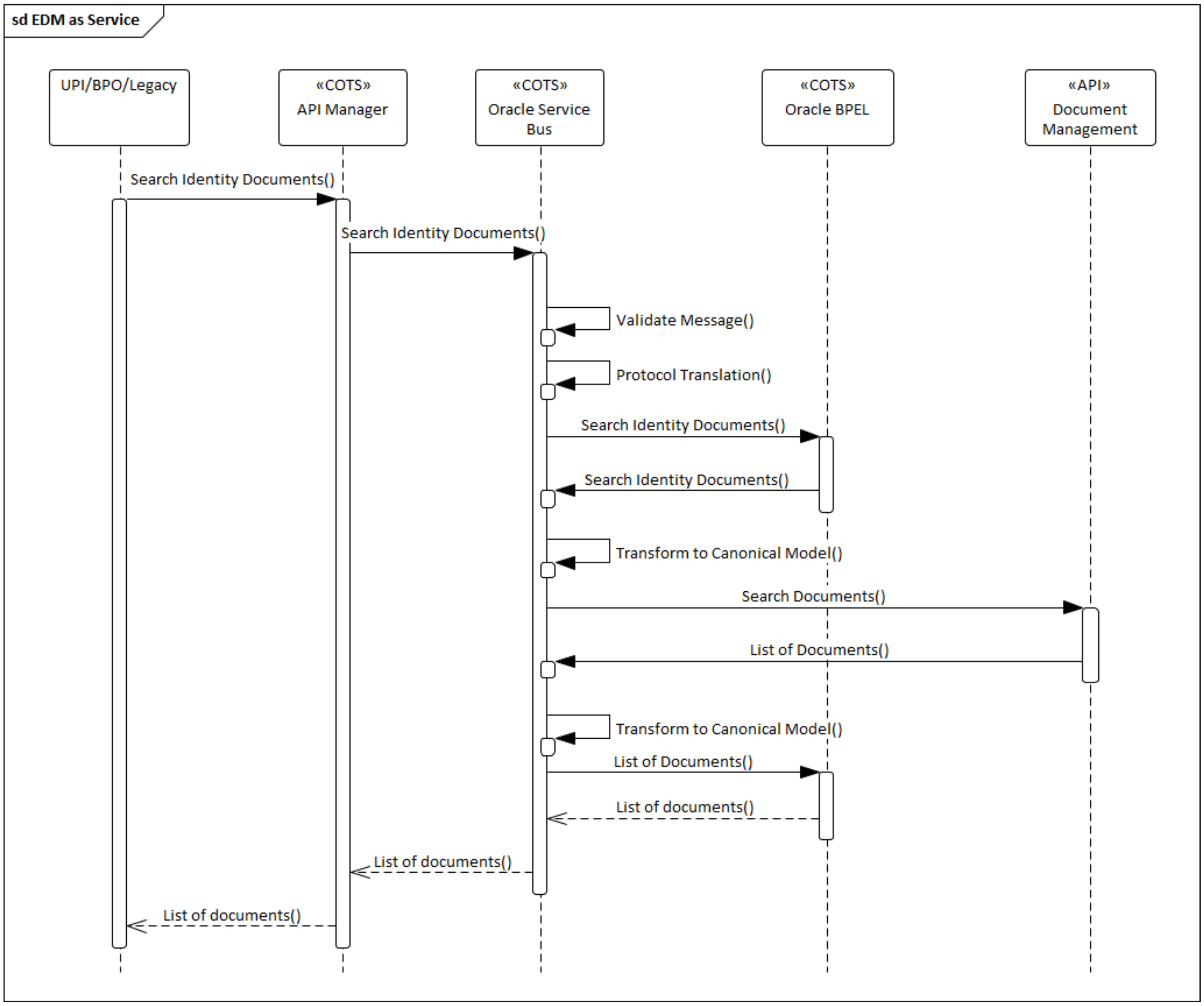
Integration Server – This is a middle-tier web service that provides communication between Perceptive Content Server and ESB. The architecture supports asynchronous and synchronous communications using standard XML and JSON Representational State Transfer (REST) message formats. Integration Server interacts seamlessly with Perceptive Content Server to increase efficiency. *The integration Server communicates with the perceptive content server using a proprietary protocol whereas it communicates with ESB using REST over HTTP. Integration server publishes a set of document management REST APIs that is leveraged by ESB to communicate with the integration server.*

ESB – This acts as a mediation layer between Perceptive Content and the integrating systems that want to use Perceptive Content for document management. ESB provides multiple services, which includes protocol translation, message conversion, auditing, logging, security, and SLA monitoring to facilitate communication between integrating systems and EDM. The ESB services are exposed using the API Manager component. The API Manager delegates the call to Business services layer through OSB. The component at business layer implements business logic for the functional call. The ESB supports Web Services for Remote Portlets (WSRP) standards to facilitate easy integration of WSRP based portal application with ESB.

Modules Interaction with the EDM Tool:

Consider a typical business scenario to understand the sequence of actions in more details. Also, consider a scenario whereby a UPI user wants to search documents managed by document management tool for a given search criterion.

Figure 6-43: Enterprise Document Management Sequence Diagram



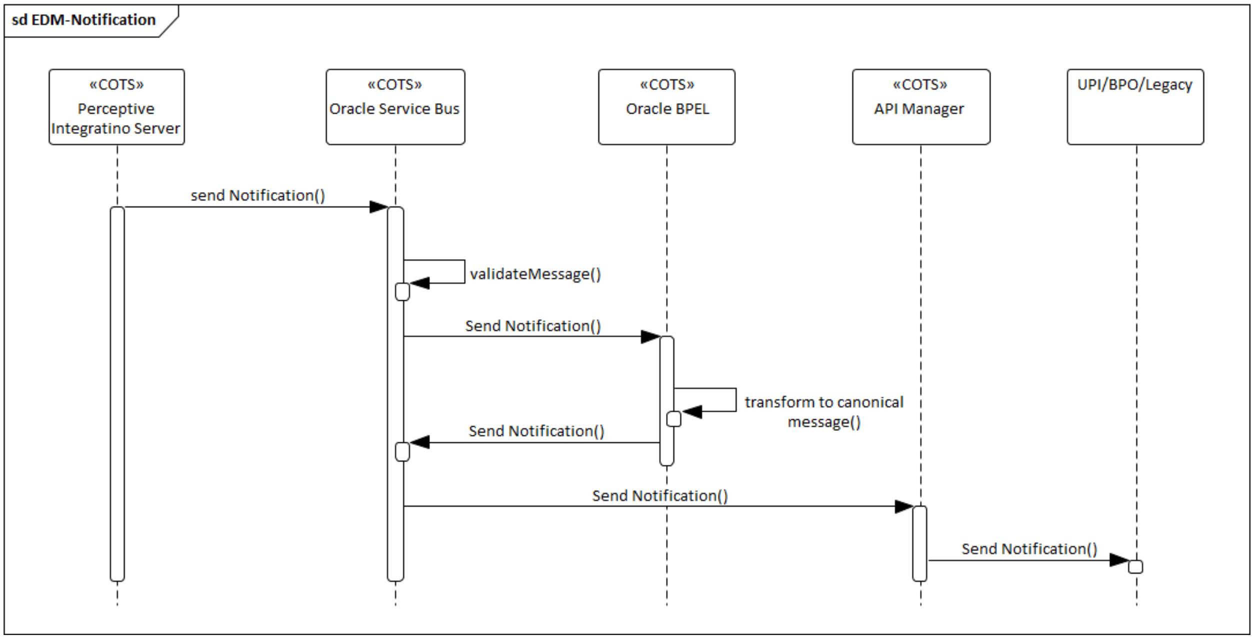
The above diagram shows an interaction between various components of ESB and document management tool when any enterprise participating in HHS 2020 wants to use enterprise document management APIs to search documents matching a given search criterion.

An enterprise application participating in HHS 2020 calls an enterprise document management API published on ESB by API Management component. The API Manager call OSB, which in turn makes a call to Business services to delegate the request to Perceptive Content tool. Business services calls proxy services published on OSB to delegate the call to API published by Integration server, a Perceptive Content component, to communicate the request to Perceptive Content Server.

Notification Propagation from EDM Tool to BPO Modules:

An EDM tool user can do certain actions, which may require to be notified to the registered modules. For example: If identity documents of a person are scanned by NM HSD case worker and submitted to EDM tool, a notification needs to be sent to multiple BPO modules for eligibility determination as well as enrollment purposes. The diagram below depicts a flow representing the above scenario.

Figure 6-44: Notification Propagation from EDM Tool



The diagram depicts Integration server sending notification to the ESB which in turn is getting propagated to the required modules, be it UPI, BPO modules or Legacy modules.

EDM User Interface Single Sign-On (SSO) integration:

Perceptive Content provides user interface for administrators and the end users. Perceptive Content provides the ability to utilize SSO Integration for URL integration. URL integration will be used to launch Perceptive Content from a link embedded in another module (for example, UPI) of the HHS 2020 system. Authentication of the Perceptive Content via the SSO will be delegated to the SSO provider (IDAM). Authorization for the EDM tool UI will be managed from Perceptive Content Management Console to determine user access privileges. In order for the SSO to work with Perceptive Content user names must exist in both Perceptive Content Server and in the SSO provider (IDAM).

Perceptive Content is designed to integrate with the SSO Provider Oracle IDAM, a stateful session is introduced outside of the Perceptive Content system. Oracle IDAM will manage the lifecycle of this session, determines if the user is logged in, and manages timeouts for the session.

The basic user story for Perceptive Content SSO integration follows this general pattern:

- User requests a Perceptive Content resource (for example, a perceptive content URL).
- An Oracle IDAM SSO provider intercepts the non-authenticated request and either prompts or redirects the user for credentials.
- An Identity Provider (IdP) Oracle IDAM authenticates the credentials against the user store. If the user authenticates successfully, the IdP directs the user to the originally requested resource.
- requested resource loads and reads the pre-authenticated username from the Oracle IDAM SSO provider. This is typically communicated via an HTTP header.
- Perceptive Content server checks for the privilege assigned to the user before rendering the requested resource.

Perceptive Content Workflows integration with Enterprise Workflow:

In-built and Custom workflows within the Perceptive Content software will be used and implemented to coordinate with HHS2020 module business processes. For documents added, updated, and deleted via the Webservice APIs or via the scanner into ImageNow where a human action is needed, a workflow will be created with tasks for case workers to view and act. These tasks will be routed to the case worker queues in the ImageNow portal and once these tasks are performed then source HHS 2020 modules business processes will be updated via ESB API calls.

The workflows in ImageNow are created by developers/administrators during design time based on specific HHS2020 module requirements. At run time these workflows will be triggered via an action like adding, updating and deleting a document via ESB APIs or when a document is captured into ImageNow from the scanner. At runtime no workflows will be created via API calls.

6.2.7.3 Enterprise Communication Management (ECM)

NM HSD uses OpenText Exstream (formerly HP Exstream) as the Enterprise Communication Management (ECM) solution. The SI Platform caters to the need of HHS 2020 enterprise by seamlessly integrating Customer Communication Management using SOAP/REST-based API calls. The integration enables various NM HSD enterprise applications to use a single application to send emails, notices, and alerts using different channels (physical, email, and text). This subsection provides an overview of the integration of OpenText Extream with the SI Platform. The tool-specific details and its integration at a functional level are covered in the SIPLT21 Design Document – CCM.

ECM is an enterprise application that will be used by NM HSD to send notices, alerts, forms, and other communications by various channels (physical mail, email, text, other) to external consumers, providers, payers, as well as internal Stakeholders.

OpenText Exstream Tool Capabilities:

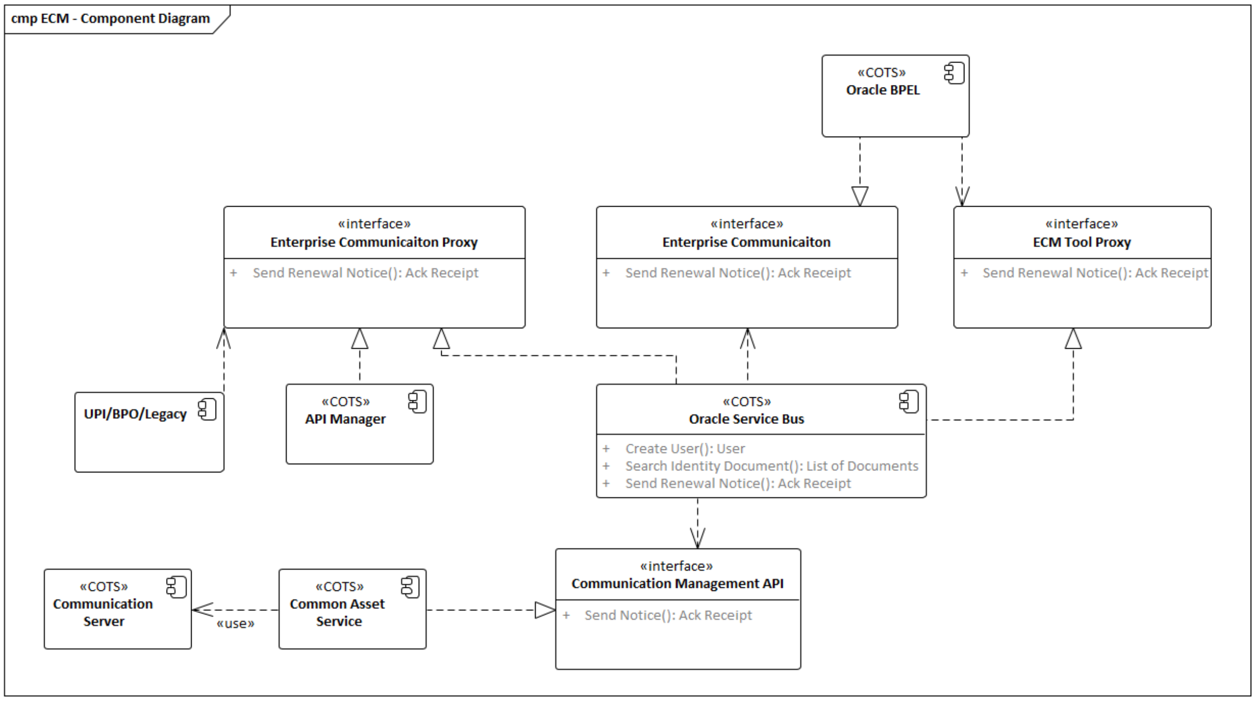
The following are the capabilities of OpenText Exstream tool for ECM.

- Send emails and notices through multiple channels including physical, social, email and text.
- Track electronic communications delivery and respond automatically to failures or re-deliver content through fail-over channels.

- Provide a tool to create custom templates that can be used for communication.
- Provide functionality to create a custom form and publish it.
- Handle transactional as well as batch communication.

The diagram below provides high-level component diagram of how ECM tool is setup and integrated with ESB.

Figure 6-45: Enterprise Communication Management Component Diagram


cmp ECM - Component Diagram
«COTS»
Oracle BPEL
«interface»
Enterprise Communicaiton Proxy
+ Send Renewal Notice(): Ack Receipt
«interface»
Enterprise Communicaiton
+ Send Renewal Notice(): Ack Receipt
«interface»
ECM Tool Proxy
+ Send Renewal Notice(): Ack Receipt
UPI/BPO/Legacy
«COTS»
API Manager
«COTS»
Oracle Service Bus
+ Create User(): User
+ Search Identity Document(): List of Documents
+ Send Renewal Notice(): Ack Receipt
«interface»
Communication Management API
+ Send Notice(): Ack Receipt
«COTS»
Communication Server
«use»
«COTS»
Common Asset Service

The components in the diagram are explained below:

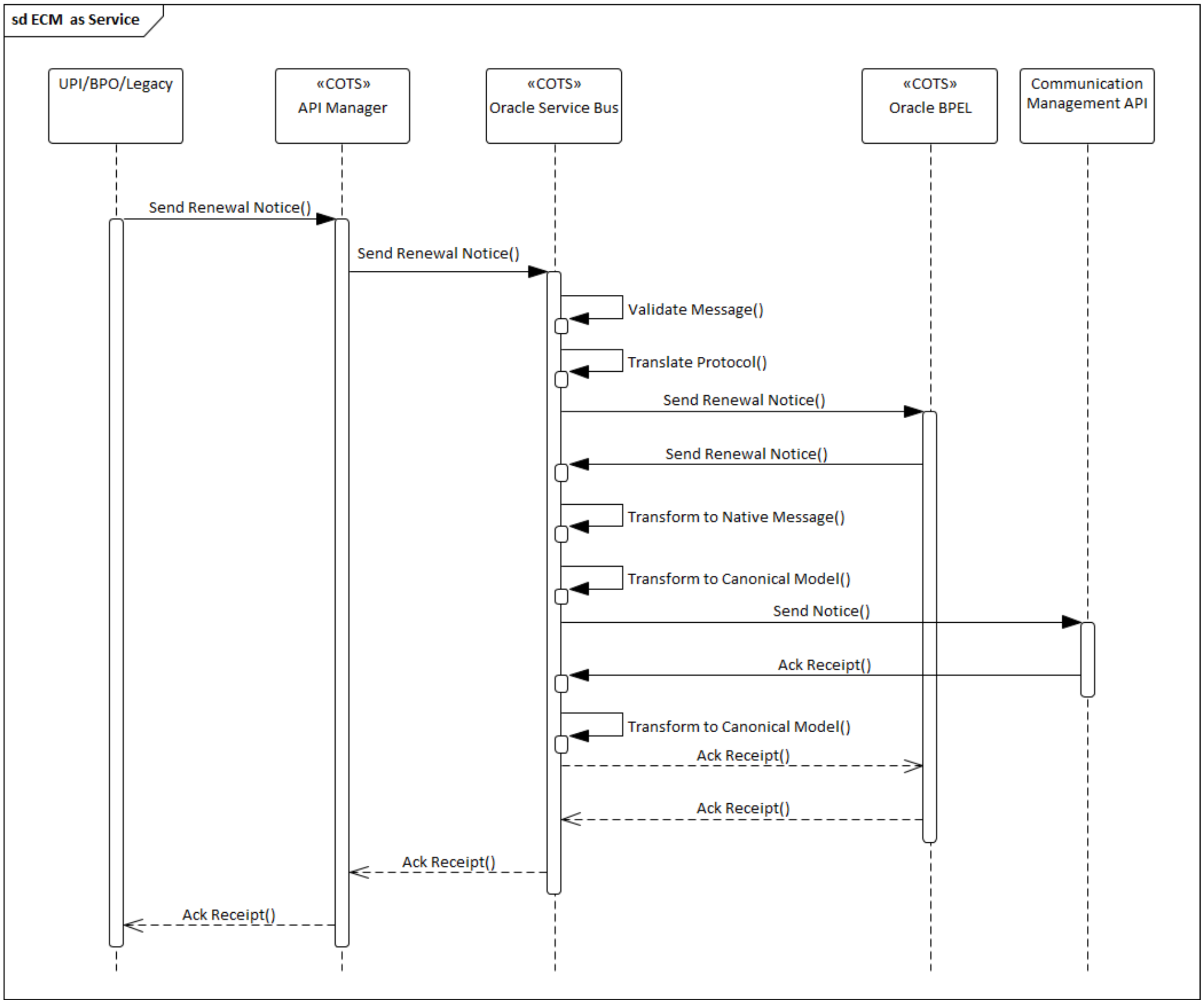
Communication Server: The Exstream Communication Server provides a highly flexible and scalable environment for event-driven integration to source systems. It can scale from small, departmental projects to complex enterprise-wide deployments. Each phase of the customer communications process orchestrated by Exstream Communication Server is staged in queues, persisting in a centralized database.

Common Asset Service (CAS): CAS is part of Extream solution. CAS is a multi-tenant content service layer which provides central access to and storage for resources used in Exstream solutions. Whenever a resource is created, updated, or accessed it is done through the CAS. CAS provides REST endpoints to access services published by Communication Server.

ESB: ESB acts as a mediation layer between CAS and clients that want to use the ECM tool for customer communication. ESB provides multiple services, which include protocol translation, message conversion, auditing, logging, security, and SLA monitoring to facilitate communication between integrating systems and the ECM tool. The ESB services are exposed using the API Manager component. The API Manager delegates the call to the business services layer through OSB. The component at business layer implement business logic for the functional call. ESB supports Web Services for Remote Portlets (WSRP) standards to facilitate easy integration of WSRP based portal application with ESB.

Consider a typical business scenario to understand the sequence of actions in more details. Also, consider a scenario whereby an enterprise needs to notify a Medicaid enrollee to renew the eligibility as his Medicaid eligibility has expired. The sequence diagram below illustrates the scenario mentioned above.

Figure 6-46: Enterprise Communication Management Sequence Diagram



The above diagram shows an interaction between various components of ESB and communication management tool when an enterprise application calls an enterprise communication proxy API to send notification to a client for renewing eligibility.

An enterprise application calls an enterprise communication proxy API published on ESB by API Management component. The API Manager call OSB, which in turn makes a call to Business services to delegate the request to communication management tool. Business services calls proxy services published on OSB to delegate the call to API published by CAS, an OpenText Extream component, to communicate the request to Communication Server.

6.2.7.4 Enterprise Address Verification Services (EAVS)

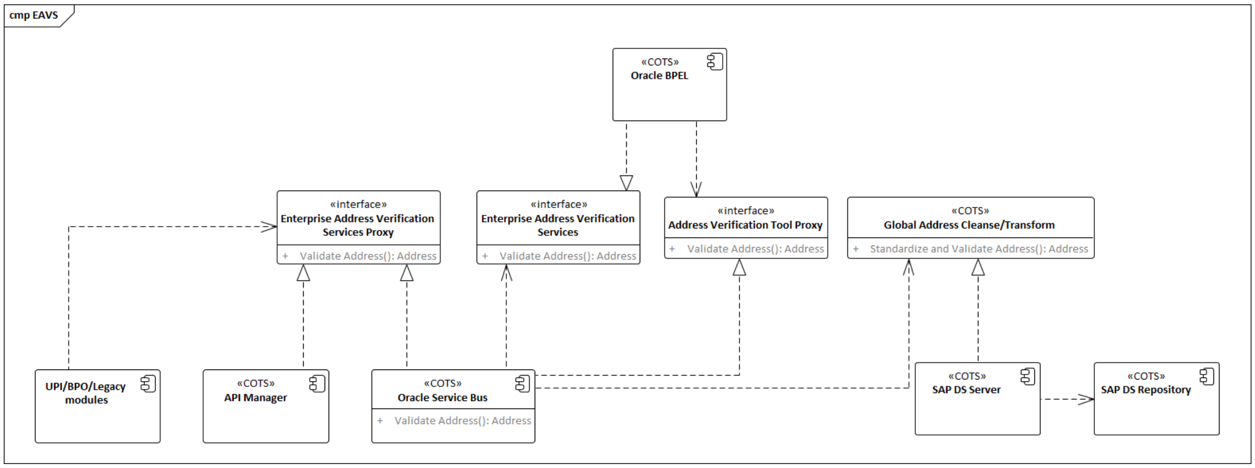
The EAVS provides address standardization, verification as an enterprise service across HHS 2020 systems. A third-party tool, SAP Data Services, enables the EUS. The following diagram provides high-level component diagram of the EAVS software modules and the realized interfaces. The EAVS enterprise solution supports both real-time, atomic transactions and batch mode involving bulk cleansing of addresses.

SAP Data Services Tool Capabilities:

The following are the capabilities of SAP Data Services for EAVS.

- **Address Standardization**:
    - The Global Address Cleanse transform of SAP Data Services takes as input any address information and matches that, using different engines, against its Address Dictionary.
    - The transform uses its internal knowledge how address lines are written to parse it into its segments and also corrects typing errors.
- **Address Validation:**
    - Customers can use Global Address Cleanse (GAC) Suggestion Lists module for real-time address validation.
    - GAC provides error-tolerant search to find a list of similar addresses if a perfect match is not found.
    - The user enters a misspelled address name, selects a record from the suggestions, enters a house number, and GAC returns the full address.
    - If the user only inputs a city name that contains more than one postcode, the GAC Suggestion List provides a postcode list.
    - When user inputs an address which contains address line information (street name, street number, building, floor/unit/wing/stairwell etc.), the GAC suggestion list also drills down step-by-step to a unique address.
    - Provides following search methods.
        - LIKE search (for example, Ber* → Berlin)
        - Phonetic search (for example, Ph → F)
        - Fuzzy search (for example, Berlen → Berlin)

Figure 6-47: Enterprise Address Verification Service Components


cmp EAVS
«COTS»
Oracle BPEL
«interface»
Enterprise Address Verification Services Proxy
+ Validate Address(): Address
«interface»
Enterprise Address Verification Services
+ Validate Address(): Address
«interface»
Address Verification Tool Proxy
+ Validate Address(): Address
«COTS»
Global Address Cleanse/Transform
+ Standardize and Validate Address(): Address
UPI/BPO/Legacy modules
«COTS»
API Manager
«COTS»
Oracle Service Bus
+ Validate Address(): Address
«COTS»
SAP DS Server
«COTS»
SAP DS Repository

Application Layer:

The ESB component, decoupled from the actual SAP data services layer, provides web services integration to send and receive data from the SAP Data services. The ESB component includes API Manager and Oracle Service Bus. The ESB component acts as the proxy for the web services published internally from the SAP Data services servers. The ESB proxy layer provides additional capabilities like security, protocol conversion on top of the core features produced by SAP data services. The web services support both real-time and batch integrations.

Administrators will configure SAP Data Services through the Administrator to publish jobs as callable web services, and then applications can start consuming these services via ESB.

SAP DS Server Layer:

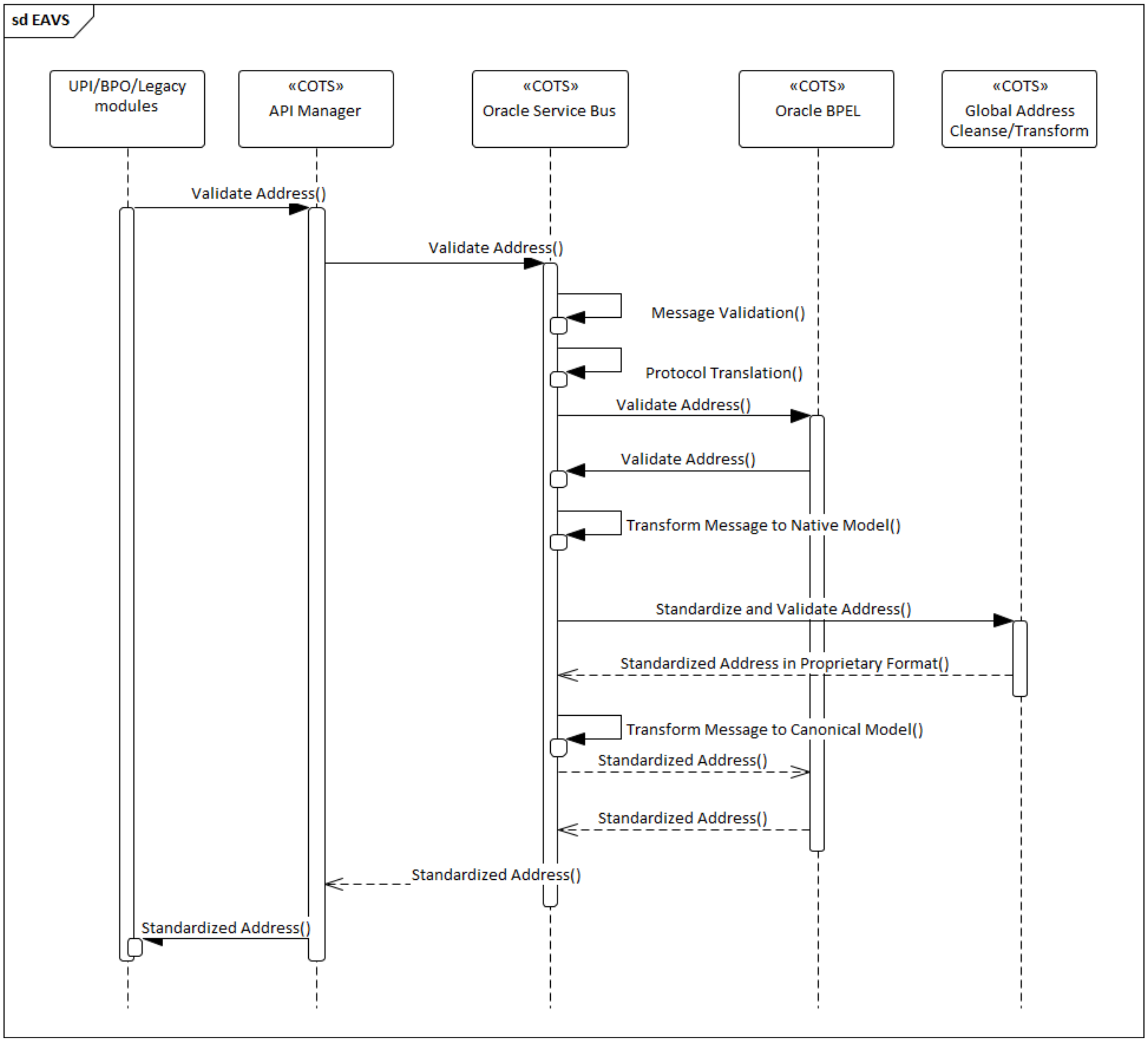
This layer performs the functions of the address standardization with the help of the following internal components:

- Job Server – The SAP Data Services Job Server starts the data movement engine that integrates data from multiple heterogeneous sources, performs complex data transformations, and manages extractions and transactions. The Job Server retrieves the job from its associated repository, then starts an engine to process the job.
- Access Server – The SAP Data Services Access Server is a real-time, request-reply message broker that collects message requests, routes them to a real-time service, and delivers a message reply within a user-specified period. The Access Server queues messages and sends them to the next available real-time service across any number of computing resources. This approach provides automatic scalability because the Access Server can initiate additional real-time services on additional computing resources if traffic for a given real-time service is high.
- Repository – The SAP Data Services repository is a set of tables that hold user-created and predefined system objects, source and target metadata, and transformation rules. Each repository is associated with one or more Job Servers, which run the jobs you create. The SAP DS repository realizes the address cleansing and transformation features provided by the suite.

Real-Time Processing of Address Validation Request:

The following sequence shows a subset of a member enrollment workflow, involving standardization of a member's address.

Figure 6-48: Real-Time Address Validation Processing



As depicted in the figure above, the communication protocol will be HTTP RESTful web services across all components as it involves new MMISR modules like UPI and the SI Platform built on top of the Oracle fusion middleware suite. REST mechanism is used consistently across the platform wherever supported and is preferred to other mechanism like SOAP. However, SOAP will still be supported by the ESB layer to provide enterprise wide support and internally performs protocol translation between SOAP and REST.

Batch Processing of Address Validation Request:

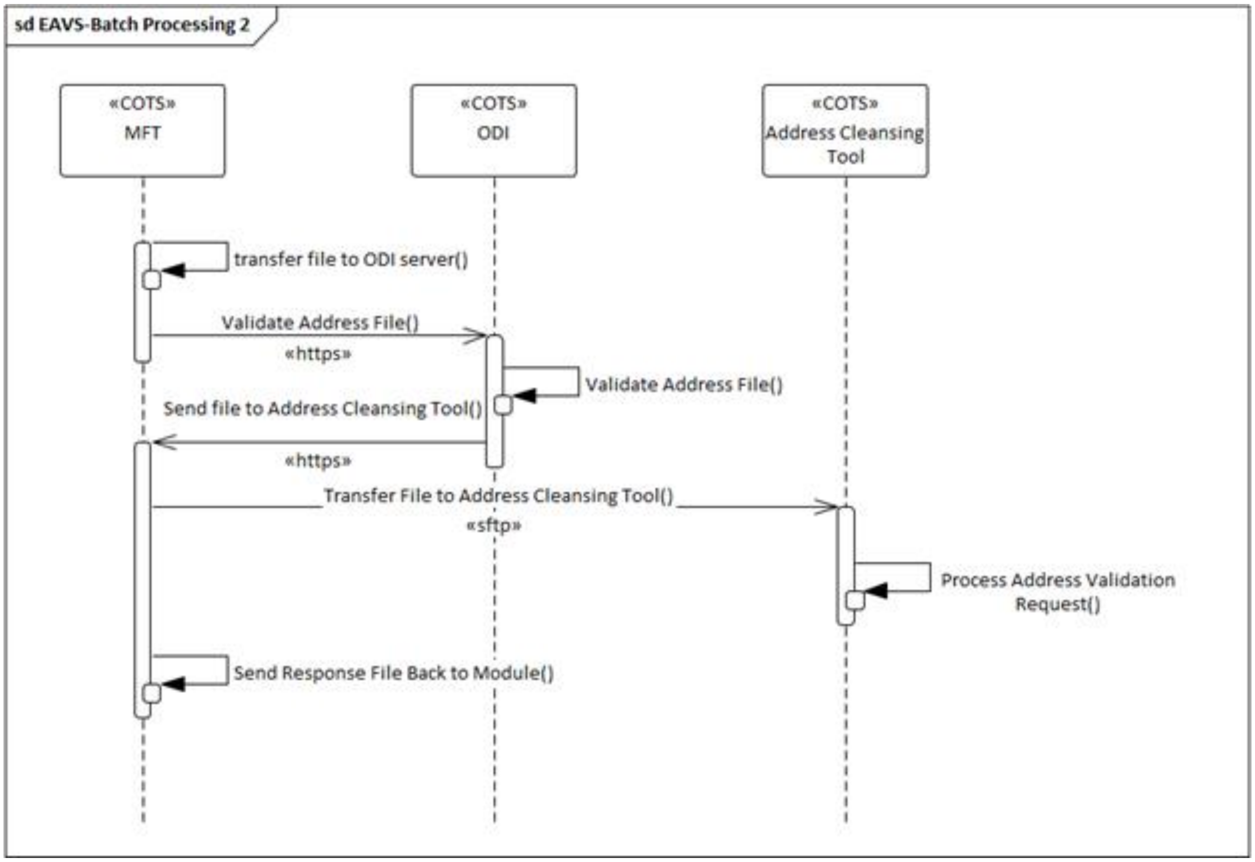
Integration platform supports batch processing of addresses. The IP receives the file containing a batch of addresses and process them using SAP Address cleansing tool. IP will leverage batch processing capabilities of the Address cleansing tool to process the batch requests.

IP expects each address to have a unique identifier in the batch file. The response file will contain the unique identifier for each response tying each response to the request. The response file will contain two kind of records:

- Validated Address: These are standardized as well as validated addresses. The tool confirms that it's a valid address in this case.
- Error Address: If an address could not be validated then the response will contain error reason returned by the tool.

The diagram below depicts a typical flow of how a request for validating a batch of address is processed and a response is returned to the requester

Figure 6-49: Batch Processing of Address File


sd EAVS-Batch Processing 2
«COTS»
MFT
«COTS»
ODI
«COTS»
Address Cleansing Tool
transfer file to ODI server()
Validate Address File()
«https»
Validate Address File()
Send file to Address Cleansing Tool()
«https»
Transfer File to Address Cleansing Tool()
«sftp»
Process Address Validation Request()
Send Response File Back to Module()

A scheduled job on MFT server transfer file to ODI server from the FTP server of the requesting system over SFTP. The MFT job triggers ODI job for validating address file once the file transfer is completed. The ODI server does basic validation to ensure it meets the IP published data dictionary and it confirms to the format that is acceptable by Address Cleansing tool. In case the requesting system does not have ability to provide the file in the format that Address Cleansing tool require, ODI can do required transformation.

The Address Cleansing tool process the request file and generate a response file, which is picked up by a schedule job of MFT and transferred back to the SFTP server of requesting system.

6.2.7.5 Enterprise Data as Service (EDAS)

The Enterprise Data as a Service is an entity service function implemented on top of the MarkLogic MDM solution. EDAS will establish relationships among entities and preclude access to information about one entity, e.g. Member by another Entity, for example, the State Employee if both entities are part of the same household or family unit.

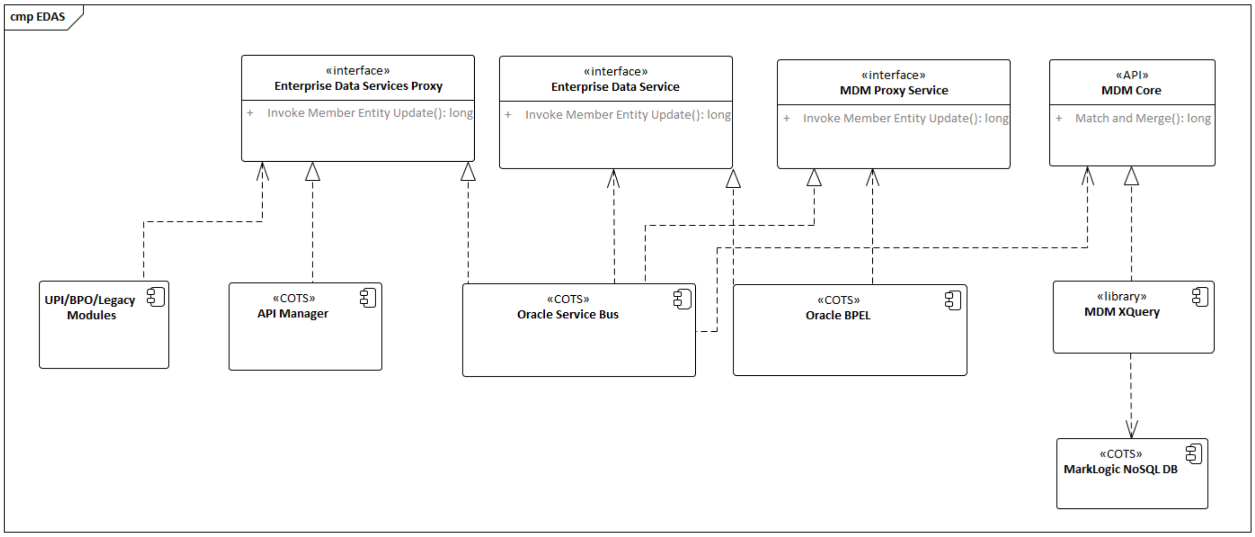
MarkLogic MDM Framework Capabilities:

The following are the MDM Framework Capabilities.

- MMISR modules queries for mastered entity data in MDM first.
- Part of every workflow, before the transaction finishes, MDM is updated.
- Data in MDM is current up to the last transaction.
- MDM notifies of data updates to each module via ESB.
- Data Stewardship happens at MDM.
- MDM becomes the Source of Truth for mastered data.
- Modules are still responsible to synchronize their information with MDM.

The diagram below provides high-level component diagram of how MDM is integrated with ESB.

Figure 6-50: Enterprise Data as Service Components



MDM External API:

The MDM external APIs are hosted on the ESB Layer. As in other cases, the ESB enables the HHS 2020 modules to consume functionality implemented in services while being oblivious of the technical details of the components involved. The ESB performs core functions at the message level to implement service transparency, and thus achieves a service-oriented architecture. These core functions are explained in detail in other subsections of this document (provide link) and will not be covered here.

MDM API Core (MAC):

All functions and algorithms of the MDM are available through the MDM Core API suite. MarkLogic provided Smart Mastering Framework include a set of RESTful API extensions[1]. The MDM core API (MAC) consists of REST API endpoints to invoke mastering and merging. These APIs are configuration driven to take parameters as input for the API. The API suite also includes services to retrieve history for documents or individual properties within the merged documents. Some of the key APIs provided by this suite are:

- Match – This API identifies the list of documents matching the given document, using JSON to define an array of objects representing potential matches.
- Merge – This API saves or provides a preview of a merge document, combining two or more other documents. The delete option on the same API will unmerge a previously merged document, restoring the original documents.
- Match-And-Merge – This API provides the convenience of calling both the match and merge in a single API. Rather than calling match and merge functions separately, you can call them together on a set of URIs. By doing so, you ensure that both happen in the same transaction and that the merges are consistent and non-redundant.
- Entity Relationship: This API saves or provides relationship between two entities. The relationship information can be the information obtained from a particular data source as well as aggregation of relationship obtained from different data sources.
- History - Smart Mastering Core tracks the history of what merge and unmerge operations have been done to a document, as well as which original documents contributed values to a merged document.
- Notifications – Notifications API identify matches that are likely but did not score high enough to automatically merge. Notifications should be presented to human users for review.

In addition to these APIs, there are other utilities and miscellaneous APIs focusing on nonfunctional aspects of the smart mastering library like statistics, dictionary, etc.

