HUMAN SERVICES DEPARTMENT

Medicaid Management Information System Replacement (MMISR) Project

Service Orchestration Management Plan

HSD Deliverable Owner: John Oliver

System Integrator (SI) Deliverable Owner: Dawn Gelle

Configuration Number: v1.7

09/11/2019

TurningPoint

TurningPoint Global Solutions
Software Services Company™

Section 508 Compliant

Table of Contents

List of Tables

List of Figures

1 Introduction

The Health and Human Services (HHS) 2020 Medicaid Management Information System Replacement (MMISR) project was initiated to enable the replacement of the existing Medicaid Management Information System (MMIS) with a new, modular enterprise. To achieve this, the system integration team will deploy a technology solution and collaborate with the New Mexico (NM) Human Services Department (HSD) to develop a framework for transforming the business processes, data management, and technology standards for HHS 2020. This framework will support the program's Medicaid Information Technology Architecture (MITA) maturity objectives.

Integration of the participating systems, modules, and technical services into the enterprise is a key component of MMISR. The System Integrator (SI) Contractor will engage with HHS 2020 stakeholders to coordinate, plan, and implement each Integration Project through the processes described in the Configuration and Continuous Integration Services (CCIS) plan. The planning, Software Development Life Cycle (SDLC), requirements gathering approach, MITA strategy, and MMIS Certification sections of this document are heavily content-dependent on the CCIS plan and are tailored to address orchestration management planning. Therefore, it is recommended that the audience of this document read it in conjunction with the CCIS plan for better context and effective reference.

This document is customized to meet service orchestration needs and provides insight into the design patterns and best practices for developing a Service-Oriented Architecture (SOA)-based solution that meets service orchestration requirements. Its various sections discuss the oversight that the SI Contractor will provide to other participating systems, and it details the dependency on other participating systems as well as dependency on different workstreams. Finally, it defines the process of maintaining the service catalog.

1.1 Overview

NM HSD has adopted the HHS 2020 vision, a transformational, enterprise-wide approach to the health and human services business. HHS 2020 will move service delivery from a program-centric approach to a person-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. HHS 2020 is technology-enabled, but includes rethinking organizational design, redesigning and streamlining business processes, and reducing barriers between organizations within the HHS enterprise.

Please see Section 1: Introduction in the Project Management Plan for a detailed MMISR project overview.

1.2 Goals of Service Orchestration

The MMISR project aims to improve system effectiveness and to comply with the Centers for Medicare and Medicaid Services' (CMS) guidelines, including MITA modularity standards. The MMISR project is part of HHS 2020 – an enterprise vision for transforming the way HHS services and programs are delivered to New Mexico's citizens. HHS 2020 is not limited to technology. It encompasses a re-evaluation of processes and organizational structures used to manage and deliver program services; efforts to work across organizational boundaries to more effectively manage and deliver all HHS

services in the State of NM; and the transition from current operating models to an outcomes-based focus. Through MMISR, NM HSD will implement the technology foundation for HHS 2020.

The HHS 2020 framework is intended to support multiple programs in the NM HHS enterprise that relate to or interact with the MMIS. The MMISR solution will be comprised of multiple modules procured through multiple contracts and service providers, and it will encompass both technology-based components and Business Process Outsourcing (BPO). Within the HHS 2020 framework, the project effort extends beyond the Medicaid Assistance Division (MAD) to incorporate a vision for NM HSD, changing the focus to a client-centered perspective and improved overall functional capacity.

1.2.1 Implementation of Modular System Architecture

CMS's MITA framework establishes the modularity standard to ensure that Medicaid technology investments provide the flexibility and extensibility necessary to support today's program needs. As well, adherence to the MITA framework ensures eligibility for enhanced Federal Financial Participation (FFP) funding. This condition requires the use of a modular and flexible approach to systems development, including the use of open interfaces and exposed Application Programming Interfaces (API), the separation of business rules from core programming, and the availability of business rules in both human and machine-readable formats.

The HHS 2020 SI solution enables a modular approach that is design-independent and allows for the flexibility to modify modules without extensive impact to other system modules. The SI solution employs an Oracle Fusion-based Enterprise Service Bus (ESB) Integration Platform (IP) that supports connections for interoperability among web services within the MMISR modules, Enterprise Shared Services (ESS), and external data sources. The SI solution provides multiple connectivity options, for example, web services – such as Representational State Transfer (REST) and Simple Object Access Protocol (SOAP) – messages queues, File Transfer Protocol (FTP), and others to enable easy system integration. These approaches all advance NM HSD's MITA maturity level goals for the enterprise.

Once the Modular System Architecture (MSA) – including the ESB and schema-agnostic MarkLogic Non-Structured Query Language (NoSQL) data solution – is established, the SI Contractor will integrate modules and services to support end-to-end business functionality through the SI framework, including ESS, legacy module integration, new module integration, and interfaces with external data trading partners. The workflows that employ these services are stitched together through service orchestration across all of the actors (services) that comprise a business scenario.

The HHS 2020 project supports the implementation of re-imagined business processes that standardize, automate, re-use, and de-duplicate services and data flows for MMISR business. The SI Contractor develops technical To-Be designs in collaboration with NM HSD and business owners. These designs leverage the automation and service re-use provided by the SI Platform's MSA within the context of the standards and governance framework established collaboratively by the SI Contractor and NM HSD.

1.2.2 Creation of Standards-based SOA and Governance Framework

The SI Contractor will collaborate with NM HSD to develop standards and governance processes for the Project Management, Business, Technology, and Data areas of the program, which will be used by all HHS 2020 module partners for successful integration into the SOA framework. This is completed through requirements and design reviews for new modules, education and training about standards and enterprise services, and APIs for service consumption and provisioning to align with HHS 2020 standards. The governance process includes ongoing monitoring to ensure conformance with enterprise standards.

Please refer to the PMO 12: Governance Standards – Technical and Architectural document for details about project standards.

1.3 Service Orchestration Definition

Service orchestration is the configuration of the ESB and the integration of services published by systems participating in the implementation of business workflows. The workflows may require one or more systems for service orchestration. The workflows are of the following three types:

- **SI Workflow**: An SI workflow is a workflow owned by the SI Contractor. In this case, service orchestration for a workflow is performed on the IP. The services can be published by any participating systems or the IP. The SI Contractor is responsible for the implementation of these types of workflows and ensuring that the implementation complies with the SI Contractor's published enterprise standards.
- **Module Workflow**: A module workflow is a workflow that is owned and executed within a given module. The modules can be any of the legacy or new modules related to the SI project. The module owner is responsible for implementing these types of workflows. The SI Contractor is responsible for providing standards and oversight at applicable stages of the SDLC.
- **Composite Workflow**: A composite workflow is a workflow that is partially executed within a module while the rest is delegated to the IP for execution, or vice versa. The SI Contractor is responsible for the implementation of a composite workflow that is implemented on the IP and has a responsibility to ensure that the implementation abides by the SI Contractor's published enterprise standards. The SI Contractor is responsible for providing standards and oversight at applicable stages of the SDLC for portions of the workflow not implemented by the SI Contractor.

Each Integration Project (that is, major business function or feature to be included in MMISR) will be decomposed into the services that will support that function. Service orchestration requests include two major functions:

- **Creation of Services (APIs)**. This supports the processing of other module contractors' services through the ESB.
- **Configuration of Services**. This involves stitching services together with services provided by other module contractors.

Service orchestration requests will be added to the Integration Backlog. These backlog items, stored in Jira, will contain Work Packages and Work Items required by each Integration Project.

As an Integration Project is selected to be worked on, the resulting service orchestration backlog items linked to that Integration Project will require a Level of Effort (LoE) work estimate to be

completed. This estimate then becomes part of the overall plan and schedule for that Integration Project.

The SI Contractor provides service orchestration as a continuous service throughout the life of the MMISR project. As such, orchestration management draws on the approach articulated in the PMO 37 - Configuration and Continuous Integration (CCIS) plan for all of the SI Contractor's continuous integration work and tailors that approach to service orchestration wherever appropriate.

1.3.1 Relationship with Other Workstreams

Service orchestration refers to the integration of services hosted by a participating system to complete an enterprise workflow.

