MES Technical Reference Architecture

Alabama Medicaid Enterprise Systems (MES) Modernization Program (AMMP)

EA-k-1

Prepared for:

Alabama Medicaid Agency

Version 7.0

September 22, 2023

Revision History

Version	Effective Date	Revision Owner	Description of Change
7.0	09/22/2023	Latoya Byas	Approved and published
6.1	08/31/2023	William Yearling/Aaron Bowman	Updates for iteration. Updated acronym AMA to “the Agency” per Agency preferences. Minor wording updates
6.0	03/25/2023	Latoya Byas	Approved and published
5.1	02/28/2023	Neil Stokes	Addressed some grammatical changes – updated the link to BPMN.org. Removed Future Updates Log
5.0	09/29/2022	Latoya Byas	Approved and published
4.2	09/15/2022	William Yearling	Updated based on Agency comments
4.1	08/31/2022	William Yearling	Updated vendor to contractor to reflect award. Update references. Changing timing from seconds to expected parameters. Update MMIS to MES for consistency. Added details for Re-use. Additional changes for consistency with other artifacts and wording changes.
4.0	04/06/2022	Latoya Byas	Approved. Removed resolved comments and published
3.1	2/28/2022	Chris Shearier	Fourth Iteration Extensive updates throughout
3.0	12/08/2021	Latoya Byas	Approved. Removed resolved comments and published
2.1	9/21/2021	Chris Shearier	Updates based on Agency feedback
2.0	8/31/2021	Chris Shearier	Third Iteration Most sections updated.
1.2	04/21/2021	Chris Shearier	Updates based on Agency feedback
1.1	03/10/2021	Chris Shearier	Second Iteration Modified Sections 3.4, 4.4.5, 6.1.1, 6.3.3
1.0	03/24/2021	Latoya Byas	Approved. Removed resolved comments and published
0.2	3/05/2021	Chris Shearier	Updates based on Agency feedback
0.2	2/12/2021	Chris Shearier	Updates based on Agency feedback
0.1	2/10/2021	Chris Shearier	Initial Draft