MDM XQuery Libraries:

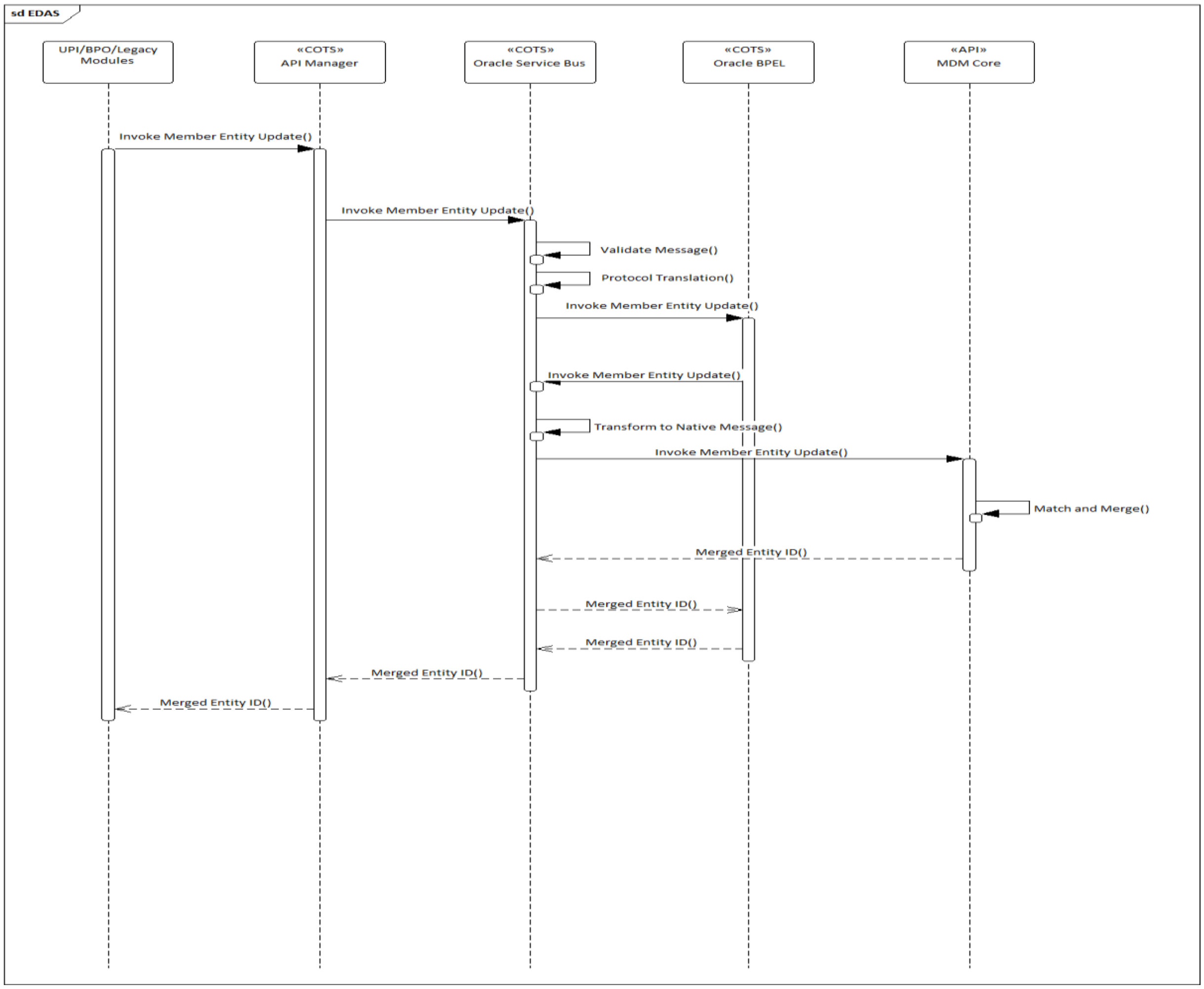
The XQuery libraries are structured to separate the API from the implementation. The APIs acts as a façade to external consumers and does not undergo drastic changes. However, the XQuery APIs continue to evolve to make the MDM operations efficient and robust. These XQuery libraries are persisted inside the MarkLogic's modules database. The modules database is an auxiliary database that is used to store executable XQuery, JavaScript, and REST code. These libraries are loaded inside MarkLogic with execute permissions.

The loading operations along with setting security privileges are handled via gradle scripts. Some of the key XQuery libraries used by the Smart Mastering Core are:

- matcher.xqy – This module provides functions to store, retrieve, delete, and list match options; find potential matches for a document; and to store, retrieve, delete, and list match blocks.
- merging.xqy – This module provides functions to build (preview), save, or remove merged documents and to store, retrieve, delete, and list merge options.
- Process-records.xqy – This module provides two functions to run through both matching and merging for a particular document.
- Match-And-Merge-Trigger.xqy – This module implements a trigger to process matching and merging any time a new document is inserted into the content collection.

The following sequence diagram shows a member demographic update, originating from UPI and culminating in the member update on the enterprise MDM solution. The diagram focuses on the sequence involving member update call propagating through the ESB layer and realized in the MDM layer.

Figure 6-51: Member Demographics Update – EDAS



6.2.7.6 Enterprise Identity and Access as Service (EIAS)

The Enterprise Identity and Access as a Service (EIAS) is an entity service function implemented on top of the Identity and Access Management (IdAM) framework. EIAS primarily provides two kinds of services

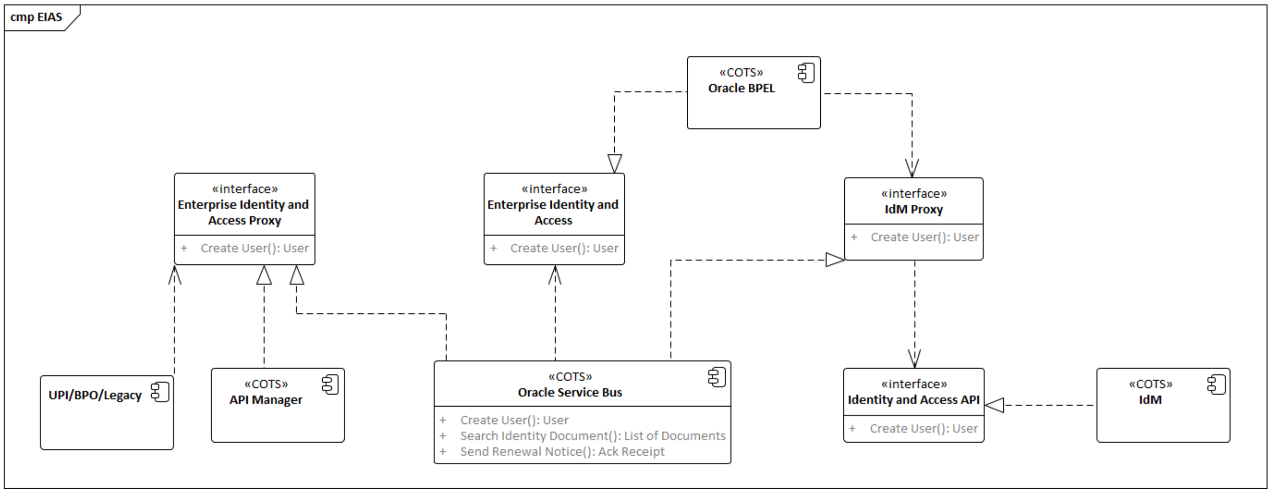
- Identity Management Services
- Access Management Services

Identity and Access Management (IdAM) Framework Capabilities:

- Identity Management
  - User functions such as create, search, modify, and delete users.
  - Role functions such as create, search, modify, and delete roles.
  - Role Administration functions for management of role members and relationships between roles.
- Access Management
  - Authenticate users by validating their credentials against Oracle Access Manager and its configured user repositories.
  - Authenticate users and check for authorization to access a resource.
  - Authenticate users and create unique Oracle Access Manager sessions represented by session tokens.
  - Validate session tokens presented by users and authorize their access to protected resources.
  - Terminate Oracle Access Manager sessions given a session token or a named session identifier.
  - Enumerate Oracle Access Manager sessions of a given user by specifying named user identifier.
  - Save or retrieve custom Oracle Access Manager session attributes.

The diagram below provides high-level component diagram of how IdAM is integrated with ESB.

Figure 6-52: Enterprise Identity and Access Service Components

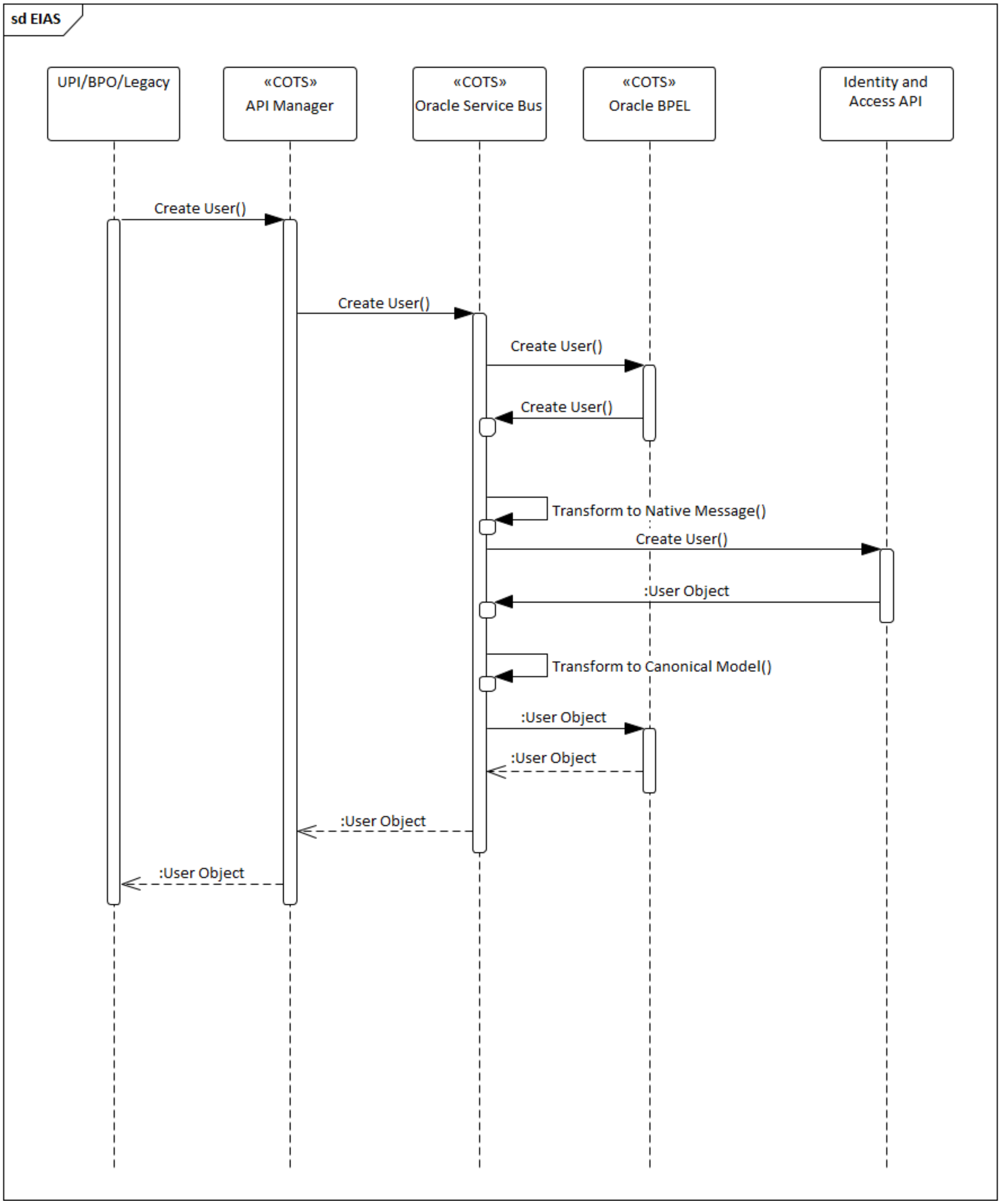


IdAM – IdAM platform serves Authentication, Multifactor Authentication, Federation, coarse-grained authorization, Provisioning, De-Provisioning, Delegated Administration, and Password Management services. Fine-grained authorization for protected resources is managed by the role-based access policies defined within each of the applications.

ESB – ESB acts as a mediation layer between different application and IdAM that want to use the Identity and Access management services for Individual Identity governance or for access control. ESB provides multiple services, which include protocol translation, message conversion, auditing, logging, security, and SLA monitoring to facilitate communication between integrating systems and IdAM. The ESB services are exposed using the API Manager component. The API Manager delegates the call to the business services layer through OSB. The component at the business layer implements business logic for the functional call.

Consider a scenario where an Individual register in UPI and UPI in turn invokes create user API in Enterprise Identity and Access Service. The sequence diagram below illustrates the scenario mentioned above.

Figure 6-53: Workflow to Create User



6.2.7.7 Enterprise Shared Service Security Design

The enterprise utility service is hosted as a shared service to be consumed by multiple MMISR modules including UPI and BPO modules like DS, FS, and BMS. Every communication between any of the MMISR module and the ESS happens through ESB. These shared services are not exposed to any external consumers or agencies.

Enabling Single Sign-on for ESS Tools:

The enterprise tools need to have following capability to support single sign-on provided by IdAM:

- **Identity Federation** – The tool needs to support identity federation using either the SAML 2.0, OpenID 2.0 or WS-Federation 1.1 specifications to participate in single sign on. A federated trust will be configured between a shared service tool and Oracle Identity Federation (OIF) with OIF acting as the identity provider.
- **OAM SSO Cookie** – The web server for the tool needs to be configured for web gate. The web gate communicates with the OAM to authenticate the user provided in tool specific logic pages. OAM returns the SSO cookie as a part of the authentication process which needs to be honored by the tool.

Transport Layer Security:

The communication between the ESB and all the ESS components are encrypted using one-way SSL. Here, the ESB validates the certificate of the ESS component server. This validation is done to make sure that it is the expected server, i.e. no Man in the Middle (MITM) attack.

- **EDM Transport Security –** The Perceptive Content's Integration server works on top of Apache Tomcat web server. The keystore configuration and certificate import operations are same as what is followed for any web server. The ESB layer working along with the OWSM component will interact with the Perceptive Content's Integration Server using one-way SSL security mechanism.
- **ECM Transport Security –**The Communication Server of OpenText Exstream will be configured to receive HTTP calls from ESB. The ESB layer working along with the OWSM component will interact with the OpenText Exstream using one-way SSL security mechanism.
- **EAVS Transport Security –** The SAP data services natively support several of security mechanism like Basic, Authorization Header (Token or key based), and OAuth 2.0 among others. The ESB layer working along with the OWSM component will interact with the SAP DS Access Server using one-way SSL security mechanism.
- **EDAS Transport Security –** In case of the EDAS, MarkLogic is the data store and MarkLogic Server uses FIPS-capable OpenSSL to implement the Secure Sockets Layer (SSL v3) and Transport Layer Security (TLS v1) protocols. When the MarkLogic Server is installed, FIPS mode is enabled by default and SSL RSA keys are generated using secure FIPS 140-2 cryptography. This implementation disallows weak ciphers and uses only FIPS 140-2 approved cryptographic functions. Note that the FIPS mode can be enabled or disabled on a running system. If FIPS mode is enabled or disabled on a running system, the OpenSSL library is reconfigured appropriately without requiring a server restart. When the FIPS mode setting changes and secure XDQP is configured, all XDQP connections are dropped and reestablished.

Data Security:

The three ESS components, SAP Data Services, Perceptive Content, and OpenText Exstream use Oracle RAC as the data store. The data stored in the database is secured using Oracle Advanced Security feature called Transparent Data Encryption (TDE), where the complete database is encrypted. The ESS applications and users authenticated to the database continue to have access to application data transparently with no application code or configuration changes required. Attacks from OS users attempting to read sensitive data from table space files and attacks from thieves attempting to read information from acquired disks or backups are denied access to the clear text data.

- **EDAS Data Security –** The EDAS implementation is MarkLogic based, and the security details including cluster configuration, role-based security, and document level security explained in Subsection 5.3 applies to EDAS as well.

6.3 Security Detailed Design

The security detailed design defines the levels and types of security and privacy controls offered by the SI Platform, including network, PKI, and SSL based infrastructure security, and user roles-based application security implemented using the Identity and Access Manager (IdAM), logging, auditing, and encryption.

The following are the architecturally significant requirements pertaining to Security of the SI Platform.

- All requests for functionality and data contained within the HHS 2020 Enterprise pass through the Security System.
- HHS 2020 EA is governed by a combination of security control requirements found in MARS-E 2.0 and FIPS 140-2 standards intended to prevent unauthorized access to system data and functionalities.
- ESB will enforce role-based authorization for service access and will carry out necessary logging of service interactions for auditing purposes.

6.3.1 Network Security

Network security prevents and monitors unauthorized access, misuse, modification, or denial of a computer network and network accessible resources. NM HSD owns and manages the network security aspects for NM MMISR project and has implemented the necessary requirements to maintain the confidentiality, integrity and availability of the network in accordance with the Minimum Acceptable Risk Safeguards for the Exchanges (MARS-E), version 2.0.

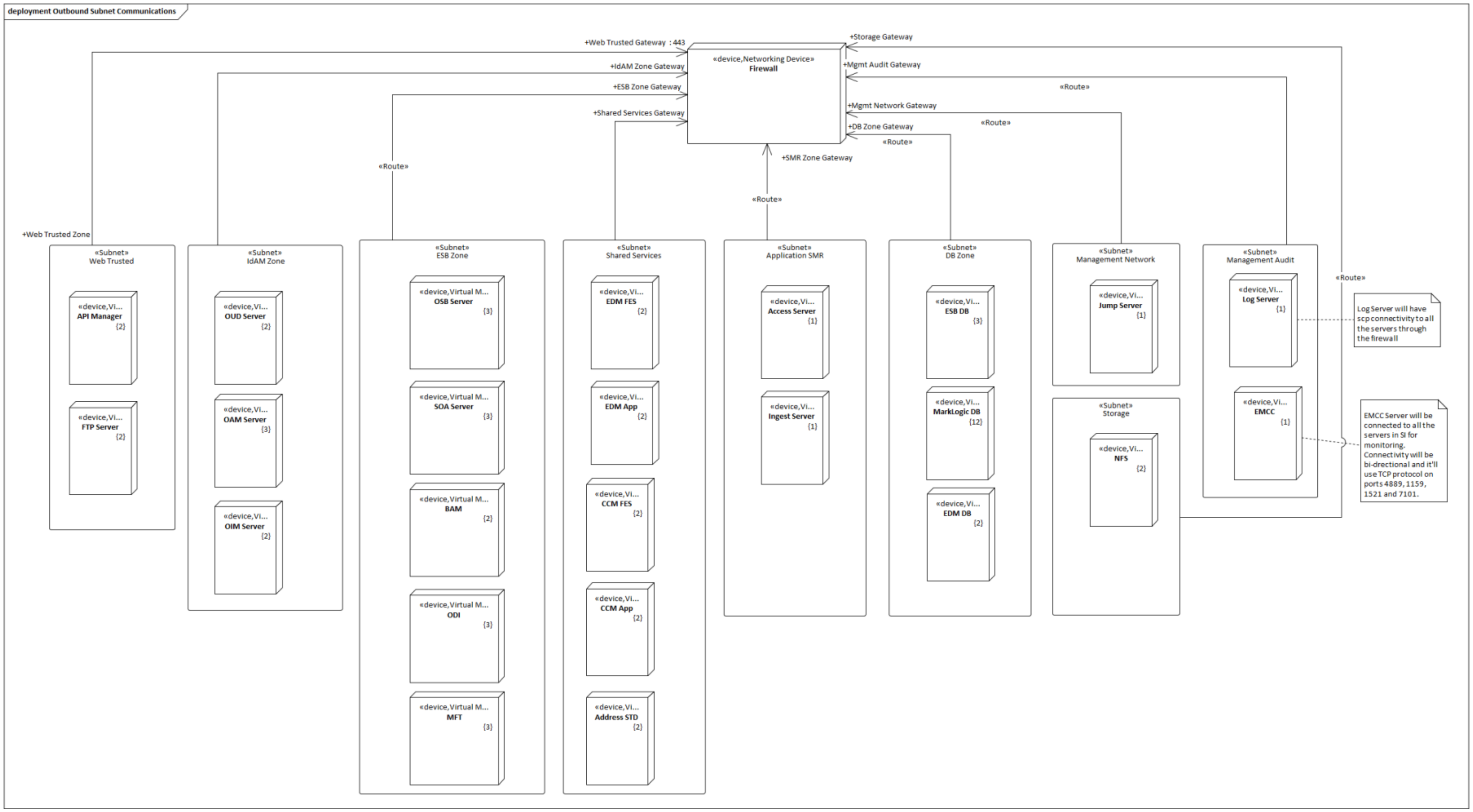
Each technical control family of the MARS-E, v2.0 is met through the solutions as noted below.

6.3.1.1 Firewall

The firewall is a key security component that protects the network boundary and manages the ingress and egress of communications within the network. As such, the firewalls provide the technical solutions for the Access Control (AC), System Communication (SC) and the System Integrity (SI) control families of the MARS-Ev2.0. The Palo Alto will be used to manage inbound and outbound traffic at the perimeter as well as across all of the subnets. The Palo Alto firewall is a next generation firewall that combines the stateful inspection, intrusion prevention, anti-malware protection and URL content filtering into one appliance. The Palo Alto firewall combines all of this security enabled functionality into a "single pass" architecture thereby allowing all of these functions to scan the traffic simultaneously.

The stateful inspection capability will use more than the traditional ports and Internet Protocol (IPs) as a means of monitoring and filtering traffic. The Palo firewall will utilize attributes of the application (App-Id), the content (Content-Id) and the user (User-ID) to accurately detect anomalous behavior and known virus signatures and will drop those before they enter the perimeter. The firewall will be configured with a deny all, allow by exception policy to ensure that only those connections that are essential are approved. The firewall will also be configured to restrict the use of unnecessary ports, functions and protocols, enhancing least functionality. The firewall will have active passive configurations such that if there is a failure in one firewall, it will failover to the passive standby device. If both devices fail, then traffic will be halted and not allowed to pass to the subnets and logical subnetworks.

Figure 6-54: Outbound Connections from Subnets through Firewall



Log Server will have scp connectivity to all the servers through the firewall

EMCC Server will be connected to all the servers in SI for monitoring. Connectivity will be bi-drectional and it'll use TCP protocol on ports 4889, 1159, 1521 and 7101.

The following are some key points depicted in the above diagram

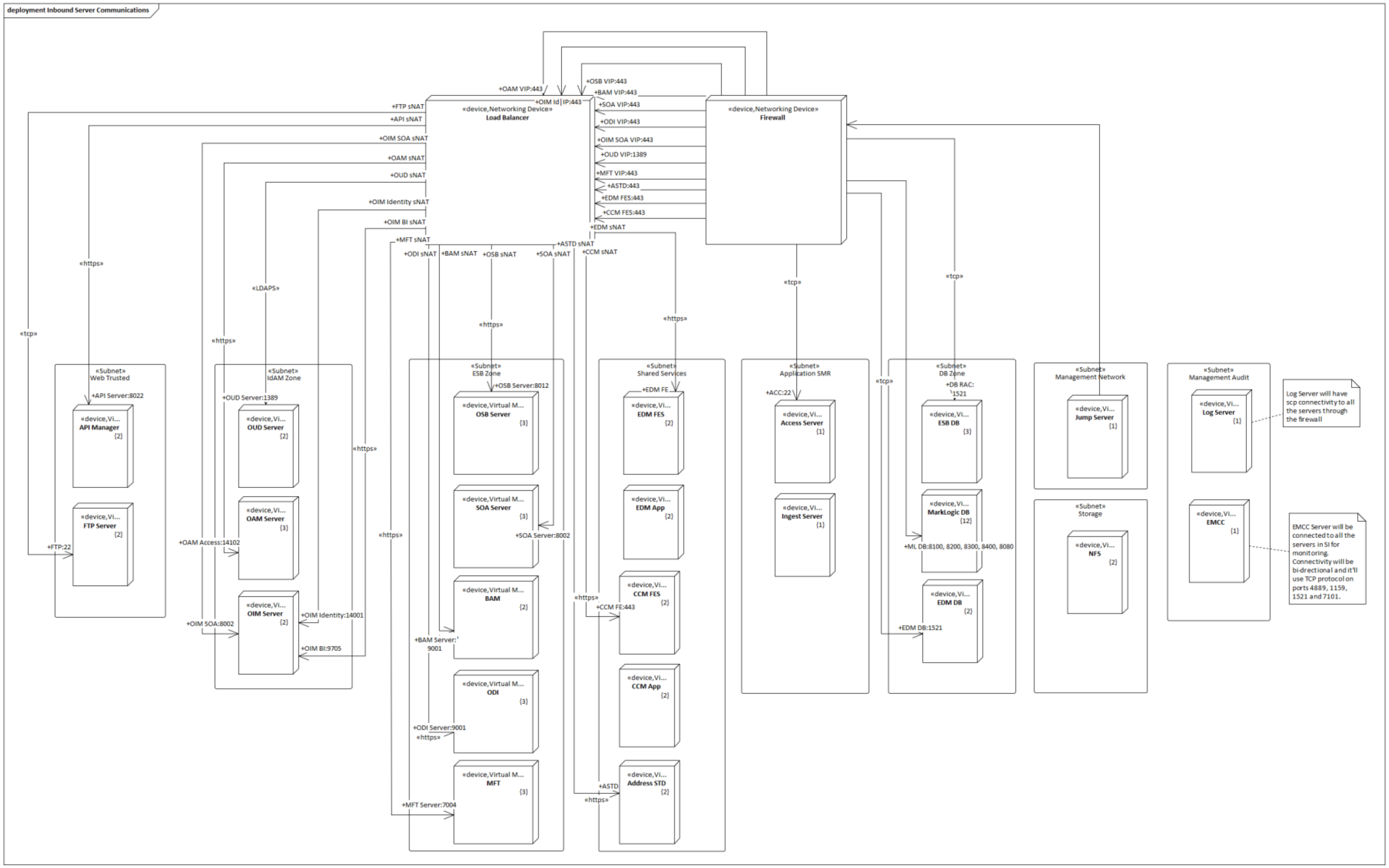
- All outgoing communication from the subnets within an SI Environment go through the firewall.
- Access control lists (ACL) on the firewall allow the traffic to different destinations based on the destination address and port.

6.3.1.2 Load Balancers

A load balancer distributes network or application traffic across various servers to ensure availability of the network. As such, load balancers help fulfill the SC control requirements of the MARS-E v2.0. The F5 Load Balancer will provide traffic distribution in a seamless fashion. It will accomplish this by distributing incoming requests evenly among VMs and thus preventing system overload and potential DoS.

The F5 load balancer will also be utilized in a HA mode which will ensure the web traffic is managed between the different components of the application. F5 also has built-in security features which can detect security attacks related DDOS.

Figure 6-55: Load Balancing Software Components through Load Balancer Virtual IPs



Log Server will have scp connectivity to all the servers through the firewall

EMCC Server will be connected to all the servers in SI for monitoring. Connectivity will be bi-drectional and it'll use TCP protocol on ports 4889, 1159, 1521 and 7101.

As shown in the above diagram, access to the application components is routed through the load balancer to distribute the traffic equally among the clustered nodes and servers along with providing failover in case of a server or software component failure.

6.3.2 Transport Layer Security

Transport layer security secures the communications channel between applications thereby protecting the confidentiality of the data in transmission. This becomes especially important in the transmission of sensitive data such as PII, PHI or FTI data. For transport-level security the SI Platform will employ Transport Layer Security (TLS) v1.2. TLS is the Internet Engineering Task Force (IETF) standardized version of SSL. This protocol creates a level of trust between the two parties involved and will aid in fulfilling many of the SI controls.

In the case of MMISR, implementing this protocol for all communication between different applications, servers, and third-party systems provide numerous benefits beyond traffic confidentiality including integrity protection, replay defenses, and server authentication. SSL Certificates from a trusted CA will be used to implement the TLS.

6.3.3 Enterprise Application Security

The HHS 2020 enterprise Identity and Access Management (IdAM) framework includes the technology to support the identity management and business rules to ensure appropriate constituent access to resources, across the MMISR. The IdAM system also provides SSO and serves as a centralized security solution for providing authentication, coarse-grain authorization, and session management for MMISR applications and for the existing NM HSD applications that participate in the IdAM framework. The COTS applications are used to build the platform and authentication across various platforms and various user types are managed by the IdAM platform.

The following are the architecturally significant requirements that are addressed in this subsection.

- Use SSO and Identity and Access Management (IdAM) to implement Authentication, Authorization and Auditing; establish, integrate and manage unique logon IDs and security profiles for Stakeholders, Users and other Contractors seeking access to the MMISR Solution.
- Use a state-produced Active Directory for State employees for full integration of managing user access to all SI components and is based on Active Directory security groups so that, for State employees, there is no secondary user management within SI components.

The IdAM platform serves Authentication, Multifactor Authentication, Federation, coarse-grained authorization, Provisioning, De-Provisioning, Delegated Administration, and Password Management services. The fine-grained authorization for protected resources is managed by the role-based access policies defined within each of the applications.

The communication between different MMISR applications are invoked through Web Services and APIs deployed on ESB and it requires both authentication and verification of the identity, and authorization and determination of permission to access data or services. ESB enforces these security controls using custom OWSM policies that communicates with IdAM along with One-Way or Two-Way SSL Certificates.

Identity Governance is part of the IdAM framework, and it provides user provisioning, self-service access, and role life cycle management along with privileged account management.

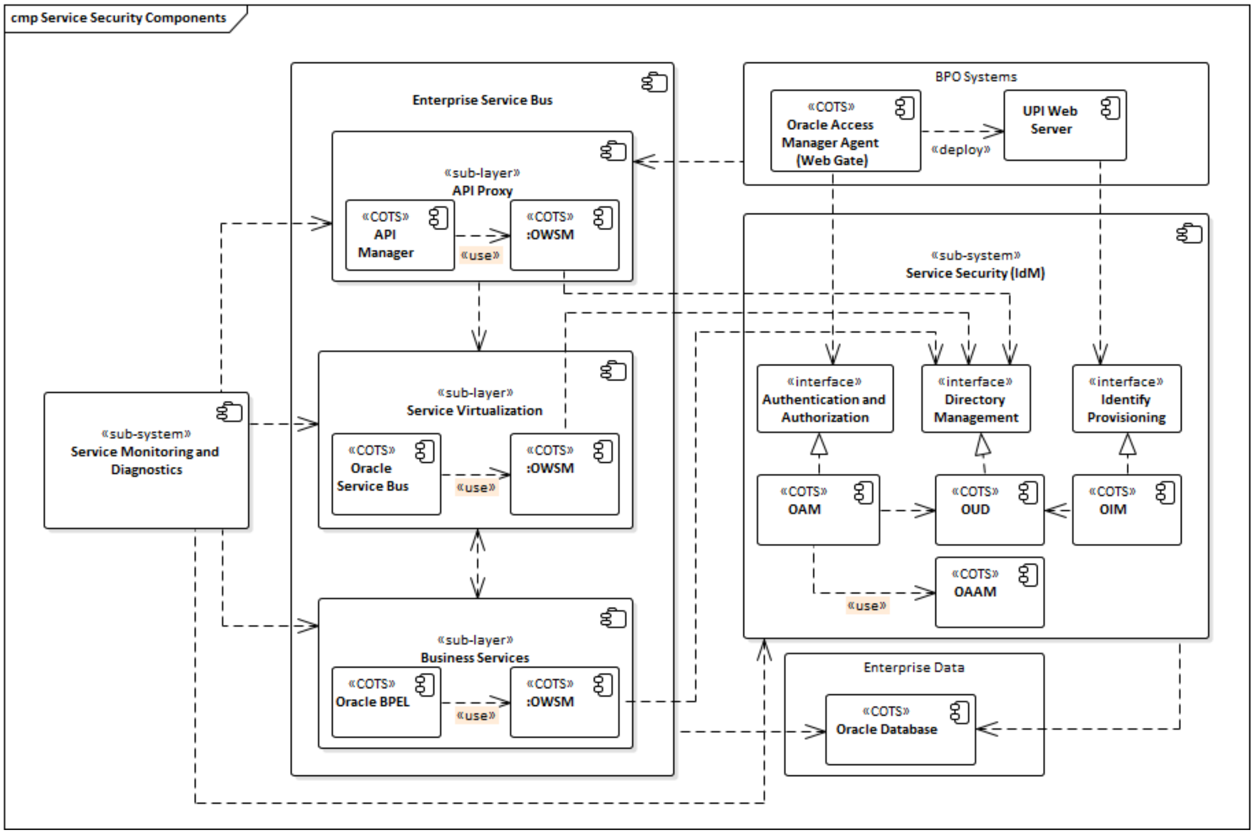
6.3.3.1 Alignment with PADU Approach and MITA Technical Strategy

SI contractor will use Oracle Identity and Access Management (IdAM) platform which is the preferred COTS tool as identified in the NM HHS 2020. Since Oracle IdAM enables to achieve maximum configuration, the alignment with PADU for this framework is deemed as "Preferred" level.

6.3.3.2 IdAM Logical View

The diagram below provides a logical view of IdAM components.

Figure 6-56: Service Security (IdAM) Components


cmp Service Security Components
Enterprise Service Bus
«sub-layer» API Proxy
«COTS» API Manager
«use»
«COTS» :OWSM
«sub-layer» Service Virtualization
«COTS» Oracle Service Bus
«use»
«COTS» :OWSM
«sub-layer» Business Services
«COTS» Oracle BPEL
«use»
«COTS» :OWSM
«sub-system» Service Monitoring and Diagnostics
BPO Systems
«COTS» Oracle Access Manager Agent (Web Gate)
«deploy»
UPI Web Server
«sub-system» Service Security (IdM)
«interface» Authentication and Authorization
«interface» Directory Management
«interface» Identify Provisioning
«COTS» OAM
«COTS» OUD
«COTS» OIM
«COTS» OAAM
«use»
Enterprise Data
«COTS» Oracle Database

Note: The interfaces like Directory Management, Authentication & Authorization etc., are depicted in the above figure for representational purpose only. These are the capabilities of the tools, which are internal to the tool itself, and not be exposed as services to other components.

6.3.3.3 Access Control Gateway

This layer is the entry point for UPI Portal users. It provides web-based access to the portal application and authenticates user access to the system. Oracle Access Manager Agent (Web Gate) will be deployed on UPI Web servers and act as a policy enforcement agent. Web Gate intercepts all the incoming requests to the UPI and performs a check whether the request requires authentication or not. Web Gate is integrated with Oracle Access Manager.

Table 6-33: Access Control Gateway and Physical Platform Tool

Name	Access Control Gateway
Description	Access Manager Agent (WebGate) is a web-server plug-in for Oracle Access Manager (OAM) that intercepts HTTP requests and forwards them to the Access Server for authentication and authorization.Access Manager Agent (WebGate) is a web-server plug-in for Oracle Access Manager (OAM) that intercepts HTTP requests and forwards them to the Access Server for authentication and authorization.
Platform Tools	Oracle Access Manager 12c WebGate
Capabilities	• Single Sign-On (SSO) Login • Session Validation • Enforce Policy Evaluation

6.3.3.4 Authentication and Coarse-Grain Authorization

This layer provides the Authentication, Access, and Federation for the users. The authentication and coarse-grain authorization decisions will also be taken for the functional COTS products. The coarse-grain access control restricts the end-user access to application/activities. This is typically a restriction based on the resource (hostname) or URI in the URL.

In addition to above, this layer is also responsible for Multi Factor Authentication (MFA) SSO within the MMISR applications as well as Federated SSO with the supporting external applications.

The authentication and coarse-grain authorization functionality of the applications is realized through the Oracle Access Manager (OAM) Component and its extensions.

Table 6-34: Oracle Access Manager Implementation View

Name	Oracle Access Manager (OAM)
Description	OAM is a component of Oracle IAM suite that provides authentication of users and authorization of user access to applications. The primary functionalities provided by Access Manager are: • Authentication Services which identifies an individual user and ensure that the user is who they claim to be. These services are also responsible for forcing password changes. • Authorization Services, which grants an Authenticated user who is authorized, access rights to applications and other objects.
Interface Definition	It communicates with OUD over LDAP protocol and with other components over HTTP(s) protocol.
Dependencies	OUD and OUD Proxy
Implementation overview	It consists of the following internal components: • Authentication: Validates user’s credentials and verifies the user is who he claims to be. • Authorization (Policy): Evaluates policies to determine whether the user has permissions to access the required resource. • Session Service: Maintains information about user session and enforce timeout limits. • · Logging Service: Tracks a user’s interactions with web applications. Creates log messages to form an audit trail of important events with the system.

Table 6-35: Oracle Adaptive Access Manager Implementation View

Name	Oracle Adaptive Access Manager (OAAM)
Description	Provides Multi-factor and risk-based authentication
Interface Definition	Adaptive access manager provides admin console from where the administrator can configure multi factor policies, rules etc. Also, it provides interfaces for user login.
Dependencies	Oracle Access Manager (OAM) and Directory Services (OUD)
Implementation overview	Adaptive Access Manager, Identity Manager & Access Manager components will be integrated in HA mode. Adaptive Access Manager handles registering user profile for security questions and answers; identity manager handles the

Name	Oracle Adaptive Access Manager (OAAM)
	password reset service; access manager is key component for authenticate the user. These components are integrated with directory servers that host user accounts for authentication.

6.3.3.5 Fine-Grain Authorization

The fine-grain access control restricts the end-user access to specific data elements within the application. This restriction may prevent certain content from appearing on a page based on a user's application defined role. Once Oracle IAM has validated the user credentials, the control is passed on to the respective applications to address the fine-grained authorization requirements.

6.3.3.6 Data Repository

The IdAM platform maintains the user repositories as well as security, policy, reporting, and session data at the back end. This layer is responsible for providing fast access to the data while maintaining the integrity and security that is required for handling sensitive eligibility information.

The user store component (LDAP based) provides:

- Centralized user account store for Individuals (OUD).
- Proxy services for accessing the Employee User Store (Active Directory).

The individuals' data is stored in a centralized secure LDAP server i.e., Oracle Unified Directory (OUD). The passwords will be stored in the hashed (3DES+SSHA) algorithm and is not readable by anyone.

The directory Server will implement attribute level Access Control Instructions (ACI) for directory server data. These ACIs will provide fine-grain access to directory server attributes. ACIs will utilize the native directory server role mechanism. Normal user access will be limited to self, minus security, and operational attributes. The application IDs will be updated with the necessary roles to provide access to the attribute.

The attributes should be aggregated in the ACIs so that they are grouped in classifications based on sensitivity and use. This generic approach still allows access to some attributes, which may be unnecessary to an application, but it is assumed that the applications will only access attributes it requires. This will still reduce exposure of attributes to applications, which are not allowed access to them.

All communication between OAM to OVD to OUD will be in SSL mode and use the self-signed server certificate.

Table 6-36: Oracle Unified Directory Server Implementation View

Name	Oracle Unified Directory Server
Description	LDAP product used to provide access to the external user data for authentication, authorization, and identity management. The directory server is based on an industry-standard server protocol called the Lightweight Directory Access Protocol (LDAP). LDAP provides a common language that client applications and servers use

Name	Oracle Unified Directory Server
	to communicate with one another. LDAP applications can easily search, add, delete, and modify directory entries.
Implementation overview	OUD will be installed as master in multiple nodes and enable replication between both nodes. Replication will provide Fault tolerance/Failover, Load balancing, Higher performance and reduced response times Applications connect to OUD through OUD Proxy.

6.3.3.7 IdAM Logging

Identity and Access Management logging is configured to provide information at various levels of granularity using the same logging infrastructure and guidelines as other oracle fusion middleware components used in SI Platform. The logging infrastructure records the following events in to the logs.

- System/Service operational events (start/stop/warnings/errors).
- Access management events (authentication, authorization, invalid login attempts).
- Identity events (user creation/modification/deletion, password resets, Role assignments).
- Directory events (binds, searches, modifications etc.).
- Federation events (Service provider and Identity provider calls, access, and assertions).

The Splunk forwarders will be installed at every component of IdAM to feed the logs generated to the Splunk enterprise for centralized log monitoring and analysis. The details of log aggregation and monitoring is explained in Subsection 4.5.4

6.3.3.8 IdAM Auditing

Identity and Access Management auditing refers to the process of collecting review specific information related to administrative, authentication, and run-time events. Auditing helps to evaluate adherence to polices, user access controls, and risk management procedures, and provides a measure of accountability and answers to the "who has done what and when" types of questions. Audit data is used to create access and Identity usage dashboards, compile historical data, and assess risks.