Service orchestration is dependent on the following workstreams:

- **ESB:** The ESB stream is essential for standing up and configuring all of the software components of the SI Contractor's IP.
- **System Migration Repository (SMR):** This stream is essential to providing data feeds during the initialization of new modules. These data feeds may not be crucial for SI testing, but they are essential for User Acceptance Testing (UAT) and higher environments.
- **New Modules**: This stream is essential as new modules participate in service orchestration, as workflow implementation is not possible without new modules.
- **Legacy Modules**: This stream is essential as legacy modules participate in service orchestration. Throughout the project, there may be a need to add or modify existing services for service orchestration. Workflow implementation is not possible without these changes to legacy modules.
- **Interfaces**: This stream deals with all the interfaces that are not part of legacy and new modules. It provides services to integrate with State and federal partners like the Social Security Administration (SSA) and Managed Care Organizations (MCOs), among others. Because there are workflows that require the IP to interact with these partners, workflow implementation is not possible without standing up these interfaces.
- **ESS**: The stream deals with standing up such services as Identity and Access Management (IdAM), Master Data Management (MDM), Customer Communication Management (CCM), Enterprise Document Management (EDM), address standardization, the Business Rule Engine (BRE), and any other services that are needed for service orchestration. Shared services need to be ready for end-to-end service orchestration implementation, thus IdAM is essential to ensure that services are secured.

1.3.2 Relationship with Other Project Plans

This section outlines how the Orchestration Management Plan relates to other SI plan deliverables, and includes a matrix with a brief description of each. This plan may further refine processes or standards defined in another plan to address service orchestration needs, where applicable.

The following table lists the deliverables referenced within the Orchestration Management Plan:

Table 1: Related Deliverables

Deliverable ID	Deliverable Name	Impact on Interfaces Management Plan
PMO37	CCIS Plan	This plan is the critical for the MMISR project, as it provides guidance and direction for how module contractors and the SI work together to deliver a fully integrated product.
SIPLT88	Interface Management Plan	This plan defines the approach for developing interfaces for MMISR. Service orchestration will be heavily leveraged to process each interface.

1.4 Service Orchestration Management Plan – Maintenance and Update

The Service Orchestration Management Plan will be updated as design patterns and strategies evolve based on the needs of the integrating systems and technology upgrades. The Quality Assurance processes as outlined in PMO 13: Quality Management Plan necessitate a minimum annual end-to-end review.

1.5 Monthly Reporting

The status of service orchestration activities is provided in the consolidated Project Management Office (PMO) report, as described in the Orchestration Management Services report. Statuses and progress related to the following areas are listed in the monthly status report:

- Status of the SDLC phase and relevant activities.
- Integration Project status:
    - Provide status of work packages from Jira.
- Activities and milestones for the previous and current months across Integration Projects.
- Risks and issues across Integration Projects.

1.6 Roles and Responsibilities

The following table outlines the roles and responsibilities for service orchestration.

Table 2: Roles and Responsibilities for Service Orchestration

Role	Responsibility
Business Analyst (BA)	Business Analysts are responsible for the following activities:

Role	Responsibility
(Both HSD and Vendor)	• Participating in Joint Application Requirements (JAR) and Joint Application Design (JAD) sessions to ensure that service orchestration requirements are captured. • Entering service orchestration requirements into Jama and ensuring traceability.
Developer (SI)	Developers are responsible for the following activities: • Implementing service orchestration work packages and work Items as assigned. • Testing service orchestration work packages and work items as assigned and ensuring the work packages meet the requirements as captured in Jama.
System Integrator	The SI is responsible for the following activities: • Providing estimates for service orchestration work packages. • Testing Integration Projects that require service orchestration. • Integrating other module vendor's services that require service orchestration within an Integration Project. • Monitoring the Integration Backlog in regards to service orchestration work packages.
Testers	Testers are responsible for the following activities: • Writing test cases that support service orchestration activities. • Executing test cases that validate the developed functions.

2 Orchestration Management Plan Approach

The following sections outline the approach to orchestration management.

2.1 Planning

The following sections outline the sequential steps involved in ensuring the successful accomplishment of orchestration management.

2.1.1 Adding to the Integration Backlog

Work Packages related to service orchestration will be added to the Integration Project Backlog. In developing these work packages, the SI Contractor and NM HSD will discuss each workflow and the following set of prioritization factors to update the work packages once the To-Be business workflows are identified. These factors will include:

1. Business stakeholders' feedback on the prioritization criteria for feature implementation.
2. Module readiness and on-boarding timeframe/schedule.
3. Validation of the readiness of the infrastructure required to implement the orchestration. The emphasis here is the integration and user acceptance testing environments, where all the systems participating in workflow execution need to interact seamlessly for end-to-end testing.
4. Level of difficulty – How many dependencies does the workflow require?
5. Level of effort – How much time or resources are required to complete the workflow?

The SI Contractor's implementation manager coordinates with the PMO for backlog grooming, their participation in this analysis exercise, and related planning and requirements activities. The prioritized requirements follow the SDLC process for end-to-end implementation.

2.1.2 Work Breakdown Structures

A Work Breakdown Structure (WBS) for each workflow will be developed in collaboration with the integrating modules and interfaces system contractors and business owners. This WBS will ensure that activities are coordinated among these stakeholders throughout the SDLC. As part of this process, the SI Contractor will create an inventory of current workflows representing NM HSD business processes. These business processes are referred to as As-Is workflows.

Any WBS for orchestration services involves development of technical To-Be workflows from As-Is workflows in coordination with new module contractors and legacy module, ESS, and interface owners. Each To-Be workflow is broken down to identify the work responsibilities of different participating systems as well as the IP. Each workflow is subsequently converted into logical tasks that can be assigned to the contractors of each participating system. As a part of the WBS task, dependencies are identified. Work Packages for orchestration services will be fed into the associated Integration Project. As additional modules or related HHS 2020 initiatives are implemented, the SI Contractor will review the approved HHS 2020 WBS to ensure that any appropriate changes or updates are incorporated.

Service orchestration Work Packages (describing how module services utilize the ESB) will be maintained in the Integration Project backlog. Once an Integration Project is scheduled, service orchestrations Work Packages are then worked through the traditional SDLC.

2.1.3 Master Schedule Management

The service orchestration schedule is developed in MS Project in conjunction with the Enterprise Project Management Office (ePMO). The resulting tasks, owners, and due dates are captured in Jira. To ensure enterprise-level visibility across stakeholders and projects for the program, the SI Contractor develops a proposed schedule for implementing new orchestrations with new module contractors, legacy module contractors, interface trading partners, and other stakeholders who are integrating into MMISR. This schedule provides the necessary inputs for the ePMO's portfolio view and is reconciled in the Enterprise Project Schedule (EPS).

A Work Package will be defined for each orchestration request and broken down into the individual Work Items necessary for the module to be implemented.

As part of the schedule management approach, various tasks and dependencies for orchestration services are identified in the orchestration services WBS. Other dependencies include availability of infrastructure, availability of the integration environment, and services from the integrating partners.

The Work Package is managed in Jira and system functionality releases are managed in the EPS. The SI Contractor team will work with the owners and contractors of systems and services included in the workflow to assign Jira items as tasks. Managing these tasks through to their completion is tracked through Jira.

The following is provided as a hypothetical example – Service Orchestration Work Package #1:

- Task #1 is the creation of the service which selects the data to be sent.
- Task #2 is the creation of the service which delivers data to a data store.
- Completion of Work Package #1 is dependent on completion of Task #1 and Task #2; it may also be dependent on other SI tasks, notably availability of the ESB and SMR. Both Tasks #1 and #2 can be worked independently, but it is not until Work Package #1 is completed that the capability can be considered available and ready for production.

2.1.4 Enterprise Governance Planning and Implementation

Implementation of service orchestration Work Packages will be compliant with business, information, and technical standards published by multiple governing bodies, including the Business Transformation Council (BTC), Data Governance Council (DGC), and Architecture Review Board (ARB).

To implement new service orchestration standards, the SI Contractor team will collaborate with the BTC on an ongoing basis to plan for the adoption of new standards as new techniques are defined. The SI Contractor Team's Functional/Business Manager will be the liaison between the SI Contractor and the BTC to highlight changes that impact the business in order to evaluate changes across business units and to plan for appropriate responses to those changes.