TABLE OF CONTENTS

LIST OF EXHIBITS

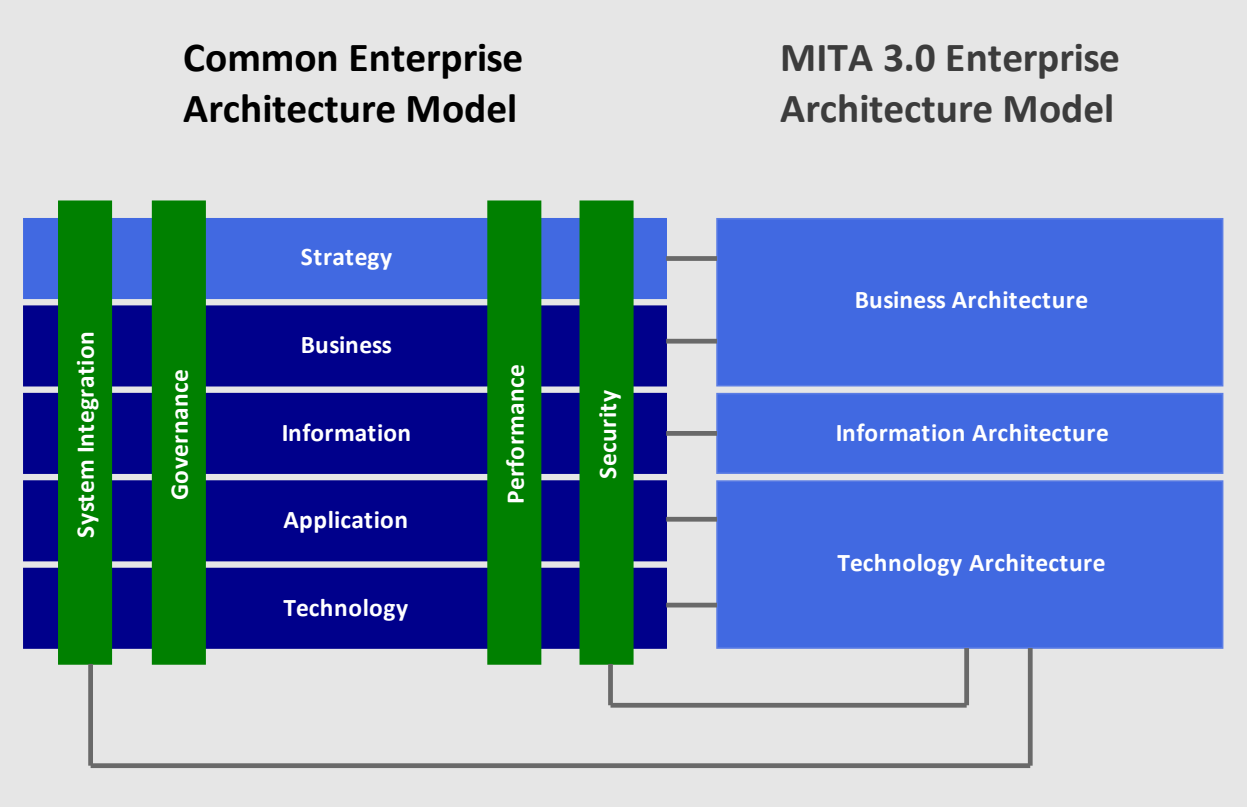
1 Introduction

The Technical Reference Architecture (TRA) provides the authoritative technical architecture approach and technical reference standards to the Alabama Medicaid Enterprise Systems (MES) Modernization Program (AMMP).

The MES Technical Reference Architecture will be built out incrementally, prioritizing the sections with the most immediate impact to MES projects and procurements. Changes will be submitted to the Alabama Medicaid Agency (the Agency) through the MES architectural governance process. The MES TRA is maintained by the Agency's Medicaid Enterprise Architecture (MEA) team.

The architectural diagrams shown in the following exhibit depict two meta-models for enterprise architecture. The right side is the Centers for Medicare & Medicaid Services (CMS) Medicaid Information Technology Architecture (MITA). The left side is a similar architecture based on other frameworks that are not specific to Medicaid. The lines between them show the mapping between the two frameworks.

Exhibit 1: Enterprise Architecture Models



This document structure is based on the framework on the left, which depicts a more detailed perspective. The TRA covers the Information, Application, and Technology layers as well as the System Integration, Governance, Performance, and Security components that span all layers.

The TRA primarily establishes a framework of architectural expectations to ensure consistent implementations of technical solutions.

1.1 Purpose

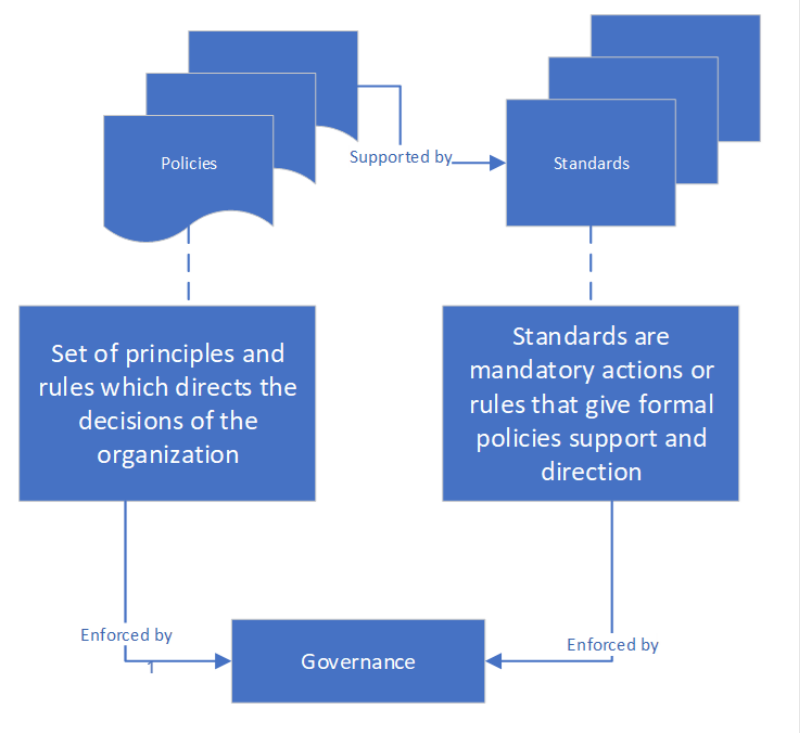
This document's purpose is to provide technical guidance and reference materials for the MEA team, MES Program Management Office (PMO), module contractors and other stakeholders in the AMMP.