The audit requirements were considered during the requirement gathering stages to appropriately build an infrastructure to support it. The level of information that needs auditing will be determined within the Policy and Processes. These includes:

- Information Access

  E.g. Successful and/or Failed Authentication/Authorization attempts
  E.g. Failed access attempts to sensitive tables within databases by privileged user

- Policy Administration

  E.g. Changes to access control policies

- User Administration

  E.g. Addition/Removal of users' roles and privileges

- Information Change

  E.g. Changes to sensitive Information

6.3.4 Identity Federation

IdAM framework includes Oracle Access Management Identity Federation engine and is used for providing SSO to HSD employees accessing UPI. A federated trust will be configured between the Employee Active Directory Federation Service (ADFS) and Oracle Identity Federation (OIF) with ADFS acting as the Identity provider and OIF as the service provider. Similarly, Oracle Identity Federation server can work with any number of Identity providers or service providers.

As shown in the sequence diagram 6.49 for Employee access to UPI, when a state employee is accessing a resource protected by Oracle Access Manager such as UPI, Oracle Identity Federation redirects the user to State ADFS for global authentication. ADFS will obtain credentials, authenticate the user, and redirect the user back to the Oracle Identity Federation server instance - which retrieves the asserted identity from the ADFS and redirects the authenticated user to the access manager which provides access to UPI Portal. Some of the existing HSD applications can also participate in the Federated SSO provided by the IdAM Federation services.

MMISR IdAM federation supports the transport and receipt of request and response messages using either the Security Access Markup Language (SAML) 2.0 specifications, SAML 1.1, OpenID 2.0 or WS-Federation 1.1. SAML uses an eXtensible Markup Language (XML) framework to define a simple request-response protocol in order to achieve interoperability between different applications using SAML assertions. SAML requester sends a SAML Request element to a responder. Similarly, a SAML responder returns a SAML Response element to the requester. SSO and Federation relies on SAML artifacts and assertions to relay authentication information.

Identity data transported using the SAML 2.0 in MMISR is secured using the following specifications:

- All outgoing Assertions will be signed.
- All outgoing requests/responses not containing Assertions will be signed.
- The signing certificate will not be included in the messages.
- Identity Federation (acting as the IdP) will not require signatures on any messages except when specified in the SP Partner metadata.
- The hashing algorithm for signatures will be configured to use SHA-256.

SAML token will have the user and application attributes that will grant the user seamless access to MMISR applications and Services. The SAML assertions will be signed and encrypted with X509 certificates so as to make it tamper proof.

6.3.5 Data Security

Sensitive data such as Federal Tax Information (FTI), Personally Identifiable Information (PII) and Personal Health Information (PHI), is received, processed, stored and transmitted in MMISR platform and must be compliant with the applicable compliance standards and guidelines. Data security in MMISR focuses implementing the System Integrity (SI) security controls to protect the data while in transit, during display, in use and at rest.

6.3.5.1 Data in Transit

Sensitive data must be encrypted when transmitted across networks to protect against eavesdropping of network traffic by unauthorized users. All types of transmission of data which includes client-to-server, server-to-server communication as well as any data transfer between the SI Platform and the third-party systems is encrypted in MMSIR as follows:

- Web traffic is encrypted using strong security protocol, Transport Layer Security (TLS 1.2+).
- The connection between the database and application is encrypted using SSL.
- Batch file transfers such as EDI communication, file transfers to and from Federal, local, or State agencies are secured by using Secure File Transfer Protocol (SFTP).

6.3.5.2 Data at Rest

Data at rest includes data stored physically in any digital form such as databases, spreadsheets, archives, tapes, offsite backups.

- Sensitive data stored in the database is secured using Oracle Advanced Security feature called Transparent Data Encryption (TDE), where the complete database is encrypted. It stops attackers from bypassing the database and reading sensitive information from storage by enforcing data-at-rest encryption in the database layer.
- Transparent Data Encryption fully supports Oracle Multi-tenant. When moving a pluggable database (PDB) that contains encrypted data, the TDE master keys for that PDB are transferred separately from the encrypted data to maintain proper security separation during transit.
- Sensitive data stored in files are secured by encrypting the files using Federal Information Processing Standards (FIPS) compliant encryption.
- Sensitive data stored in archives, tapes and offsite backups are also encrypted using FIPS compliant encryption algorithm.

6.3.5.3 Data in Use

All employees undergo the security awareness training to ensure the data they handled only for business purposes and understand the implications of any misuse

Sensitive data must be displayed only to authorized users on a need to know basis. It is achieved by designing and implementing a Role Based Access to view sensitive data. Fine-grain access controls are implemented within the respective applications to ensure only authorized users are granted access to view sensitive data and every view of sensitive data is audited and logged. In most cases, privileged users are authenticated using a two-factor authentication wherein, the user must enter username/password followed by answering security questions that were set up during the account creation process.

6.3.5.4 Data de-identification

Data de-identification is an important step in securing the breach of confidential data in lower environments. The production data will be used only in the Production environment, access to which will be controlled.

However, in lower environments such as Development, QAT, and SIT, which may contain read/write permission to non-admin users, the data must be secured following the HIPAA guidelines. De-identification of production data is an important step in order to achieve this, also enabling the test teams to access production-like data. The data de-identification helps the cross-functional teams test the integration capabilities across the modules with production-like dataset.

Three things to configure in a data de-identification program are:

- The source dataset – Identifying the sensitivity of each attribute.
- What to de-identify – Identify all the attribute to be de-identified.
- The destination dataset – Where will the de-identified data be stored.

The de-identification of source data does not impact the original dataset or its data. Instead, the original data is copied to the destination dataset, applying the de-identification rules on it before saving the de-identified data.

There are different means of achieving data de-identification. The following is an inexhaustible list of such options:

- Using a COTS de-identification programs, such as Google Cloud Healthcare API.

- Customized program that leverages the SI Platform defined HHS 2020 enterprise common data models along with the element classification.

6.4 Performance Detailed Design

This subsection elaborates on how the SI Platform design complies with requirements for performance and high availability. The SI Platform solution is designed for high availability to meet the challenges of a large user and transaction base anticipated in the MMISR solution with real-time response, but ongoing performance monitoring enables us to maintain a well-tuned system.

The following are the architecturally significant requirements that are addressed in this subsection.

- Implement systemic Performance Management. (ConOps/Reference architecture).
- Provide performance standards for accountability and planning. (MITA Technical strategy).

6.4.1 Server Performance

The SI Platform performance architecture begins with the VxRack hardware which is a hyper-converged infrastructure system that delivers extreme performance, resiliency and flexibility and VMWare Infrastructure.

- SI Platform components are built with high-availability using VMware infrastructure.

- All the servers will have the NIC teaming enabled and all the storage controllers will have multipathing configured for high-availability.
- Every node in the VMWare Infrastructure is used in the processing of I/O operations, making all I/O and throughput accessible to any application for eliminating any performance bottlenecks.
- The VxRack system automatically manages and optimizes data layout, preventing performance hot spots.

6.4.2 Application Performance

The SI Platform application performance architecture includes configuring scalable application infrastructure and incorporating performance and scalability in the application development lifecycle.

6.4.2.1 Application Infrastructure

IP Components are configured using several standard features associated with scalable and highly available systems to address peak usage and response times: a load balancer (Existing NM HSD F5 Big IP Devices are used), redundant Web and Application tier hosts and services, clustered databases, and shared storage. Some of the key features of application Infrastructure are listed below.

- All IP Applications are installed and configured on VMWare infrastructure that enables easy server scale-out.
- IP Platform’s tiered architecture allows for adding more hardware resources and increasing processing capacity of one tier without impacting another tier. This ability contributes to making the application highly scalable.
- Application components are built as multi-node clusters to enable seamless session failover.
- Each cluster consists of multiple application instances running simultaneously and working together to provide increased scalability and reliability.
- Application components are configured to support both horizontal and vertical scaling of server instances. Administrators can easily add more hardware without modifying application code to support the load.
- WebLogic Automatic Service Migration is configured for the JMS Service in fusion middleware components used by Integration Platform.
- F5 load balancer is configured to route application traffic to all available application instances and support persistence of session’s state.
- F5 load balancer is configured to detect service and node failures and to stop directing traffic to a failed node.
- Oracle data sources in applications are configured with Oracle Single Client Access Name (SCAN) addresses for Fast Application Notification (FAN) and Fast Connection Failover (FCF).
- Shared storage will be configured for persistence data such as transaction logs and JMS Stores
- The MarkLogic solution is also a node-based architecture, which allows for scaling by adding additional nodes with minimal effort.

6.4.2.2 Application Service Performance

The service response times will be optimized for all the application components in the IP Platform by:

- Configuring appropriate caching mechanisms available in the Oracle Fusion Middleware products, Oracle RAC Database and MarkLogic.
- Tuning the Java Virtual Machine (JVM) settings by allocating sufficient heap memory and configuring the optimal Garbage collection thresholds.
- Configuring Session timeouts, JTA timeout and Idle timeouts based on the application requirements to avoid long-running transactions.
- Configure optimum connection pools in Application Server, to manage resources such as Database connections, JNDI connections, Adapter outbound connection pools, bean pools and service objects.
- Ensuring that excessive logging is not configured in the production environment.
- Configuring the security providers with optimized user and group search filters.
- Limit JMS Queues and Topics retries as per the SLA's.
- Asynchronous Services implementation for long-running transactions by implementing JMS Queues and Topics.
- Optimize the application thread pools and service engine threads in different application components to meet the throughput and performance requirements.
- Extensive Performance Testing to make sure there are no memory leaks due to bad coding and configurations.

6.4.3 Oracle Database Performance

SI Platform databases are configured to provide high throughput and fast response times. Below are some of the key performance considerations for these databases.

- Purging and archiving the unused data on a regular basis.
- Avoiding multi-level nested views and joins in the application SQLs.
- Providing a larger block/buffer size for I/O to reduce the number of disk accesses.
- Increasing the number of operations that run in parallel (increasing concurrency).
- Setting up optimal number of cursors, processes, buffer and share pool sizes.
- Creating and managing the indexes per application requirements.
- Application sufficient redo/undo spaces on the disk.

Using the monitoring tools such as Log Insight, Oracle Enterprise Manager Cloud Control (EMCC) and Splunk, history of the IP Applications performance over time is maintained and is used to make baseline comparisons. With data about actual resource consumption for a range of loads, we can predict the resource requirements for anticipated load volumes and tune the IP Platform to meet the throughput and performance requirements.

6.4.4 MarkLogic Database Performance

The SI Platform leverages MarkLogic's performance monitoring and database efficiency-enhancing tools and design guidelines.

6.4.4.1 Performance Monitoring

For the database performance monitoring, MarkLogic provides OpsDirector tool, which SMR will configure to monitor the live performance of various SMR nodes and databases. For application performance, MarkLogic Monitoring Dashboard and MarkLogic admin console provide dashboards providing a holistic view into the SMR application.

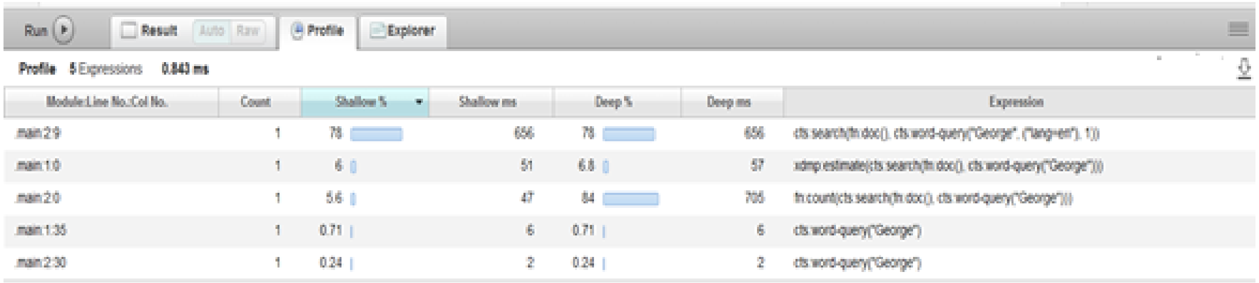
6.4.4.2 Performance Tuning / Enhancement

The SI Platform utilizes the various performance tuning options available in MarkLogic. All the configurations to enhance the performance and database efficiency are made using the MarkLogic Admin Console. Some of the elements and tools that could be configured to enhance the performance are listed below:

- **Thread Count Configuration** – The database administrator can control, without restarting the server, the maximum number of threads to run, based on the expected efficiency to be achieved.
- Usage of Element Range Index – The SMR and MDM databases will utilize the efficiency provided by MarkLogic element range indexes by creating them, especially for all date and id fields to increase the ability to search on these frequently searched fields. More information on range indexes can be found here.

  The Database administrator can create and manage the range indexes without restarting the server, on the MarkLogic Admin Console**.**
- **Query Console, Profiling the Queries** – The developers can control the performance of queries by profiling the requests on the query console. This is a supplementary tool available for the developers to tune their queries for optimal performance. The following figure shows the query profiling example. For details on profiling, please read this information.

Figure 6-57: Query Profiler



Profile 5 Expressions 0.843 ms

Module:Line No.:Col No.	Count	Shallow %	Shallow ms	Deep %	Deep ms	Expression
.main:2:9	1	78	656	78	656	cts:search(fn:doc(), cts:word-query("George", ("lang=en"), 1))
.main:1:0	1	6	51	6.8	57	xdmp:estimate(cts:search(fn:doc(), cts:word-query("George")))
.main:2:0	1	5.6	47	84	705	fn:count(cts:search(fn:doc(), cts:word-query("George")))
.main:1:35	1	0.71	6	0.71	6	cts:word-query("George")
.main:2:30	1	0.24	2	0.24	2	cts:word-query("George")

6.4.4.3 Data Export Performance

The SMR and MDM support bulk export as files, as well as, real-time data exports. In addition, the databases leverage various export options available in MarkLogic.

MarkLogic provides the following export options:

- Exporting selected documents as independent XML files to a directory.
- Exporting selected documents to a compressed file.
- Exporting to an archive.
- Exporting an entire collection to a compressed file.
- Exporting an entire collection as XML files to a directory.
- Exporting the database snapshot.
- Scheduled data exports.

The performance impact during data exports and transfers will be resolved by using the MarkLogic's scalability, availability, and failover guidelines, explained in the following subsections.

Along with these options, the SMR database administrator can increase the number of threads operating on the MarkLogic CORB program to achieve the desired data export or delivery performance.

CORB allows configuration of a range of property options that can be configured to suit the application requirements. THREAD-COUNT option is one of those parameters that the database administrators can use to control the number of worker threads to be used for a particular operation. Based on the load and the size of the content to be exported, the administrator can set it to run an optimal number of threads. By default, CORB sets it to 1; but SMR is designed to use 10 as a default, with an option to increase, or decrease, before the export programs are executed.

6.4.4.4 Scalability

The SMR utilizes MarkLogic's capabilities to scale. MarkLogic Server is built with solid foundations derived from both database and search engine architectures. Consequently, updates become available for querying as soon as they commit and queries against extremely large content sets return quickly.

See MarkLogic documentation library for more information.

6.4.4.5 Availability

The SMR leverages MarkLogic Server's high availability features enabling fast and reliable performance, and also provide recovery from power outages, application errors, or software failures. There are many features in MarkLogic Server designed to keep the server running and available.

- Fast automatic restart.

- Automatic, concurrent forest recovery.

- Tunable database parameters, such as memory limit, in memory list size, and in memory range index size.

- Online database backup operations.

- Many configuration changes in MarkLogic Server happen without the need to restart the server. This makes it easier to make changes to environments while keeping all running applications available.

Please see on the MarkLogic documentation library for more information.

6.4.4.6 Failover

Failover provides both high levels of availability and data integrity in the event of a data node or forest failure. Failover maintains data and transactional integrity during failure events. For example, it allows a single host to attempt any writing or recovery operations to a given forest at any particular time.

Please see MarkLogic documentation library for more information.

6.5 Internal Communications Detailed Design

This subsection provides detailed diagrams depicting internal communication between various components of Integration platform. The diagrams provide a template for internal communication for IP. The number of servers in a cluster may different from environment to environment. The lower environments may not need load balancer or clustered server configuration.

Internal communication is depicted through multiple logical diagrams to provide better readability.

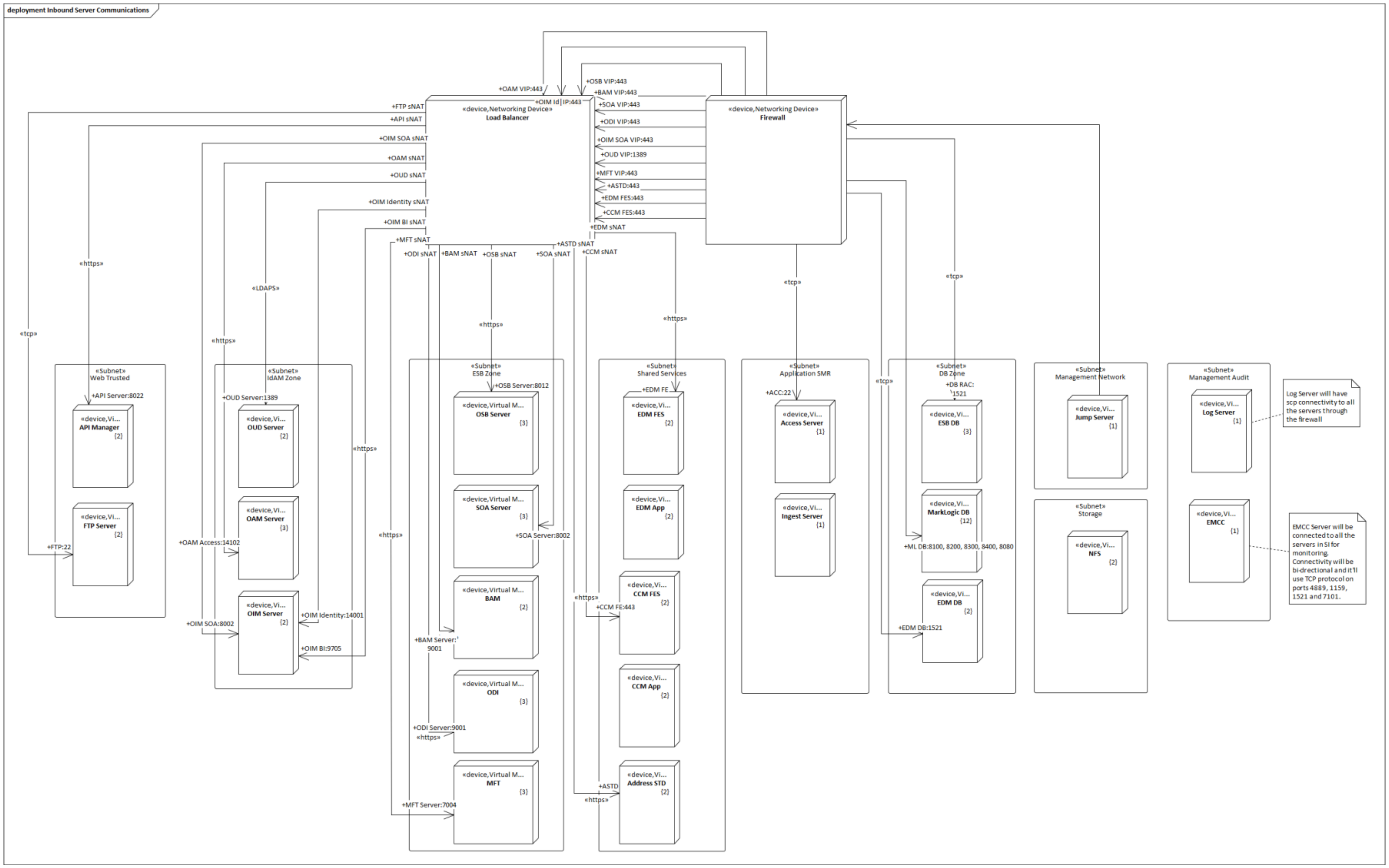
The internal communication diagram consists of following diagrams:

- Inbound traffic communication – The diagram depicts that any inbound traffic to any server constituting IP passes through the firewall and load balancer.
- Outbound traffic communication – The diagram depicts that any outbound call from any subnet of Integration platform passes through the firewall.
- IDAM internal communication – The diagram depicts any internal communication needed by IDAM components to cater to inbound requests.
- ESB internal communication – The diagram depicts any internal communication needed by ESB components to cater to inbound requests.
- SMR internal communication – The diagram depicts any internal communication needed by SMR components to cater to inbound requests.
- ESS internal communication – The diagram is pending and will be completed in iterative fashion as and when more information is available through discussion with the tool vendors.

6.5.1 Inbound Traffic Communication

The diagram depicts flow of inbound traffic to all the servers that constitute Integration Platform. It shows that all the inbound traffic passes through firewall to enforce required security controls. Apart from that load balancer is used to distribute the traffic properly across clustered servers.

Figure 6-58: Inbound Traffic Communication



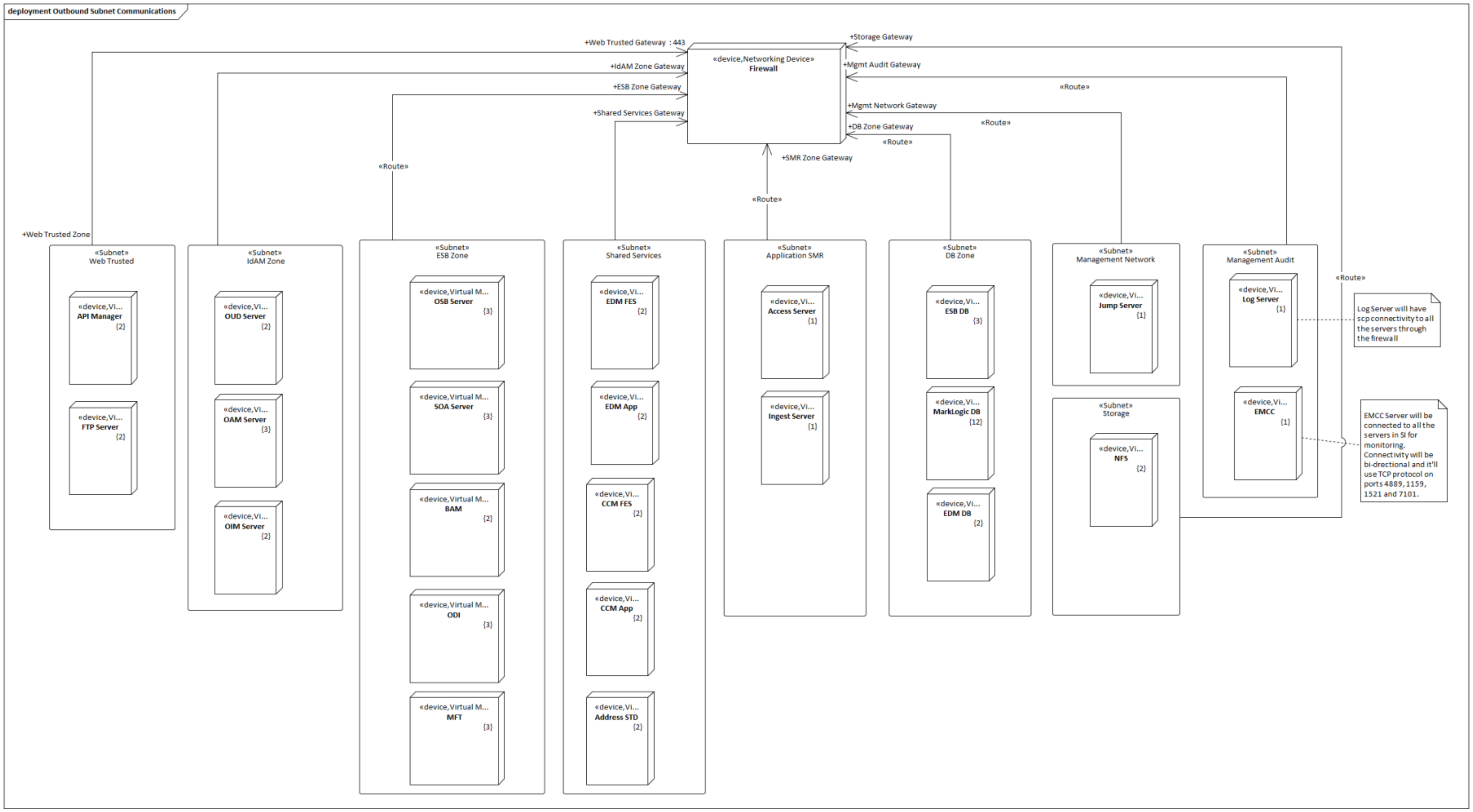
Log Server will have scp connectivity to all the servers through the firewall

EMCC Server will be connected to all the servers in SI for monitoring. Connectivity will be bi-drectional and it'll use TCP protocol on ports 4889, 1159, 1521 and 7101.

6.5.2 Outbound Traffic Communication

The diagram depicts flow of outbound traffic originating from any subnet. It shows that all the outbound traffic passes through firewall to enforce required security controls.

Figure 6-59: Outbound Traffic Communication



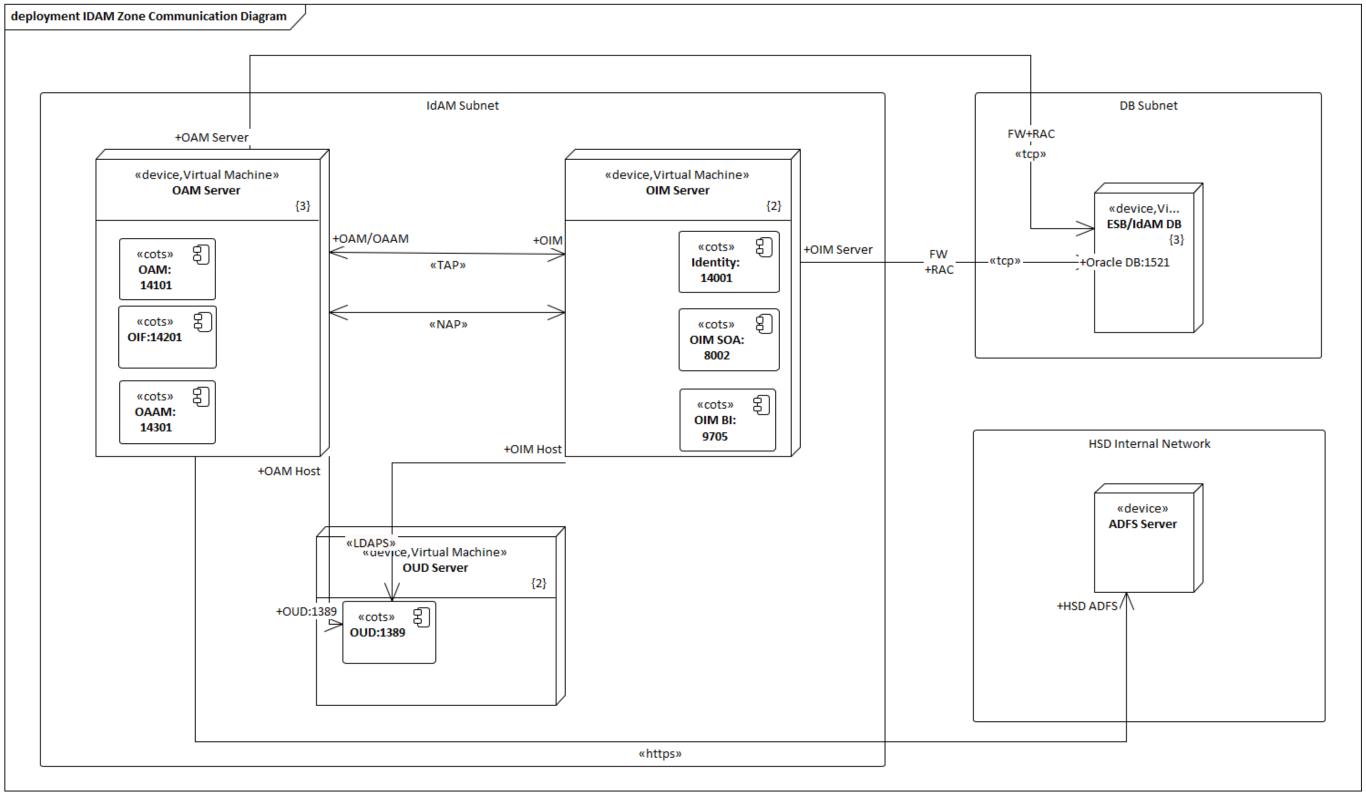
Log Server will have scp connectivity to all the servers through the firewall

EMCC Server will be connected to all the servers in SI for monitoring. Connectivity will be bi-drectional and it'll use TCP protocol on ports 4889, 1159, 1521 and 7101.

6.5.3 IDAM internal communication

The diagram depicts internal communication between components that are required to fulfill inbound request to IDAM subnet. Inbound requests are received by either OAM server or OIM server.

Figure 6-60: IdAM Zone Communication Diagram



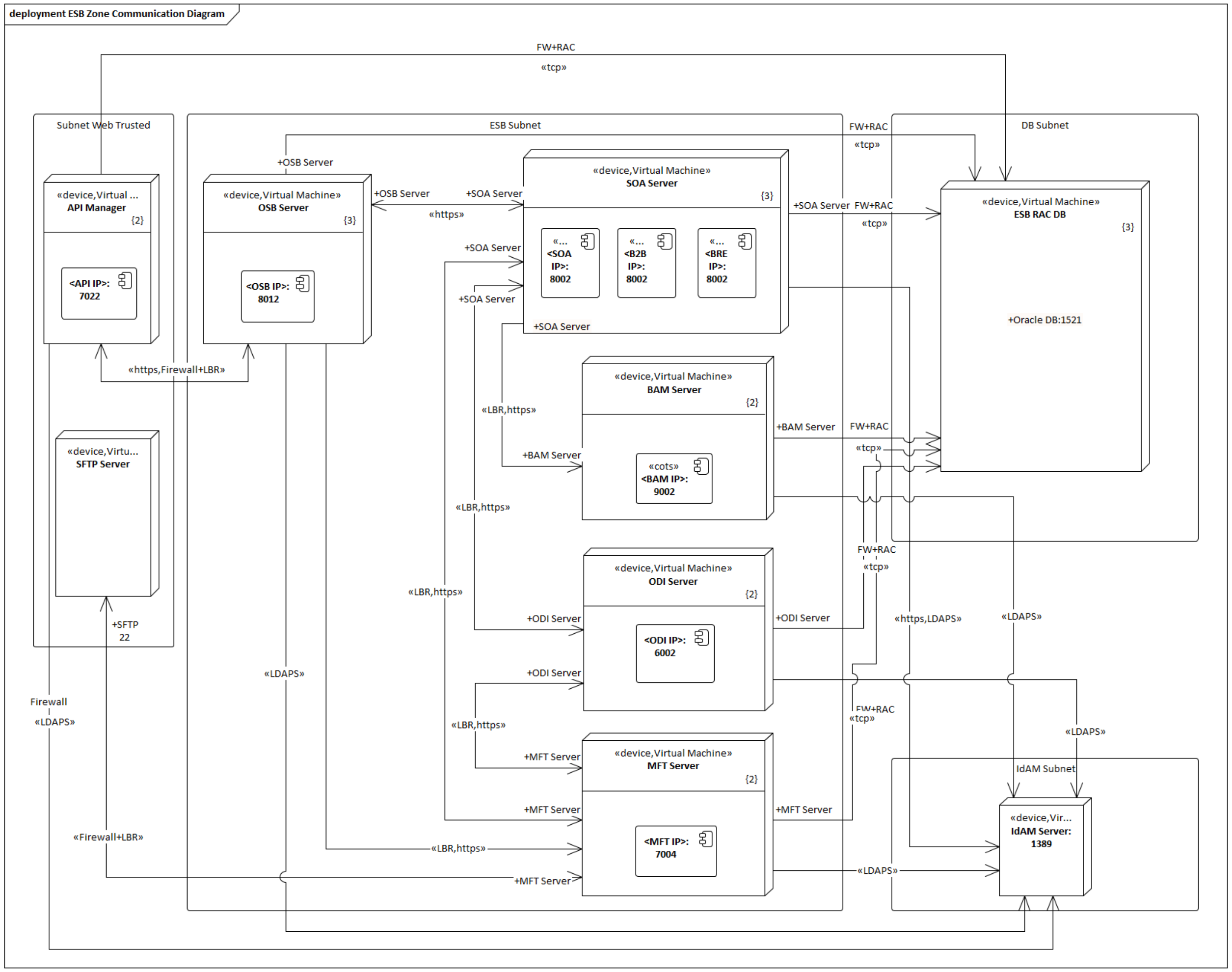
6.5.4 ESB Internal Communication

The diagram depicts internal communication between components that are required to fulfill inbound request to ESB platform which is divided into two subnets:

- Web Trusted
- ESB subnet

Either the API Manager or OSB server receives inbound requests. For file communications, SFTP server is used.

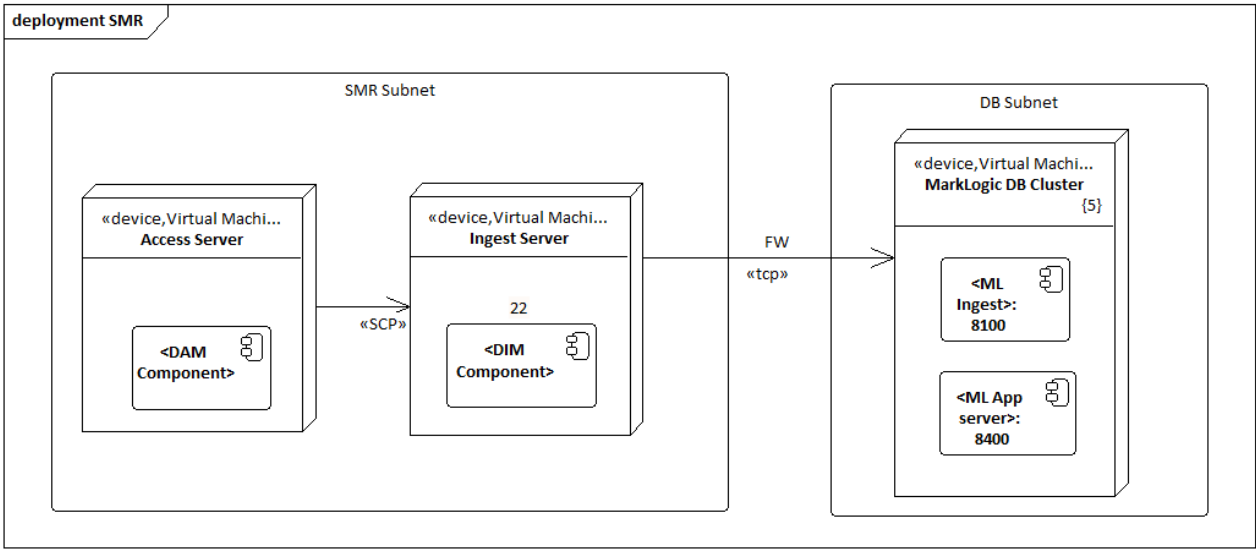
Figure 6-61: ESB Zone Communication Diagram



6.5.5 SMR Internal Communication

The figure below shows the internal communications of SMR.

Figure 6-62: SMR Components Internal Communications



6.5.6 ESS Internal Communication

The diagram will be completed in iterative fashion based on information gathered from tool vendor discussions and tool documentation.

7 System Integrity Controls

The SI Platform has implemented processes and solutions to ensure that the data that is created, processed, and stored is not subject to unauthorized modifications. The SI System uses cryptographic algorithms to maintain the integrity of data in transit and at rest. The implementation of the encryption will meet many of the SI and privacy controls.

These solutions include:

- Access controls to restrict access of critical data items to only those access types required by users/operators.
- The Integrated Platform application ensures all interfaces are encrypted to protect all data being exchanged with outside entities. Communication and messages exchanged, whether with systems or system users, are secured via TLS v1.2. TLS uses a checksum to protect the integrity of the data in transit.
- The Oracle Exadata utilizes Transparent Data Encryption (TDE), which protects the integrity of the data at rest.
- Auditing and logging captured by Splunk to meet control, reporting, and retention period requirements for operational and management reports.

- Application audit trails to dynamically audit retrieval access to designated critical data.
- Standard tables to be used or requested for validating data fields.
- Verification processes for additions, deletions, or updates of critical data.
- Ability to identify all audit information by user identification, network terminal identification, date, time, and data accessed or changed.

8 External Interfaces

External interfaces are data exchanges between the HHS 2020 systems and any external entities. An entity could be (1) another government agency like CMS, SSA etc. (2) two systems within the same agency (CMS MBES, CMS T-MSIS) or (3) an external business partner (e.g. a provider, MCOs). These interfaces replace the legacy interfaces with a standards-based approach that enables an SOA based, MITA compliant approach.

The following are the architecturally significant requirements that are addressed in this section.

- Facilitate integration with access to services for data sharing between applications and entities, in accordance with service contracts and security policies.
- Integration Requirements will enforce service designs (protocols used, description of services, naming conventions used, standards for asynchronous vs synchronous invocations, message encoding/markup, exception management) that are consistent with other module implementations as well as standards of the SI Platform.
- Interface Integration approach/patterns
    - Web services
    - Batch
    - ETL
    - Secure file transfer
- Source and target definitions

The following sections cover the high-level interface architecture along with detailed design elements of various interface integration patterns. This section also describes how the architecture handles the cross-cutting concerns like security, monitoring, transaction management, performance, and error handling. This section also covers some common use cases where the interfaces platform is leveraged for communication with external entities.

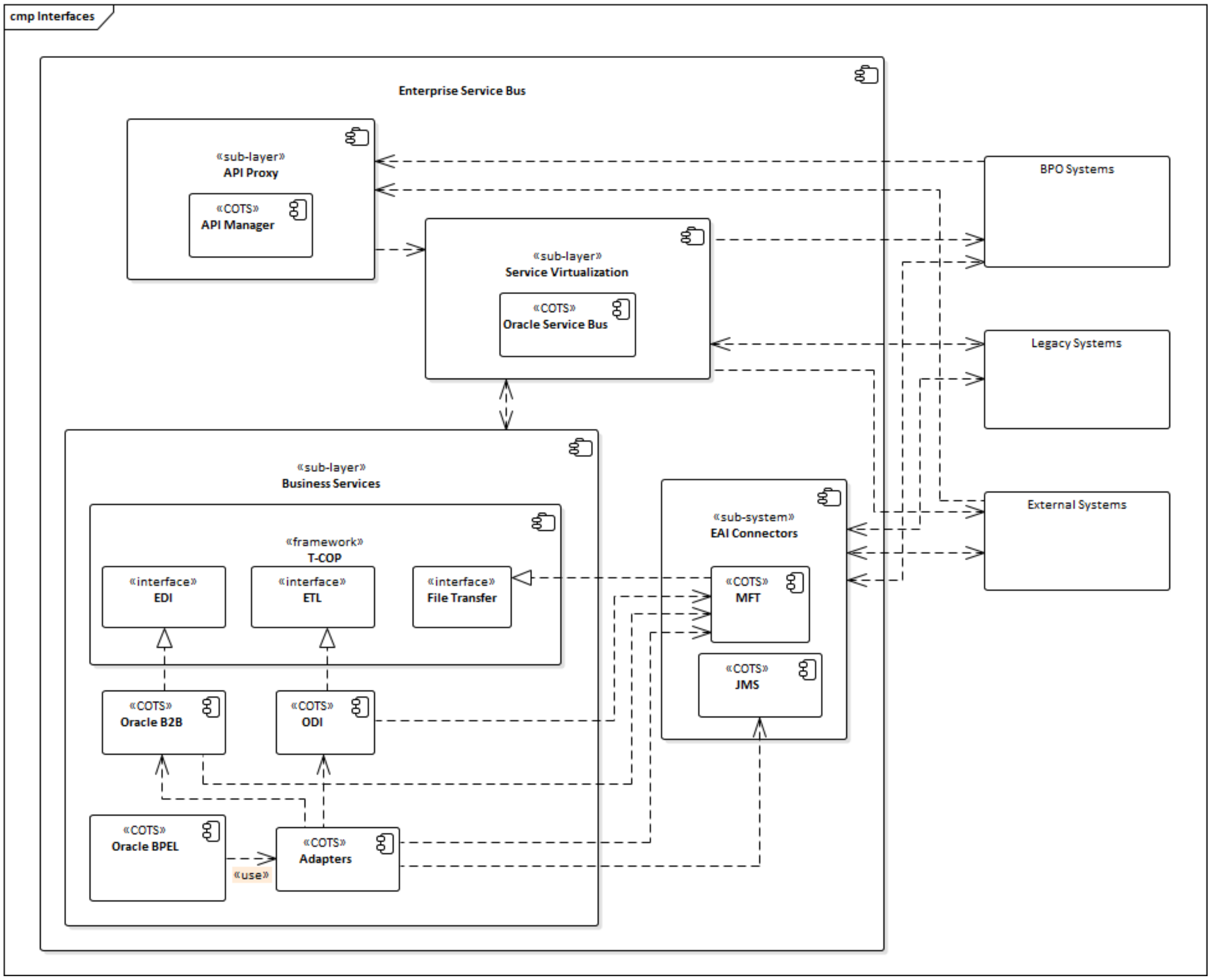
8.1 Alignment with PADU Approach and MITA Technical Strategy

The external interfaces will be built using Oracle Fusion Middleware COTS which is the preferred technology stack as identified in the NM HHS 2020 MITA Technical Strategy document and implement a standards-based interface to the external modules. Since Oracle Fusion enables to achieve maximum configuration and minimum effort for developing WSDLs, XSLT transformations, the alignment with PADU for this framework is deemed as “Preferred” level.