For service orchestration, the following technology standards are considered:

- **Infrastructure Standards**
  - Virtualization
- **Interfacing and Interoperability Standards**
  - SOA-centric standards and frameworks/patterns
  - ESB and module integration strategies involving ESB-driven patterns/best practices
  - Health Level 7 (HL7)
  - Web services using SOAP and REST

  - JavaScript Object Notation (JSON) as a data format specification for non-eXtensible Markup Language (XML) usage
  - XML as the default data format specification
  - XML schema and namespaces for interface definitions
  - Workflow engines driven by Business Process Expression Language (BPEL) and Business Process Management Notation (BPMN)
- **Security Standards**
  - Federal Information Security Management Act (FISMA)
  - Single Sign-on (SSO) and Security Assertion Markup Language (SAML) token-based security
  - Lightweight Directory Access Protocol (LDAP)
  - Minimum Acceptable Risk Standards for Exchanges (MARS-E)
  - National Institute of Standards and Technology (NIST)

To implement the data standards, the SI Contractor will collaborate with the ARB and DGC to plan for adoption of the technology standards.

2.2 SDLC Execution for Orchestration Workflows

From an orchestration management standpoint, SDLC implies establishing a standardized approach for the identification, management, and implementation of workflow requirements. Service orchestration aims to facilitate the integration of services published by multiple systems across multiple work streams to complete workflows representing state business processes.

The service orchestration implementation is a two-step process:

- Defining requirements for technical To-Be workflows.
- Implementing the technical To-Be workflow requirements.

2.2.1 Defining Requirements for Technical To-Be Workflows

The requirements gathering for service orchestration is comprised of, first, understanding and documenting current workflows (As-Is workflows), and, then, designing modified workflows (To-Be workflows) based upon the input of NM HSD business owners, system contractors, and the HSD project governance.

2.2.2 Implementation of Technical To-Be Workflow Requirements

The technical To-Be workflow requirements are the basic system touchpoints requiring design and development. Each technical To-Be workflow defines the services that need to be integrated for workflow automation. They are represented as requirements in Jama and as an individual task for the SI Contractor and other contributors in Jira.

Service orchestration Work Packages are identified and prioritized for implementation in iterations. Using this approach may take multiple iterations to complete an entire Work Package implementation.

The Work Items that make up the service orchestration work packages are managed through the iterative waterfall SDLC. Additionally, the detailed schedule of activity for completing the integration of services for a work package may be not be defined at the outset of the work package, as the exact timing of the availability of external dependencies on the modules and services may be unknown.

Therefore, it is necessary to continuously refine and manage the expectations for completing the integration of all the modules and services of a work package through the iterative integration process.

2.3 Requirements Gathering Approach

The objective of service orchestration is to implement technical workflows through the IP in order to access business modules and services that enable streamlining and automation of business objectives. The IP provides centralized access to ESS, legacy modules, new modules, and external Interfaces through the SOA framework. It also facilitates integration into the IP by applying enterprise technical, data, and business architecture standards established by enterprise governance bodies like the DGC, BTC, and ARB.

Requirements management will follow the processes outlined in PMO 15: Requirements Management Plan.

2.3.1 To-Be Technical Requirements

The SI Contractor analyzes As-Is business workflows and To-Be business process recommendations from the BTC in order to propose technical To-Be requirements that leverage the modules and services of the HHS 2020 road map. Taken together these support HHS 2020 goals for achieving MITA maturity levels for those business processes and optimizing the potential for re-use and increasing enterprise efficiency.

The SI Contractor team then proposes a technical To-Be workflow, which is vetted through the HSD governance for its impact on multiple business owner stakeholders, and further analyzed for impact to business process re-engineering, organizational change management needs, and training.

For technical workflows to be implemented within the HHS 2020 enterprise services and modules, the service orchestration team – a matrixed team of Business Analysts (BAs), engineering, and data resources – collaborates with HHS 2020 business and system owners to elicit lower-level requirements and design specifications that support To-Be business processes through the IP. The lower-level requirements are derived by further elaboration of the high-level To-Be workflow to identify the technical services required for end-to-end implementation.

There are two kinds of business services that technical workflows enable:

- Existing services from legacy modules and interfaces.
- Services that need to be developed, such as new module Commercial off-the-shelf (COTS) offerings and the services they provide that can be exposed on the ESB.

It is important to identify any existing services that can be reused in a workflow to reduce implementation costs and time to delivery.

Business services that do not yet exist require the engineering, data, and business team members to analyze their high level To-Be requirements. Recommendations for technical services are based on adherence to the system design as outlined in SIPLT 1: System Design Document and enterprise governance standards.

Lower-level requirements for the enterprise workflow lists services that need to be provided by each integrating system in order to complete a workflow. These requirements identify the existing technical services and those services that need to be developed for workflow execution. The SI

Contractor conducts JAR sessions with system owners, NM HSD business owners, and various governance bodies to reach agreed upon lower-level requirements.

Please refer to PMO 15: Requirements Management Plan for a detailed description of stakeholder identification and engagement, along with the complete requirements management process.

2.4 Design Approach

As part of the design effort, the To-Be lower level technical requirements are converted to designs depicting services and inter-service integrations that constitute an orchestrated technical workflow. The data format of the message exchanges and communication protocols between the participating systems are depicted in the workflow.

The SI Contractor will conduct multiple JAD sessions with stakeholders to discuss the services and service orchestrations needed to execute a workflow. The documentation of the To-Be design involves detailing the technical services and their interactions for end-to-end workflow execution. The design needs to address security, auditing, and logging, as well as Service-Level Agreement (SLA) monitoring in compliance with enterprise design standards.

The service orchestration design leverages the following features from the ESB for seamless integration of services and workflow implementation. These design features are also explained in detail in Section 1.2 of the System Design Document:

- **Service Mediation**: The mediation layer primarily facilitates communication across different services. An important aspect of this is that the mediation layer makes services independent of each other so that even if a particular service is replaced or removed, the other services can seamlessly interact with new services. The Oracle Service Bus (OSB) acts as an ideal software component for facilitating service mediation, as it provides different protocol connectors “out of the box.”
- **Service Versioning**: A service may change over a period of time to accommodate new functional requirements originating from new integration needs. If an older integration service cannot be modified to consume the new service, it is imperative that the older and newer versions of the service coexist. Service versioning helps achieve this objective. Service versioning is a standard web services concept and is supported by Oracle SOA components.
- **Message Transformation**: This relates to the transformation of messages into different formats when integrating different services and modules. Typically, interfaces and operations of disparate services are not identical and the message from the source needs to be transformed into a format that can be accepted by the target. For example, this occurs when converting a native message format from a source module to a canonical message format. Oracle OSB provides functionality to configure message transformation rules to facilitate message transformation.
- **SLAs**: SLAs are a commitment between a service provider and a client. Particular aspects of the service, such as quality, availability, and responsibilities are agreed on between the service provider and the service user. The SLAs are measured using Key Performance Indicators (KPIs) that are defined during the requirement and design phases. There can be two kinds of SLAs:
    - **Business SLAs**: SLAs for completion of an end-to-end workflow.
    - **Technical SLAs**: SLAs for execution of a standalone service.

To measure SLAs, the Oracle BPM suite allows for the creation of sampling points for a project or a particular process. These data can be passed to a Business Activity Monitor (BAM) dashboard using data objects.

- **Audit and Logging**: Auditing provides the capability to track service calls and data as received from, or sent to, integrating partners. Logging is required for the operation team to address operation issues. It can provide analytical data as well as data regarding process performance. Oracle BPM tracks all workflow executions and provides a graphic display of each activity performed for the execution of a workflow. Oracle BPM supports multiple levels of logging that can be configured during installation of the software. Post installation, the level of logging can be modified using Oracle fusion middleware controls or the WebLogic Scripting Tool (WLST).
- **Exception Handling**: Allows service components to handle error messages or other exceptions. It generates error messages in response to business or runtime faults, and defines how exceptions will be communicated back to integrating systems.