The key consumers of the TRA are:

- The Agency change control and technical governance boards
- Future MES Module Contractors, System Integration (SI) Contractor, or other organizations interfacing with MES
- Development teams designing or implementing MES
- Teams operating or supporting MES
- The Agency teams that may use these architecture standards in non-MES projects

MES Change Advisory and Technical Review Boards will utilize the TRA as compliance materials to govern the MES program.

Exhibit 2: How policies, standards and governance are facilitated by the TRA



1.2 Document Scope

The TRA is an abstract blueprint to provide a framework for the MES architecture, rather than documentation of the MES solution architecture itself. Essentially, strategy documentation provides the 'what,' 'when' and 'why' of the MES approach, where the TRA provides the 'how'.

The TRA will define expectations for requirements and frameworks, but implementation details will be deferred to the other architecture and system integration documents. The architecture and system integrator documents will also be made available as reference material for procurement of systems from vendors.

This document, along with other architectural and SI deliverables will include policies and standards to be used in the design, construction, and implementation of MES.

Requirements defined in any Request for Proposal/Request for Bid (RFP/RFB) supersede any conflicting requirements in this document.

1.3 Referenced Deliverables

The following deliverables are cited in this document:

- EA-b: Enterprise Architecture Governance
- EA-f: MMIS Concept of Operations
- EA-k-7: Interface Control Document Template

Links to external artifacts is restricted to those with a business need and the required level of access.

2 TRA Overview

This section describes the scope of the TRA and the processes for publishing, revising, and maintaining the TRA and TRA documentation.

2.1 TRA Scope

It is not expected that existing components will be retrofitted to ensure TRA compliance, but all technical architecture decisions arriving through the MES' governance intake process will be reviewed and approved by the Technical Review Board (TRB) and Enterprise Architecture Board (EAB).

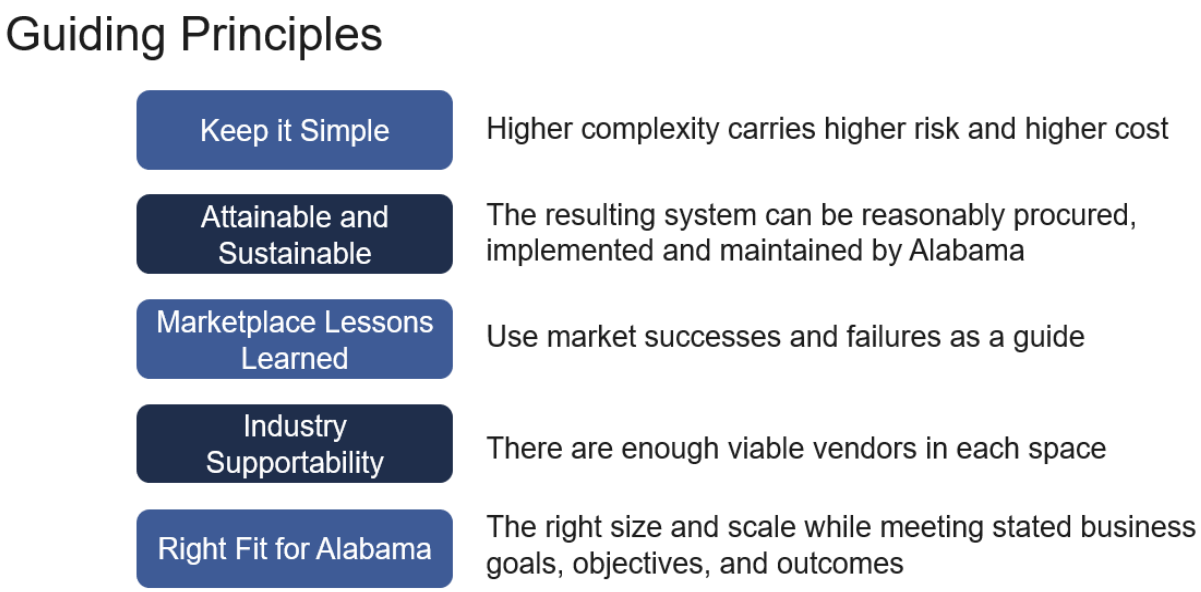
The scope of the TRA includes design, procurement, and implementation of the MES. The TRA standards define architectural requirements, but do not identify specific products to implement the architecture.