8.2 Interface Architecture

The SI Platform acts as the single window gateway for all interfaces that are external in nature. The following figure illustrates the external interfaces in the context of the system architecture. The external interfaces architecture is based on and a logical extension to the ESB architecture. The interfaces use SOA backplane as the foundation for web service communication, file transfer, policy enforcement, monitoring, and governance. The Integration platform enables other legacy and MMISR modules (DS, QA, BMS, and FS) to communicate with an external entity in a robust, standardized, and secure fashion. This architecture enables separation of concern for all participating modules, allowing them to focus on core business areas and leaving the transport, security, and other cross-cutting concerns to the IP.

Figure 8-1: External Interfaces – System Architecture



Note: The interfaces like File Transfer, ETL, EDI etc. depicted in the above figure are for representational purpose only. These are the capabilities of the tools, which are internal to the tool itself and not be exposed as services to other components. The service contracts are the actual services/proxies for the service integration, which will make use of the capabilities depicted in the model above.

8.3 Interface Goals

The goals of re-engineering the interfaces on top of the SI Platform are as follows:

- Modular MMIS – Evolving from a monolithic approach to a modular approach has several benefits like separation of concerns, component reusability, standardization, and many others. Dedicated modules for handling interfaces, categorized by their nature will help centralize and streamline the solution.
- Service Orientation – By embracing SOA enablement, the solution enables interoperability via the ESB-mediated service invocation. All service consumers will request the ESB to invoke the appropriate service endpoints rather than having any direct end-point knowledge. The ESB will enforce role-based authorization for service access and will carry out necessary logging of service interactions for auditing purposes.

8.4 Interface High-Level Design

The SI solution will implement interfaces using Oracle Fusion Middleware Product stack, which includes Service Oriented Architecture (SOA) suite modules, database adaptors, Oracle B2B, Oracle Data Integrator (ODI), Oracle Managed File Transfer (MFT), Oracle Service Bus (OSB), Oracle API Manager, and other adaptors and connectors. The description of these components is already covered as part of the ESB Platform Design section. The following are the high-level functional objectives of the interfaces layer of the SI Platform.

- The SI Platform will transfer the output file from any of the source systems to the target systems using MFT. It will host the FTP connectivity of several target servers where the files need to be transferred periodically, on-demand, from any of the source systems.
- Several interfaces are currently transferring the output files to DMZ (MoveIT), where the tool handles the job of transferring the file to respective targets. The SI Platform will be the single source for all DMZ related transfers using Oracle MFT.
- Some interfaces of the MMIS systems are web services based. The SI Platform will leverage the API Manager and Oracle Service Bus components to handle these invocations in a centralized fashion.
- The interfaces that require transformations and processing of the output files before reaching target systems, can utilize the hybrid exchanges where the records from source systems are transformed or translated by the SI Platform and the desired output files are created as per target specifications.

8.4.1 Integration Patterns

Interfaces are broadly classified into two categories: file-based exchanges that are mostly non-real time in nature, and web services-based exchanges that are mostly real-time in nature. The interfaces architecture leverages the core ESB components of the SI Platform. As most of the exchanges between the systems are file-based, Oracle MFT is heavily used as the centralized platform for all file-based interfaces with external entities. The web services-based exchanges use API manager and Oracle Service Bus to manage and publish web services to external entities. This pattern supports both SOAP and REST based web services. All the patterns described below will leverage the SOA framework to orchestrate the end-to-end service. Thus, they reuse the same underlying error logging, monitoring, security, and performance framework.

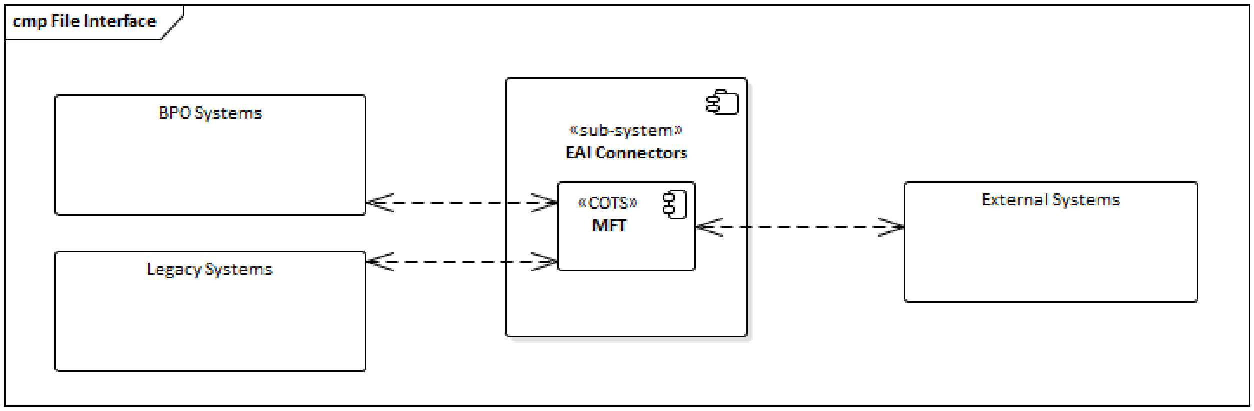
8.4.1.1 File-based Interfaces

File Transfer Interface:

Oracle MFT is an integral part of the Oracle fusion middleware stack provided by the SI Platform. Oracle MFT enables the modular contractors to communicate with external agencies using the Secure File Transfer Protocol (SFTP) protocol.

The MFT based ESB integration is explained in detail in Subsection 6.2.5.3.

Figure 8-2: File Interface Pattern



EDI Interface:

Several interfaces in the HHS 2020 ecosystem use EDI file exchanges. For example, the Dental and Healthcare Claims (837s) and Benefits Claim and Eligibility inquiry (270 and 271). These transactions use the format established to meet HIPAA requirements for the electronic submission of healthcare information. Furthermore, the HIPAA standards mandate that they are compliant with the version 5010 of the HIPAA EDI standards.

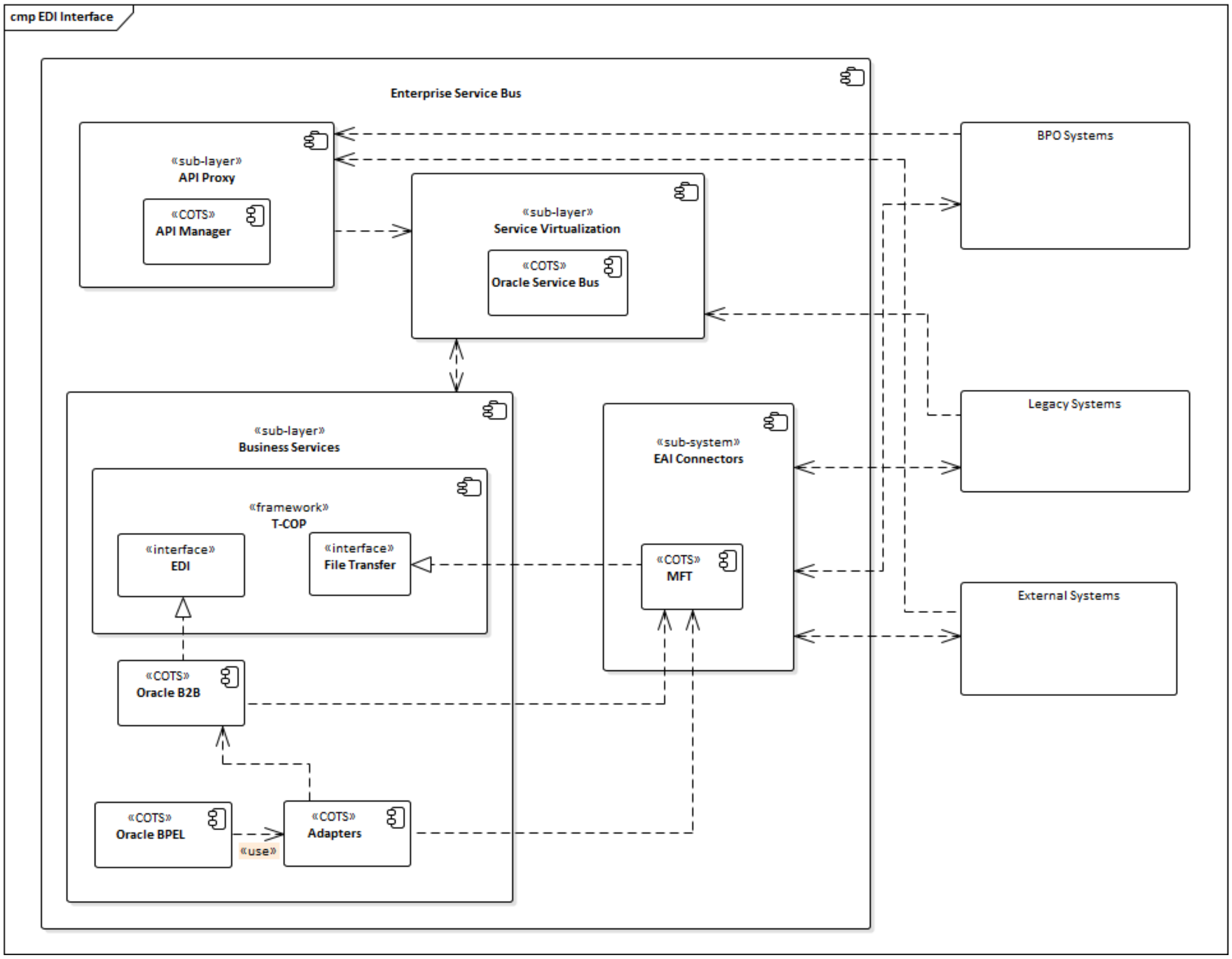
SI Platform’s Oracle B2B enables HIPAA standards compliant exchanges of EDI messages between external trading partners and MMISR modules. Oracle B2B supports both the EDIFACT and X12 flavor of EDI. The SI Platform acts as the transport and validation layer for EDI exchanges.

The following are two types of validation supported by the B2B and enabled SI Platform:

- Type 1 EDI Standard Integrity validation: Validate basic syntactical integrity of the EDI message.
- Type 2 HIPAA Implementation Guide Requirement validation: Validate HIPAA requirement-guide-specific syntax requirement by checking limits on repeat counts, used or not used qualifiers, code, elements, and segments.

Type 3 through Type 7 validation are performed by the respective MMISR module. The MMISR modules will configure their respective trading partner profiles on the SI provided B2B platform.

Figure 8-3: EDI Interface Pattern



File Exchange – Process Steps

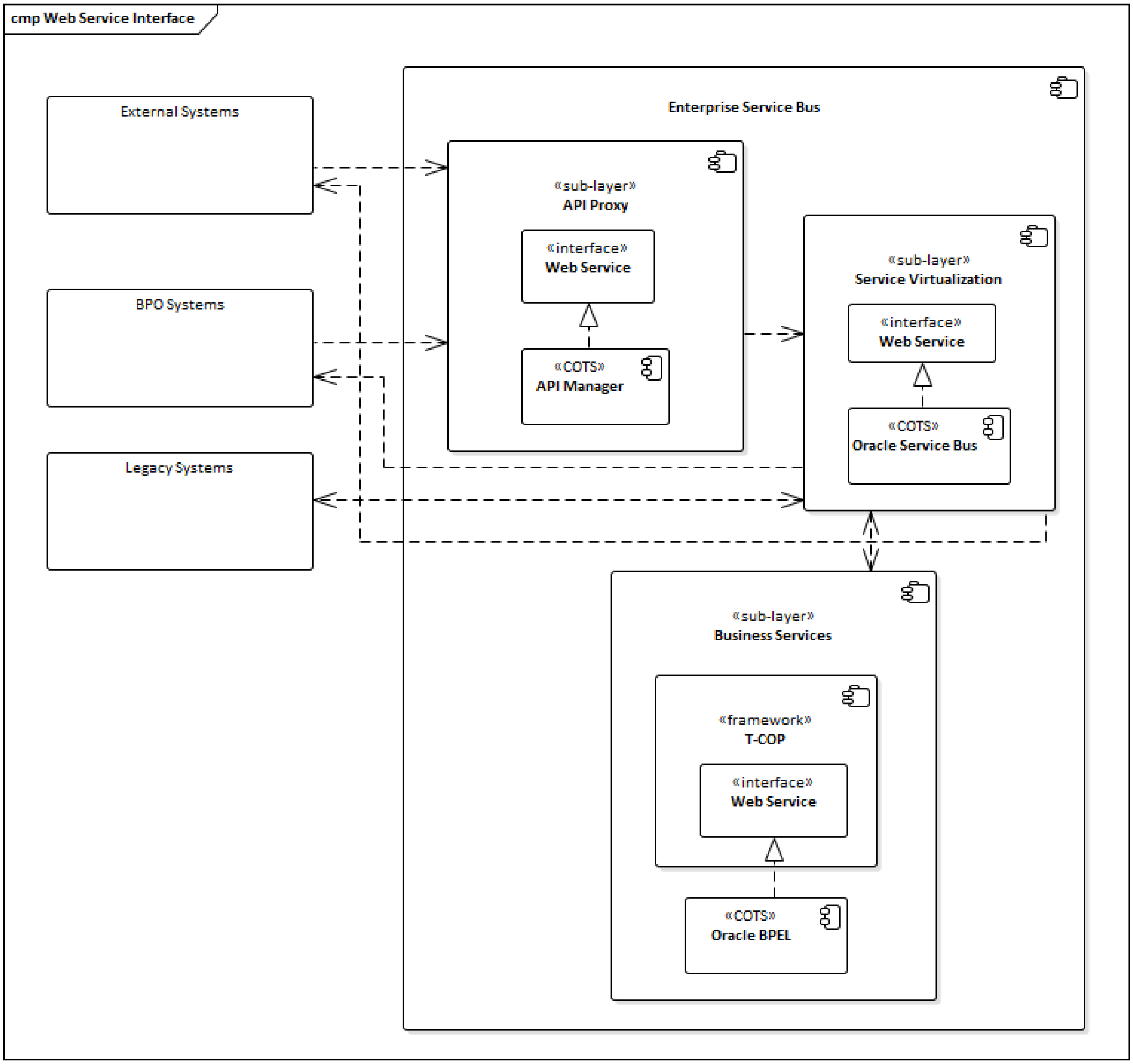
The ESB supports the exchange, transfer, decoding, and processing of files. These files include:

- Electronic Data Interchange (EDI) file transfers, encoded in the X-12 format, between NM HSD and its trading partners.
- Delimited-and fixed-width file exchanges between NM HSD and external agencies.
- Structured files such as JSON and XML received from web services or systems.

8.4.1.2 Web Service Interfaces

The SI Platform exposes business functionality for consumption by trading partners and external agencies. Web services support interoperable machine-to-machine interaction over an IP network through a set of open standards-based. The web services description, expressed in the Web Services Description Language (WSDL), describes the rules for communication between the web services consumer (e.g., the trading partner) and the web services provider (e.g., the SI Platform). The description includes message formats, datatypes, transport protocols, and transport serialization formats.

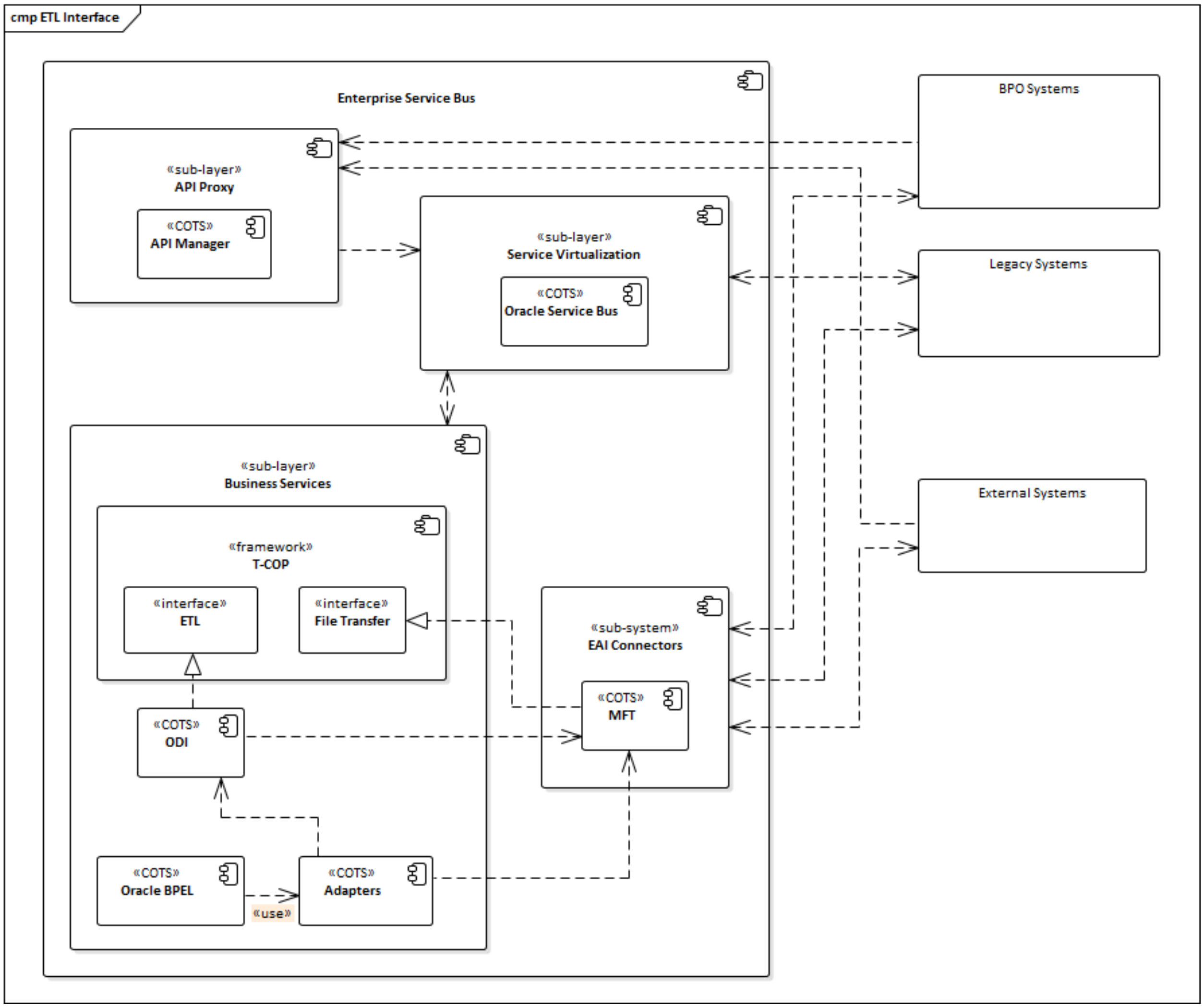
Figure 8-4: Web Services Interface Pattern



8.4.1.3 ETL Interfaces

ETL exchanges are essentially file or web-based interfaces, however, these interfaces require some additional processing in the form of protocol translation or message transformation.

Figure 8-5: ETL Interface Pattern



8.4.2 Transaction Management

Several current interfaces in the HHS 2020 systems are batch based, non-real time, in nature. These interfaces follow a one-way fire and forget approach. In this approach, the source system sends a message to the interface layer to process and does not wait for a response. If the interface processing results in some errors, the source system will receive an error file at the same location after a pre-determined number of minutes or hours.

8.4.2.1 Real-Time Transactions

The real-time transactions are either SOAP or REST based web services. Regardless of the type, these web services support stateless mechanism, where transactions are atomic in nature. These web services leverage the XA compliant 2-Phase commit mechanism to ensure that the transactions execute as an atomic unit and prevents inconsistencies.

For further details about real-time transaction management, refer to Subsection 6.2.4.1.

8.4.2.2 Non-Real Time Transactions

The MFT application provides a monitoring dashboard for viewing real-time transfer sessions and user statistics. This includes Oracle B2B based transfer sessions as well. This dashboard graphical view contains information about the file transfer instances, such as the source, target, transfer status, size, etc. Typically, an administrator accesses the monitoring dashboard in the event of unusual application behavior. The monitoring functionality also provides the application administrators a means of control to start, pause, resume, or cancel file transfers. Control of file transfers is especially useful when the administrative discretion is needed while transferring important files, very large files, and a large volume of files.

8.4.3 Monitoring

The SI Platform provides tools such as administrative web pages and application consoles to monitor, track, and manage the runtime parameters of all service instances. For instance, ESB administrators will be able to:

- See request messages to the ESB proxy business services.
- See the status of service instances (processing, completed, error, or waiting).
- Use the ESB tools to manage and influence the completion of a service flow. For example, an administrator can resubmit a failed service request.
- Collect and view runtime statistical information about job sequences, proxy services, and service endpoints. This information provides an effective way of determining the runtime performance of the ESB.
- Use the collected statistical data to analyze network traffic patterns.
- Use the collected statistical data to estimate the service load and the rate of use.

In addition, the SI team will monitor the day-to-day operations and associated metrics using Oracle BAM Monitor Express. The Monitor Express offering from Oracle BAM provides high-level instrumentation of BPEL process; automatically handling Oracle BAM data object deployment and population. BAM provides end-users with the ability to create real-time business dashboards.

Oracle BAM 12c supports, in real-time, changing dashboards that update without having to refresh the browser and tactical dashboards that allow a user to change parameters to see a new perspective of the data without having to develop a database query.

8.4.4 Security

Security and Privacy controls will be implemented for secure, steady, and accurate facilitation of centralized interface management between all identified Interface partners and the SI Platform. These controls will allow the SI Contractor to provide oversight, design and develop interfaces with external partners to replace the legacy interfaces in a secure manner.

These controls will be implemented in compliance with all applicable MARS – E Version 2.0 suite of security controls. Interface management will be done in a manner to ensure that the Confidentiality, Integrity, and Availability (CIA) of data is protected at all times.

The SI Platform needs to transfer files from/to a number of interface partners at various intervals like daily, weekly, monthly, quarterly and yearly. Some interface partners have specific file transfer requirements like Secure File Transfer Protocol (SFTP) and CyberFusion for Social Security Administration (SSA). The SI Platform will ensure the encryption of data at rest and in transit using encryption standards AES-256 or PGP. In addition, FTP folder level security controls, per interface partner, will be established to protect unauthorized access.

File server users access the MFT embedded file server to send and receive files. These users log into the application using FTP client software. The users can also access the file transfer API provided by the MFT to upload/download the files programmatically.

Upon registration, the IP provides each file server user with the following:

- Unique username and password.
- Directories with appropriate access rights to read and/or write.
- Required disk space in the directories.
- Encryption or decryption keys.

Security mechanisms are classified as shown below.

Table 8-1: Security Mechanism

Data Category	Security Mechanism
Data at rest	• Batch files at rest will be either FIPS-140-2 compliant Advanced Encryption Standard (AES - 256) or Pretty Good Privacy (PGP) encrypted. • Folder level security controls will be implemented. • Interface Partner-specific security controls will be implemented for file transfers (CyberFusion, SFTP, etc.).
Data in Transit	• 2-way SSL (Secure Sockets Layer) / Transport Layer Security (TLS) at the transport layer. • WS Security Username Token at the message level. • HTTPS Basic Authentication.

The Enterprise Application Security is covered in detail as part of Subsection 6.3.3.

8.4.5 Performance

Performance metrics are classified into two categories. The first is the Technical Performance Metrics, where health and metrics to Oracle Fusion Middleware components are displayed for administrators to consume and publish. The second is the Business Performance Metrics, where business SLAs and associated metrics are captured and displayed via centralized charts and dashboard.

8.4.5.1 Technical Performance Metrics

All health and performance related metrics are displayed as part of the Oracle Enterprise Manager Cloud Control (EMCC). The EMCC acts as a single pane of window for all technical performance-related metrics.

The EMCC leverages the following out of box components to collect and report metrics:

- Oracle Dynamic Monitoring Service (DMS) – The performance metrics of the Oracle Fusion Middleware components are captured by a component called DMS. The DMS captures the data related to each component's performance and pushes them to EMCC.
- Java Management Extensions (JMX) – The JMX monitoring API exposes an MBean to provide all required operations to get statistical information for any monitored services.
- Oracle Integrated Workload Statistics (IWS) – This component periodically saves the system resource usage, composite SOA metrics, and endpoint statistics to the database. The collected metrics are available in the form of CSV or HTML reports. Note that IWS will be enabled only when there is a need to investigate fusion middleware performance issues, as this puts a significant burden on the database.

Business Performance Metrics:

The BAM Monitor Express feature helps collect key performance indicators and display them on an out-of-the-box dashboard. The BAM helps to track analytics before, during, and after a risk has occurred and required course corrections can be made as and when appropriate.

For further details on overall performance, refer to Subsection 6.4

8.4.6 Validation and Error Handling

A common Integration Platform Error Handling Framework will be implemented to handle all exceptions and errors in the interfaces layer. The framework will manage all technical and business errors raised by the interfaces. The framework will include a centralized error logging and notification mechanism for the administrator to audit and take corrective actions as per the error. This framework will also have out of box fault policies configured as applicable and will differ based on each interface requirement.

For each interface, data validation will be implemented based on the schema and file layout definition. If data validation fails, then the common error handling framework will be invoked to log the validation error and for the administrator to review and take further action if needed. If data validation succeeds, then the message will be forwarded for further processing.

There is a provision of a general fault-handling framework in the ESB layer for handling faults in services. If a fault occurs during an invoke activity runtime process, the framework catches the fault and performs a user-specified action defined in a fault policy file associated with the activity. The following figure shows an example of configuring faults as part of the BPEL workflow.

If the MFT, B2B, BPEL, Mediator, or OSB service throws an error, the respective module will invoke a web service call to the error-handling framework, which will ultimately persist it in a table. Now you have a log and history of all errors encountered to which you can implement a notification.

For further details on error handling framework, refer to the Subsection 6.2.4

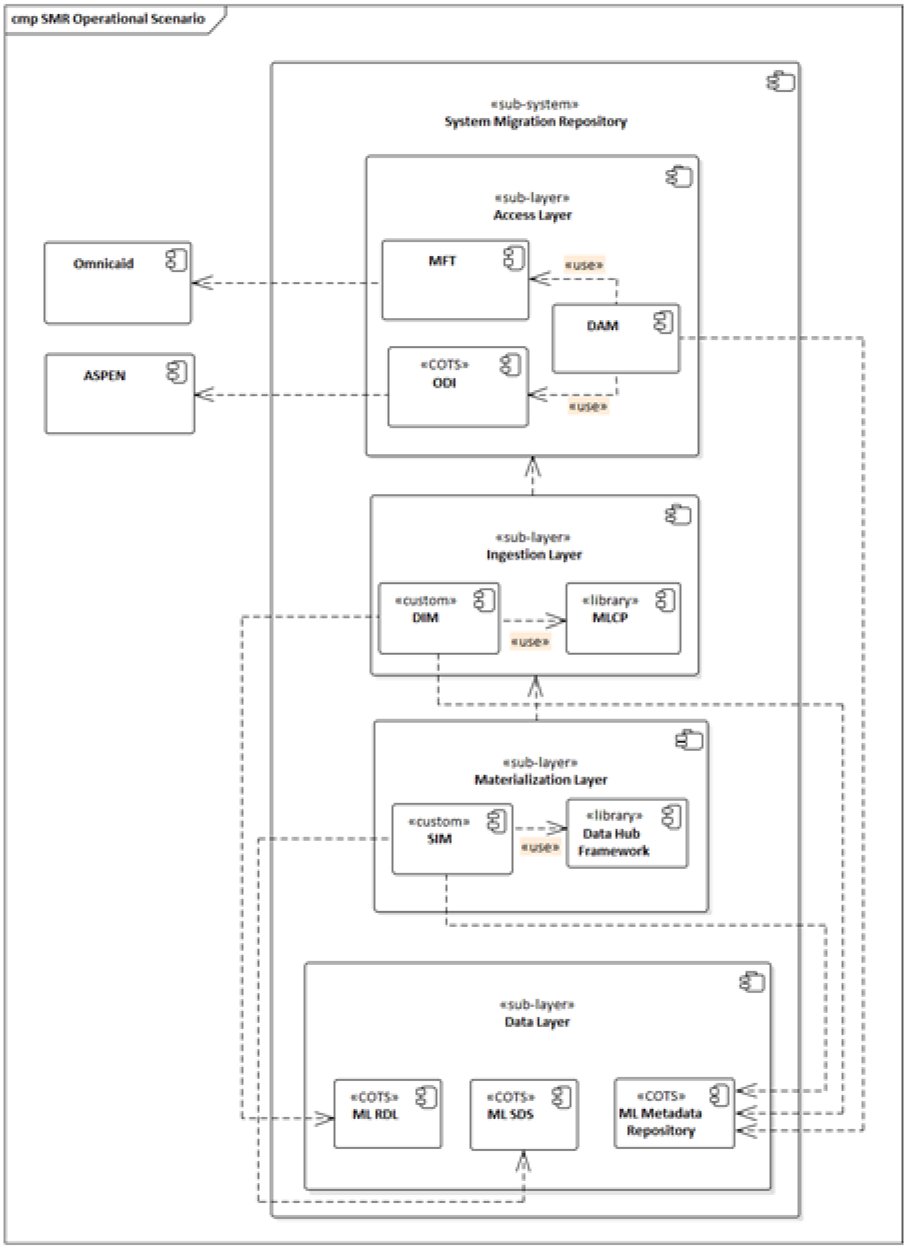
9 Operational Scenarios

The following are some of the Operational Scenarios of SI Platform.

9.1 SMR

The figure below shows the operational scenario for the SMR involving data ingestion from Omnicaid and ASPEN, through the DAM, DIM and SIM components. The data from each source is independently materialized in SMR into the standard SMR data model described in Subsection 5.1.1.

Figure 9-1: SMR Operational Scenario



9.2 ESB

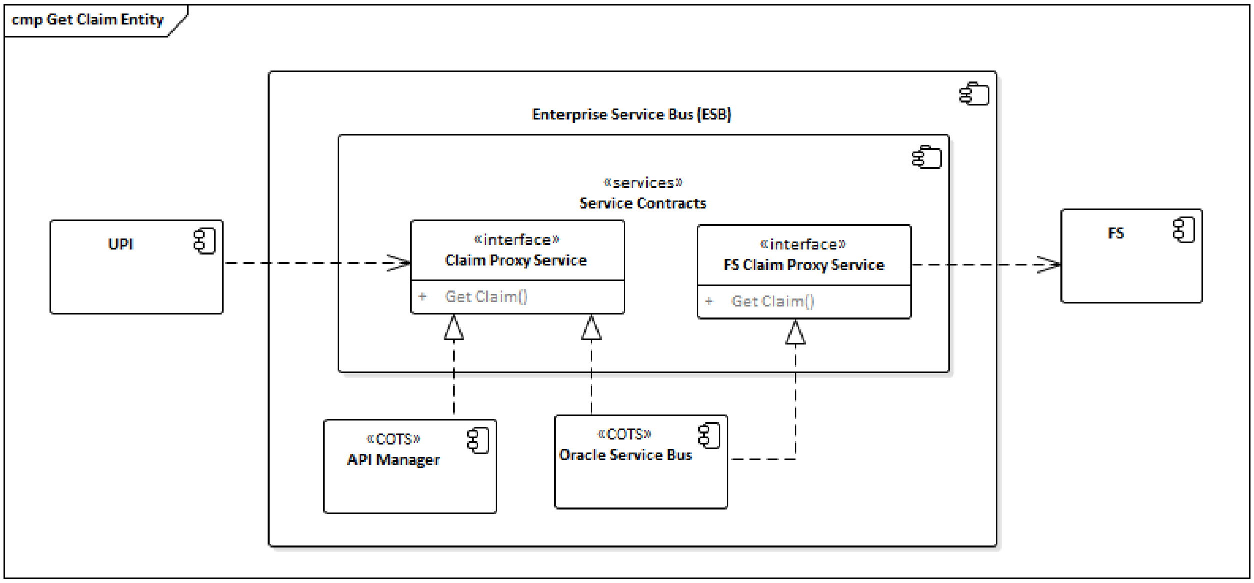
The following are the sample workflows to show service integration and orchestration capabilities of the ESB Platform.

9.2.1 Get Claim Details

This sample workflow assumes that a user of the UPI portal wanting to see the details of the claim submitted. The figure below shows the component diagram of execution when UPI module makes a call to the SI Platform get the claim entity. Since the purpose of the service call is to get the claim status

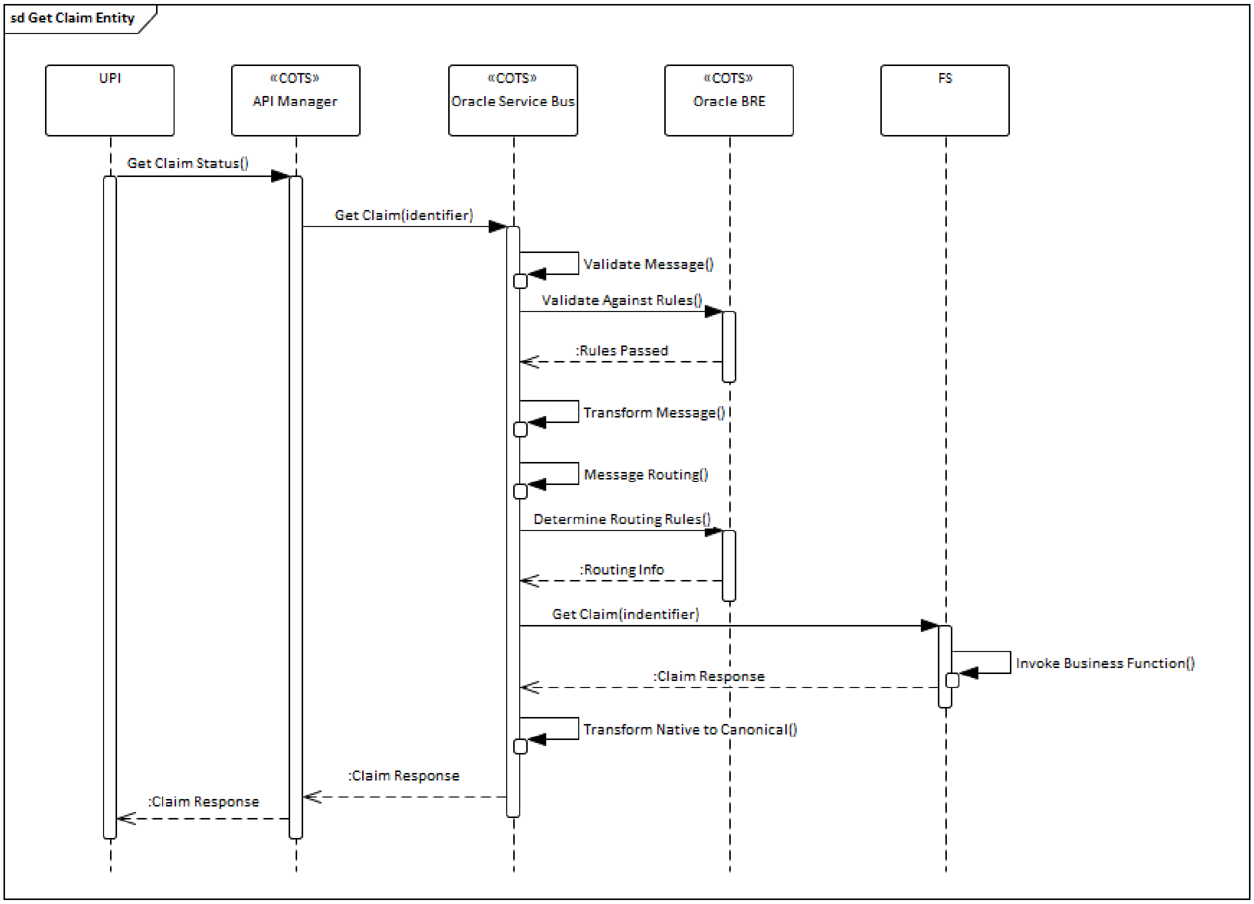
(Entity) which does not involve any business functions or business orchestration with in the ESB, the Oracle service bus proxy will make a call to the service provider directly (Assuming FS is the provider) instead of making call through Business Service layer.

Figure 9-2: Get Claim Entity - Component Diagram



The figure below shows the sequence diagram of execution when UPI module makes a call to the SI Platform to get the claim status.

Figure 9-3: Get Claim Entity – Sequence Diagram



9.2.2 Member Enrollment for Benefits

The following is the list of steps that illustrate how the ESB orchestrates services when a member applies for benefits:

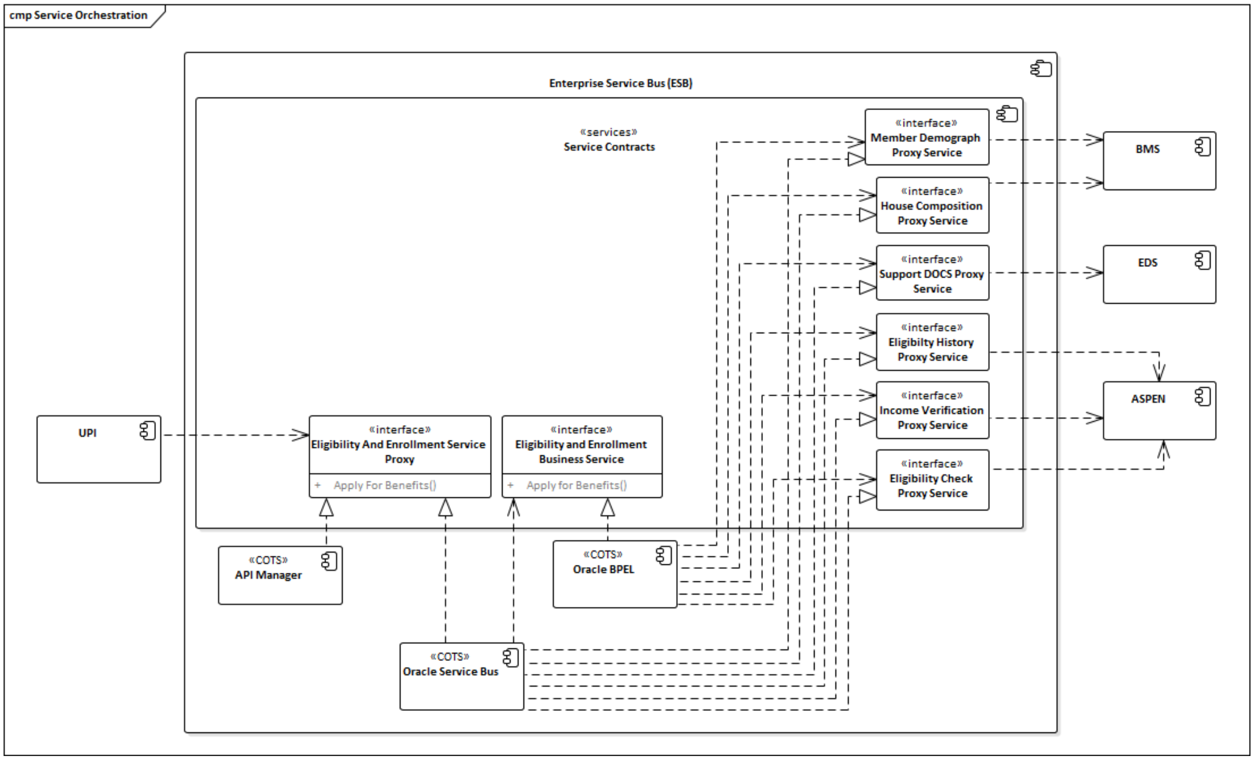
- The member enters all required information in a UPI system's portal and submits the application.
- The portal calls the EligibilityAndEnrollmentAPIService exposed through Oracle API Manager and call reached ESB and waits for a response.
- The ESB, in turn, breaks the application data into smaller fragments and orchestrates the calling of each of the services (exposed by partner applications) that comprise the EligibilityAndEnrollmentRequest service. The services exposed by partner applications include:

  1. Get member demographic information.
  2. Get member household composition.
  3. Verify member income.
  4. Get member eligibility history.
  5. Get member supporting documentation.
  6. Determine member eligibility.