  Oracle BPM provides advanced error handling and recovery that provides the following features.
  - **Force Commit After Execution:** To avoid re-executing non-idempotent activities.
  - **Skip and Back Error Recovery:** A declarative feature for choosing whether to re-execute a faulted flow object or just skip it and move to the next flow object as defined in the process flow.
  - **Fault Policy Editor:** A graphical editor for creating fault policies**.**

The design should follow the SOA guidelines for implementation of orchestration services:

- **Abstraction via Use of Policies:** Policies hide implementation details and constraints of services from outside service clients. Details of service implementation are hidden completely from all service clients. Clients operating in a .NET environment that may be restricted to only using the Hyper Text Transfer Protocol (HTTP) protocol should not be prevented from interacting with services operating in a Java Message Service (JMS) stack. Services that can use a particular encryption/decryption standard, or that use SOAP versus JSON messages, should still be able to interact. The ESB stores, manages, and enforces all service policies.
- **Boundaries Defined by Contracts:** Contracts describe the purpose of a service interface. They also serve the critical goal of decoupling services by eliminating legacy integration methods of sharing APIs, classes or memory references – Remote Procedure Calls (RPC), Remote Method Invocation (RMI), and/or remote Java/Oracle Database Connections (JDBC/ODBC). This provides decoupled software assets that make no assumptions about each other but rather interact based on what the services advertise about themselves.
- **Loose Coupling:** Loose coupling ensures that service integration eliminates to the highest degree possible the interdependencies between services. Taken to the maximum extent possible, services can become autonomous and able to defend their integrity in the face of unforeseen technical and business events. Loose coupling involves intentional focus on the following:
  - **Asynchronous versus Synchronous Communications**: Asynchronous messaging between services helps ensure services are independent from one another and are stateless and idempotent where possible, while synchronous messages wait for a response before continuing.
  - **Location Independence**: Service invocations should be identified at run-time, not when the software is compiled. This ensures that the physical deployment, replication, and scaling services occur via a virtual connection instead of a specific IP address, memory address, or Uniform Resource Locator (URI).

  - **Version Independence**: Services are envisaged to change over a period of time as they take on more functional responsibilities. It is therefore imperative that existing interfaces change as little as possible and that they support backward-compatibility with existing APIs.
  - **Schema Adherence**: Services should be invoked and should reply to requests with DGC-approved shared schemas. This ensures implementation details are hidden from other client services.
- **Autonomy:** Taken as an ultimate goal of service design, autonomy ensures that a service is resilient and available regardless of the other services they may use. This also ensures that the underlying mechanisms used within a service may be changed without causing any changes in its clients. Because of the run-time binding employed by SOA, as implemented through an ESB, services should be able to evolve in real time without causing any failures to clients, as long as their contracts and policies remain the same.
- **Message-Encapsulated, Document-Based Integration:** This kind of integration relies on passing data encapsulated in documents between services, usually referred to as messages. A document in the context of SOA is a rigorously structured human-readable file of name-value pairs encoded as XML or JSON files. As opposed to sharing classes between tightly coupled APIs, messages enable data to be shared via decoupled interfaces. Messages are managed via an ESB that helps ensure decoupling as it manages contracts and policies that enable message flow.
- **Reusability:** Good service design ensures that the long-term evolution of service capabilities are architected and designed into the service from the start. This applies to both services as well as messages. Even if the future capabilities are not yet utilized, they should be designed into the service to alleviate ripple effects when those capabilities are released. This is part of loose coupling and autonomy in an SOA ecosystem.
- **Stateless Processing**: This supports idempotency (that is, produces the same results every time) and reduces coupling.
- **Composability:** This supports the orchestrations of coarser-grained services from finer-grained services.

The design process will leverage existing service orchestration patterns to ensure better design. These design patterns are detailed in Section 1.2.5.5 of the System Design Document.

2.5 Development Approach

The development approach for service orchestration is the same as described in the Development Management Plan. There are no specific exceptions or additions for service orchestration work.

2.6 Testing Approach

Testing is essential to ensuring that all the functional and non-functional requirements for service orchestration are met by the developed solutions. All the phases of testing – the environments, the stakeholders, and the tools used – are discussed in detail in PMO 14 Test Management Plan.

The sections below reflect multiple scenarios pertaining to service orchestration testing:

- **System Testing in Absence of Services**. At the start of the implementation of the SI solution, services may not exist that are required to complete a workflow execution. In the absence of service endpoints, either from the source system or the target system, mock services will be created to simulate these systems. These mock services will be invoked to perform system

testing. SoapUI, an API test tool, will be used for mock creation and automation of web service-based services. As well, custom tools will be developed to test services that are not web services-based, and data will be prepared based on design documentation and interface control documents for use in testing positive and negative test cases.

- **Integration Testing**. Integration testing requires services to be available to test service orchestration. Because of the staggered nature of the implementation timeline for services across multiple systems and contractors, integration testing integrates with the services that are already in place and also uses mock services that have been developed as a part of system testing for testing service orchestration.

  One aspect of the integration-testing environment is that it may not have synchronized data. For example, the claims system testing environment may be trying to process a claim for a client that may not exist in the client management system. To manage this, the SI Contractor testing team will coordinate with the integrating system contractors to ensure that a testing data sample is synched up between the participating systems during integration testing.

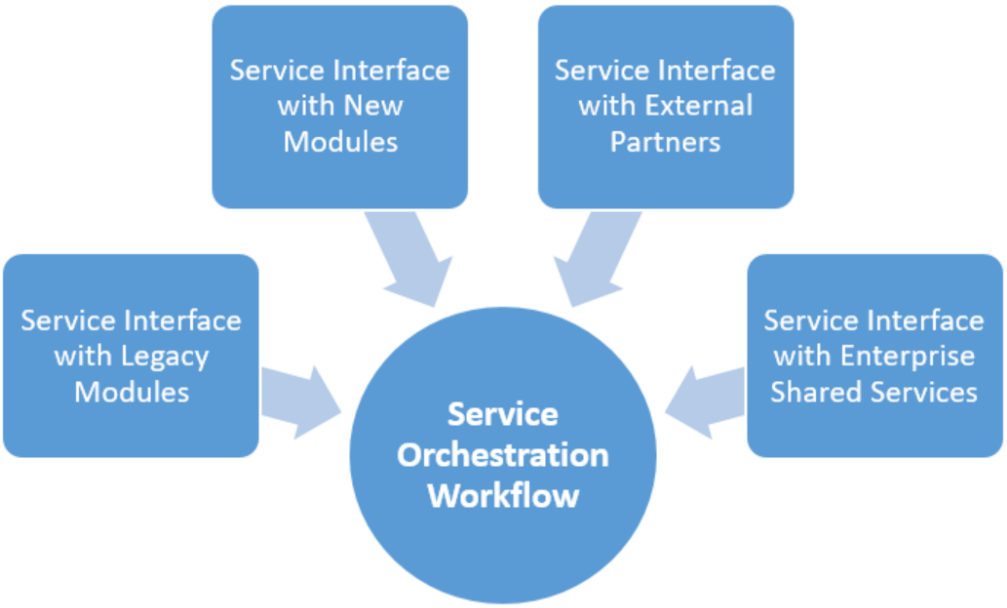
  Before starting integration testing, the SI Contractor test team will identify stakeholders for systems that will integrate with the SI solution. The team will provide the stakeholders with the test scenarios and test cases that will be executed during integration testing. The participating teams will create test data for these test cases which will be used during test execution. The SI Contract or test team will also coordinate with the module's Database Administrator (DBA) for test data maintenance (that is, backup, restore, and deletion) so that the test data can be reused for future cycles of testing. The testers will verify and validate requests and responses while performing service orchestration testing.

- **User Acceptance Testing**. The UAT is conducted once all of the participating systems are done with their implementation and integration testing is completed. Prior to UAT, the SI Contractor Testing Team will identify stakeholders for modules (test teams, data teams, and development teams) who will participate in UAT. The SI Contractor Testing Team will support UAT business owners in the creation of user test cases and test scripts for execution. They will also coordinate with the module's DBA for database backup so that the test data can be reused for future cycles. During UAT, the test teams of all the participating systems will engage in end-to-end testing of the business process-driven workflows.

2.7 Implementation Approach

Service orchestration focuses on stitching together several services, interfaces, and enterprise shared services into a technical workflow that supports business processes. The following figure illustrates the relationship between the individual services and how they form a cohesive workflow as part of a Work Package implementation.

Figure 1: Service Orchestration Workflow



The deployment procedure varies depending on the deployment environment. The lower testing environments need deployment of stubs and test scripts to automate testing, but they may not require a very comprehensive backup and rollback strategy. On the other hand, the higher environments may not need mock data, but may require a comprehensive backup and a rollback plan. This ensures that deployments can be rolled back in case of any last minute glitches in the production environment during new release deployment.