At its inception, the scope of the TRA is limited to MES within the Agency.

2.2 Architecture Guiding Principles

The image below provides a set of guiding principles that were established for Alabama Medicaid to aid in decision-making.

Exhibit 3: Architecture Guiding Principles



2.3 TRA Publishing and Maintenance

This section provides a framework for updating, reviewing, and publishing the TRA throughout the life of the program as the Agency's business needs evolve and new technologies become available.

The TRA will be published on the Alabama MES Modernization Program (AMMP) SharePoint Procurement Site.

The source files will be version controlled and maintained on the AMMP – Program Wide – Project Site under Deliverables.

The TRA will also be included in the procurement libraries of future MES modular procurements to provide a reference for the permissible technologies, standards, patterns, and tools for components of the MES architecture.

2.3.1 TRA Modifications

Changes to the TRA will be submitted to the Agency through the MES architectural governance process.

Triggers for TRA modifications include:

- If the TRA does not provide enough direction to the TRB for a solution or topic under review
- As new technologies and contractor offerings emerge
- When Federal or State technology mandates are published or updated
- When MES modules request deviations from TRA standards
- One year from the date of the last approved modification

The TRB and EAB approves changes to the TRA before they are adopted.

2.3.2 Maintenance of TRA Structure

The Agency MEA team maintains the MES TRA Structure.

2.3.3 TRA Maintenance Considerations

Since new MES procurements will require alignment with the MES TRA, careful consideration must be given when changes are made in order to understand the impact to existing procurements and contracts.

2.4 TRA Governance

The TRA will be governed via the TRB, according to processes laid out in EA-b: Enterprise Architecture Governance.

3 Information Architecture

This section presents a summary of the aspects of the Information Architecture related to system integration. It does not mandate any approach to Information Architecture used by contractors ***within the contractors' systems***.

The technical implementation of the interfaces is covered in the Section 4 - Application Reference Architecture of this document.

The following terms are used to define the information architecture:

Data Exchange (Interface) – An exchange of information from a sender to one or more receivers; an interface. Each interface has a specification for the content and structure of the data exchanged. An interface may be a push of information or a request for information.

Data Entity – Represents a "class" of data; a grouping of data elements by conceptual business use. The Conceptual Data Model defines the Data Entities and their relationships.

Data Element – A specific element of data that indicates specific pieces of information contained in a Data Entity. The Logical Data Model defines the Data Elements contained in each Data Entity.

System-Level Interfaces – This model shows the interface(s) that exchange data between two systems. This includes a high-level view of the flow of Data Entities, and an Interface Control Document (ICD) that defines the technical specification for the interface.

Service-Level Interfaces Model – This model shows the flow of Data Entities required to support specific Business Services, which ultimately support the ability to perform a Business Process. This model is System-agnostic and focuses on the business needs for data. Therefore, this model enables visualization of potential System interface needs when splitting functionality into separate systems (modularity).

System Integration Platform (SIP) – A group of services that pass information from one system to another. It will consist of a service bus, file transfer capabilities, and interfaces to shared services. It may also include transformation capabilities.

System of Record – The authoritative source of data. The System of Record is typically the originator of the data, but in certain circumstances may receive the data from an outside source for use in the MES.

3.1 Interface Creation Process

The following steps must be completed for each interface, both internal and external. The steps are defined in the following sections of the document, and they must be completed for all interfaces whether utilizing batch files or Service Oriented Architecture (SOA) interfaces.

1. Write the Interface Control Document (ICD)
2. Obtain approval from the EAB/TRB
3. Publish ICD
4. Deploy interface for testing
5. Obtain approval of test results
6. Deploy interface to Production

Each data interface requires an ICD describing the interface. The organization that serves as the System of Record is responsible for creating and maintaining the ICD when exchanging data with MES modules. For external and third-party data sources, the organization that receives the data is responsible for creating and maintaining the ICD. The ICD will be in MS-Word format but may be subject to change in the future to integrate with other tools.