- The ESB returns a list of potential benefits for which the member is eligible and can be enrolled in the client browser (portal).

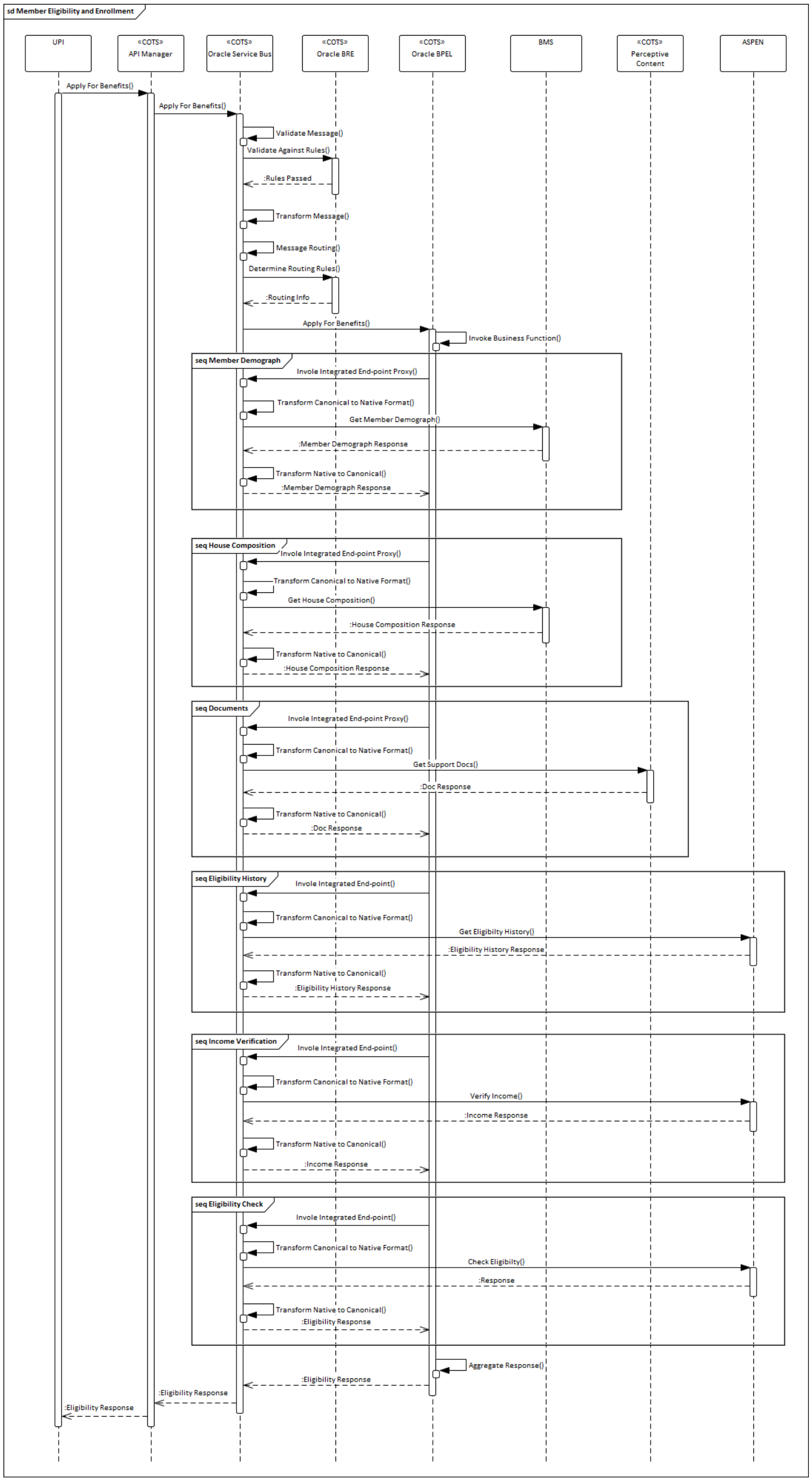
The figure below is the component diagram that represents the flow for member eligibility and enrolment for benefits.

Figure 9-4: Member Eligibility and Enrollment – Component Diagram


cmp Service Orchestration
Enterprise Service Bus (ESB)
«services»
Service Contracts
«interface»
Member Demograph Proxy Service
«interface»
House Composition Proxy Service
«interface»
Support DOCS Proxy Service
«interface»
Eligibilty History Proxy Service
«interface»
Income Verification Proxy Service
«interface»
Eligibility Check Proxy Service
BMS
EDS
ASPEN
UPI
«interface»
Eligibility And Enrollment Service Proxy
+ Apply For Benefits()
«interface»
Eligibility and Enrollment Business Service
+ Apply for Benefits()
«COTS»
API Manager
«COTS»
Oracle BPEL
«COTS»
Oracle Service Bus

The figure below is the sequence diagram that represents the flow for member eligibility and enrollment for benefits.

Figure 9-5: Member Eligibility and Enrollment – Sequence Diagram


sd Member Eligibility and Enrollment
UPI
«COTS» API Manager
«COTS» Oracle Service Bus
«COTS» Oracle BRE
«COTS» Oracle BPEL
BMS
«COTS» Perceptive Content
ASPEN
Apply For Benefits()
Apply For Benefits()
Validate Message()
Validate Against Rules()
:Rules Passed
Transform Message()
Message Routing()
Determine Routing Rules()
:Routing Info
Apply For Benefits()
Invoke Business Function()
seq Member Demograph
Invole Integrated End-point Proxy()
Transform Canonical to Native Format()
Get Member Demograph()
:Member Demograph Response
Transform Native to Canonical()
:Member Demograph Response
seq House Composition
Invole Integrated End-point Proxy()
Transform Canonical to Native Format()
Get House Composition()
:House Composition Response
Transform Native to Canonical()
:House Composition Response
seq Documents
Invole Integrated End-point Proxy()
Transform Canonical to Native Format()
Get Support Docs()
:Doc Response
Transform Native to Canonical()
:Doc Response
seq Eligibility History
Invole Integrated End-point()
Transform Canonical to Native Format()
Get Eligibilty History()
:Eligibility History Response
Transform Native to Canonical()
:Eligibility History Response
seq Income Verification
Invole Integrated End-point()
Transform Canonical to Native Format()
Verify Income()
:Income Response
Transform Native to Canonical()
:Income Response
seq Eligibility Check
Invole Integrated End-point()
Transform Canonical to Native Format()
Check Eligibilty()
:Response
Transform Native to Canonical()
:Eligibility Response
Aggregate Response()
:Eligibility Response
:Eligibility Response
:Eligibility Response

9.3 IdAM

The following use case diagrams and descriptions that are organized by the main functional areas of MMISR IdAM environment:

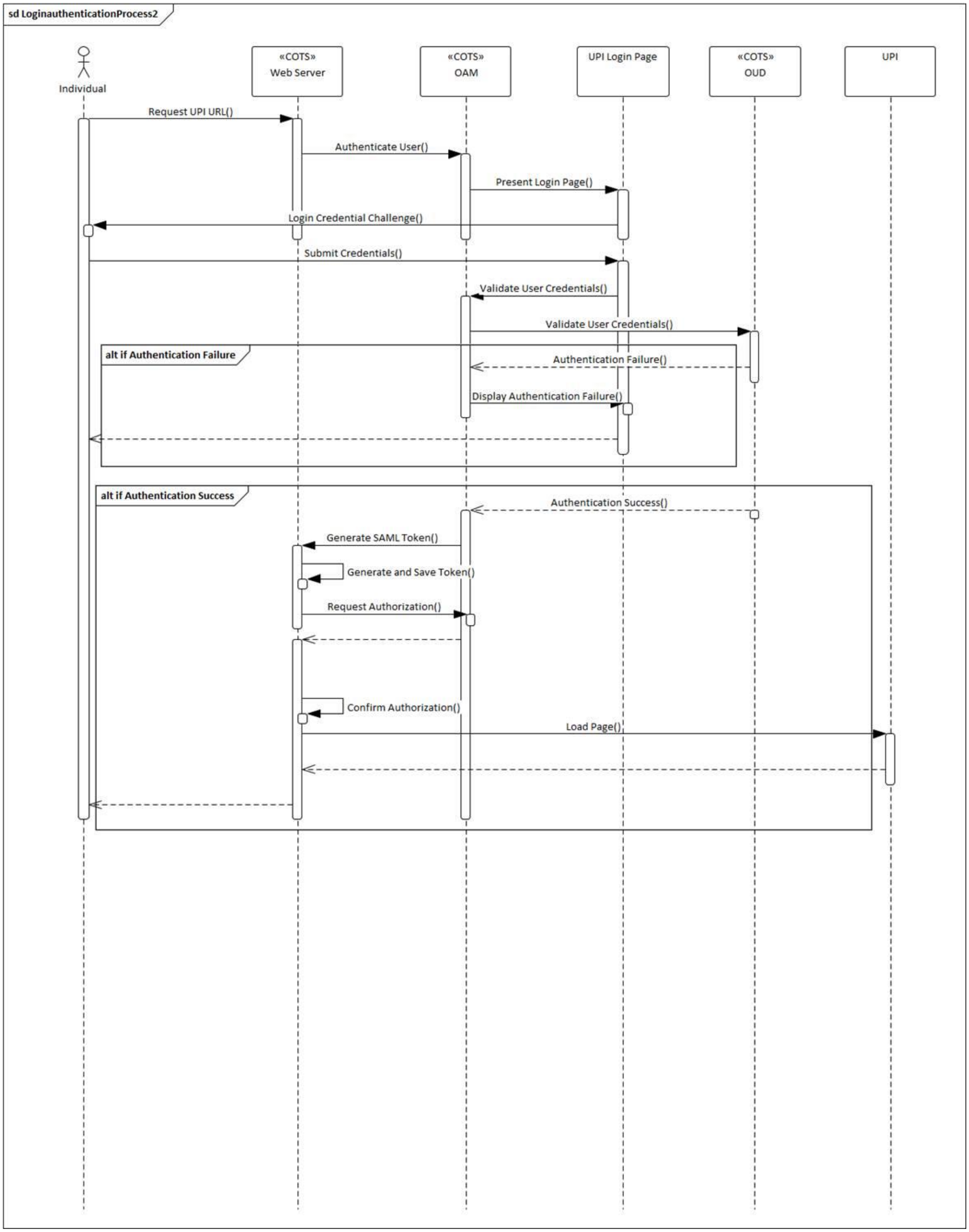
1. The runtime functionality that provides authentication for Individual login to the UPI.
2. The runtime functionality that provides authentication for Employee login to the UPI.
3. The runtime functionality that provides authentication for the system to system calls within MMISR.

9.3.1 Individual Login to UPI – Authentication

The following sequence diagram illustrates the Login authentication process for Individual accessing the UPI Portal. Below is a high-level sequence of events as depicted in the diagram

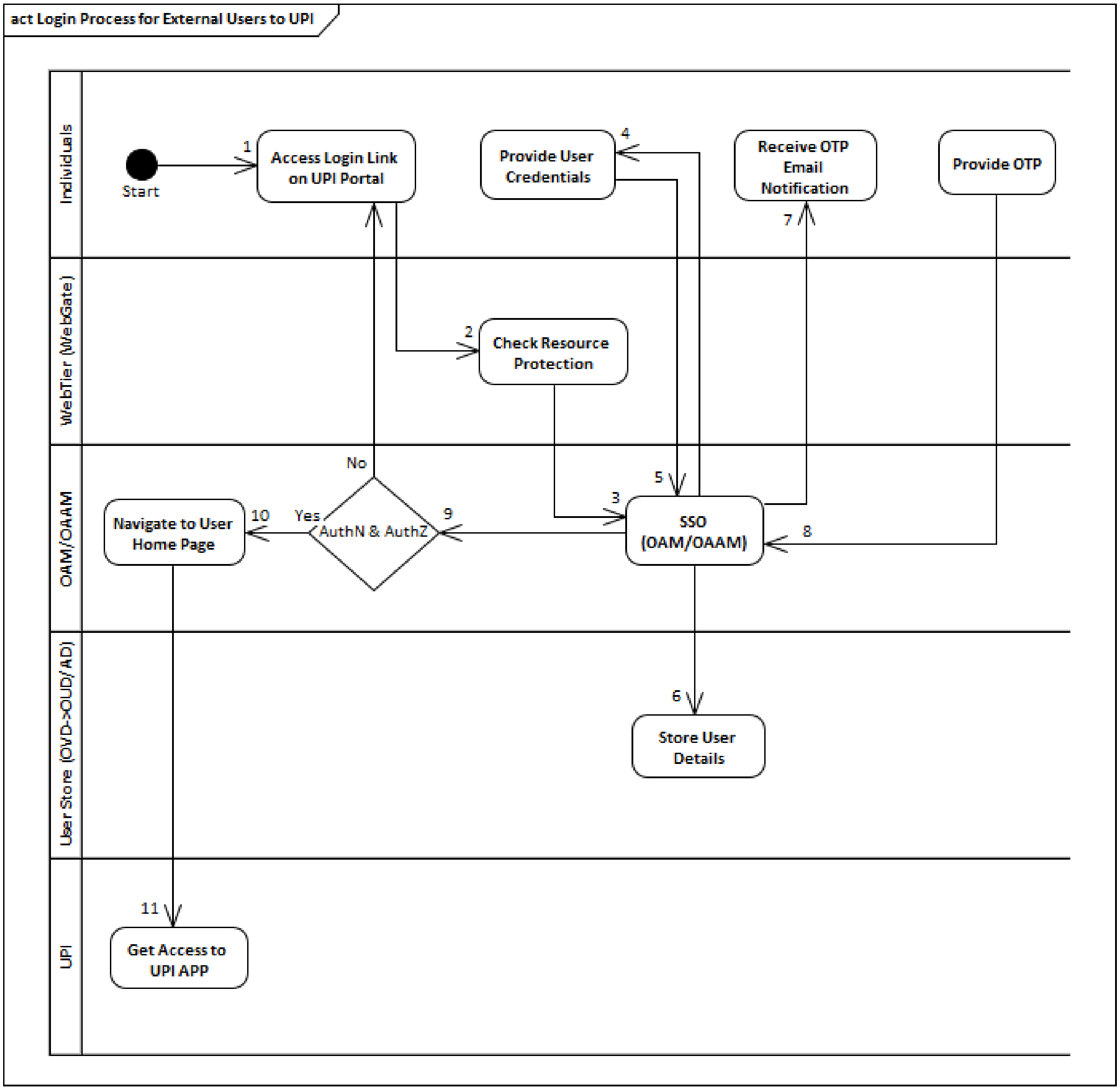
- Individual launches the UPI Portal URL which is protected by OAM Web gate and gets challenged with a login page presented by the OAM.
- User enters the credentials and OAM passes them to OUD for authentication.
- Upon successful authentication, OAM routes the users to the UPI portal with default authorization.
- Upon a failed authentication, OAM web gate routes them back to the login page.

Figure 9-6: Individual Login to UPI – Sequence Diagram



Following data flow diagram depicts the login process for External Users to UPI.

Figure 9-7: External Users Login to UPI – Sequence Diagram



The following are the process activities depicted in the figure above.

1. Individuals access the login page of the UPI portal.
2. The request goes to WebGate agent on OHS.
3. WebGate checks resources protection with OAM.
4. OAM sends the login page to the individual.

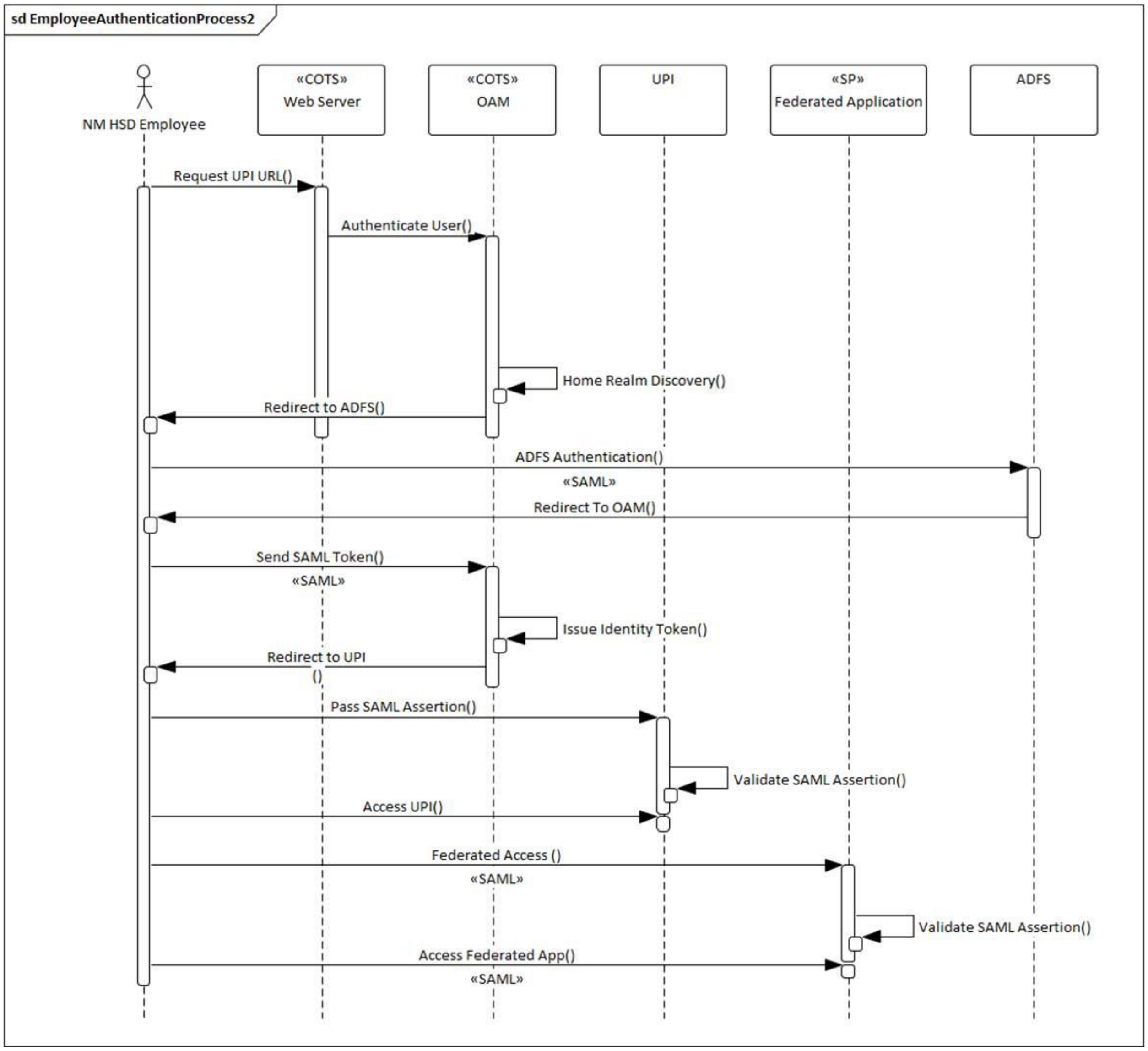
5. The user submits credentials to OAM.
6. OAM authenticates individual with the User store (OVD/OUD/AD).
7. OAAM send an e-mail notification with OTP.
8. The individual provides OTP to OAAM.
9. OAAM checks for AuthN & AuthZ with the details provided.
10. If AuthN & AuthZ is successful, the individual gets access to the user home page navigation application. If not, redirects to login age with a message.
11. Individual get access to the application.

9.3.2 Employee Login to UPI – Authentication

The following sequence diagram illustrates the Login authentication process for HSD Employees accessing the UPI Portal. Below is a high-level sequence of events as depicted in the diagram:

- Employee launches the UPI Portal URL from the HSD issued laptop.
- Based on the resource policies OAM redirects them to the ADFS for authentication and requests a SAML token.
- ADFS sends a response SAML token and is validated by OAM Federation policies.
- The SAML assertion is passed to UPI Portal and the employee is allowed to access the UPI Portal with the default role,
- Upon a failed authentication, OAM reroutes them back to ADFS login page,

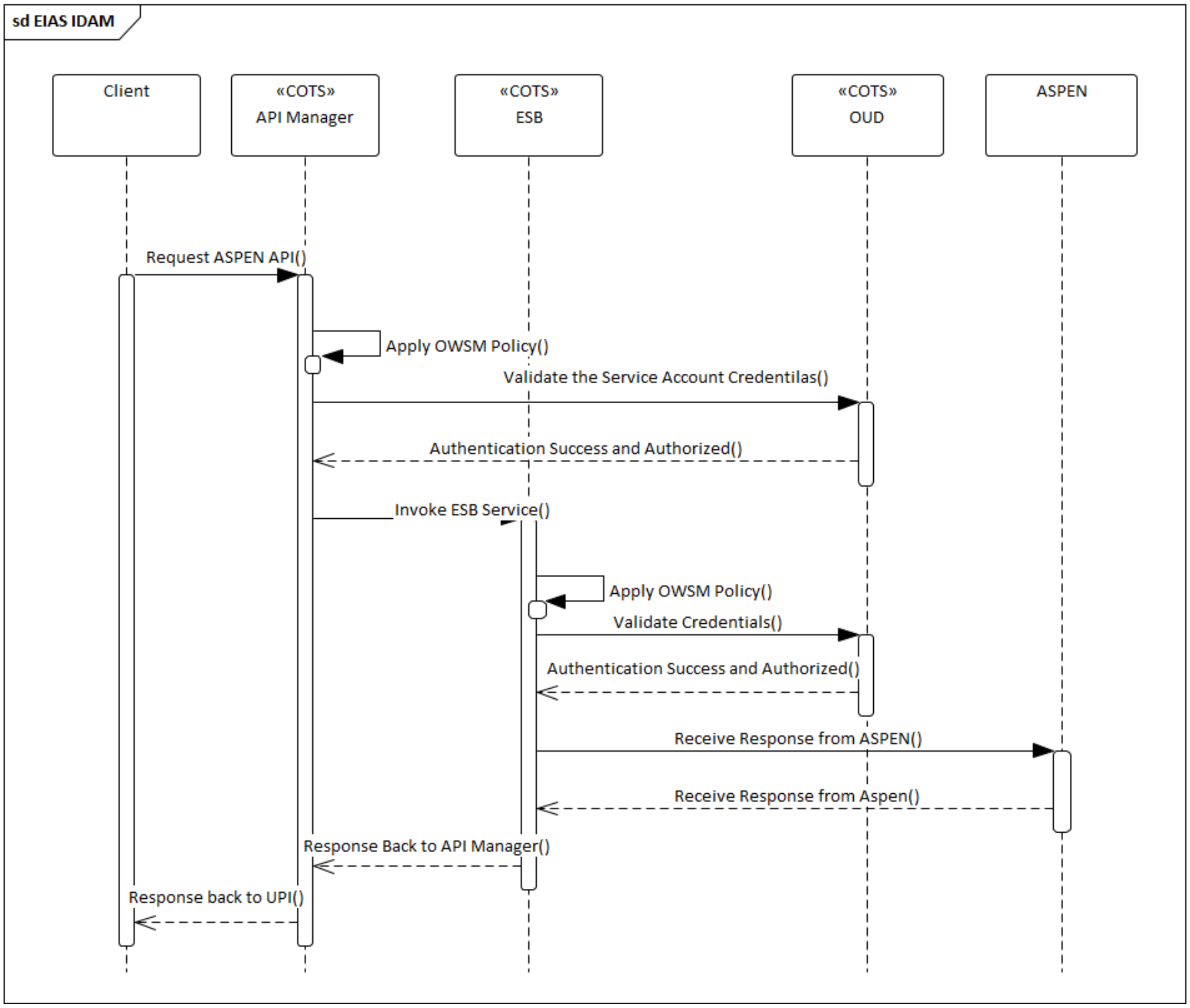
Figure 9-8: Employee Login to UPI – Sequence Diagram



9.3.3 System to System Calls - Authentication

The following sequence diagram shows the Authentication and authorization process for UPI application calling ASPEN.

Figure 9-9: System to System Calls – Sequence Diagram


sd EIAS IDAM
Client
«COTS» API Manager
«COTS» ESB
«COTS» OUD
ASPEN
Request ASPEN API()
Apply OWSM Policy()
Validate the Service Account Credentilas()
Authentication Success and Authorized()
Invoke ESB Service()
Apply OWSM Policy()
Validate Credentials()
Authentication Success and Authorized()
Receive Response from ASPEN()
Receive Response from Aspen()
Response Back to API Manager()
Response back to UPI()

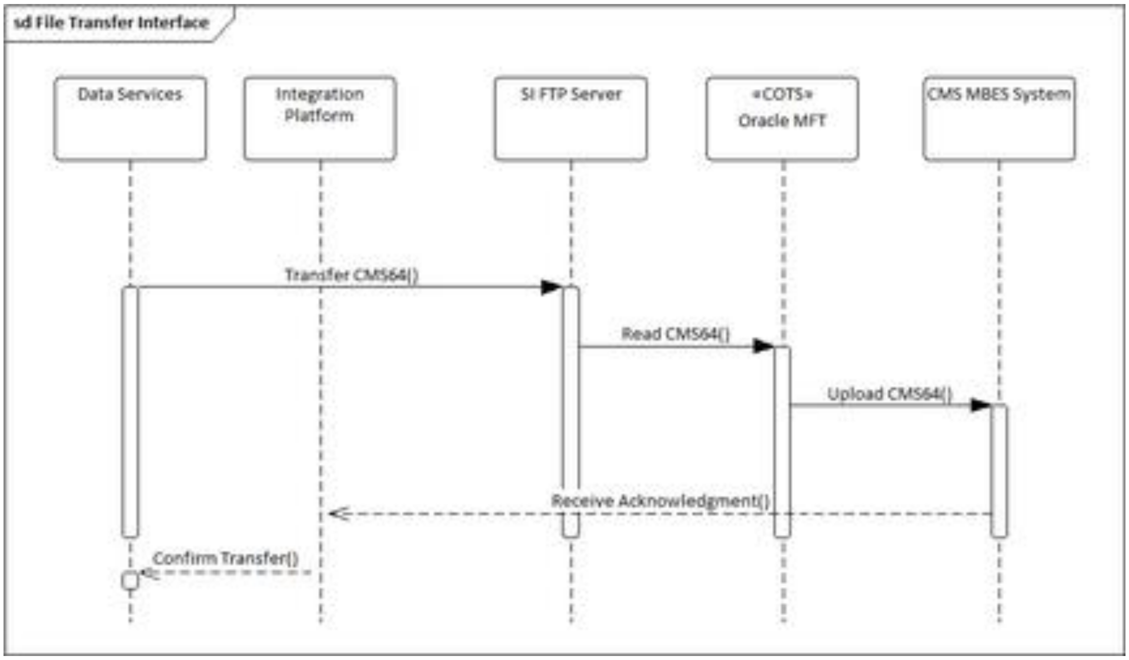
9.4 External Interfaces

The following subsections capture the various use cases that the SI Platform supports when interfacing with external partners or agencies.

9.4.1 Simple File Transfer Use Case

The figure below depicts a simple file transfer use case for a CMS64 file, generated by the data services module and transported to CMS by the IP. The IP hosts an FTP server, which the MFT will monitor for file presence and upload to the CMS system. This entire process is part of a larger Oracle SOA Composite service. The service will also ensure that the acknowledgment from CMS is relayed to the Data services module and thus, concluding the workflow.

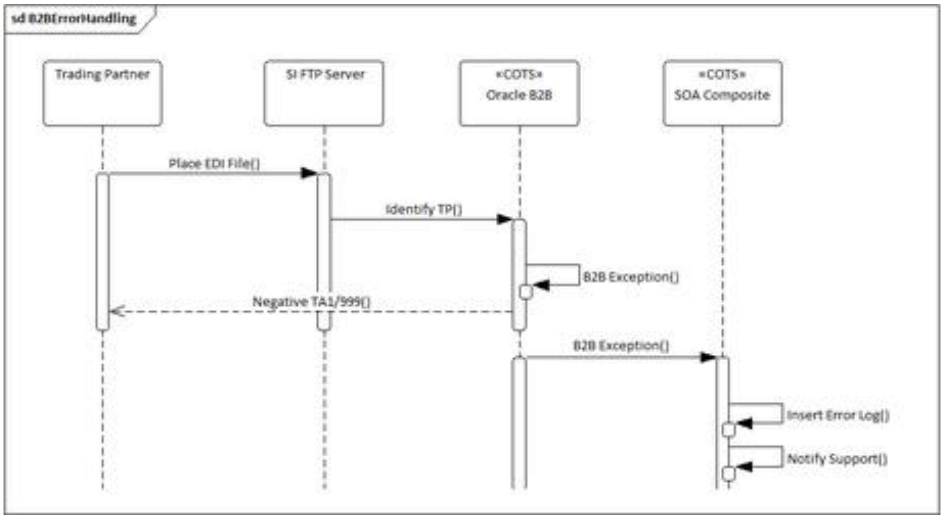
Figure 9-10: File Transfer Use Case



9.4.2 EDI Error Handling Use Case

The following figure illustrates a use case where a trading partner places an EDI file on the FTP server for processing. The B2B application attempts to identify the trading partner from the EDI file and proceed with other stages of validations. However, the process results in error and a negative acknowledgment in the form of Interchange Acknowledgment (TA1) and document type 999 Functional Acknowledgment.

Figure 9-11: EDI Error Handling Use Case



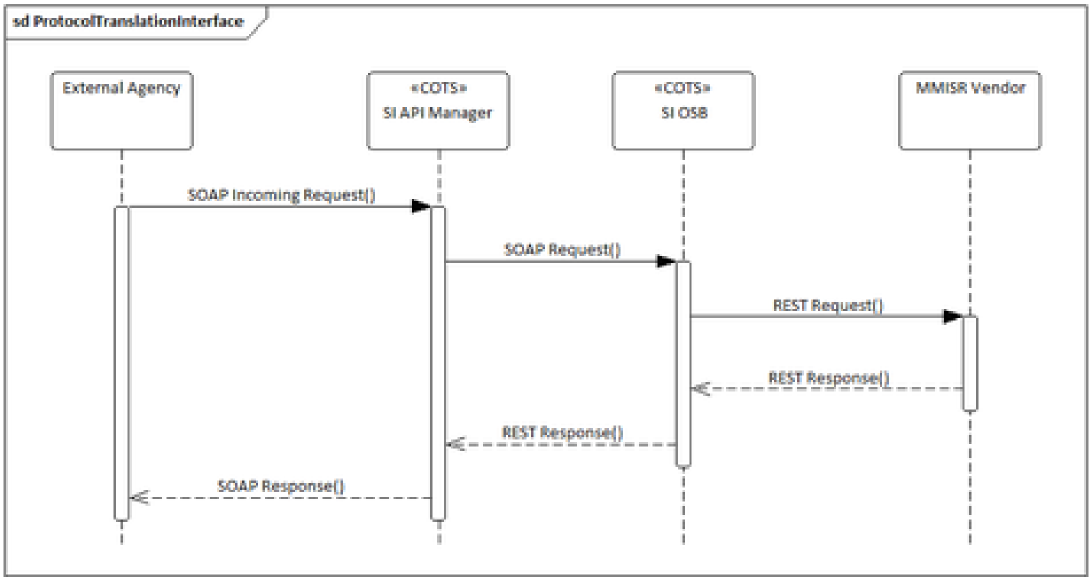
9.4.3 Protocol Translation Use Case

Some external agencies are restricted to SOAP-based web services. However, the new MMISR modules can choose to leverage the REST based web services within the HHS 2020 ecosystem. The IP acts as the protocol translation hub between external agencies and internal MMISR vendors.

The ESB also supports the following translations:

- Electronic Data Interchange (EDI) to XML.
- Flat file to XML.

Figure 9-12: Protocol Translation Use Case

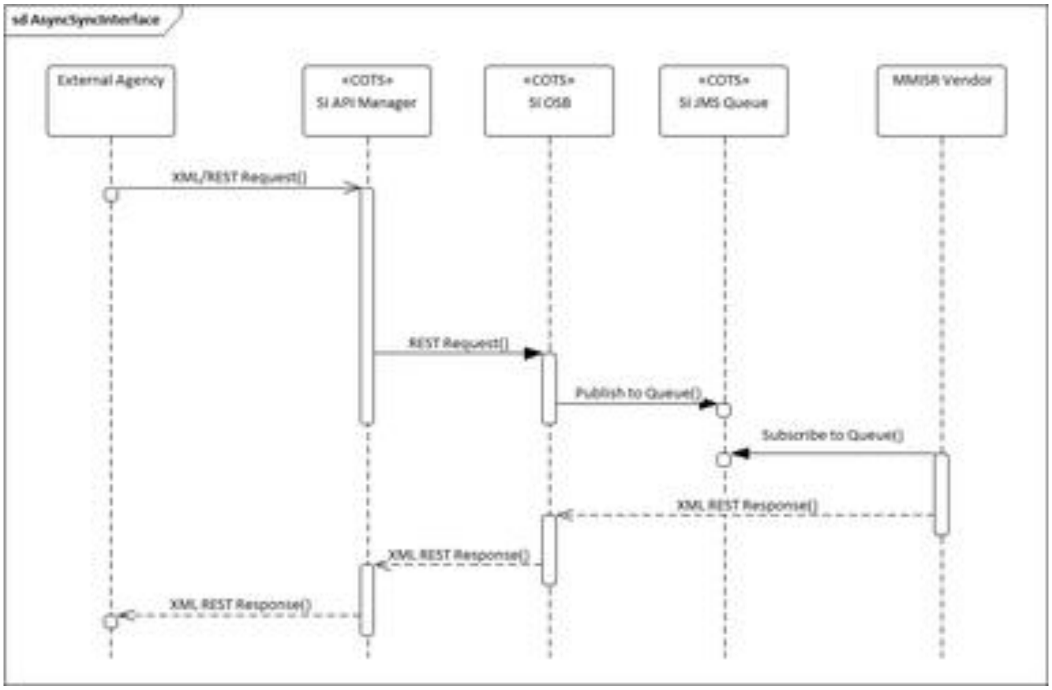


9.4.4 Asynchronous Processing with JMS Use Case

In this use case, an external agency initiates an XML REST asynchronous request, and the request is acknowledged by the IP with the HTTP status code 202/Accepted. This code conveys to the agency that the request is accepted, but not processed and the results of processing will be notified via a subsequent, independent web service call.

Upon acknowledging the receipt of the request, the IP posts the request to the JMS queue. JMS resources are created on the OSB server for the errors/process logs to be published to queues.

Figure 9-13: Asynchronous Interface Use Case



10 CMS Certification

The following are the approach to the Medicaid Management Information System (MMIS) Certification concerning the system design. The CMS will formally review this data conversion plan during the Medicaid Enterprise Certification Life Cycle (MECL) as part of the following reviews:

- R2, Operational Milestone Review
- R3, MMIS Certification Final Review

CMS may also review this deliverable during informal reviews, including consults and gate reviews.

The Certification Process Guide will be located in SharePoint and contains detailed information about MMIS Certification.

Appendix E in this document provides the checklist items for the SMR design.

10.1 Critical Success Factors

The MECL incorporates two types of Critical Success Factors (CSFs) into the certification process, programmatic and functional. The programmatic CSFs identify activities the State Project Management Office (PMO) will need to perform in managing the MMIS project. These guidelines are found in the Programmatic Tab of the Independent Verification and Validation (IV&V) Progress Report Template, which the IV&V contractor fills out as part of the regular progress reports.

11 Applicable Standards

The system design is in compliance with and supports all applicable federal, State, or other applicable regulations, guidance, and laws. This includes the applicable standards and protocols listed below and identified in Addendum 14 – HHS 2020 Security and Privacy Standards in the Request for Proposal (RFP) and in line with HHS 2020 Technical Reference Architecture and Concept of Operations documents.

- Americans with Disabilities Act (ADA).
- American National Standards Institute (ANSI) Accredited Standards Committee (ASC) X12.
- ASTM International Continuity of Care Record.
- XML/JSON
- Electronic Data Interchange for Administration, Commerce, and Transport (EDIFACT).
- e-Business XML (ebXML).
- The Open Group Standard for Service-Oriented Architecture Ontology, Version 2.0
- The Open Group Standard for SOA Reference Architecture (SOA RA)
- The Open Group Standard for the Service-Oriented Cloud Computing Infrastructure (SOCCI) Framework
- The Open Group SOA Integration Maturity Model (OSIMM)
- ISO/IEC/IEEE 12207:2017, Systems and software engineering -- Software life cycle processes
- Institute of Electrical and Electronics Engineers (IEEE) 1074-1995, IEEE standard for developing software life cycle processes
- DoD Mil-Std 2915, Systems and software engineering -- Software life cycle processes
- Federal Information Processing Standards (FIPS).
- Federal Risk and Authorization Program (FedRAMP) Certification.
- Health Level 7 (HL7):
- Specifically: HL7 Quality Reporting Document Architecture and HL7 Continuity of Care Document.

- CMS Minimal Acceptable Risk Standards for Exchanges (MARS-E) 2.0, 2015.
- FIPS 140-2, Security Requirements.
- Federal Information Security Management Act (FISMA) of 2002.
- Health Information Technology for Economic and Clinical Health (HITECH) Act.
- Health Insurance Portability and Accountability Act (HIPAA) of 1996.
- Internal Revenue Service (IRS) Publication 1075.
- International Organization for Standards (ISO) 27002, ISO 27003, ISO 27005 and ISO 27034, Information Security.
- ISO 15408, Information Security.
- National Institute of Science and Technology (NIST) 800-100, Information Security Handbook: A Guide for Managers, 2007.
- NIST Special Publication 800-131A, Revision 1.
- NIST SP800-53A, Assessing Security and Privacy Controls in Federal Information Systems and Organizations Building Effective Assessment Plans, 2014.
- SSA Office of Systems Security Operations Management Guidelines.
- Payment Card Industry Data Security Standard (PCI DSS).
- New Mexico Administrative Code (NMAC) 1.12.20, Information Security Operation Management.
- Medicaid Information Technology Architecture version 3 (MITA 3.0).
- All standards identified in the Statement of Work (SOW) Subsection 2.1.5, Appendix G, Appendix H, and requirement 5.26 of the RFP.
- CMS Seven Conditions and Standards as outlined in RFP Addendum 10.
- Industry standards, best practices, and experience for information exchange and interoperability.
- CMS and NM Department of IT (DoIT) IV&V standards.
- HHS 2020 Security and Privacy Standards as outlined in Request for Proposal (RFP) Addendum 14 and NIST Special Publications: http://csrc.nist.gov/publications/PubsSPs.html.
- Standards for all ESB connections for web service interoperability among all MMISR modules.
- Project Management Plan (PMP), Contractual, and Project Performance standards.
- Substance Abuse and Mental Health Services Administration (SAMHSA).
- National Human Services Interoperability Architecture (NHSIA).

- National Information Exchange Model (NIEM).
- Fast Healthcare Interoperability Resources (FHIR).

12 Requirements Traceability

This deliverable meets the following requirements:

Request for Proposal (RFP):

- Page 93. Subsection 2.1.2.5

From RFP – Appendix H:

- Page122. ID: 1.11
- Page 128. ID: 1.21
- Page 165. ID: 2.06
- Page 191. ID: 2.42
- Page 199. ID: 3.01-3.07
- Page 221. ID: 5.03
- Page 124. Subsection 2.3

Proposal:

- Page 74. Subsection 1.3.6

Statement of Work (SOW):

- SIPLT1

13 Appendices

Appendix A. Record of Changes

The deliverable includes a record of changes as outlined in below.

Table 13-1: Record of Changes

Version Number	Date	Author/Owner	Description of Change
v0.1	1/8/18	Satya Govindu	Draft Deliverable
v0.2	1/22/18	Satya Govindu	Updated Diagrams and content
V0.3	02/19/18	Satya Govindu	Final submitted
V1.0	05/22/18	Satya Govindu	Updated the document to address review comments
V1.1	06/14/19	Satya Govindu	Updated the document to address review comments

Appendix B: Acronyms

The deliverable will include a List of Acronyms in the following form. For a comprehensive, project-wide list of acronyms, consult the Master Acronyms list on the SI Contractor team SharePoint site at Shared Resources on SharePoint.

The table below lists all acronyms used in this document.