The SI Contractor DevOps team will be responsible for deploying the build for the IP in all seven environments, (that is, Development, Quality Assurance/System Integration Testing (QAT/SIT), UAT/Performance, Patch, Production Support, Production, and Disaster Recovery (DR)). The contractors of participating systems will participate in deployment planning using an integrated communication and engagement strategy. The SI Contractor will coordinate with the participating systems contractors for deployment activities and sanity testing post-deployment.

3 CMS Certification

This section documents the approach to Centers for Medicare and Medicaid Services (CMS) Certification related to this deliverable. The documented processes are followed, and the changes will be documented and tracked throughout the entire project life cycle. This deliverable will be reviewed by CMS during the following Medicaid Enterprise Certification Life Cycle (MECL) reviews:

- R2, Operational Milestone Review
- R3, CMS Certification Final Review

This deliverable may also be reviewed by CMS during informal reviews, including Consults and Gate Reviews.

Appendix E: MECT Checklist and Programmatic CSF contains the MECT and Critical Success Factor items that are attributable this deliverable.

The Certification Process Guide contains detailed information regarding the CMS Certification approach.

Critical Success Factors (CSFs)

The MECL incorporates critical success factors (CSFs) into the certification process. There are two types of CSFs—programmatic and functional. Programmatic CSFs identify activities the state PMO will need to perform in managing its MMIS project. They are found in the Programmatic Tab of the IV&V Progress Report Template, which the IV&V contractor fills out as part of the regular progress reports.

4 Applicable Standards

PMO 12 – The Governance Standards – Technical and Architectural document describes the standards applicable across the entire project; Link is provided here. There are no other specific standards identified associated solely with Orchestration Management.

5 Assumptions / Constraints / Risks

The following sections provide an initial set of assumptions, constraints, and risks appropriate for this deliverable.

5.1 Assumptions

The assumptions listed below may be updated over time:

1. The subject matter experts from integrating modules contractors and interface contractors, as well as the NM HSD team, will be available to the SI Contractor at various stages of the SDLC.
2. External interface partners and associated contractors will be available to support Integration projects.

5.2 Constraints

The constraints listed below may be updated over time:

1. **Data Repository and Distribution Requirements:** When the vendor needs to share with or accept data from any other service, they must convert their private schema to a shared schema as enforced by the Integration Platform. These messages and the schemas they contain will conform to HHS 2020 Technology Architecture standards and guidelines.
2. **Autonomous and Decoupled System Assets:** All vendor services that are exposed for sharing throughout the HHS 2020 ecosystem must be decoupled from other services, and must be autonomous. Autonomy will be partially guaranteed by each service managing its own internal persistence, as opposed to using shared data stores, as this creates coupling and interdependencies that hinder service autonomy.
3. **Shared SOA Capabilities:** Contractors will share the same SOA infrastructure, which may be hosted in a completely different environment, for sharing and integration via SOA. Contractors will be encouraged to leverage the same SOA tools that other systems/projects within HHS 2020 will leverage. These may include a common BPM tool/platform, a common workflow tool/platform, a common Business Rules Engine, as well as a common XML mapping tool.
4. **Contractors will use the same security services hosted in the IP:** These will provide consistent Single Sign-On (SSO) capabilities as well as services for securing data in transit such as encryption/decryption, compression/decompression, monitoring and auditing.

5.3 Risks

The risks listed below are managed in SharePoint:

1. Availability of NM HSD personnel, new module contractors, and business partners may cause re-prioritization of work and delay deployment.
2. If new module procurement is delayed, then implementation of workflows that include those modules cannot be completed in a timely manner.
3. If a Work Package requires a module or data trading partner to implement a change in order to complete the service integration and that partner has a release schedule that does not align with the SI release schedule, then the service integration implementation may be delayed.
4. **Technology Risks:** Each of the interfacing partners and agencies are constrained by their native technology stack, which will partially or completely restrict any interfacing with them. This risk will prevent consolidation of the interfaces to a handful of protocols, toolsets, and frameworks.
5. **Budgetary Risks:** While the IP is standards-driven and can integrate with any external partners supporting these standards, there is a risk of some partners requiring legacy or custom protocols, thereby increasing the budgetary requirements for the Integration work streams.
6. **Quality Risks:** Legacy interfaces may not be adequately documented. If this is the case, as new modules come on line, there may be issues with message interactions and the data in these messages.
7. **Testing Risks:** Legacy module partners may sometimes not be able to provide a testing/integration environment, leading to the inability to test the integration thoroughly before promoting it to the Production environment.

8. **Schedule Risks:** Integration work stream activities are directly dependent on the rollout of the SI platform and the new modules.
9. **Security Risks:** Though the IP recommends security standards, protocols, and practices, some MMISR business partners may not be able to support these standards and instead may support only a predecessor/deprecated version. For example, some interface partners may only support Secure Sockets Layer (SSL), while the IP supports the successor to SSL, which is Transport Layer Security (TLS). Deprecated versions of protocols will lead to such security risks as Man-in-the-Middle-Attacks (MITM).

6 Requirements Traceability

Service Orchestration projects manage requirements discovery, documentation, and traceability as prescribed in PMO 15: Requirements Management Plan and PMO 16: Requirements Traceability Matrix.

7 Appendices

7.1 Appendix A: Deliverable Record of Changes

The following table provides the Deliverable Record of Changes:

Table 3: Deliverable Record of Changes

Version Number	Date	Author / Owner	Description of Change
0.1	5/7/2018	Ashish Kumar	Initial draft
0.2	2/15/2019	Tom Costa	Updates after internal TP discussion
1.0	3/06/2019	Tom Costa	Updates after discussions with HSD
1.0	3/28/2019	Tom Costa	Re-delivered Final-Draft after modification due to expected contract modifications regarding SI responsibilities.
1.1	4/29/2019	Tom Costa	Re-delivered Final-Draft after responding to all NM HSD comments
1.2	7/15/2019	Henry Huston/ Dawn Gelle	Edited for clarity in relationship to iteration S.O.P. and added use case addendum.
1.3	8/22/2019	Dawn Gelle	Updated per HSD comments
1.4	8/26/2019	Dawn Gelle	Finished updates to all Comments
1.5	8/27/2019	Linda Frankish	QA Review
1.6	8/27/2019	Dawn Gelle	Review prior to posting for HSD
1.7	9/11/2019	Linda Frankish	Final Review

7.2 Appendix B: Service Orchestration Use Cases

The following sections provide explanations of types of service orchestration use cases:

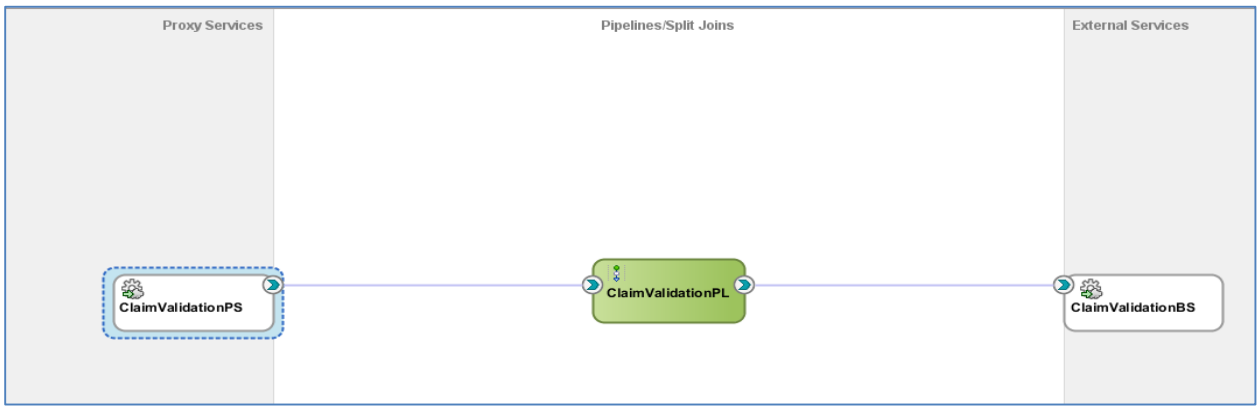
- **Fine-grained Service**: Smaller, atomic, and reusable services of which larger ones are composed. Usually, these services perform a subset of tasks in a complex workflow.
- **Coarse-grained Service**: Larger components than fine-grained, they simply wrap one or more fine-grained services together into a more coarse-grained operation. Usually, these services represent an entire business workflow. By definition, a composite service (that is, the single combination of three services) would be considered coarser-grained than its components.