The interface architecture must be designed for each interface to serve a distinct business purpose. The ICD should clearly define a set of applicable Data Entities. If completing the transaction requires data from Data Entities outside the scope of the interface, additional interfaces may need to be defined.

In some scenarios, a system may need to obtain data from multiple Data Entities that originate from different System(s) of Record and may need to combine the data. For example, the Transformed Medicaid Statistical Information System (T-MSIS) exchange with CMS uses the same strategy of exchanging one business area of data at a time. However, the interface requires separate files for provider data, recipient data, and claims data. CMS then combines the data as needed for analysis. CMS can change the way the T-MSIS data is combined without requiring interface changes.

The ICD will describe the interface by including the following information:

1. Business purpose of the interface
2. Business description of the data content (Data Entities)
3. Use of Protected Health Information (PHI), Personally Identifiable Information (PII) and Federal Tax Information (FTI) in data content
4. Sending system
5. Receiving system(s)
6. Exchange method (file, request / response, or publish / subscribe)
7. The business or technical process that triggers the exchange
8. Action taken after exchange is complete
9. Existing interfaces considered and selected for reuse

10. Standards considered and selected for use
11. Interface pattern (See Section 4.2)
12. Expected exchange frequency
13. Expected average and peak volumes of data
14. Performance Requirements
15. Contact Information
16. Interface Control Document Approvals (where applicable)

The SI will be responsible for working with the module contractors to develop the ICDs for interfaces that pass through the SIP (aka internal interfaces). The Agency will approve or reject these ICDs and make final decisions when the expectations of the organizations exchanging data do not align.

The following link is the latest ICD template.

MES_NTT_EA-k-7_Interface_Control_Document_Template.docx

Exhibit 4: Example Information in the ICD

Required Information	Example
Business purpose of the interface	Add new provider to MMIS
Business description of the data content	Provider Enrollment for newly enrolled provider
Sending system	Provider Enrollment
Receiving system(s)	MMIS
Exchange method (file, request / response, or publish/subscribe)	Publish / Subscribe
The process that triggers the exchange	Staff approves a provider enrollment request
Standards to be used	US Core Implementation Guide (v3.2.0: STU3) based on Fast Healthcare Interoperability Resources (FHIR) R4
Action taken after exchange is complete	Begin using data to pay claims
Expected exchange frequency	On-demand
Expected average and peak volumes of data	Five (5) per hour average 15 per hour peak
Performance Requirements	Responding systems must process the published information with two (2) minutes.
Contact Information	email@domain.exc, 555-555-5555
Interface Control Document Approvals	Signature, Date

3.2 Data Standards

Interface developers must use industry standard definitions and data field structures when standards are available.

Determining when to use standards, as well as selecting from multiple standards for the same information, should be based on flexibility of the standard and the level of adoption within the industry. The EAB can provide guidance on standards at any time during the interface creation process. The EAB will make the final decision if stakeholders cannot agree.

3.2.1 Health Care Data Standards

There are multiple sources for standards applicable to state healthcare programs, such as:

- Centers for Medicare and Medicaid Services (CMS) Interoperability and Patient Access Rule (CMS-9115-F)
- Health Insurance Portability and Accountability Act of 1996 (HIPAA)
- Health Level Seven International (HL7) Fast Healthcare Interoperability Resources (FHIR)
- United States Core Data for Interoperability (USCDI)
- Office of the National Coordinator (ONC) / Health Information Technology (HIT) Interoperability Standards Advisory
- HL7 Reference Information Model
- Affordable Care Act (ACA) 1104 Core Operating Rules

The following table provides examples of data element standards. The examples are for illustrative use only.

Exhibit 5: Example Data Elements and Applicable Standards

Data Element	Data Standard to Use
Race and ethnicity	Centers for Disease Control and Prevention (CDC) Race and Ethnicity Code Set Version 1.0 as specified by ONC/HIT 2020 Interoperability Standards Advisory
Health Record	US Core Implementation Guide (v3.2.0: Standard for Trial Use (STU) 3) based on FHIR
Care Plans	HL7 Implementation Guide for Clinical Document Architecture (CDA®) Release 2
Provider Directory	US Core Implementation Guide (v3.2.0: STU3) based on FHIR

3.2.2 General Data Standards

Standards exist for general information included in interfaces. In addition to the sources listed in section 4.4.1, sources of standards include:

- National Institute of Standards and Technology (NIST)
- International Organization for Standardization (ISO)
- American National Standards Institute (ANSI)

The following table shows examples of data standards that may be used. The examples are illustrative only.