Table 13-2: List of Acronyms

Acronyms	Definition
AAT	Agency Acceptance Test
AC	Access Control
ACL	Access Control Lists
AD	Active Directory
ADA	Americans with Disabilities Act
ADFS	Active Directory Federation Service
AES	American Encryption Standard
ALTYSD	Aging and Long-Term Services Department
AMP	Advanced Management Pod
ANSI	American National Standards Institute

API	Application Programming Interface
APM	Application Performance Management
ARB	Architecture Review Board
ARB	Architectural Review Board
ASC	Accredited Standards Committee
ASPEN	Automated System Program and Eligibility Network
ASRs	Architecturally Significant Requirements
AWS	Amazon Web Services
B2B	Business to Business
BA	Business Analyst
BAM	Business Activity Monitoring
BHSD	Behavioral Health Enforcement Division
BMS	Benefits Management Services
BPEL	Business Process Execution Language
BPM	Business Process Management
BRE	Business Rules Engine
CA	Certificate Authority
CAS	Common Asset Service
CCB	Change Control Board
CCIS	Configuration and Continuous Integration Service
CDC	Change Data Capture
CDC	Change Data Capture
CDM	Conceptual Data Model
CDM	Common Data Model
CFF	Custom Fault Handling Framework
CIA	Confidentiality, Integrity, and Availability
CMS	Centers for Medicare and Medicaid Services
CMS	Center for Medical and Medicaid Services
CMSO	Center for Medicaid and State Operations
ConOps	Concept of Operations
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CRUD	Create, Read, Update, and Delete
CSES	Child Support Enforcement System
CSF	Critical Success Factors
CSF	Critical Success Factor
CSV	Comma Separated Values
DAM	Data Access Module
DB	Database
DDR	Detailed Design Review
DED	Deliverable Expectation Document

DEK	Database Encryption Key
DIM	Data Ingest Module
DMS	Dynamic Monitoring Service
DMZ	Demilitarized Zone
DOH	Department of Health
DoIT	Department of Information Technology
DPI	Deep Packet Inspection
DRAMS	Drug Rebate Analysis and Management System
DRS	Distributed Resource Scheduling
DS	Data Services
DVS	Distributed Virtual Switch
EA	Enterprise Architecture
EAI	Enterprise Application Integration
EAVS	Enterprise Address Verification Service
ECFR	Electronic Code of Federal Regulations
ECM	Enterprise Communication Management
ECS	Endpoint Certificate Store
EDAS	Enterprise Data as a Service
EDI	Electronic Data Interchange
EDIFACT	Electronic Data Interchange for Administration, Commerce, and Transport
EDM	Enterprise Data Model
EDM	Enterprise Document Management
EIAS	Enterprise Identity Access Service
EIP	Enterprise Integration Pattern
ELT	Extract Load Transfer
EMCC	Enterprise management cloud controller
EML	Extensible Markup Language
EPSDT	Early and Periodic Screening, Diagnostic and Treatment
ERD	Entity Relationship Diagram
ESB	Enterprise Service Bus
ESS	Enterprise Shared Services
ETL	Extract, Transform, Load
ETL	Extract Transfer Load
EVC	Enhanced vMotion Compatibility
FAN	Fast Application Notification
FCF	Fast Connection Failover
FCR	Federal Case Registry
FedRAMP	Federal Risk and Authorization Program
FHIM	Federal Health Information Model
FHIM	Federal Health Information Model
FIPS	Federal Information Processing Standards

FISMA	Federal Information Security Management Act
FPLS	Federal Parent Locator Service
FS	Financial Services
FTI	Federal Tax Information
FTP	File Transfer Protocol
GAC	Global Address Cleanse
GB	Gigabyte
HA	High Availability
HCI	Hyper-Converged Infrastructure
HHS	Health and Human Services
HIPAA	Health Insurance Portability and Accountability Act
HIPAA	Health Insurance Portability and Accountability
HITECH	Health Information Technology for Economic and Clinical Health
HL7	Health Level Seven International
HL7	Health Level 7
HSD	Human Services Department
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
I&C	Installation and Configuration
IdAM	Identity and Access Management
IdM	Identity Management
IdP	Identity Provider
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IIF	Individually Identifiable Information
IIF	Individually Identifiable Information
IKE	Internet Key Exchange
IP	Integration Platform
IP	Internet Protocol
IRC	Internal Revenue Code
IRS	Internal Revenue Service
ISO	International Organization for Standards
ITD	Information Technology Division
ITIL	Information Technology Infrastructure Library
IV&V	Independent Verification and Validation
IVR	Interactive Voice Response
IWS	Oracle Integrated Workload Statistics
JDBC	Java Database Connectivity
JMS	Java Messaging Service
JMX	Java Management Extensions
JSON	JavaScript Object Notation

JVM	Java Virtual Machine
KMIP	Key Management Interoperability Protocol
KMS	Key Management Service
LDAP	Lightweight Directory Access Protocol
LDM	Logical Data Model
MAC	MDM API Core
MAD	Medical Assistance Division
MARS-E	Minimum Acceptable Risk Standards for Exchanges
MCO	Managed Care Organizations
MDM	Master Data Management
MECL	Medicaid Enterprise Certification Life Cycle
MECL	Medicaid Enterprise Certification Lifecycle
MECT	Medicaid Enterprise Certification Toolkit
MFA	Multi Factor Authentication
MFT	Oracle Managed File Transfer
MITA	Medicaid Information Technology Architecture
MITA TMS	Medicaid Information Technology Architecture Technical Management Strategy
MITM	Man-In-the-Middle-Attack
MLCP	MarkLogic Content Pump
MMIS	Medicaid Management Information System
MMISR	Medicaid Management Information System Replacement
MOM	Message-Oriented Middleware
MS SQL	Microsoft SQL
NARA	National Archives and Records Administration
NDNH	National Directory of New Hires
NHSIA	National Human Services Interoperability Architecture
NIEM	National Information Exchange Model
NIST	National Institute of Science and Technology
NIST	National Institute of Standards and Technology
NM	New Mexico
NMAC	New Mexico Administrative Code
NoSQL	Not Only Structured Query Language
NoSQL	Not Only Structured Query Language
OAAM	Oracle Adaptive Access Manager
OAM	Oracle Access Manager
ODI	Oracle Data Integrator
ODS	Operational Data Store
OFMW	Oracle Fusion Middleware
OGC	Oracle Government Cloud
OHS	Oracle HTTP Server
OIF	Oracle Identity Federation

OIM	Oracle Identity Manager
OS	Operating System
OSB	Oracle Service Bus
OUD	Oracle Unified Directory Server
OWSM	Oracle Web Services Manager
PA	Prior Authorization
PADU	Preferred-Acceptable-Discouraged-Unacceptable
PBM	Pharmacy Benefits Management
PCI	Payment Card Industry
PCI DSS	Payment Card Industry Data Security Standard
PDB	Pluggable Database
PDF	Portable Document Format
PDM	Physical Data Models
PDR	Preliminary Design Review
PDR	Preliminary Design Review
PGP	Pretty Good Privacy
PHI	Personal Health Information
PHI	Personal Health Information
PII	Personally Identifiable Information
PII	Personally, Identifiable Information
PKCS	Public Key Cryptography Standards
PMO	Project Management Office
PMO	Project Management Office
PMP	Project Management Plan
PSC	Platform Service Controller
QA	Quality Assurance
QA	Quality Assurance Services
QAT	Quality Assurance Test
QMP	Quality Management Plan
RA	Reference Architecture
RACI	Responsible, Accountable, Consulted, and Informed
RCM	Release Certification Matrix
RDL	Raw Data Lake
REST	Representational State Transfer
RFP	Request for Proposal
RFPs	Request for Proposal
RHEL	Red Hat Enterprise Linux
RMP	Requirements Management Plan
RTM	Requirements Traceability Matrix
SAML	Security Assertion Markup Language
SC	System Communication

SCAN	Single Client Access Name
SDD	System Design Document
SDLC	Software Development Life Cycle
SDP	System Design Plan
SDS	Standardized Data Store
SFTP	Secure File Transfer Protocol
SI	System Integrity
SI	System Integration or System Integrator
SIEM	Security Information and Event Management
SIM	Source-specific Integration Module
SIM	Source-specific Integration Module (SIM)
SIT	System Integration Test
SLA	Service-Level Agreements
SMA	State Medicaid Application
SME	Subject Matter Expert
SMEs	Subject Matter Experts
SMR	System Migration Repository
SMR	System Migration Repository
SMTP	Simple Mail Transfer Protocol
SOA	Service-Oriented Architecture
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOCCI	Service-Oriented Cloud Computing Infrastructure
SOW	Statement of Work
SQL	Structured Query Language
SSA	Social Security Administration
SSH	Secure Shell
SSL	Secure Sockets Layer
SSO	Single Sign-on
SSP	System Security Plan
TARC	Technical Architecture Review Committee
TBD	To Be Discussed
T-COP	Traffic Cop
TDE	Transparent Data Encryption
TDE	Template Driven Extraction
TLS	Transport Layer Security
TMP	Test Management Plan
TOR	Top of Rack
TPA	Third-party administrator
TPL	Third Party Liability
TRA	Technical Reference Architecture

UAT	User Acceptance Test
URI	Uniform Resource Identifiers
USGCB	United States Government Configuration Baseline Standards
VM	Virtual Machines
VPC	Virtual Port Channel
VPN	Virtual Private Network
VSAN	Virtual Storage Area Network
VUM	vSphere Update Manager
WS	Web Service
WSDL	Web Services Description Language
WSRP	Web Services for Remote Portlets
XML	eXensible Markup Language
Xquery	eXensible Markup Language Query
ZIP	Zone Improvement Plan

Appendix C: Glossary

A glossary of project-specific terminology is maintained on the SI Contractor team SharePoint site at Shared Resources on SharePoint.

Appendix D: Referenced Documents

This section provides a list of referenced artifacts, standards, and tools.

Table 13-3: Referenced artifacts, Standards, and Tools

Document Name	Document Location and/or URL	Issuance Date
SAP Data services – Address Cleansing	https://wiki.scn.sap.com/wiki/display/EIM/Address+Cleansing	
SAP Real-Time Address Validation	https://wiki.scn.sap.com/wiki/display/EIM/Real-Time+Address+Validation	
REST API – MarkLogic Smart Mastering	https://www.marklogic.com/product/marklogic-database-overview/database-features/smart-mastering/	
MarkLogic Range Indexes	https://docs.marklogic.com/guide/admin/range_index	
MarkLogic Scalability, Availability and Failover	https://docs.marklogic.com/guide/cluster/scalability	

Document Name	Document Location and/or URL	Issuance Date
Oracle Fusion Middleware Voluntary Product Accessibility Template (VPAT)	http://www.oracle.com/us/corporate/accessibility/vpats-162843.html	
Process to Enable Accessibility Mode for Oracle Fusion Middleware Administrative Consoles	https://docs.oracle.com/middleware/1221/core/ASADM/accessible.htm#ASADM161	
New Mexico Administrative Code	http://164.64.110.134/nmac/home	
Electronic Code of Federal Regulations	https://www.ecfr.gov/cgi-bin/text-idx?SID=2b00792abddeb5364ddcac13934667c0&mc=true&tpl=/ecfrbrowse/Title36/36cfrv3_02.tpl#1200	

Appendix E: MECT Checklist

The deliverable contains the Medicaid Enterprise Certification Toolkit (MECT) Checklist items that correspond to this deliverable as listed in the table below.

Table 13-4: MECT Checklist

Checklist ID	Requirement Text / System Review Criteria (SRC)	MITA Business Area Module Checklist Set	Business Process	Module Owner	CMS Guidance	Location
TA.FR.1	The system of interest supports retrieval and presentation of data associated with geographic indicators such as State, county, and ZIP code.	Access and Delivery	Technical Service Classification: Forms and Reporting	All	For R1, evidence could include acquisition documents, requirements specifications, a Concept of Operations (ConOps) that explains how this functionality will be implemented, or other planning documents that demonstrate plans to incorporate this capability. For R2 and R3, evidence should include screenshots showing retrieval of data through searches against State, county, and ZIP code. For R2 (if	

Checklist ID	Requirement Text / System Review Criteria (SRC)	MITA Business Area Module Checklist Set	Business Process	Module Owner	CMS Guidance	Location
					not a desk review) and R3, the State should be prepared to demonstrate this functionality. Enterprise: The State ensures that data can be associated with geographic indicators across all relevant modules. Module: This applies to modules that edit, store, retrieve, present, and/or report data that have geographic indicators associated with them.	

Appendix F: Critical Success Factors

The table below provides the Programmatic Critical Success Factors:

Table 13-5: Critical Success Factors

Checklist ID	Requirement Text/ System Review Criteria (SRC)	MITA Business Area Module Checklist Set	Module Owner	Business Process	CMS Guidance	Location
N/A	N/A	N/A	N/A	N/A	N/A	N/A

Appendix G: List of Software, Tools and Libraries

The following are the list of necessary software, tools and libraries to build SI Platform components. This list is subject to change based on current, ongoing, and future needs. For further details please refer to the IP Resource sheet stored in SI Configuration Documents on the NM HSD SharePoint site.

Infrastructure Foundation Software and Tools:

The following table details the foundation software and tools that are required building, configuring and maintain the infrastructure.

Table 13-6: Infrastructure Foundation Software Tools

Category	Name	Installation
EMC Software	• Vision for VxRack FLEX. • VxRack Manager – Management Control (AMP).	Management Software for DELL-EMC Hardware installed on the management server within the VxRack
VMware Virtual Infrastructure	• ESXi. • vSphere/vCenter. • vSAN. • vRealize NSX. • vRealize Operations. • vRealize Automation. • vRealize Orchestration.	These software/appliances will be installed on the DELL-EMC VxRack virtual infrastructure
Operating System	• RedHat Enterprise Linux (RHEL).	RHEL OS will be installed on the VMs of IP as needed
Operating System	• Windows Server 2016/2012.	Windows will be installed on the VM's of IP as needed
Development Kit	• Java. • Java Development Kit (JDK). • Java Runtime Environment (JRE).	Java will be installed on the VMs as needed.

Enterprise Service Bus (ESB):

The following major software components are required to install and configure the ESB.

Table 13-7: Enterprise Service Bus

No	Software Required for ESB	Function/Purpose
1.	Oracle Real Application Cluster (RAC) Database	Oracle RAC DB will be installed on individual VMs, which will be used by the ESB and IdAM to store and process custom and runtime data.
2.	OFM – Service Oriented Architecture (SOA) Suite	SOA Suite will be used to build, monitor, test, and report Simple Object Access Protocol (SOAP) and Representational State Transfer (REST)-based web services on the ESB platform using Oracle Service Bus (OSB), Business Process Execution Language for Web Services (BPEL), Oracle Business Activity Monitoring (BAM) and Oracle Web Services Manager (OWSM) components. These Web services will be built using the industry standard SOA, Message Oriented Middleware (MOM), and web service patterns.
3.	Oracle Data Integrator (ODI)	ODI will be used by the ESB, the Systems Migration Repository (SMR) and other interfaces to process bulk batch file data, which can be in different file formats. It will act as a translator for all the bulk data coming from all new modules and state agencies and will transform it into different file and database formats.
4.	Oracle Managed File Transfer (MFT)	MFT will be used to manage, monitor, report, and transfer batch file transfers from source to targets. The Interfaces will mostly use this work stream to transfer batch files in different file transfer modes such as File Transfer Protocol (FTP), Secure FTP (SFTP, FTPS), etc.
5.	Oracle B2B	Oracle B2B will be used to manage, monitor, report, and process all EDI B2B transactions with the providers. This tool supports major health care EDI transactions like 834, 820, 270, 271, etc.
6.	Oracle Service Bus (OSB)	OSB will be used as a proxy and virtualization layer for all web services deployed on the ESB platform. The OSB will also provide protocol translation, native to canonical xsd translation, security, and message transformations.
7.	Oracle Web Services Manager (OWSM)	This will be used to manage and secure all authentication and authorization policies for all SOA Web services built on the ESB platform. This tool includes many default client and server-side policies to protect Web services.
8.	Oracle Business Activity Monitoring (BAM)	BAM will be used to manage, monitor, and report all real time business transactions. This tool will work in conjunction with BPEL, OSB, and OWSM to provide real time updates of a particular transaction in its lifecycle.

No	Software Required for ESB	Function/Purpose
9.	Business Process Execution Language for Web Services (BPEL)	BPEL will be the business layer of all the SOA Web services on the ESB platform. This layer will do most of the business processing on the ESB platform. The message, once it is authenticated, validated, and translated, will be passed by the OSB layer to the BPEL layer to orchestrate and apply all the business logic.
10.	Oracle Enterprise Manager Cloud Control (EMCC)	EMCC will be used to manage, monitor, troubleshoot, and report all the transactions processed by the BPEL, OSB, OWSM, and BAM layers. It provides a console to do all of these activities and take necessary actions.
11.	Oracle Fusion Middleware (OFM) Infrastructure	WebLogic installed on the SOA servers
12.	Oracle Application Programming Interface (API) Manager	Oracle API Manager facilitates the creation of APIs that expose the functionality of backend systems and services. These APIs are published for use by application developers and are managed and monitored at runtime. Oracle API Manager provides the following capabilities: • Allows users to easily create APIs. • Provides the ability to secure APIs. • Enables easy API editing and publishing. • Facilitates the discovery and use of APIs. • Controls the access to APIs at runtime.
13	Oracle Business Rule Engine	Oracle Business Rules is a high-performance lightweight business rules product that addresses the requirements for agility, business control, and transparency. It is part of the Fusion Middleware stack and integrates seamlessly across the entire Oracle SOA Suite and BPM Suite stack.

Identity and Access Management (IdAM)

The following major software components are required to install and configure IdAM. For further details please refer the IP Resource sheet stored in SI Configuration Documents in NM HSD SharePoint site.

Table 13-8: Identity and Access Management

No	Software Required for IdAM	Purpose
1	Oracle Real Application	Oracle RAC DB will be installed on individual VMs, which will be used by the ESB and IdAM to store and process custom and runtime data.

No	Software Required for IdAM	Purpose
	Cluster (RAC) Database	
2	Oracle Identity Management (OIM)	OIM provides interfaces for system administrators, self-service, and design (configuration) consoles. New unified portal user registration and provisioning is performed through the identity module in OIM.
3	Oracle Access Management (OAM)	OAM provides authentication of users and authorization of user access to various applications participating in the IdAM framework. OAM is responsible for authentication and authorization of individuals accessing the unified portal, providing federated Single Sign-On (SSO) access to employees accessing the portal and other shared resources, as well as individual user authentication against social media providers and maintaining SSO access tokens.
4	Oracle Unified Directory (OUD)	OUD is the Lightweight Directory Access Protocol (LDAP) product used to store individual portal user data, which is used for authentication, authorization, and identity management.
5	Oracle HTTP Server (OHS)	OHS is the entry point for unified portal users. It provides web-based access to the portal system and authenticates user access to the system. Oracle Access Manager Agent (Web Gate) will be deployed on OHS servers and act as a policy enforcement agent. Web Gate intercepts all the incoming requests to the unified portal and checks whether the request requires authentication. Web Gate is integrated with Oracle Access Manager.
6	Oracle Application Programming Interface (API) Manager	Oracle API Manager facilitates the creation of APIs that expose the functionality of backend systems and services. These APIs are published for use by application developers and are managed and monitored at runtime. Oracle API Manager provides the following capabilities: • Allows users to easily create APIs. • Provides the ability to secure APIs. • Enables easy API editing and publishing. • Facilitates the discovery and use of APIs. • Controls the access to APIs at runtime.

System Migration Repository (SMR):

The following major software components are required to install and configure the SMR.

Table 13-9: System Migration Repository

No	Software Required for SMR	Purpose
1	MarkLogic	MarkLogic NoSQL database.
2	MarkLogic Content Pump (MLCP)	MLCP will provide the fastest way to import, export, and copy data to or from MarkLogic databases.
3	Content-Reprocessing in Bulk (CORB)	CORB is a Java tool designed for bulk content-reprocessing of documents stored in MarkLogic.
4	Gradle	Gradle will be used to build the MarkLogic code.
5	MarkLogic Ops Director	This is MarkLogic's monitoring tool. MarkLogic Ops Director provides a clear view to see and manage your entire MarkLogic infrastructure – whether it's across multiple clusters, cloud and on-premises systems, or production, test and development environments.
6	· redhat-lsb. · glibc. · gdb. · cyrus-sasl-lib. · glibc.i686 (required for PDF/MS-Office extraction pipeline).	RHEL OS Packages for MarkLogic.
7	Access Server	This server will host custom programs to extract contents of the legacy system databases to enable ingestion into the SMR.
8	Ingest Server	This server will host custom programs based on MLCP, Data movement SDK to feed legacy system database records into the SMR.

Master Data Management (MDM):

The following major software components are required to install and configure the MDM.

Table 13-10: Master Data Management

No	Software Required for MDM	Purpose
1	MarkLogic	MarkLogic No SQL database.
2	MarkLogic's Smart Mastering Toolkit	Smart Mastering makes it possible to intelligently harmonize data and automate away time consuming and brittle parts of data integration.
3	Content-Reprocessing in Bulk (CORB)	CORB is a Java tool designed for bulk content-reprocessing of documents stored in MarkLogic.
4	Gradle	Gradle will be used to build the MarkLogic code.
5	• redhat-lsb. • glibc. • gdb. • cyrus-sasl-lib. • glibc.i686 (required for PDF/MS-Office extraction pipeline).	RHEL OS Packages for MarkLogic.

Electronic Document Management (EDM)

The following major software component required to install and configure the EDM.

Table 13-11: Electronic Document Management

No	Software Required for EDM	Purpose
1.	ImageNow/Perceptive Content	Perceptive Content is a scalable content services platform that manages the entire content lifecycle, from capture to disposition. Flexible functionality across multiple business applications, integration with virtually any business application and a simple-to-use interface help Perceptive Content transform internal processes and the customer experience.

Customer Communication Management (CCM):

The following major software component required to install and configure the CCM.

Table 13-12: Customer Communication Management

No	Software Required for CCM	Purpose
1	OpenText ExStream	OpenText Exstream is a multichannel customer communication management (CCM) solution that empowers you to make the most of every customer touch point by creating insightful, impactful, real-time customer communications.

Address Standardization:

The following major software component required to install and configure the CCM.

Table 13-13: Address Standardization

No	Software Required for Address Standardization	Purpose
1	SAP Data Services	SAP Data services is an ETL tool which gives a single enterprises level solution for data integration, Transformation, Data quality, Data profiling and text data processing from the heterogeneous source into a target database or data warehouse

Miscellaneous:

The following are the miscellaneous software required for installing, configuring and Managing SI Platform components.

Table 13-14: Miscellaneous

No.	Miscellaneous Infra Components	Function/Purpose	Product
1	Secure File Transfer Protocol (SFTP)	SFTP that support send and receive files from and to Data sources. FTP Servers serving the file transfer for external trading partners.	ProFTPD FTP server will be used. Two SFTP servers are required for Production and one SFTP server is required for Non-Prod environment.
2	Log Analytics	Capture logs of ingress and egress traffic between various segments.	Splunk and EMCC will be used for Log analytics.

No.	Miscellaneous Infra Components	Function/Purpose	Product
3	Application Performance Monitoring (APM)	APM provides development and operations teams with the information that they need to find and fix application issues.	EMCC To be determined
4	Firewall	Virtual Appliance - Network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules.	To be determined (VMware NSX, Palo Alto
5	Repository	Allow teams in one place to plan projects, collaborate on code, test, and deploy.	Atlassian BitBucket
6	Build Server	It is a Continuous Integration and Continuous Deployment tool (CI/CD).	Atlassian Bamboo/Jenkins
7	Load Balancer	Virtual Appliance that acts as a reverse proxy and distributes network or application traffic across a number of servers.	For On-perm- BIG IP - F5 For OGC- To be determined

Appendix H: VMware NSX – Reference & Future Implementation

NSX Hardware Architecture

For the SI Platform, NSX for VMware vSphere provides virtualized networking components (including the Distributed Firewall and the Load Balancer) that will be configured to implement software-defined networking. One NSX Manager will be installed to support the functions. VMware uses NSX APIs to configure a VXLAN with the IP parameters specified for the Virtual Machine, and the hardware management service (HMS) to configure the ToR ports associated with the servers. Data center administrators will use the vSphere Web Client to perform additional NSX configuration required by the specific VMs deployed within the VxRack Flex 1000.

Figure 13-1: VxRack AMP

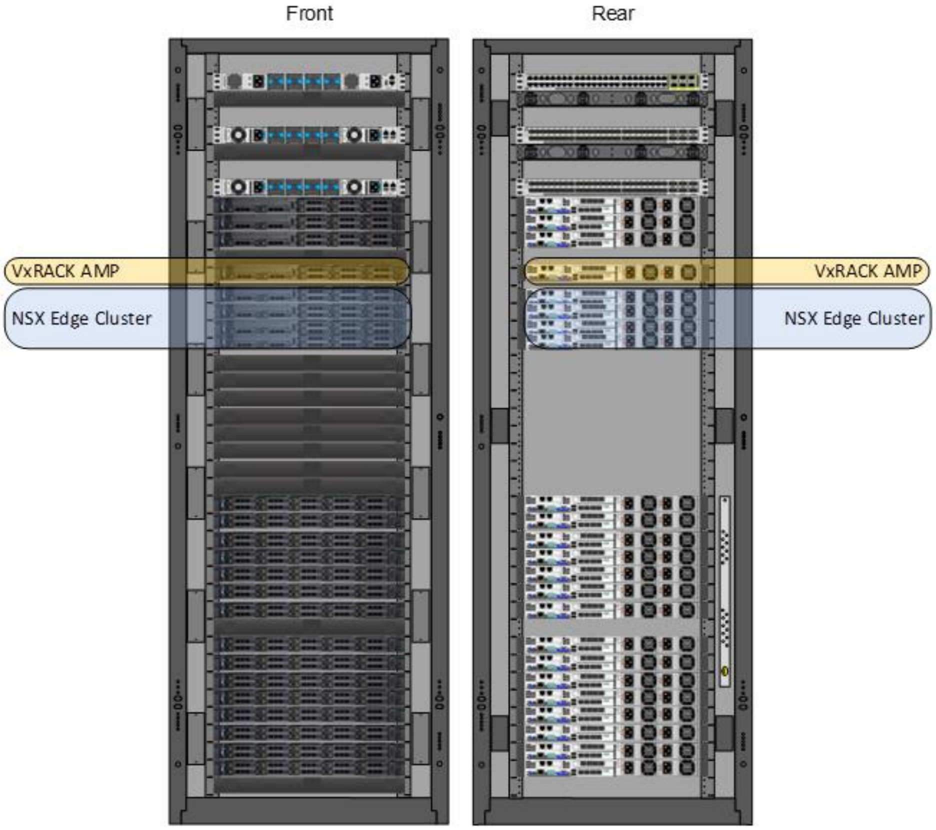
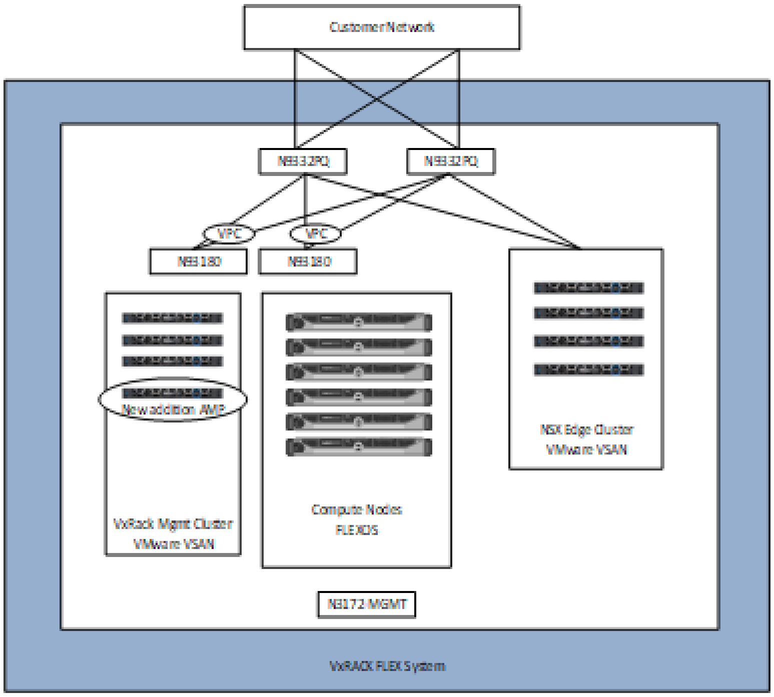


Table 13-15: NSX Hardware

Compute	Storage
(4) Physical VxRACK FLEX Management Servers to create a separated VSAN Cluster to house NSX Edge Servers (4 x NSX Edge Virtual Edition & (1) NSX Controller VM).	VMware VSAN
Management	**Network**
(4) Physical VxRACK FLEX Management Server to provide addition resources for NSX Management components (NSX Manager). This solution will leverage existing VxRACK FLEX AMP vCenter for both AMP VSAN Cluster and NSX Edge VSAN Cluster.	(2) Layer 3 Advance Licenses for Cisco Nexus 9332PQ Switches

Figure 13-2: NSX Edge Nodes and Custer Construct



The table below lists each the management VMs that are needed for NSX.

Table 13-16: NSX Management cluster VMs

VM Description	CPU (vCPUs)	Memory (GB)	Storage (GB)	Network bandwidth	High availability
NSX Manager (1) Management Cluster	4	12	60	1 Gbe	vSphere HA
NSX Controller Management Cluster	4	4	20	1 Gbe	Built-in/vSphere re HA
NSX Manager (2) Edge and Compute Cluster	4	12	60	1 Gbe	vSphere HA

Table 13-17: NSX SW Licensing

NSX SW Licensing	License QTY
VMware vSphere 6.0 Enterprise Plus Edition (per CPU socket) – AMP	1
VMware vSphere 6.0 Enterprise Plus Edition (per CPU socket) – NSX Edge VSAN Cluster	8
VMware VSAN 6.0 Enterprise Edition (per CPU Socket) AMP	1
VMware VSAN 6.0 Enterprise Edition (per CPU Socket) – NSX Edge VSAN Cluster	8
VMware NSX Enterprise Edition (per CPU Sockets)	TBD

VMware NSX Firewall Architecture

VMware NSX plays a major role in network virtualization and micro-segmentation for the SI Platform architecture for the MMISR Project. VMware NSX add-on will provide software-defined network what it has already delivered for compute and storage. VMware NSX network virtualization programmatically creates, snapshots, deletes, and restores software-based virtual networks. The SI Platform for MMISR Project can build multi-tier application networks and implement micro-segmentation to mitigate against threats that penetrate through the perimeter firewall. It will be deployed on an IP based network, including both types of existing traditional networking models and next-generation fabric architectures from any vendor, NSX is totally non-disruptive process of provisioning. NSX will be deployed over the physical network infrastructure that already exists at the existing data center.

Network virtualization is a functional equivalent of a “network hypervisor” that reproduces the complete set of layers 2 to layer 7 networking services software such as switching, routing, firewalling, and load balancing.

As a result, these services will be programmatically assembled in any arbitrary combination, to produce unique, isolated virtual networks in a matter of seconds.

Virtual networks do not depend on the underlying SI Platform network hardware and treat the physical network layer as a pool of transport capacity that can be repurposed on demand. Unlike legacy architectures, virtual networks can be changed, deleted, restored, and provisioned programmatically without disturbing or reconfiguring the underlying physical hardware.

NSX Components

The following table provides NSX components that will be implemented for the SI Platform.

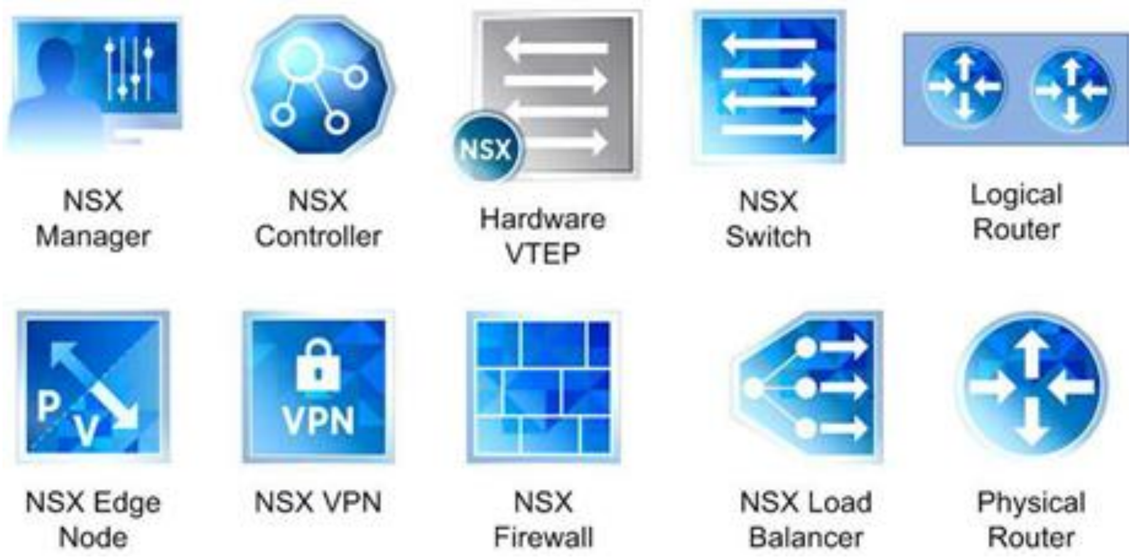
Table 13-18: NSX Components

Component	Description	Logical Network	Default
NSX Manager	Provides the single point in which the entire SDN solution is deployed. From this single appliance, the administrator can configure and deploy multiple services for the overlay network.	Management Plane	1
NSX Controller	Contains slices of information about the overlay network, such as the logical switches, VXLANs, and Virtual Machines. NSX controllers are deployed in odd numbers with a minimum of three.	Control Plane	3
Hardware VTEP	VMkernel interface that is created by the NSX manager during the initial preparation of the ESXi Host to participate in the overlay network.	Data Plane	1
Edge Services Gateway	The Edge Services Gateway gives access to all NSX Edge services, such as firewall, NAT, DHCP, VPN, load balancing, and high availability. Each Edge Services Gateway can be configured for single or multiple services and have a total of 10 uplink and internal network interfaces. The internal interfaces connect to secured port groups and act as the gateway for all protected virtual machines in the port group.	Data Plane	0
Physical Router	A physical router that is logically connected to each ESXi host in the data center.	Data Plane	1

Component	Description	Logical Network	Default
Logical (Distributed) Router	This service is a special edge service that handles east-west traffic to reduce the amount of traffic received by the Edge Gateways. This service is also called a logical distributed router (LDR).	Data Plane	0

The figure below shows the standard set of icons that are defined by VMware to represent the various NSX

Figure 13-3: NSX Standardized Icons

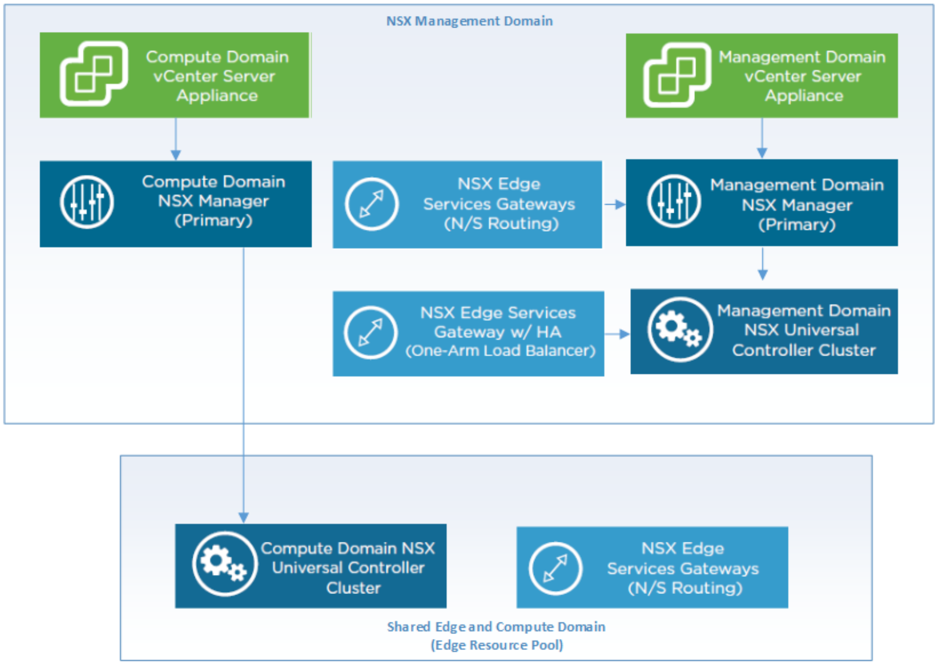


NSX Functional Components

The logical network provided by NSX uses two types of access layers as described below.

- The Hypervisor Access Layer: This represents the point of attachment to the logical networks for Virtual systems.
- The Gateway Access Layer: This provides the L2 and L3 connectivity to the logical space of physical devices and the virtual systems provisioned in the physical network infrastructure.

Figure 13-4: NSX Logical Network


NSX Management Domain
Compute Domain vCenter Server Appliance
Management Domain vCenter Server Appliance
Compute Domain NSX Manager (Primary)
NSX Edge Services Gateways (N/S Routing)
Management Domain NSX Manager (Primary)
NSX Edge Services Gateway w/ HA (One-Arm Load Balancer)
Management Domain NSX Universal Controller Cluster
Compute Domain NSX Universal Controller Cluster
NSX Edge Services Gateways (N/S Routing)
Shared Edge and Compute Domain (Edge Resource Pool)

For the SI Platform, NSX is being deployed to bridge the physical and software-defined network infrastructure and provides a considerable degree of flexibility, scalability, and agility through logical networks, along with the ability to create networks. NSX edge services will be leveraged to deploy and implement networks services such as load balancer, Firewall rules, and micro-segmentation on the existing or new networks. Primarily NSX will be deployed in a logical layer and can create networks which can also be delivered by deploying the physical layer where switching, routing, firewall, and load balancing is deployed.

NSX Edge Services Gateway

The below NSX components are being implemented for the SI Platform.

- Routing and Network Address Translation (NAT): The NSX Edge provides centralized on-ramp/off-ramp routing between the logical networks deployed in the NSX domain and the external physical network infrastructure. The NSX Edge supports various routing protocols (OSPF, iBGP, and eBGP) and can also communicate leveraging static routing. NAT can be performed for traffic flowing through the Edge and both source and destination NAT is supported.

- Firewall: NSX Edge is a capable stateful firewall which complements the Distributed Firewall (DFW) working in congestion with the kernel of the ESXi hosts. DFW is used to enforce the security policies with the workloads (east-west traffic), whereas the Edge firewall services are deployed to handle the North-South bound traffic between the Virtual systems deployed in the clusters to the external network.

- Load Balancing: the NSX Edge can perform load-balancing services for server farms of workloads deployed in logical space. The load-balancing functionalities natively supported in the Edge cover most of the typical requirements found in real-life deployments.

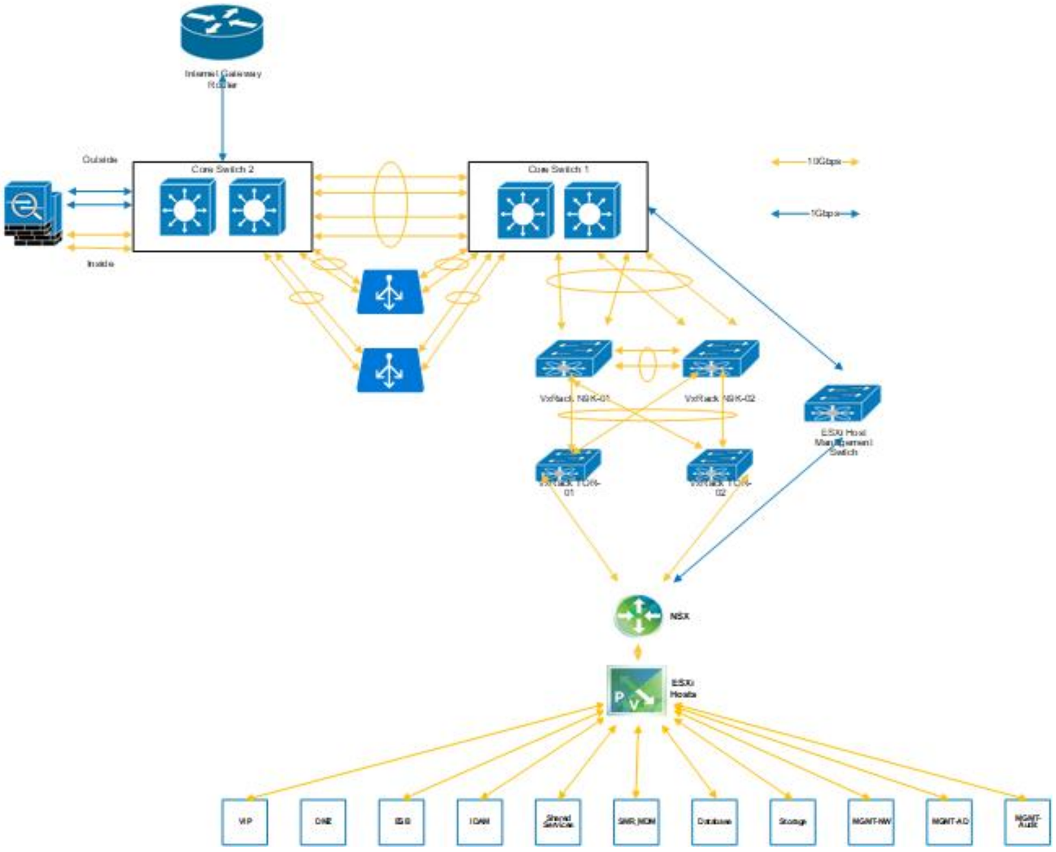
Micro-Segmentation

For the SI Platform micro-segmentation, the NSX DFW is deployed and configured to provide L2-L4 stateful firewall services to any workload in the environment. DFW is configured in the kernel space and performs real-time network traffic protection. DFW add-on is deployed onto the host as part of the NSX deployment as required. Every vNICs instance on a given VM is deployed with its own DFW instance. DFW policy (L2) is enforced before L3/L4 rules.