7.2.1 Fine-Grained Service

The example of a fine-grained service in Figure 2 shows an incoming claim that is validated and transformed before it is sent forward for further processing. Claim validation is an atomic service that can be reused in several business workflows; hence, this service qualifies as a fine-grained service. This service is built on top of an Oracle Fusion Middleware (OFMW) component, the OSB. An OSB includes:

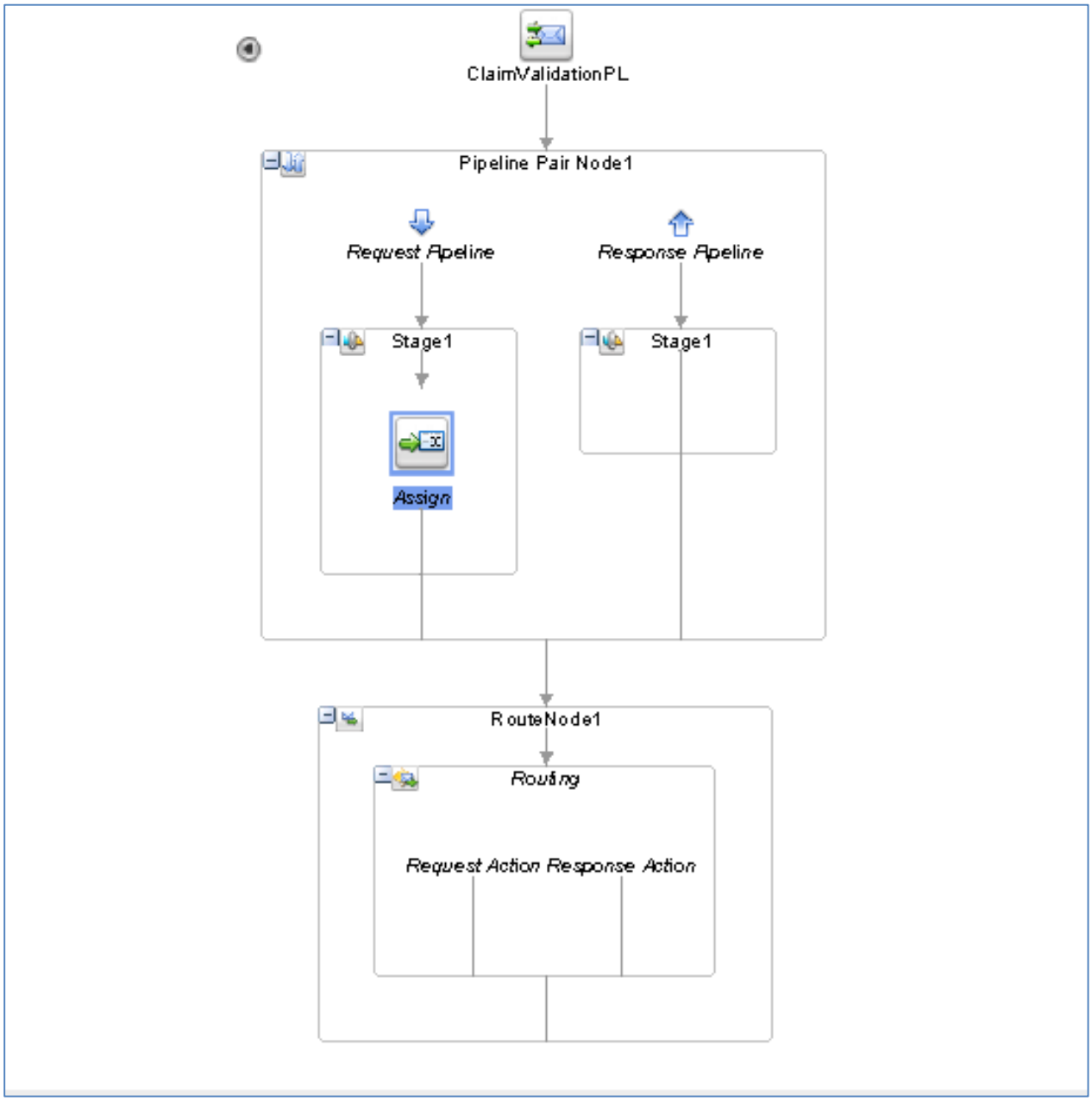
- **Proxy Services:** These services provide the intermediary interface needed to access back-end services through the OSB. The proxy services facilitate sending messages between service clients and business services, using Web Services Description Language (WSDL) or Web Application Definition Language (WADL). The services also define the type of communication the interface uses, the transport type, the transport settings, and the security type. The service then utilizes a pipeline, or message flow, to transform and route messages to one or more business services.
- **Pipeline:** The pipeline dictates how messages are controlled as they flow through a service bus. If a proxy service is based on a WSDL document, the configuration includes a WSDL port or binding. If the proxy service is REST-based, the configuration includes the WADL. The components of a pipeline define the logic for routing and manipulating messages.
- **Business Services:** These are service bus definitions that the Enterprise system uses to exchange messages while conducting a business process. A service bus configuration includes the service type interface and the configuration and transport type needed to connect with service producers, security requirements, message handling, performance tuning, and SLA alert rules. A business service also specifies the endpoint Uniform Resource Identifier (URI), or multiple endpoints, for load balancing and high availability.

Figure 2: Fine-Grained Service



Claim Validation Pipeline Operation: This simple pipeline operation involves transforming/manipulating the incoming message into the canonical model and forwarding the message to a separate validation service for further processing.

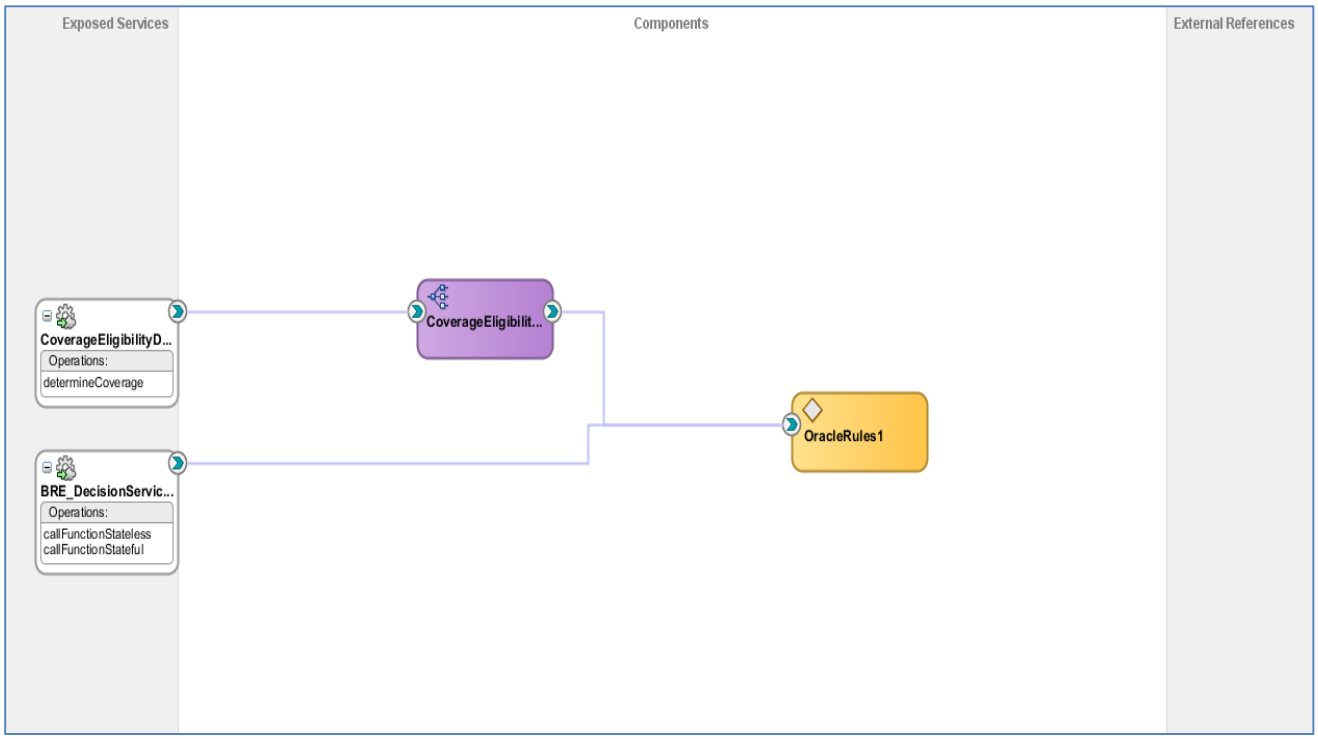
Figure 3: Simple Pipeline Operation



7.2.2 Coarse-Grained Service

The coarse-grained service example in Figure 4 is a composite service used to check coverage eligibility based on several parameters. The composite service is built using the Oracle SOA Suite. The Oracle SOA Suite enables services to be created, managed, and orchestrated into composite applications and business processes. Composites enable easy assembly of multiple technology components into one SOA composite application.