Exhibit 6: Example Data Standards

Data Element	Data Standard to Use	Standards Body
State and County identification	United States Geological Survey (USGS) Geographic Names Information System (formerly FIPS)	US Geological Survey
Languages	ISO 639 Language Codes	ISO
Location Data Structure	US Core Implementation Guide (v3.2.0: STU3) based on FHIR	HL7

3.3 Required Data Standards

The required Health Care Data Standards will be specified in the interface control document and during the interface development process.

3.3.1 Required Standards for General Data Content

The standards in the following table apply to all interfaces unless defined otherwise by an interface specification in an ICD.

Exhibit 7: Standards for General Data Content

Data Element	Data Standard to Use	Standards Body
Language	Latin-1 character set ISO 8859-1	ISO
Demographics, Race and Ethnicity, Gender	ONC Interoperability Advisory	ONC

Data Element	Data Standard to Use	Standards Body
Units of Measure	The Unified Code for Units of Measure (UCUM)	The UCUM Organization
Dates	Date only with no time value	
Time	Central Time and offset	

These standards apply to interface specifications defined in an ICD. The Agency does not prescribe standards for data content or structure within a system. However, use of standards within a system may be required to support The Agency business rules.

3.4 Information Specifications

3.4.1 Reuse

Interfaces must be reused where possible to minimize use of specialized interfaces. For example, interfaces that require similar but not identical data should use a common interface with sufficient data content to meet the needs of both. Each system will use the data elements needed and ignore the unneeded content.

Each schema must use the same structure for any data used in more than one interface within the business area. For example, the same structure for address must be used each time address information is exchanged.

If data is updated in a System of Record, a method of providing the same update to all systems that receive that data is required. When an interface passes updates to existing data from one system to another, the interface must ensure the same data is updated in each applicable system.

The Agency prefers that all interfaces use natural keys to identify individual elements. A natural key is comprised of unique elements within the data itself. The data elements that comprise the natural key must be managed data elements that are part of the Agency's Master Data Management (MDM) program. Managed elements mean the elements have been defined and logged within the MDM application and a process is in place to control changes to the data elements. The Data Governance Committee and Data Governance Council implement and control the process to manage master data elements. All additions, changes, and deletes for master data elements are routed through the Data Governance Committee and approved by the Data Governance Council. If an element is used in a natural key of a data interface, it must automatically become a master data element.

A unique key that has no meaning relative to the data itself and is internally unique to the System of Record should not be used. If use of this type of unique key was allowed, downstream systems would be required to implement customized solutions that store the keys for reference. Use of this type of key adds unnecessary complexity to the integration of systems and decreases the effectiveness of modularity. Therefore, the use of system specific keys is strongly discouraged. Proposed use of system specific keys shall be defined on the ICD and reviewed by the Agency.

3.4.2 Data Transformation

The System of Record for a Data Entity should produce and accept data for that Data Entity in a standard interface format. If this system cannot produce and accept data in a standard format, the contractor is responsible for working with the SI Contractor to define transformation operations in order to complete data exchange as defined in the ICD.

The Agency expects systems other than the System of Record for a Data Entity may need to transform data to or from interface standards. These systems may also support a capability to produce and accept the standard data structures and content without SI transformations.

3.5 Defining and Publishing Interface Information

Upon completion of interface specification development, the contractor developing the interface must publish the specifications. Interface specifications will be published in a common ICD Repository to be specified by the Agency and designed to manage this content.

4 Application Reference Architecture

This section addresses the application architecture components that impact the integration frameworks required for MES contractors. These requirements only apply to interfaces that communicate with other systems outside of a contractor's solution. Contractors are free to use any framework for interfaces internally within their applications, provided their assigned business requirements are met.

MES contractors should prioritize the use of interface architecture that follows SOA patterns. This includes both web services and event-driven SOA. However, it is understood that in some scenarios file exchanges will also be used.