DFW is a VMware NSX component configured to protect network traffic within Workloads configured on the Hardware (Virtual-to-Virtual or Virtual-to Physical). For the SI Platform, DFW's main goal is to protect east-west traffic; that said, since DFW policy enforcement is configured and applied to the VMnics, it will also be used to block communication between the VMs and the external network. Edge Service gateway (ESG) is used to protect north-south traffic (Virtual-to-Physical) and as such is the first entry point to the software-defined network.

Micro-segmentation with NSX DFW will be leveraged to implement network segments for VIP/DMZ/ESB/IdAM/Shared Services/ SMR_MDM/Database/Storage/MGMT-NW/MGMT-AD/MGMT-AUDIT for an 11-tier application where multiple modules share the same logical network topology.

Figure 13-5: Zone NSX Firewall Layout



The SI Platform network design will have 11 virtual switches (DMZ/ESB/IdAM/Shared Services/ SMR_MDM/Database/Storage/MGMT-NW/MGMT-AD/MGMT-AUDIT) interconnected through a Distributed

Logical Router (DLR). IP addresses defined on DLR are the default gateway for VM connected to any of the Logical Switch. DLR is connected to the Edge Services Gateway (GW) that provides connectivity to the physical network.

The micro-segmentation provides flexible security posture using DFW for servers of the same function or role to be connected to the same virtual switch irrespective of the application and it is no more a barrier for security enforcement. DFW supports any type of topology with the same level of stringent traffic access control, and the Service Composer allows grouped VMs of the same role as objects in a logical container to manage the traffic flow easily.

The DFW default policy rule will be modified from Action = Allow to Action = Block. This is because we are going to use a positive security model, where only the allowed traffic flows are defined in the security rules table and all other communication is denied.

Table 13-19: Firewall Rule Table

Name	Source	Destination	Service	Action
Default rule	ANY	Any	Any	Block

The vCenter server, NSX Manager and NSX Controllers for the SI Platform will be in the DFW exclusion list in order not to lose connectivity and ensure that they are operational for the environment.

NSX Detailed Design

The NSX (Network Virtualization) will be used by NM HSD to manage the workload and the components of NSX onto the ESXI hosts where Network and Edge services are required.

NM HSD will use three primary functions of NSX that include load balancers, a distributed firewall, and VxLan, to be deployed at the VMKernel Level:

- The load balancers function will be used at the web server layer to create redundancy and to meet the SLA of the application/system.
- The distributed firewall will be used to achieve micro-segmentation of the networks and to keep the applications siloed in their network segment. The IP address/hashing NAT rules and protocol to allow or block any traffic to the virtual systems will also be used to allow east-west traffic, achieving a faster performance of network traffic.
- The VxLan will be used on a stretch network of systems across the datacenter to create logical switches.

In addition to internal deployment, NSX Edge services for north-south traffic will consist of the same set of components and include a load balancer, DFW, and VxLan where ESXI hosts will be deployed:

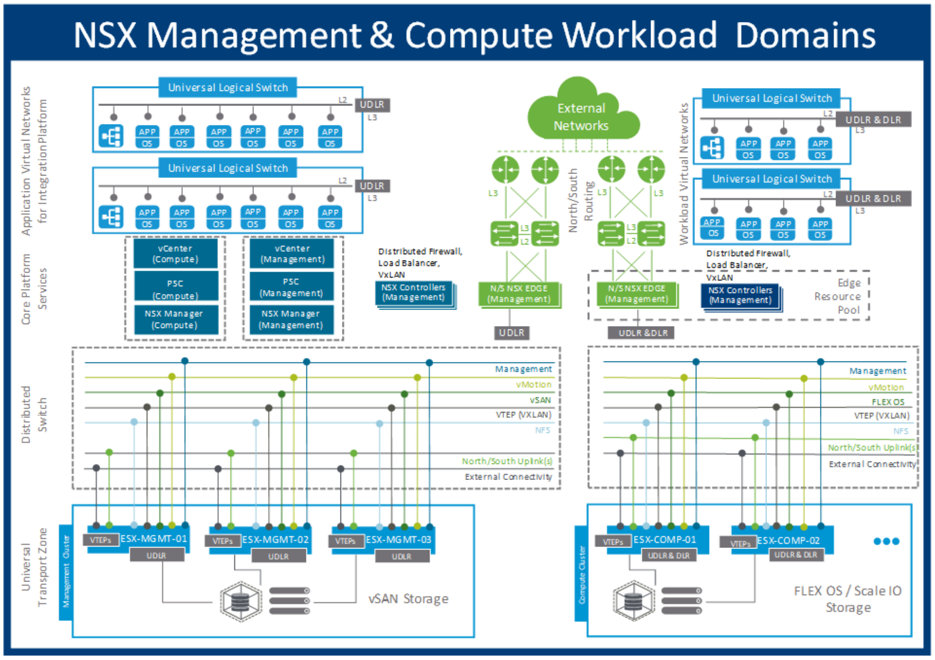
- The distributed firewall on the Edge Service containers will be exposed to the external traffic and the IP based NAT rules to allow specific protocol, as agreed upon for external addresses. To translate traffic to internal servers, hosting the application will also be used.

- The load balancers in the Edge Services and will be exposed to external traffic behind the enterprise firewall or the NSX distributed firewall on the Edge services.
- Load balancers will be used for the web servers, when the servers are redundant or have minimal to no downtime.

An overview of NSX deployment and functions are explained below and in the following figure:

- The NSX Manager is the component where an administrator can configure and control the configurations.
- The NSX Controller is the component that holds the information about the overlay network; NM HSD will deploy three controllers to follow this standard procedure.
- The VDS is the virtual distributed switch configured to handle east-west traffic.
- The Edge Service is the network configuration that contains the firewall, load balancing to the external network of the organization.

Figure 13-6: NSX Deployment Overview and Functions



Previous: Section 12.

NSX Security:

The core of the infrastructure security will be in the VMware NSX (future state). The NSX contains virtualized networking components including the Edge Services Gateway. The Edge services gateway provides firewalls, and load balancing to protect the perimeter of the SI Platform and ensure secured communications within and outside of the network.

Within the NSX, the Edge Services gateway has integrated virtual firewalls that will provide comprehensive security external to internal (North-South) Traffic. The firewalls will be configured so that traffic that does not match the firewall rules will be dropped. All ports on the firewalls not specifically required for operations will be closed by default.

The Virtual Distributed Switch will control the internal flow of information within the system and between interconnected systems East-West) in accordance with MMISR information flow policies. The logical switches will be connected to the logical router. The logical router will be configured to manage excess to minimize Denial of Service (DoS) attacks. The logical router will also be configured to enable neighbor router authentication to ensure that rogue routers under the command of a hacker cannot send incorrect routing updates. The NSX will utilize distributed switch functions as a single switch across all associated hosts. Each host is configured with each of the switches in the environment thereby bolstering inherent port security.

The NSX Load Balancer will provide traffic distribution in a seamless fashion. It will accomplish this by distributing incoming requests evenly among VMs and thus preventing system overload and potential DoS).

Appendix I: Architecturally Significant Requirements

The following table identifies the architecturally significant requirements applicable to this SDD.

Table 13-20: Architecturally Significant Requirements

High-level Requirements	Source
HHS 2020 Designs will be Standards-Based.	NM HSD 2020 Reference Architecture
Consumers will communicate with services via messages routed to the appropriate end points by the ESB and abstract the implementation details from other services.	NM HSD 2020 Reference Architecture
Enable the creation and evolution of composite applications through the use of Business Process Management (BPM), Workflow, Orchestration, and Business Rules Engine (BRE) tools.	NM HSD 2020 Reference Architecture

High-level Requirements	Source
Implement systemic Performance Management.	NM HSD 2020 Reference Architecture
Abstraction via Use of Policies - Policies hide implementation details and constraints of services from outside service clients. Details of a service's implementation are hidden completely from all service clients.	NM HSD 2020 Reference Architecture
Loose Coupling • Services are independent of one another and are stateless and idempotent where possible. • Service invocations should be bound at run-time, not at the compilation of software. • The existing interfaces change as little as possible and the services support backward-compatibility with existing APIs. • Services should be invoked and should reply to requests with DGC-approved shared schemas.	NM HSD 2020 Reference Architecture
HHS 2020 Will Continuously Improve Data Quality.	NM HSD 2020 Reference Architecture
Every architectural and design decision must be traceable to at least one corresponding business requirement, statement of system capability or detailed system requirement.	NM HSD 2020 Reference Architecture
All requests for functionality and data contained within the HHS 2020 Enterprise pass through the Security System.	NM HSD 2020 Reference Architecture
SOA Toolkit • Services will be autonomous and decoupled from other services. • Services will be discoverable through the IP. • Services will be composable. • Services will interoperate via sending/receiving asynchronous messages • Services and messages will be built to be reusable. • Services will be stateless and idempotent, as much as possible. • Services will communicate via asynchronous messages. • Services will hide their implementation details from other services.	NM HSD 2020 Reference Architecture

High-level Requirements	Source
Services will store metadata about their Contracts and Policies in a repository where the services are discoverable by other services at design, test and run-time.	ConOps
Services will be discoverable through the IP.	ConOps
ESB should support Composability that supports orchestrations of coarser-grained services from finer-grained services	ConOps
To externalize business logic controlling various long-running (process orchestrations), short running (service compositions) and system-level (ESB message validation and routing) functionalities HHS 2020 enterprise makes use of commercial BRE technology.	NM HSD 2020 Reference Architecture
Enforce versioning of services and messages and the proper retirement of outdated services	TP Statement of Work
User Interface layer depends on the Services layer for all data acquisition, transformation, and processing needs. Additionally, legacy and 3rd-party UI layer sub-systems in use post-HHS 2020 go-live will continue direct connections to their respective back-ends, thus necessitating the overall dependency between the UI and the Back-end layers.	NM HSD 2020 Reference Architecture
Services layer depends on the Back-end layer for the fulfillment of all data-centric and certain processing-centric requirements.	NM HSD 2020 Reference Architecture
Business Process Orchestration layer depends on the UI and the Services layers since the contents of both are used during process orchestration.	NM HSD 2020 Reference Architecture
Changes to Services would have to be evaluated against both the UI and the Orchestrations, while Back-end Layer would be unaffected.	NM HSD 2020 Reference Architecture
All service consumers that are aware of service interfaces will request the ESB to invoke the appropriate service end-points rather than having any direct end-point knowledge.	NM HSD 2020 Reference Architecture
Consumers will communicate with services via messages routed to the appropriate endpoints by the ESB.	NM HSD 2020 Reference Architecture

High-level Requirements	Source
The message will adhere to Shared/Canonical Schemas and validation for both schema and content as well as context-based routing will occur with assistance from a BRE enforcing appropriate sets of message validation and routing business rules.	NM HSD 2020 Reference Architecture
ESB will provide encoding (XML, JSON) and protocol (HTTP, JMS) translations to handle messages serving the client of varying technological capabilities and needs.	NM HSD 2020 Reference Architecture
ESB will enforce role-based authorization for service access and will carry out necessary logging of service interactions for auditing purposes.	NM HSD 2020 Reference Architecture
Shared services implement shared behavioral and non-behavioral functional requirements and are intended for broad re-use across business processes.	NM HSD 2020 Reference Architecture
Facilitate integration with access to services for data sharing between applications and entities, in accordance with service contracts and security policies	TP Statement of Work
HHS 2020 has mandated adherence to Web Services for Remote Portlets or WSRP 2.0 standard for its BPO partner systems	NM HSD 2020 Reference Architecture
The standard method for distributed transaction management is the use of Open XA-compliant data sources in service-implementing EAI orchestrations. The transaction boundaries are to be defined explicitly in BPML flows so that the Distributed Transaction Coordinator available as part of the EAI engine takes care of all commits and rollback logic based on the collective outcome of all functionality invocations within transaction’s scope.	NM HSD 2020 Reference Architecture
To externalize business logic controlling various long-running (process orchestrations), short running (service compositions) and system-level (ESB message validation and routing) functionalities HHS 2020 enterprise makes use of commercial BRE technology.	NM HSD 2020 Reference Architecture
HHS 2020 EA is governed by a combination of security control requirements found in MARS-E 2.0 and FIPS 140-2 standards intended to prevent unauthorized access to system data and functionalities.	NM HSD 2020 Reference Architecture

High-level Requirements	Source
HHS 2020 enterprise will expose multi-factor security functionality to 4 categories of users: • State employees using sanctioned devices to perform business functions. • BPO partner employees using their respective employer sanctioned. devices to access their solutions, which invoke HHS 2020 services. • External users, such as constituents and providers, accessing Unified Web Portal screens from devices of generally unknown origin. • Other systems accessing HHS 2020 services without involving any human user interactions.	NM HSD 2020 Reference Architecture
As a general standard, HSH 2020 Enterprise will favor multi-factor authentication over a single factor, even if multiple forms of a single factor are involved.	NM HSD 2020 Reference Architecture
State employees accessing HHS 2020 Enterprise from state-sanctioned personal computers over a private network (in office or VPN) will be authenticated via MS Active Directory server with the same user ID and password credentials as those used to login to state Windows Domain.	NM HSD 2020 Reference Architecture
BPO/Partner users will primarily access their employer-provided solutions on employer-provided/sanctioned host devices.	NM HSD 2020 Reference Architecture
Constituents, providers, partners, and any other external users of HSS 2020 web UI layer will be subject to a minimum 2-factor authentication.	NM HSD 2020 Reference Architecture
Leverage MMISR to build out HHS 2020 and populate with reusable services and COTS capabilities.	ConOps
Meet MITA guidelines, the Seven Conditions, and Standards, and enable achieving of MITA Maturity Level 4.	ConOps
Implement standardized interfaces and message schemas.	ConOps
Establish master data management for Client, Provider, Employer, and other data.	ConOps
Improve the quality of data systemically and at the point of capture wherever possible.	ConOps
Achieve wider data sharing by implementing consistent data models and shared message schemas, compliant with Information Architecture and SOA.	ConOps
Implement Business Analytics as a Service internally and move away from costly, brittle, traditional Data Warehouse environments.	ConOps

High-level Requirements	Source
Implement reusable, more adaptable, real-time reporting, analytics, and business intelligence tools to Enterprise users that leverage highly shared, cross-program information.	ConOps
Implement systemic security, auditing, logging, application performance management, and Business Activity Monitoring (BAM).	ConOps
SMR should: • A copy of the legacy system's data in a complete raw form will be obtained. • Complete raw legacy data set will be placed into a data lake repository. • Convert the raw data into a canonical format, standardize reference values and enhance data quality through de-duplication. • Place canonical, standardized, and de-duplicated data into the NoSQL database of the System Migration Repository. • Make NoSQL database contents available for ingestion by new BPO Partner systems in order to enable their operations within HHS 2020 Enterprise post-go-live.	NM HSD 2020 Reference Architecture
Establish master data management for Client, Provider, Employer, and other data.	TP Statement of Work
Implement and manage common services, such as logging/auditing services, event handling, data transformation and mapping and message and event queuing and sequencing.	TP Statement of Work
Monitor usage and maintain a record of resource levels and consumption within the solution.	TP Statement of Work
Support automated and integrated service checkpoints to monitor service accuracy and completeness before proceeding to the next step or application batch process.	TP Statement of Work
Perform SOA-related business process and service management.	TP Statement of Work
Capture performance data (e.g., elapsed time, dates) to support continuous improvement.	TP Statement of Work

High-level Requirements	Source
Provide the ability to suspend processing of erroneous transactions until the error is resolved and provide notification of the error and resolution.	TP Statement of Work
Provide structured exception and error handling.	TP Statement of Work
Distinguish between errors (stop process) and exception conditions (skip transaction and continue process).	TP Statement of Work
Services will interoperate via sending/receiving synchronous and asynchronous messages.	NM HSD 2020 Reference Architecture
Support common SOA and Enterprise integration patterns, including publish/subscribe, broadcast, intermediaries, splitter/aggregator, parsing and validating messages and others as recommended by the Vendor.	TP Statement of Work
Facilitate integration with access to services for data sharing between applications and entities, in accordance with service contracts and security policies.	TP Statement of Work
All requests for functionality and data contained within the HHS 2020 Enterprise pass through the Security System.	NM HSD 2020 Reference Architecture
Integration Requirements will enforce service designs (protocols used, description of services, naming conventions used, standards for asynchronous vs synchronous invocations, message encoding/markup, exception management) that are consistent with other module implementations as well as standards of the SI Platform.	TP Statement of Work
Interface Integration should support following approach/patterns • Web services • Batch • ETL • Secure file transfer	TP Statement of Work

Appendix J: Data Models

Please refer to SIDAM1 - Conceptual Data Model document for details.

1. **High-Level Conceptual Entity Relationship Diagram across Data Domains.**
2. **Conceptual Entity Relationship Diagram by Data Domain**

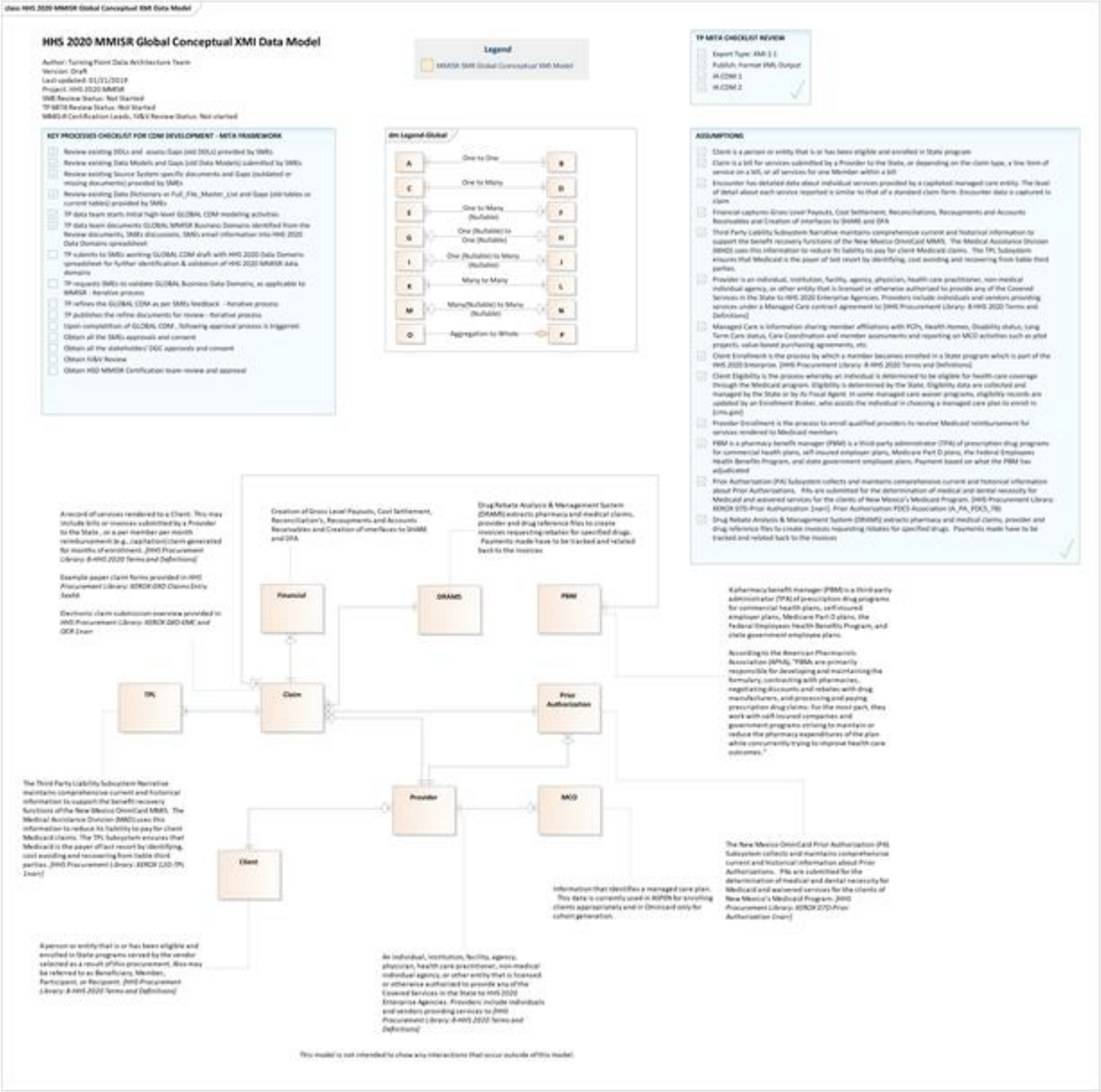
    a) Client
    b) Claim
    c) Drug Rebate Analysis and Management System
    d) Financial
    e) Managed Care Organization
    f) Pharmacy Benefits Management
    g) Prior Authorization
    h) Provider
    i) Third Party Liability

High-Level Conceptual Entity Relationship Diagram across Data Domains

SharePoint link for this ERD

- PDF

Figure 13-7: High-Level Conceptual Entity Relationship Diagram


HHS 2020 MMISR Global Conceptual XMI Data Model
Author: Turning Point Data Architecture Team
Version: Draft
Last updated: 01/11/2019
Project: HHS 2020 MMISR
Legend
KEY PROCESSES CHECKLIST FOR CDM DEVELOPMENT - MITA FRAMEWORK
dm Legend-Global
A
B
One to One
C
D
One to Many
E
F
One to Many (Nullable)
G
H
One (Nullable) to One (Nullable)
I
J
One (Nullable) to Many (Nullable)
K
L
Many to Many
M
N
Many(Nullable) to Many (Nullable)
O
P
Aggregation to Whole
TP MITA CHECKLIST REVIEW
ASSUMPTIONS
Financial
DRAMS
PBM
TPL
Claim
Prior Authorization
Provider
MCO
Client
This model is not intended to show any interactions that occur outside of this model.

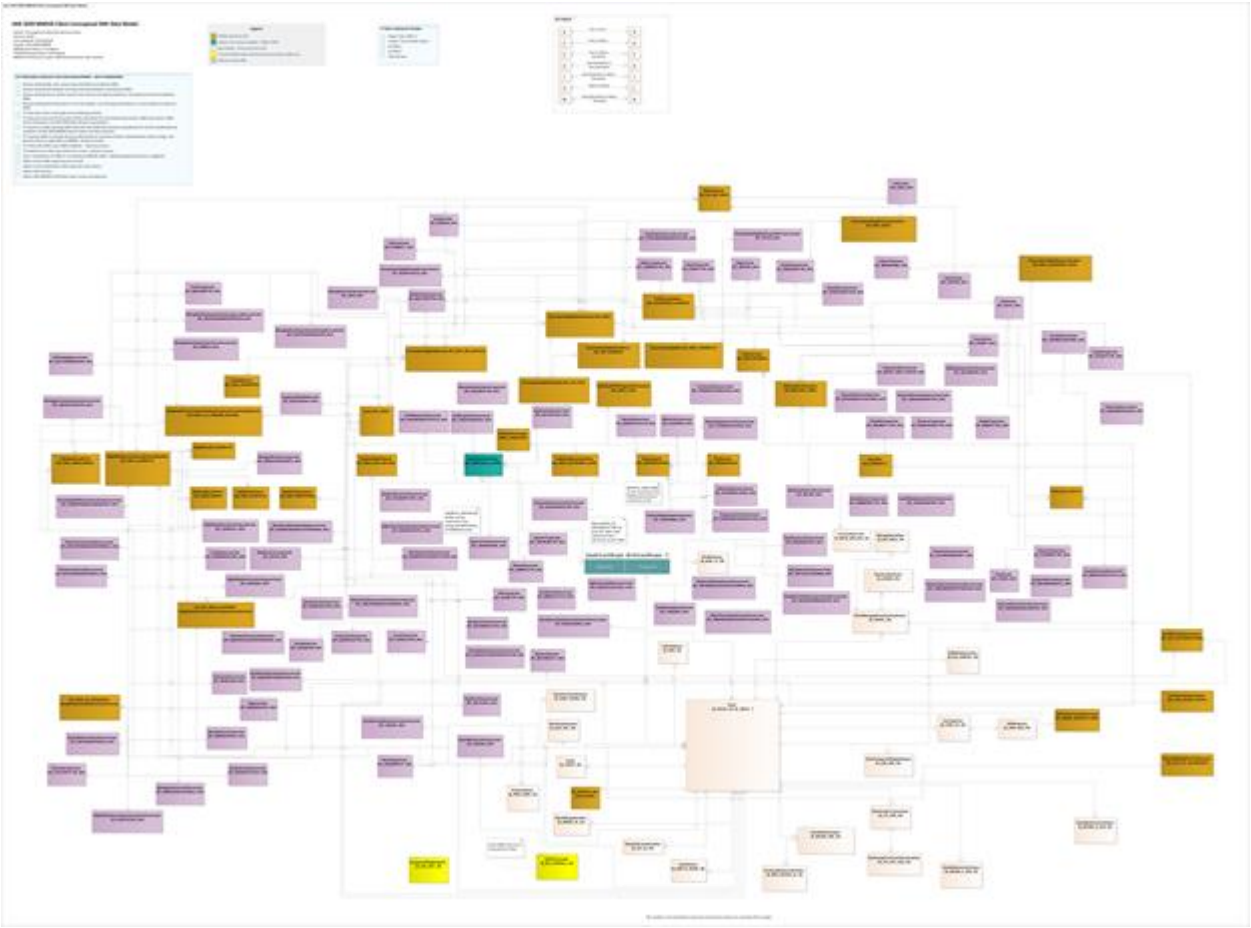
Conceptual Entity Relationship Diagram by Data Domain

The ERDs depicted within the sections represent each of the HHS 2020 data domains. Each subsection under this section contains an ERD depicting the business entities and their relationships within the domain in question.

Client – SharePoint link for this ERD:

- PDF
- XMI

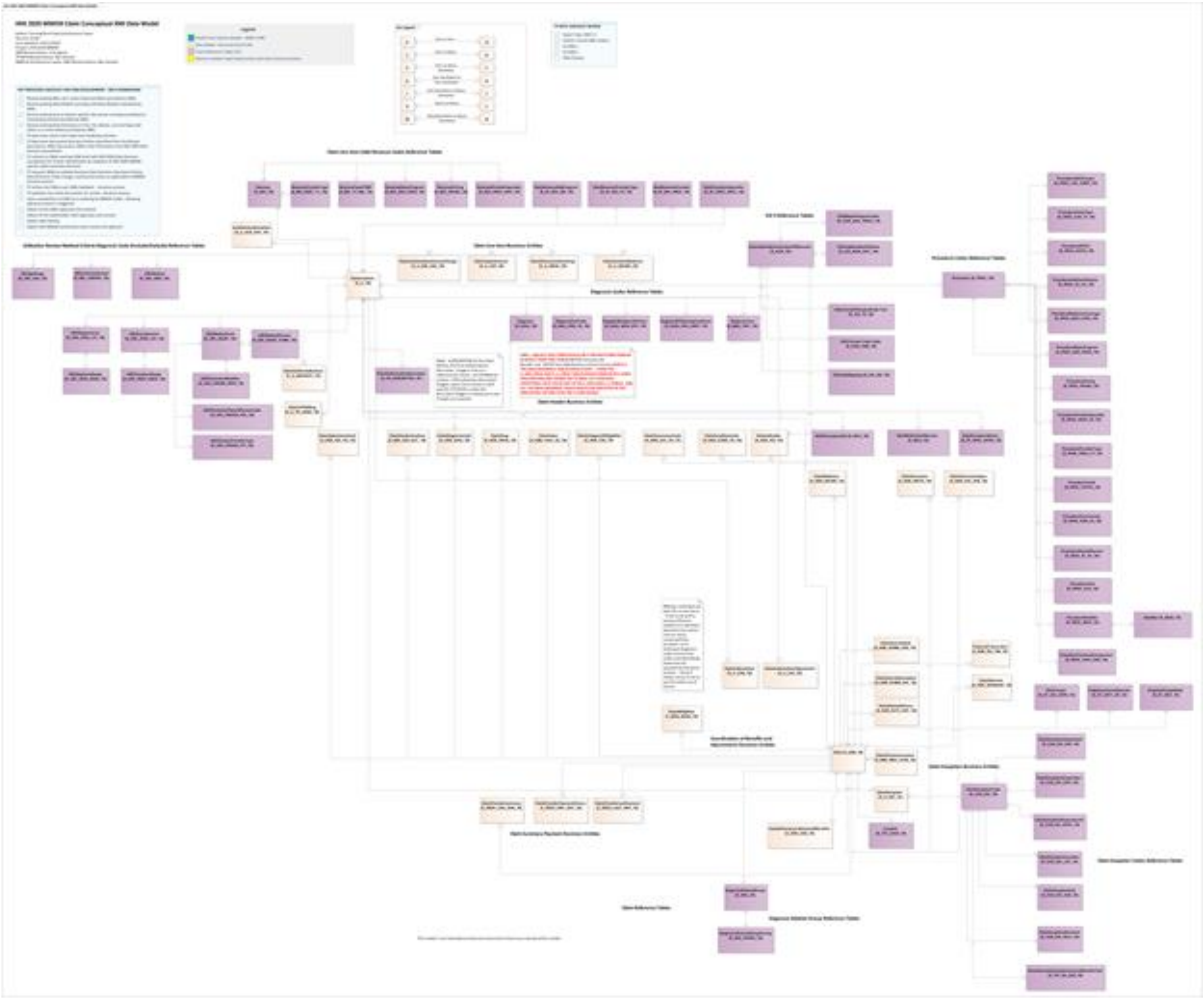
Figure 13-8: Conceptual Entity Relationship Diagram



Claim – SharePoint link for this ERD

- PDF
- XMI

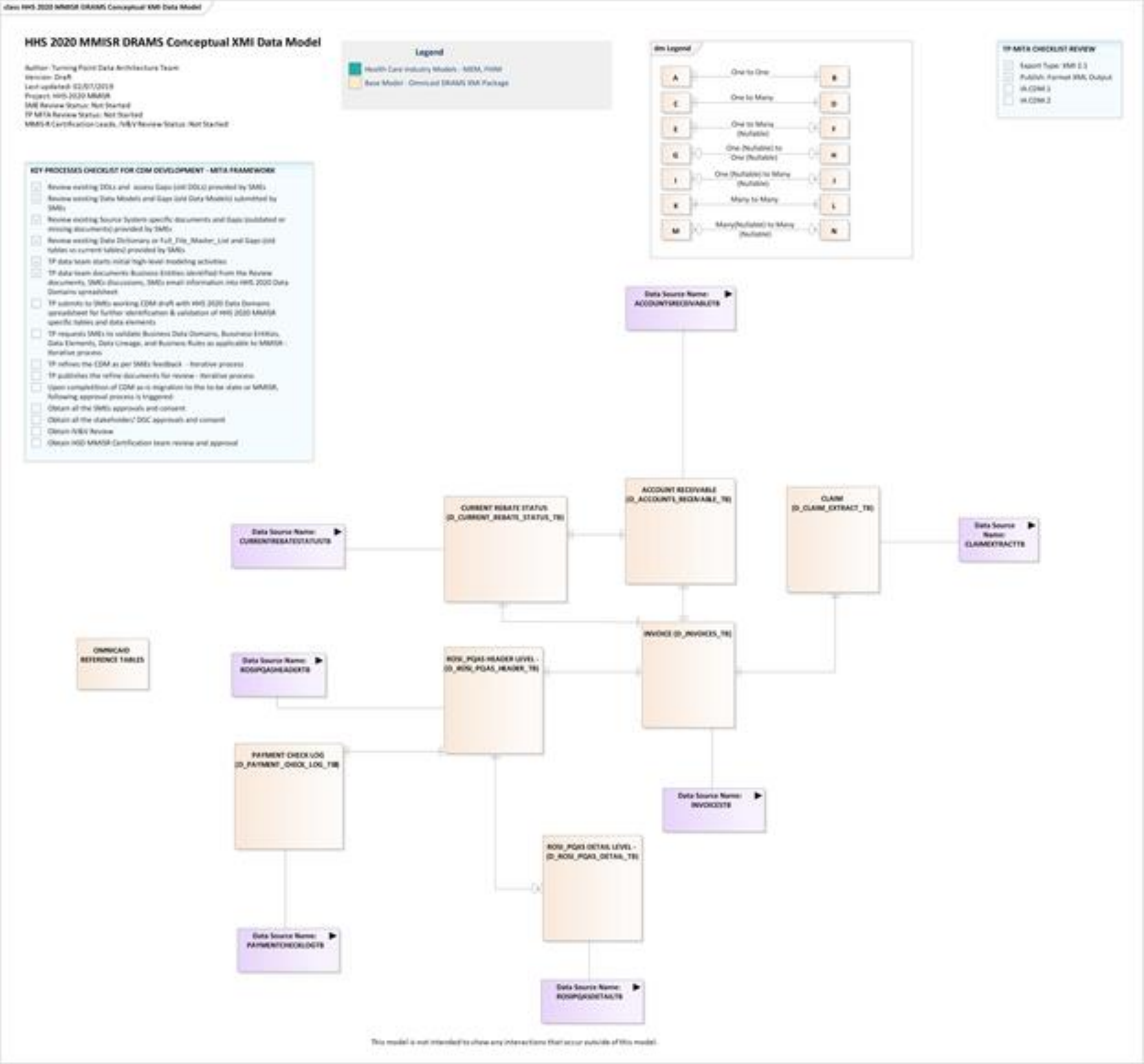
Figure 13-9: Claim Conceptual Data Model



Drug Rebate Analysis and Management System – SharePoint link for this ERD:

- PDF
- XMI

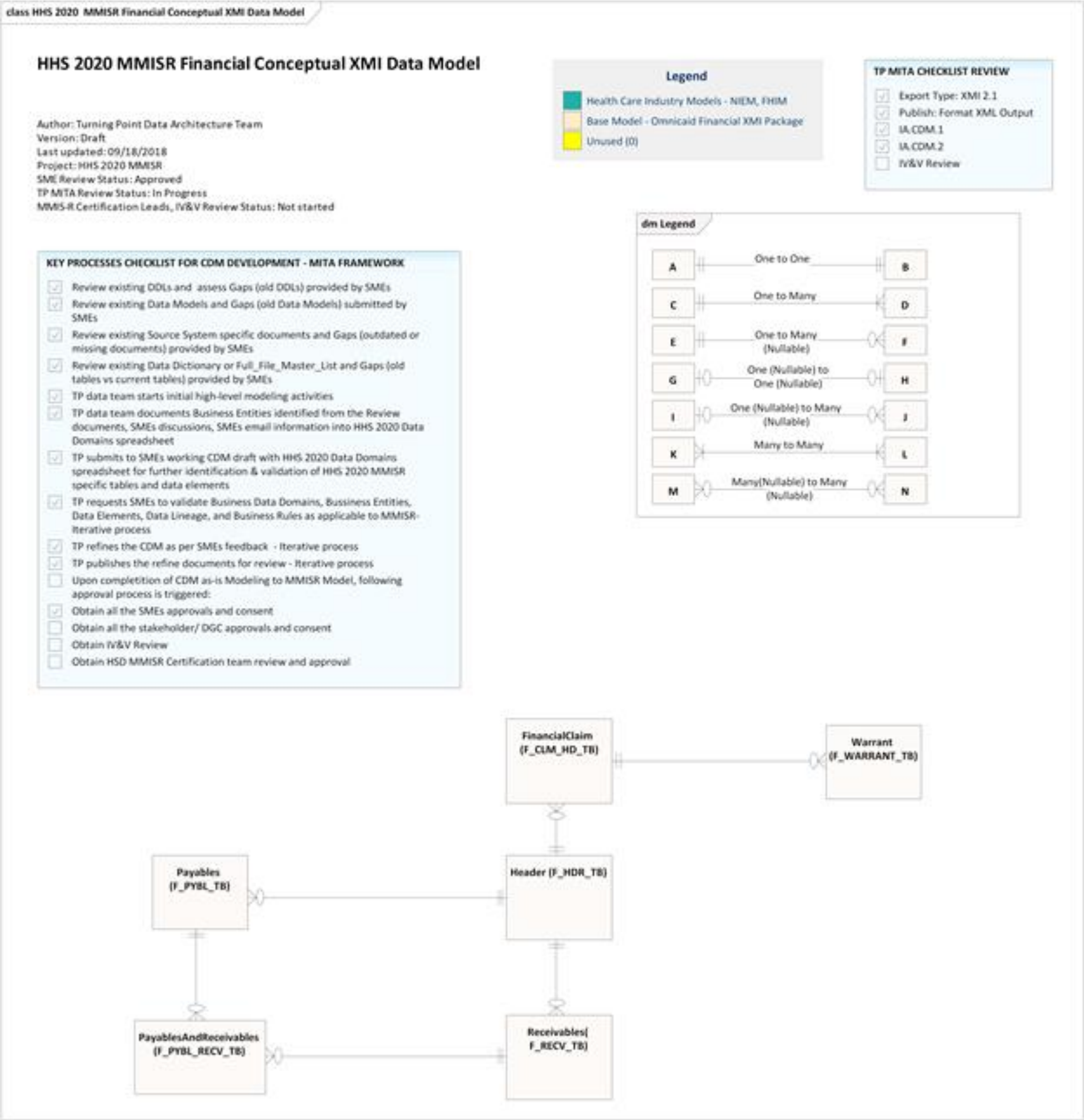
Figure 13-10: Drug Rebate Analysis and Management System


class HHS 2020 MMISR DRAMS Conceptual XMI Data Model
HHS 2020 MMISR DRAMS Conceptual XMI Data Model
Author: Turning Point Data Architecture Team
Version: Draft
Last updated: 02/07/2019
Project: HHS-2020 MMISR
SME Review Status: Not Started
TP MITA Review Status: Not Started
MMIS-R Certification Leads, IV&V Review Status: Not Started
Legend
Health Care Industry Models - MEM, FHIM
Base Model - Omnicaid DRAMS XMI Package
dm Legend
A
B
One to One
C
D
One to Many
E
F
One to Many (Nullable)
G
H
One (Nullable) to One (Nullable)
I
J
One (Nullable) to Many (Nullable)
K
L
Many to Many
M
N
Many(Nullable) to Many (Nullable)
TP MITA CHECKLIST REVIEW
Export Type: XMI 2.1
Publish: Format XML Output
IA.CDM.1
IA.CDM.2
KEY PROCESSES CHECKLIST FOR CDM DEVELOPMENT - MITA FRAMEWORK
Review existing DDLs and assess Gaps (old DDLs) provided by SMEs
Review existing Data Models and Gaps (old Data Models) submitted by SMEs
Review existing Source System specific documents and Gaps (outdated or missing documents) provided by SMEs
Review existing Data Dictionary or Full_File_Master_List and Gaps (old tables vs current tables) provided by SMEs
TP data team starts initial high-level modeling activities
TP data team documents Business Entities identified from the Review documents, SMEs discussions, SMEs email information into HHS 2020 Data Domains spreadsheet
TP submits to SMEs working CDM draft with HHS 2020 Data Domains spreadsheet for further identification & validation of HHS 2020 MMISR specific tables and data elements
TP requests SMEs to validate Business Data Domains, Business Entities, Data Elements, Data Lineage, and Business Rules as applicable to MMISR - Iterative process
TP refines the CDM as per SMEs feedback - Iterative process
TP publishes the refine documents for review - Iterative process
Upon completion of CDM as-is migration to the to-be state or MMISR, following approval process is triggered
Obtain all the SMEs approvals and consent
Obtain all the stakeholders' DGC approvals and consent
Obtain IV&V Review
Obtain HSD MMISR Certification team review and approval
Data Source Name: ACCOUNTSRECEIVABLETB
ACCOUNT RECEIVABLE (D_ACCOUNTS_RECEIVABLE_TB)
CURRENT REBATE STATUS (D_CURRENT_REBATE_STATUS_TB)
Data Source Name: CURRENTREBATESTATUSTB
CLAIM (D_CLAIM_EXTRACT_TB)
Data Source Name: CLAIMEXTRACTTB
OMNICAID REFERENCE TABLES
Data Source Name: ROSIPQASHEADERTB
ROSI_PQAS HEADER LEVEL - (D_ROSI_PQAS_HEADER_TB)
INVOICE (D_INVOICES_TB)
PAYMENT CHECK LOG (D_PAYMENT_CHECK_LOG_TB)
Data Source Name: INVOICESTB
ROSI_PQAS DETAIL LEVEL - (D_ROSI_PQAS_DETAIL_TB)
Data Source Name: PAYMENTCHECKLOGTB
Data Source Name: ROSIPQASDETAILTB
This model is not intended to show any interactions that occur outside of this model.