Figure 4: Coarse-Grained Service



Exposed Services: This swim lane provides an entry point to the SOA composite application, which are usually web services or Java Connector Architecture (JCA) adaptors. In this case, the coverage eligibility is exposed as two SOAP web services. The first service is orchestrated using a Mediator component, while the second service directly exposes Oracle BRE rules as a web service.

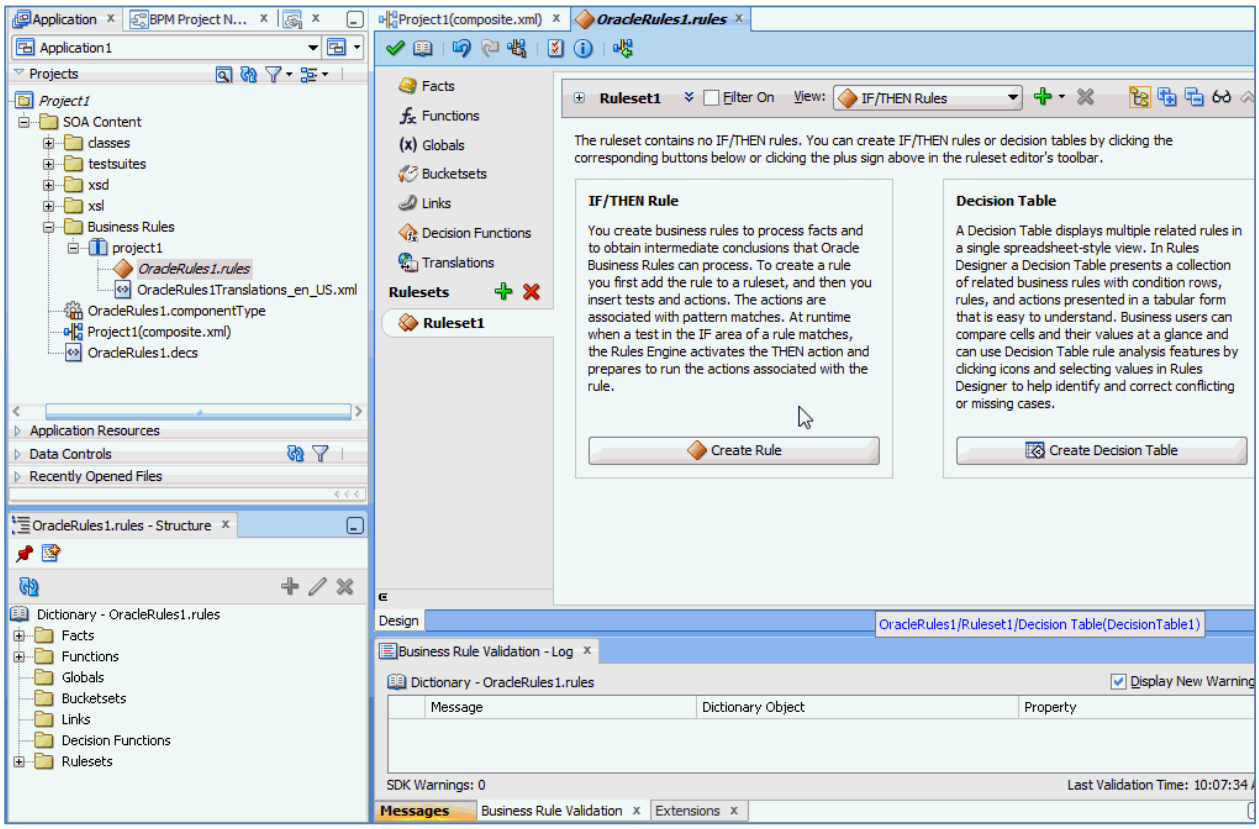
Components: The components used in this application are:

1. **Oracle Mediator:** This provides the framework needed to mediate between various components within a composite application. The Oracle Mediator transforms the incoming web service message fields to business rules specific elements. Other applicable benefits in using the Oracle Mediator, in this case, are:
    - It supports synchronous interaction, which happens when a client requests a service and waits for a response to their request. It also provides an asynchronous interaction which happens when a client requests a service but does not wait for a response to their request. A timeout period for an asynchronous interaction can be specified to perform an action, such as to raise an event or start a process.

- It supports the validation of the incoming message payload by using an XSD file.
- It supports fault policy-based actions, which consist of conditions and actions as well as manual error handling. A fault policy condition specifies the action to be carried out for a particular error condition.

2. **Oracle Rules:** These are business rule service components, which are also referred to as decision components. Business rules may be added as part of an SOA composite application or as part of a Business Process Execution Language (BPEL) process.

Figure 5: Oracle Rules



7.3 Appendix C: List of Acronyms

The following table provides a list of acronyms used throughout this deliverable:

Table 4: List of Acronyms

Acronym	Definition
API	Application Programming Interfaces
ARB	Architecture Review Board
BA	Business Analyst
BAM	Business Activity Monitor
BPEL	Business Process Expression Language
BPMN	Business Process Management Notation
BPO	Business Process Outsourcing
BRE	Business Rule Engine
BTC	Business Transformation Council
CCIS	Configuration and Continuous Integration Services
CCM	Customer Communication Management
CMS	Centers for Medicare and Medicaid Services
COTS	Commercial off-the-shelf
CSF	Critical Success Factor
DBA	Database Administrator
DGC	Data Governance Council
DR	Disaster Recovery
EDM	Enterprise Document Management
ESB	Enterprise Service Bus
ESS	Enterprise Shared Services
FFP	Federal Financial Participation
FISMA	Federal Information Security Management Act
FTP	File Transfer Protocol
HHS	Health and Human Services
HL7	Health Level 7 (HL7)

Acronym	Definition
HSD	Human Services Department
HTTP	Hyper Text Transfer Protocol
IdAM	Identity and Access Management
IMS	Integrated Master Schedule
IP	Integration Platform
IV&V	Independent Verification and Validation
JAD	Joint Application Design
JAR	Joint Application Requirements
JCA	Java Connector Architecture
JDBC	Java Database Connections
JMS	Java Message Service
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LDAP	Lightweight Directory Access Protocol
LoE	Level of Effort
MAD	Medicaid Assistance Division
MCO	Managed Care Organization
MDM	Master Data Management
MECL	Medicaid Enterprise Certification Life Cycle
MECT	Medicaid Enterprise Certification Toolkit
MITA	Medicaid Information Technology Architecture
MITM	Man-In-the-Middle-Attack
MMIS	Medicaid Management Information System
MMISR	Medicaid Management Information System Replacement
MSA	Modular System Architecture
NM	New Mexico
NoSQL	Non-Structured Query Language
ODBC	Oracle Database Connections

Acronym	Definition
OFMW	Oracle Fusion Middleware
OSB	Oracle Service Bus
PMO	Project Management Office
QAT	Quality Assurance Testing
REST	Representational State Transfer
RMI	Remote Method Invocation
RPC	Remote Procedure Calls
SAML	Security Assertion Markup Language
SDLC	Software Development Life Cycle
SI	System Integrator
SIT	System Integration Testing
SLA	Service-Level Agreement
SMR	System Migration Repository
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SRC	System Review Criteria
SSA	Social Security Administration
SSL	Secure Sockets Layer
SSO	Single Sign-on
TLS	Transport Layer Security
UAT	User Acceptance Testing
URI	Uniform Resource Locator
WADL	Web Application Definition Language
WBS	Work Breakdown Structure
WLST	WebLogic Scripting Tool
WSDL	Web Services Description Language
XML	eXtensible Markup Language

7.4 Appendix D: Glossary

The SI project employs many terms that have particular meaning to this project. A complete Glossary of Terms document is maintained in SharePoint at the following link:

REDACTED DUE TO SECURITY CONCERNS

The following sample of terms will provide the reader with a contextual basis for this document:

Table 5: Glossary

Term	Definition
Configuration	The way a system is put together; a specific set and arrangement of internal and external components, including hardware, software, and devices.
Integration Lifecycle	A methodology employing an iterative waterfall approach to completing integration of modules, services, and workflows through the SI solution.
Integration	Integration is the activity to enable a module or service or Interface to be accessed and available through the IP.
Integration Project	A logical grouping of Work Items across different work streams that are combined to meet a business objective.
Interface	An Interface is a shared boundary across which two or more separate components of an enterprise system exchange information; for the SI project, the term Interfaces refers only to those data trading partners that are external to the HHS 2020 enterprise.
Module	A module is a system that is a component of the HHS 2020 enterprise. This can refer to legacy systems (i.e., ASPEN) or new modules for MMISR (i.e. Data Services) or other module that may be defined in future modular system re-design.
Service	A Service is any capability offered or consumed by a user, system, module, component or Interface. This service can be a web service like REST or SOAP or a file service offered as a batch transfer or a message in a message queue or any other technical ways of offering and consuming a service. Every service has a specification that defines the input to the service, output of the service and exception or errors, which can be technical in nature or business centric.
Service Endpoint	A service endpoint is an entity, processor, or resource that can be referenced and to which services messages can be addressed. Endpoint references convey the information needed to address a service endpoint. Clients need to know this information before they can access a service.
Service Orchestration	Service Orchestration is the implementation of business workflows across modules, services, and Interfaces, that flows through the IP.
System	A working combination of hardware, software, and data communications devices.

Term	Definition
Workflow	A workflow consists of an orchestrated and repeatable pattern of business activity enabled by the systematic organization of resources into processes that transform materials, provide services, or process information through the IP to the HHS 2020 modules and services.
Work Stream	A SI work stream is a set of activity that is a component of the SI team's scope of work. These include: • Interfaces • SMR • IP • Enterprise Shared Services • Interfaces • Legacy Module Integration • New Module Integration • Service Orchestration

7.5 Appendix E: MECT Checklist and Programmatic CSF

This appendix contains the MECT Checklist items and programmatic CSFs applicable to Orchestration Management.

Table 6: MECT Checklist

Checklist ID	Requirement Text / System Review Criteria (SRC)	MITA Business Area Module Checklist Set	Business Process	CMS Guidance
IA.LDM.1	The system of interest accepts, records, stores, and retrieves documents (free-form or in HIPAA attachment format) submitted with or in reference to claim submission activity, and auto-archives	Information Architecture	IA Component Name: Logical Data Model (LDM)	This criterion does not apply to E&E. This criterion applies to modules that intake claims. They should be able to attach and retrieve claims-related documents such as operative reports; occupational, physical, and speech therapy reports; durable medical equipment and warranty data; manufacturer's tracking data for implants; waivers and demonstration-specific requirements; etc. For R1, evidence could include acquisition documents, requirements, a ConOps, or other planning documents that demonstrate intent to

Checklist ID	Requirement Text / System Review Criteria (SRC)	MITA Business Area Module Checklist Set	Business Process	CMS Guidance
	or forwards to appropriate operational area for processing.			implement HIPAA requirements. For R2 and R3, evidence should include screenshots showing how documents sent with claims are stored and are retrievable by system users. For R3, the evidence should demonstrate this capability back to go-live. For R2 (if not a desk review) and R3, the state should be prepared to demonstrate and discuss.
S&C.ISC.6	The system of interest complies with standards and protocols adopted by the Secretary under sections 1104 and 1561 of the Affordable Care Act.	Standards and Conditions	S&C: Industry Standards Condition	This criterion speaks to health information enrollment standards and protocols to promote the interoperability of systems for the enrollment of individuals in federal and state health and human services programs as well as the adoption of uniform standards and operating rules for the electronic transactions that occur between providers and health plans that are governed under HIPAA. Establishes a process to regularly update the standards and operating rules for electronic transactions and requires health plans to either certify compliance or face financial penalties. The goal of this section is to make the health system more efficient by reducing the clerical burden on providers, patients, and health plans. For R1, evidence could include acquisition documents, requirements, or a ConOps that explains how the state plans to adopt standards. For R2 and R3, evidence could include test reports of successful data

Checklist ID	Requirement Text / System Review Criteria (SRC)	MITA Business Area Module Checklist Set	Business Process	CMS Guidance
				exchange between modules and/or external systems. **Enterprise:** The state should have an architecture that supports this capability. **Module:** This applies only to modules involved in data exchange with human services systems. These should be able to support the state's data exchange goals.
TA.DC.5	The system of interest interfaces with the pharmacy prior authorization database.	Pharmacy	Technical Service Classification: Data Connectivity	This criterion does not apply to E&E. **Enterprise**: The state should have designs that indicate which modules will need to interface with the pharmacy prior authorization database. **Module**: This applies only to modules that should interface with the pharmacy prior authorization database. For R1, evidence could include acquisition documents, requirements, a ConOps that explains how this will be implemented, or other planning documents that demonstrate plans to incorporate this capability. For R2 and R3, evidence could include an System Design Document (SDD) showing how the module(s) interfaces with the pharmacy prior authorization database. Evidence should also include screenshots of successful prior authorizations and an example of a denied authorization request. For R3, the evidence should show that the interface was functional at go-live. For R2 (if not a desk review) and R3, the

Checklist ID	Requirement Text / System Review Criteria (SRC)	MITA Business Area Module Checklist Set	Business Process	CMS Guidance
				state should be prepared to demonstrate this capability.
TA.DC.6	The system interfaces with electronic authorization for retail pharmacy drug referral certification and authorization.	Pharmacy	Technical Service Classification: Data Connectivity	This criterion does not apply to E&E. Enterprise: The state should ensure that the relevant interfaces between the relevant modules are working. Module: This applies only to modules that support pharmacy drug referrals and authorizations. For R1, evidence could include acquisition documents, requirements, a ConOps that explains how this will be implemented, or other planning documents that demonstrate plans to incorporate this capability. For R2 and R3, evidence could include an SDD showing how the module(s) interfaces with the pharmacy prior authorization database. Evidence should also include screenshots of successful prior authorizations and an example of a denied authorization request. For R3, the evidence should show that the interface was functional at go-live. For R2 (if not a desk review) and R3, the state should be prepared to demonstrate this capability.
TA.SOA.1	The system of interest adopts MITA-recommended ESB, automated arrangement, coordination, and	Intermediary and Interface	Technical Service Classification: Service Oriented Architecture	This criterion means that the Medicaid system uses an enterprise service bus (ESB). For R1, evidence could include acquisition documents, requirements, a ConOps that explains how this will be implemented, or other planning documents that demonstrate plans to incorporate this

Checklist ID	Requirement Text / System Review Criteria (SRC)	MITA Business Area Module Checklist Set	Business Process	CMS Guidance
	management of system.			capability. For R2 and R3, evidence could include enterprise system diagrams like those found in a System Design Document that show how the ESB architecture is integrated in the overall solution. Evidence could also include test reports and demonstrations showing that individual modules are successfully integrated with the ESB. For R2 (if not a desk review) and R3, the state should be prepared to discuss. Module: Modules should be configurable to plug into the state's ESB.
TA.SE.2	The system of interest uses RESTful and/or SOAP-based web services for seamless coordination and integration with other U.S. Department of Health & Human Services (HHS) applications and intrastate agencies, including the Health Insurance Exchange (HIX).	Intermediary and Interface	Technical Service Classification: System Extensibility	For R1, evidence could include acquisition documents, requirements, a ConOps that explains how this will be implemented, or other planning documents that demonstrate plans to incorporate this capability. For R2 and R3, evidence could include enterprise system diagrams like those found in a System Design Document that explain how integration with the other systems are achieved. For R2 (if not a desk review) and R3, the state should be prepared to discuss. Module: This criterion applies to modules that must integrate with the Department of Health and Human Services and intrastate agencies.

7.5.1 Critical Success Factors

The following table provides the Programmatic Critical Success Factors:

Table 7: CSFs

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