Financial – SharePoint link for this ERD:

- PDF
- XMI

Figure 13-11: Financial Conceptual Data Model



class HHS 2020 MMISR Financial Conceptual XMI Data Model

HHS 2020 MMISR Financial Conceptual XMI Data Model

Author: Turning Point Data Architecture Team
Version: Draft
Last updated: 09/18/2018
Project: HHS 2020 MMISR
SME Review Status: Approved
TP MITA Review Status: In Progress
MMIS-R Certification Leads, IV&V Review Status: Not started

KEY PROCESSES CHECKLIST FOR CDM DEVELOPMENT - MITA FRAMEWORK

- ☑ Review existing DDLs and assess Gaps (old DDLs) provided by SMEs
- ☑ Review existing Data Models and Gaps (old Data Models) submitted by SMEs
- ☑ Review existing Source System specific documents and Gaps (outdated or missing documents) provided by SMEs
- ☑ Review existing Data Dictionary or Full_File_Master_List and Gaps (old tables vs current tables) provided by SMEs
- ☑ TP data team starts initial high-level modeling activities
- ☑ TP data team documents Business Entities identified from the Review documents, SMEs discussions, SMEs email information into HHS 2020 Data Domains spreadsheet
- ☑ TP submits to SMEs working CDM draft with HHS 2020 Data Domains spreadsheet for further identification & validation of HHS 2020 MMISR specific tables and data elements
- ☑ TP requests SMEs to validate Business Data Domains, Bussiness Entities, Data Elements, Data Lineage, and Business Rules as applicable to MMISR-Iterative process
- ☑ TP refines the CDM as per SMEs feedback - Iterative process
- ☑ TP publishes the refine documents for review - Iterative process
- □ Upon completition of CDM as-is Modeling to MMISR Model, following approval process is triggered:
- ☑ Obtain all the SMEs approvals and consent
- □ Obtain all the stakeholder/ DGC approvals and consent
- □ Obtain IV&V Review
- □ Obtain HSD MMISR Certification team review and approval

Legend

- 🟩 Health Care Industry Models - NIEM, FHIM
- ▨ Base Model - Omnicaid Financial XMI Package
- 🟨 Unused (0)

TP MITA CHECKLIST REVIEW

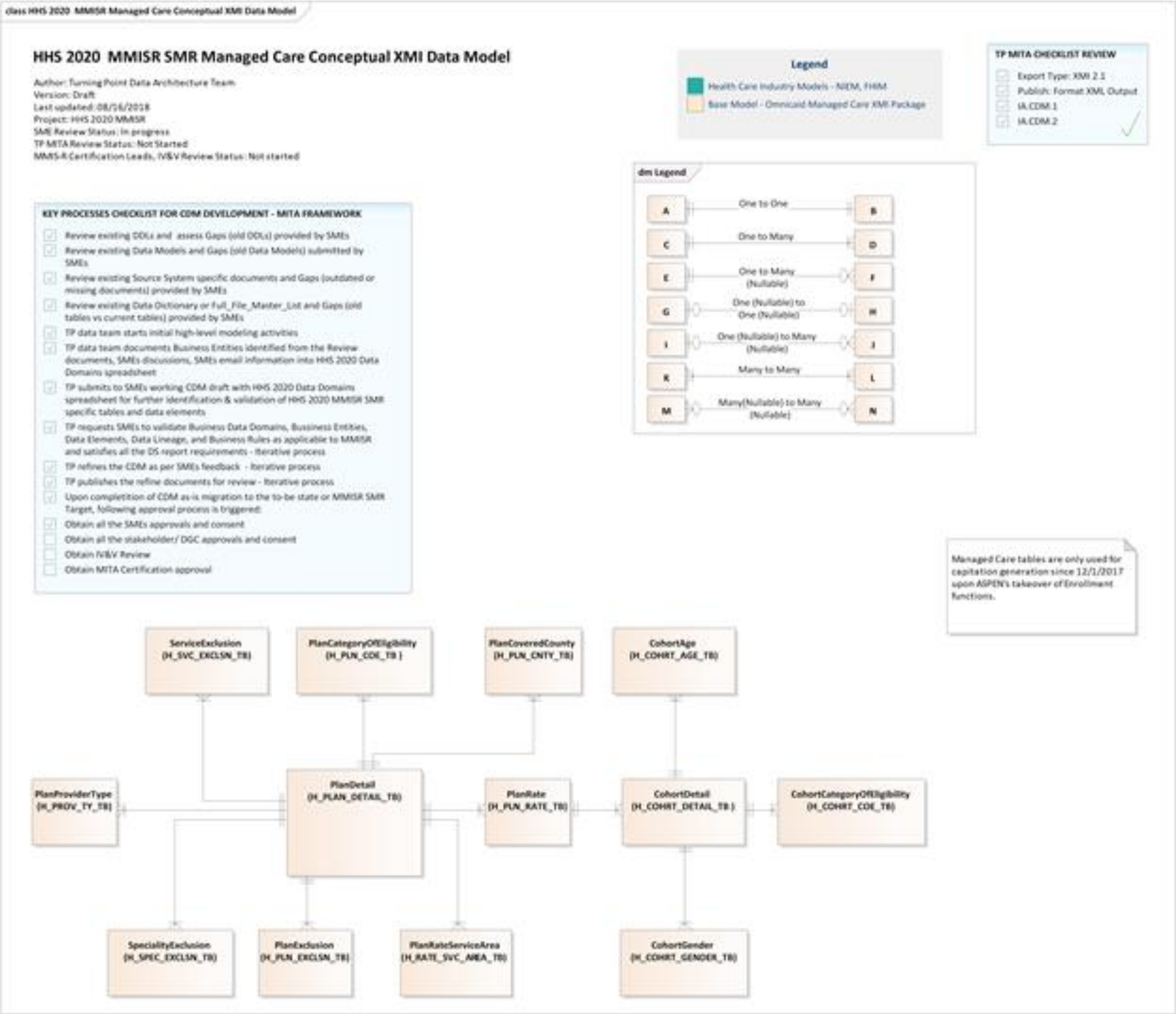
- ☑ Export Type: XMI 2.1
- ☑ Publish: Format XML Output
- ☑ IA.CDM.1
- ☑ IA.CDM.2
- □ IV&V Review

dm Legend
A
One to One
B
C
One to Many
D
E
One to Many (Nullable)
F
G
One (Nullable) to One (Nullable)
H
I
One (Nullable) to Many (Nullable)
J
K
Many to Many
L
M
Many(Nullable) to Many (Nullable)
N
FinancialClaim (F_CLM_HD_TB)
Warrant (F_WARRANT_TB)
Payables (F_PYBL_TB)
Header (F_HDR_TB)
PayablesAndReceivables (F_PYBL_RECV_TB)
Receivables(F_RECV_TB)

Managed Care Organization – SharePoint link for this ERD

- [PDF]()
- [XMI]()

Figure 13-12: Managed Care Organization Conceptual Data Model



class HHS 2020 MMISR Managed Care Conceptual XMI Data Model

HHS 2020 MMISR SMR Managed Care Conceptual XMI Data Model

Author: Turning Point Data Architecture Team
Version: Draft
Last updated: 08/16/2018
Project: HHS 2020 MMISR
SME Review Status: In progress
TP MITA Review Status: Not Started
MMIS-R Certification Leads, IV&V Review Status: Not started

KEY PROCESSES CHECKLIST FOR CDM DEVELOPMENT - MITA FRAMEWORK

- ☑ Review existing DDLs and assess Gaps (old DDLs) provided by SMEs
- ☑ Review existing Data Models and Gaps (old Data Models) submitted by SMEs
- ☑ Review existing Source System specific documents and Gaps (outdated or missing documents) provided by SMEs
- ☑ Review existing Data Dictionary or Full_File_Master_List and Gaps (old tables vs current tables) provided by SMEs
- ☑ TP data team starts initial high-level modeling activities
- ☑ TP data team documents Business Entities identified from the Review documents, SMEs discussions, SMEs email information into HHS 2020 Data Domains spreadsheet
- ☑ TP submits to SMEs working CDM draft with HHS 2020 Data Domains spreadsheet for further identification & validation of HHS 2020 MMISR SMR specific tables and data elements
- ☑ TP requests SMEs to validate Business Data Domains, Bussiness Entities, Data Elements, Data Lineage, and Business Rules as applicable to MMISR and satisfies all the DS report requirements - Iterative process
- ☑ TP refines the CDM as per SMEs feedback - Iterative process
- ☑ TP publishes the refine documents for review - Iterative process
- ☑ Upon completition of CDM as-is migration to the to-be state or MMISR SMR Target, following approval process is triggered:
- ☑ Obtain all the SMEs approvals and consent
- □ Obtain all the stakeholder/ DGC approvals and consent
- □ Obtain IV&V Review
- □ Obtain MITA Certification approval

Legend

🟩 Health Care Industry Models - NIEM, FHIM
🟧 Base Model - Omnicaid Managed Care XMI Package

TP MITA CHECKLIST REVIEW

- □ Export Type: XMI 2.1
- ☑ Publish: Format XML Output
- ☑ IA.CDM.1
- ☑ IA.CDM.2

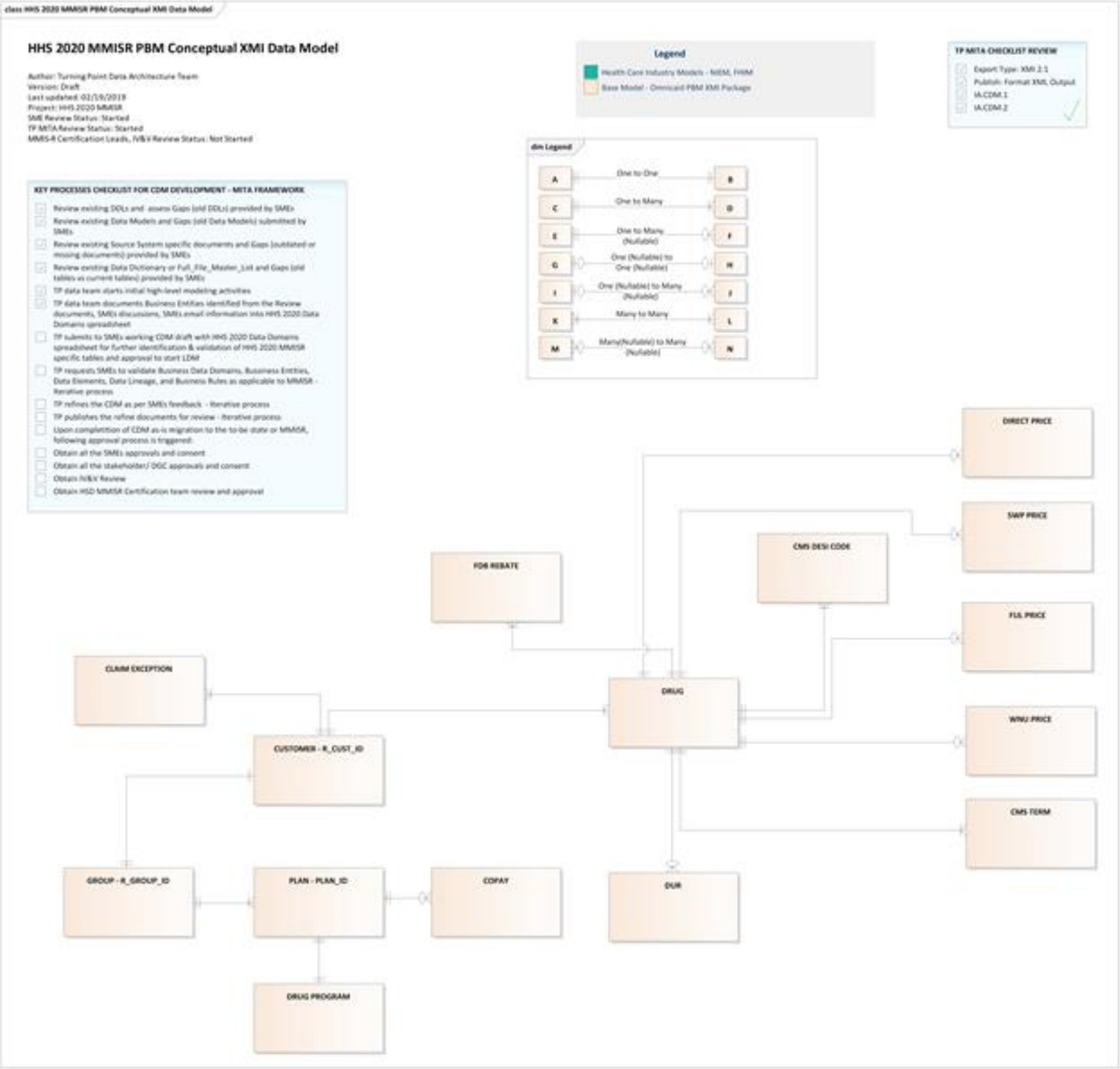
Managed Care tables are only used for capitation generation since 12/1/2017 upon ASPEN's takeover of Enrollment functions.

✓
dm Legend
A
One to One
B
C
One to Many
D
E
One to Many (Nullable)
F
G
One (Nullable) to One (Nullable)
H
I
One (Nullable) to Many (Nullable)
J
K
Many to Many
L
M
Many(Nullable) to Many (Nullable)
N
ServiceExclusion (H_SVC_EXCLSN_TB)
PlanCategoryOfEligibility (H_PLN_COE_TB)
PlanCoveredCounty (H_PLN_CNTY_TB)
CohortAge (H_COHRT_AGE_TB)
PlanProviderType (H_PROV_TY_TB)
PlanDetail (H_PLAN_DETAIL_TB)
PlanRate (H_PLN_RATE_TB)
CohortDetail (H_COHRT_DETAIL_TB)
CohortCategoryOfEligibility (H_COHRT_COE_TB)
SpecialityExclusion (H_SPEC_EXCLSN_TB)
PlanExclusion (H_PLN_EXCLSN_TB)
PlanRateServiceArea (H_RATE_SVC_AREA_TB)
CohortGender (H_COHRT_GENDER_TB)

Pharmacy Benefits Management – SharePoint link for this ERD

- PDF
- XMI

Figure 13-13: Pharmacy Benefits Management Conceptual Data Model



class HHS 2020 MMISR PBM Conceptual XMI Data Model

HHS 2020 MMISR PBM Conceptual XMI Data Model

Author: Turning Point Data Architecture Team
Version: Draft
Last updated: 01/19/2019
Project: HHS 2020 MMISR
SME Review Status : Started
TP MITA Review Status: Started
MMIS-R Certification Leads, IV&V Review Status : Not Started

KEY PROCESSES CHECKLIST FOR CDM DEVELOPMENT - MITA FRAMEWORK

☑ Review existing DDLs and assess Gaps (old DDLs) provided by SMEs
☑ Review existing Data Models and Gaps (old Data Models) submitted by SMEs
☑ Review existing Source System specific documents and Gaps (outdated or missing documents) provided by SMEs
☑ Review existing Data Dictionary or Full_File_Master_List and Gaps (old tables vs current tables) provided by SMEs
☑ TP data team starts initial high-level modeling activities
☑ TP data team documents Business Entities identified from the Review documents, SMEs discussions, SMEs email information into HHS 2020 Data Domains spreadsheet
□ TP submits to SMEs working CDM draft with HHS 2020 Data Domains spreadsheet for further identification & validation of HHS 2020 MMISR specific tables and approval to start LDM
□ TP requests SMEs to validate Business Data Domains, Business Entities, Data Elements, Data Lineage, and Business Rules as applicable to MMISR - Iterative process
□ TP refines the CDM as per SMEs feedback - Iterative process
□ TP publishes the refine documents for review - Iterative process
□ Upon completition of CDM as-is migration to the to-be state or MMISR, following approval process is triggered:
□ Obtain all the SMEs approvals and consent
□ Obtain all the stakeholder/ DGC approvals and consent
□ Obtain IV&V Review
□ Obtain HSD MMISR Certification team review and approval

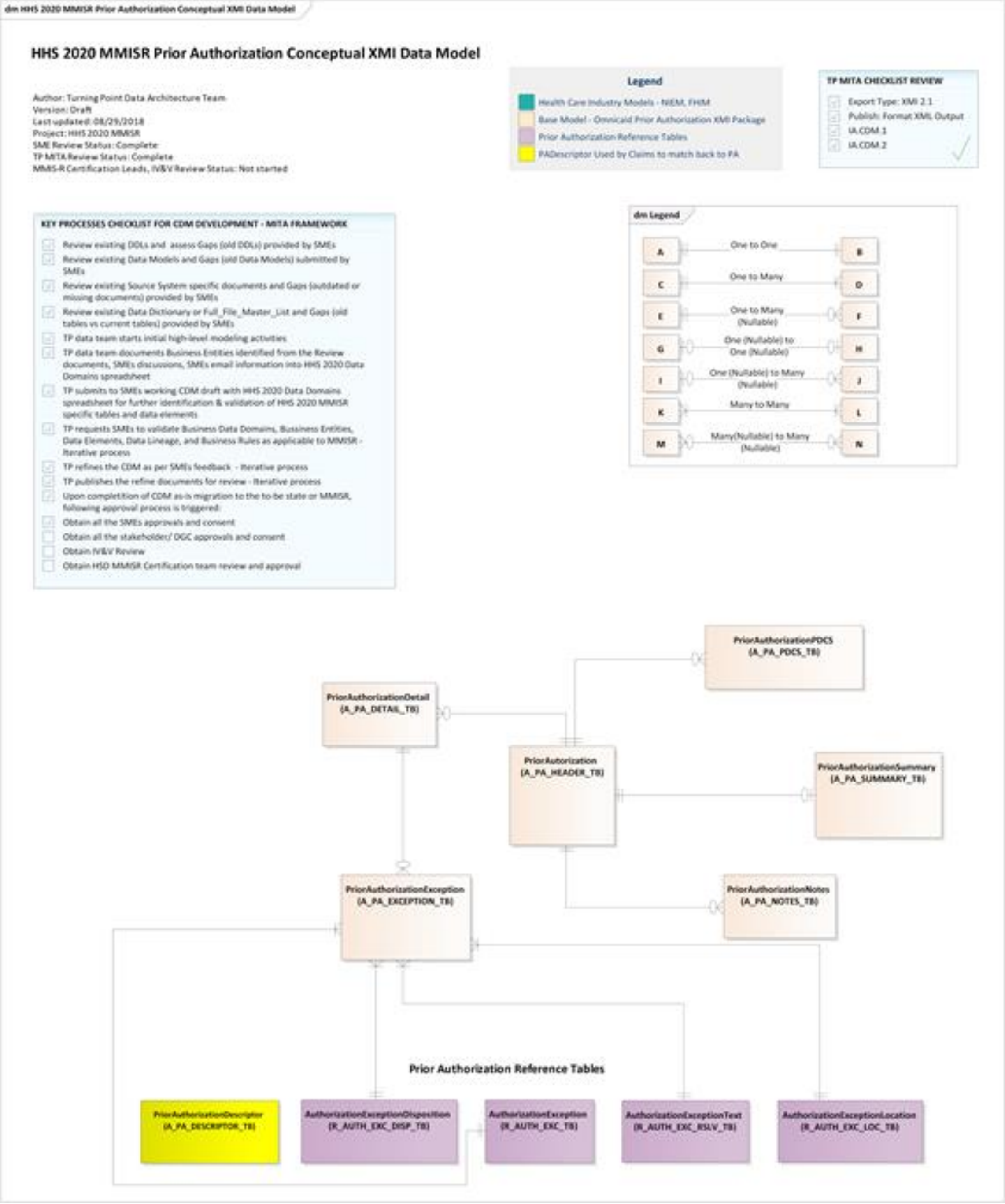
Legend
🟩 Health Care Industry Models - NIEM, FHIM
🟧 Base Model - Omnicaid PBM XMI Package

TP MITA CHECKLIST REVIEW
☑ Export Type: XMI 2.1
☑ Publish: Format XML Output
☑ IA.CDM.1
☑ IA.CDM.2

Prior Authorization – SharePoint link for this ERD

- PDF
- XMI

Figure 13-14: Prior Authorization Conceptual Data Model



dm HHS 2020 MMISR Prior Authorization Conceptual XMI Data Model

HHS 2020 MMISR Prior Authorization Conceptual XMI Data Model

Author: Turning Point Data Architecture Team
Version: Draft
Last updated: 08/29/2018
Project: HHS 2020 MMISR
SME Review Status: Complete
TP MITA Review Status: Complete
MMIS-R Certification Leads, IV&V Review Status: Not started

Legend

- 🟩 Health Care Industry Models - NIEM, FHIM
- 🟧 Base Model - Omnicaid Prior Authorization XMI Package
- 🟪 Prior Authorization Reference Tables
- 🟨 PADescriptor Used by Claims to match back to PA

TP MITA CHECKLIST REVIEW

- ☑ Export Type: XMI 2.1
- ☑ Publish: Format XML Output
- ☑ IA.CDM.1
- ☑ IA.CDM.2

KEY PROCESSES CHECKLIST FOR CDM DEVELOPMENT - MITA FRAMEWORK

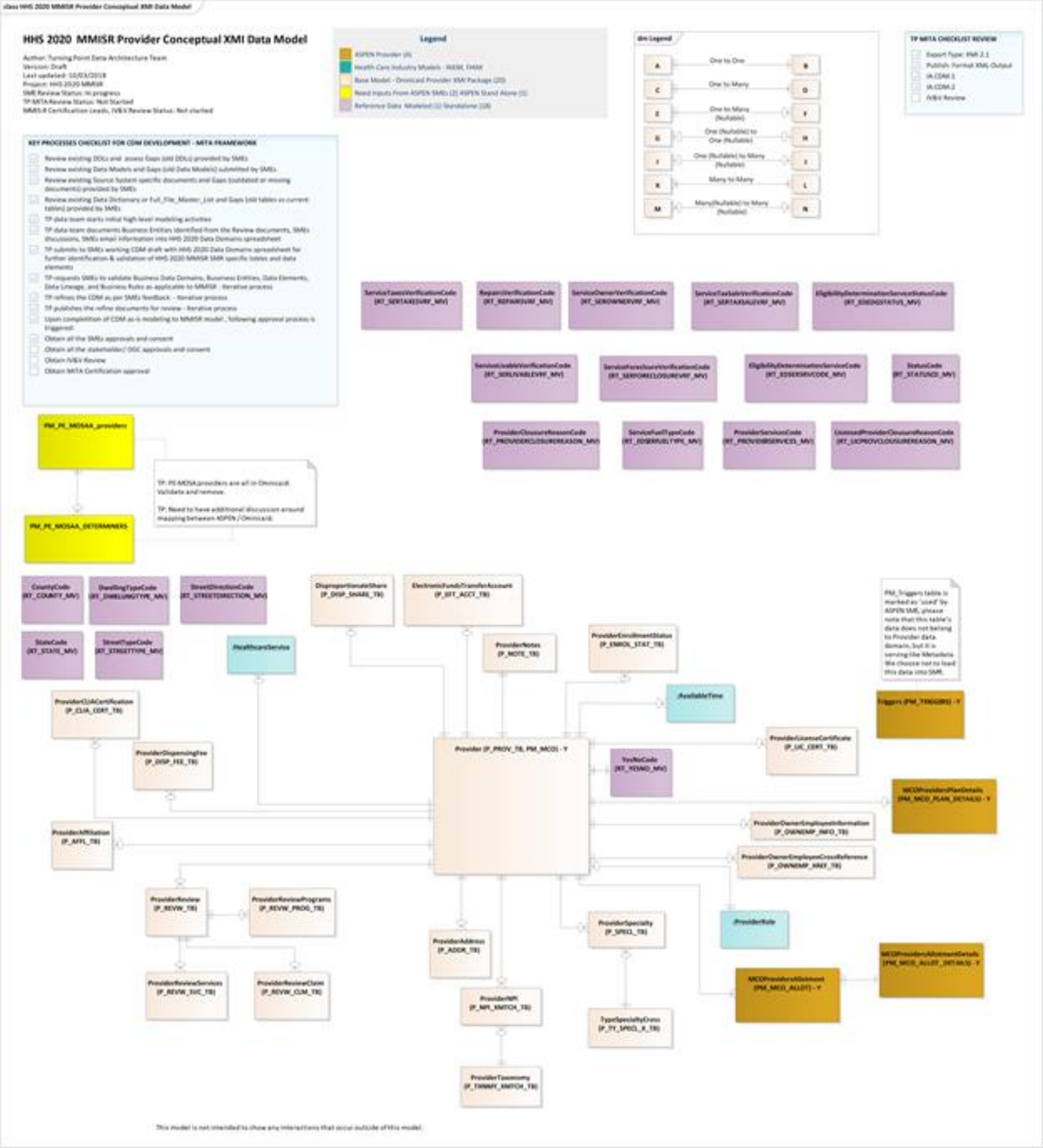
- ☑ Review existing DDLs and assess Gaps (old DDLs) provided by SMEs
- ☑ Review existing Data Models and Gaps (old Data Models) submitted by SMEs
- ☑ Review existing Source System specific documents and Gaps (outdated or missing documents) provided by SMEs
- ☑ Review existing Data Dictionary or Full_File_Master_List and Gaps (old tables vs current tables) provided by SMEs
- ☑ TP data team starts initial high-level modeling activities
- ☑ TP data team documents Business Entities identified from the Review documents, SMEs discussions, SMEs email information into HHS 2020 Data Domains spreadsheet
- ☑ TP submits to SMEs working CDM draft with HHS 2020 Data Domains spreadsheet for further identification & validation of HHS 2020 MMISR specific tables and data elements
- ☑ TP requests SMEs to validate Business Data Domains, Bussiness Entities, Data Elements, Data Lineage, and Business Rules as applicable to MMISR - Iterative process
- ☑ TP refines the CDM as per SMEs feedback - Iterative process
- ☑ TP publishes the refine documents for review - Iterative process
- ☑ Upon completition of CDM as-is migration to the to-be state or MMISR, following approval process is triggered:
- ☑ Obtain all the SMEs approvals and consent
- □ Obtain all the stakeholder/ OGC approvals and consent
- □ Obtain IV&V Review
- □ Obtain HSD MMISR Certification team review and approval

dm Legend
A
One to One
B
C
One to Many
D
E
One to Many (Nullable)
F
G
One (Nullable) to One (Nullable)
H
I
One (Nullable) to Many (Nullable)
J
K
Many to Many
L
M
Many(Nullable) to Many (Nullable)
N
PriorAuthorizationPDCS (A_PA_PDCS_TB)
PriorAuthorizationDetail (A_PA_DETAIL_TB)
PriorAutorization (A_PA_HEADER_TB)
PriorAuthorizationSummary (A_PA_SUMMARY_TB)
PriorAuthorizationException (A_PA_EXCEPTION_TB)
PriorAuthorizationNotes (A_PA_NOTES_TB)
Prior Authorization Reference Tables
PriorAuthorizationDescriptor (A_PA_DESCRIPTOR_TB)
AuthorizationExceptionDisposition (R_AUTH_EXC_DISP_TB)
AuthorizationException (R_AUTH_EXC_TB)
AuthorizationExceptionText (R_AUTH_EXC_RSLV_TB)
AuthorizationExceptionLocation (R_AUTH_EXC_LOC_TB)

Provider – SharePoint link for this ERD

- PDF
- XMI

Figure 13-15: Provider Conceptual Data Model



class HHS 2020 MMISR Provider Conceptual XMI Data Model
HHS 2020 MMISR Provider Conceptual XMI Data Model
Legend
ASPEN Provider (4)
Health Care Industry Models - IMEM, FHIM
Base Model - Omnicaid Provider XMI Package (20)
Need Inputs From ASPEN SMEs (2) ASPEN Stand Alone (1)
Reference Data Modeled (1) Standalone (18)
dm Legend
A
One to One
B
C
One to Many
D
E
One to Many (Nullable)
F
G
One (Nullable) to One (Nullable)
H
I
One (Nullable) to Many (Nullable)
J
K
Many to Many
L
M
Many(Nullable) to Many (Nullable)
N
TP MITA CHECKLIST REVIEW
Export Type: XMI 2.1
Publish: Format XML Output
IA CDM 1
IA CDM 2
IV&V Review
KEY PROCESSES CHECKLIST FOR CDM DEVELOPMENT - MITA FRAMEWORK
ServiceTaxesVerificationCode (RT_SERTAXESVRF_MV)
RepairsVerificationCode (RT_REPAIRSVRF_MV)
ServiceOwnerVerificationCode (RT_SEROWNERVRF_MV)
ServiceTaxSaleVerificationCode (RT_SERTAXSALEVRF_MV)
EligibilityDeterminationServiceStatusCode (RT_EDSDSTATUS_MV)
ServiceLivableVerificationCode (RT_SERLIVABLEVRF_MV)
ServiceForeclosureVerificationCode (RT_SERFORECLOSUREVRF_MV)
EligibilityDeterminationServiceCode (RT_EDSERSRVCODE_MV)
StatusCode (RT_STATUSCD_MV)
ProviderClosureReasonCode (RT_PROVIDERCLOSUREREASON_MV)
ServiceFuelTypeCode (RT_EDSERFUELTYPE_MV)
ProviderServicesCode (RT_PROVIDERSERVICES_MV)
LicensedProviderClosureReasonCode (RT_LICPROVCLOSUREREASON_MV)
PM_PE_MOSAA_providers
PM_PE_MOSAA_DETERMINERS
CountyCode (RT_COUNTY_MV)
DwellingTypeCode (RT_DWELLINGTYPE_MV)
StreetDirectionCode (RT_STREETDIRECTION_MV)
StateCode (RT_STATE_MV)
StreetTypeCode (RT_STREETTYPE_MV)
::HealthcareService
DisproportionateShare (P_DISP_SHARE_TB)
ElectronicFundsTransferAccount (P_EFT_ACCT_TB)
ProviderNotes (P_NOTE_TB)
ProviderEnrollmentStatus (P_ENROL_STAT_TB)
::AvailableTime
Triggers (PM_TRIGGERS) - Y
ProviderCLIACertification (P_CLIA_CERT_TB)
ProviderDispensingFee (P_DISP_FEE_TB)
Provider (P_PROV_TB, PM_MCO) - Y
YesNoCode (RT_YESNO_MV)
ProviderLicenseCertificate (P_LIC_CERT_TB)
MCOProvidersPlanDetails (PM_MCO_PLAN_DETAILS) - Y
ProviderAffiliation (P_AFFL_TB)
ProviderOwnerEmployeeInformation (P_OWNEMP_INFO_TB)
ProviderOwnerEmployeeCrossReference (P_OWNEMP_XREF_TB)
ProviderReview (P_REVW_TB)
ProviderReviewPrograms (P_REVW_PROG_TB)
ProviderAddress (P_ADDR_TB)
ProviderSpecialty (P_SPECL_TB)
::ProviderRole
MCOProvidersAllotmentDetails (PM_MCO_ALLOT_DETAILS) - Y
MCOProvidersAllotment (PM_MCO_ALLOT) - Y
ProviderReviewServices (P_REVW_SVC_TB)
ProviderReviewClaim (P_REVW_CLM_TB)
ProviderNPI (P_NPI_XMITCH_TB)
TypeSpecialtyCross (P_TY_SPECL_X_TB)
ProviderTaxonomy (P_TXNMY_XMITCH_TB)

Author: Turning Point Data Architecture Team
Version: Draft
Last updated: 10/03/2018
Project: HHS 2020 MMISR
SME Review Status: In progress
TP MITA Review Status: Not Started
MMIS-R Certification Leads, IV&V Review Status: Not started

- □ Review existing DDLs and assess Gaps (old DDLs) provided by SMEs
- □ Review existing Data Models and Gaps (old Data Models) submitted by SMEs
- □ Review existing Source System specific documents and Gaps (outdated or missing documents) provided by SMEs
- □ Review existing Data Dictionary or Full_File_Master_List and Gaps (old tables vs current tables) provided by SMEs
- □ TP data team starts initial high level modeling activities
- □ TP data team documents Business Entities identified from the Review documents, SMEs discussions, SMEs email information into HHS 2020 Data Domains spreadsheet
- □ TP submits to SMEs working CDM draft with HHS 2020 Data Domains spreadsheet for further identification & validation of HHS 2020 MMISR SMR specific tables and data elements
- □ TP requests SMEs to validate Business Data Domains, Business Entities, Data Elements, Data Lineage, and Business Rules as applicable to MMISR - Iterative process
- □ TP refines the CDM as per SMEs feedback - Iterative process
- □ TP publishes the refine documents for review - Iterative process
- □ Upon completion of CDM as-is modeling to MMISR model , following approval process is triggered:
- □ Obtain all the SMEs approvals and consent
- □ Obtain all the stakeholder/ DGC approvals and consent
- □ Obtain IV&V Review
- □ Obtain MITA Certification approval

TP: PE-MOSA providers are all in Omnicaid. Validate and remove.

TP: Need to have additional discussion around mapping between ASPEN / Omnicaid.

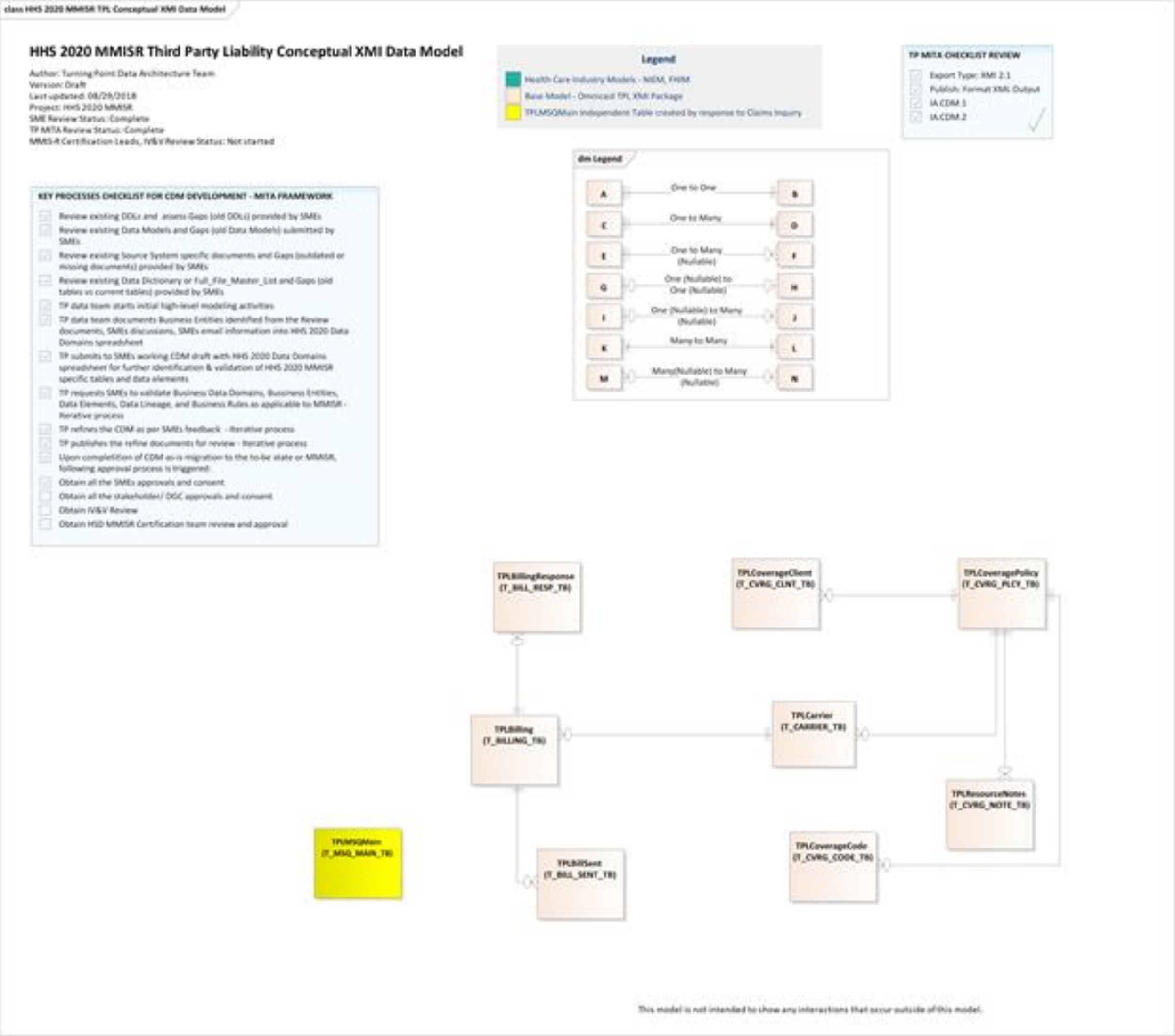
PM_Triggers table is marked as 'used' by ASPEN SME, please note that this table's data does not belong to Provider data domain, but it is serving like Metadata. We choose not to load this data into SMR.

This model is not intended to show any interactions that occur outside of this model.

Third Party Liability – SharePoint link for this ERD:

- PDF
- XMI

Figure 13-16: Third Party Liability Conceptual Data Model



class HHS 2020 MMISR TPL Conceptual XMI Data Model

HHS 2020 MMISR Third Party Liability Conceptual XMI Data Model

Author: Turning Point Data Architecture Team
Version: Draft
Last updated: 08/29/2018
Project: HHS 2020 MMISR
SME Review Status : Complete
TP MITA Review Status: Complete
MMIS-R Certification Leads, IV&V Review Status: Not started

KEY PROCESSES CHECKLIST FOR CDM DEVELOPMENT - MITA FRAMEWORK

- ☑ Review existing DDLs and assess Gaps (old DDLs) provided by SMEs
- ☑ Review existing Data Models and Gaps (old Data Models) submitted by SMEs
- ☑ Review existing Source System specific documents and Gaps (outdated or missing documents) provided by SMEs
- ☑ Review existing Data Dictionary or Full_File_Master_List and Gaps (old tables vs current tables) provided by SMEs
- ☑ TP data team starts initial high-level modeling activities
- ☑ TP data team documents Business Entities identified from the Review documents, SMEs discussions, SMEs email information into HHS 2020 Data Domains spreadsheet
- ☑ TP submits to SMEs working CDM draft with HHS 2020 Data Domains spreadsheet for further identification & validation of HHS 2020 MMISR specific tables and data elements
- ☑ TP requests SMEs to validate Business Data Domains, Bussiness Entities, Data Elements, Data Lineage, and Business Rules as applicable to MMISR - Iterative process
- ☑ TP refines the CDM as per SMEs feedback - Iterative process
- ☑ TP publishes the refine documents for review - Iterative process
- ☑ Upon completiton of CDM as-is migration to the to-be state or MMISR, following approval process is triggered:
- ☑ Obtain all the SMEs approvals and consent
- ☐ Obtain all the stakeholder/ DGC approvals and consent
- ☐ Obtain IV&V Review
- ☐ Obtain HSD MMISR Certification team review and approval

Legend

- 🟩 Health Care Industry Models - NIEM, FHIM
- ▨ Base Model - Omnicaid TPL XMI Package
- 🟨 TPLMSQMain Independent Table created by response to Claims Inquiry

TP MITA CHECKLIST REVIEW

- ☑ Export Type: XMI 2.1
- ☑ Publish: Format XML Output
- ☑ IA.CDM.1
- ☑ IA.CDM.2

dm Legend
A
One to One
B
C
One to Many
D
E
One to Many (Nullable)
F
G
One (Nullable) to One (Nullable)
H
I
One (Nullable) to Many (Nullable)
J
K
Many to Many
L
M
Many(Nullable) to Many (Nullable)
N
TPLBillingResponse (T_BILL_RESP_TB)
TPLCoverageClient (T_CVRG_CLNT_TB)
TPLCoveragePolicy (T_CVRG_PLCY_TB)
TPLBilling (T_BILLING_TB)
TPLCarrier (T_CARRIER_TB)
TPLResourceNotes (T_CVRG_NOTE_TB)
TPLMSQMain (T_MSQ_MAIN_TB)
TPLBillSent (T_BILL_SENT_TB)
TPLCoverageCode (T_CVRG_CODE_TB)

This model is not intended to show any interactions that occur outside of this model.