The application architecture for interfaces is based on the following requirements:

1. Utilize reusable Application Programming Interfaces (APIs)
2. Use APIs to reduce operational dependencies between systems
3. Use an integration layer for interfaces rather than direct connections from one module to another

APIs used by MES contractors must follow industry standard web service architectures utilizing Simple Object Access Protocol (SOAP) and/or Representational State Transfer (REST).

All changes to the production environment shall go through the Change Control Board (CCB).

4.1 Service-Oriented Architecture Patterns

This section presents the framework for SOA integration patterns and the additional Agency standards used with the patterns.

The following are the three acceptable patterns for SOA interfaces:

1. Synchronous Request / Response – make a request and wait for response
2. Asynchronous Request / Response – make a request but do not wait for response

3. Publish / Subscribe – Send data once and let an integration layer publish it to one or more systems

4.1.1 Synchronous Request / Response

Use a synchronous request / response pattern to:

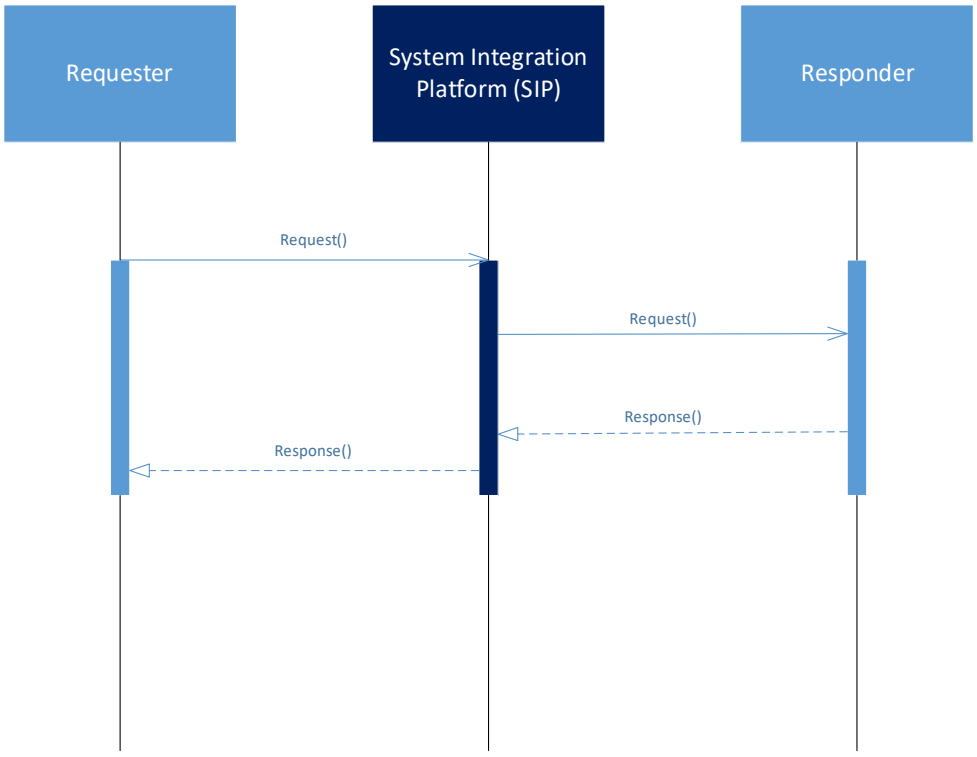
- Retrieve data from another system
- Request another system to perform an action and wait for a response about the status of the action

Retrieving data from another system is often called a query or a request query although the request must **not** include Structure Query Language (SQL). The requesting system provides some data to describe the information requested. For example, the requester may supply the National Provider Identifier (NPI) when requesting information about the service locations for the provider with that NPI. The responding system will complete the request and reply within expected parameters. Failure to respond within expected parameters will usually cause problems for users or result in a queue of requests to build at the responding system.

A synchronous request / response is also appropriate for requesting a system to perform an action. As with a request query, the action should complete within expected parameters. Processing a claim in 'real time' is an example that should use a synchronous request / response. Any action that takes more than a few seconds to complete should be performed asynchronously or redesigned to have adequate performance.

The following diagram depicts a synchronous request / response pattern.

Exhibit 8: Synchronous Request / Response Pattern



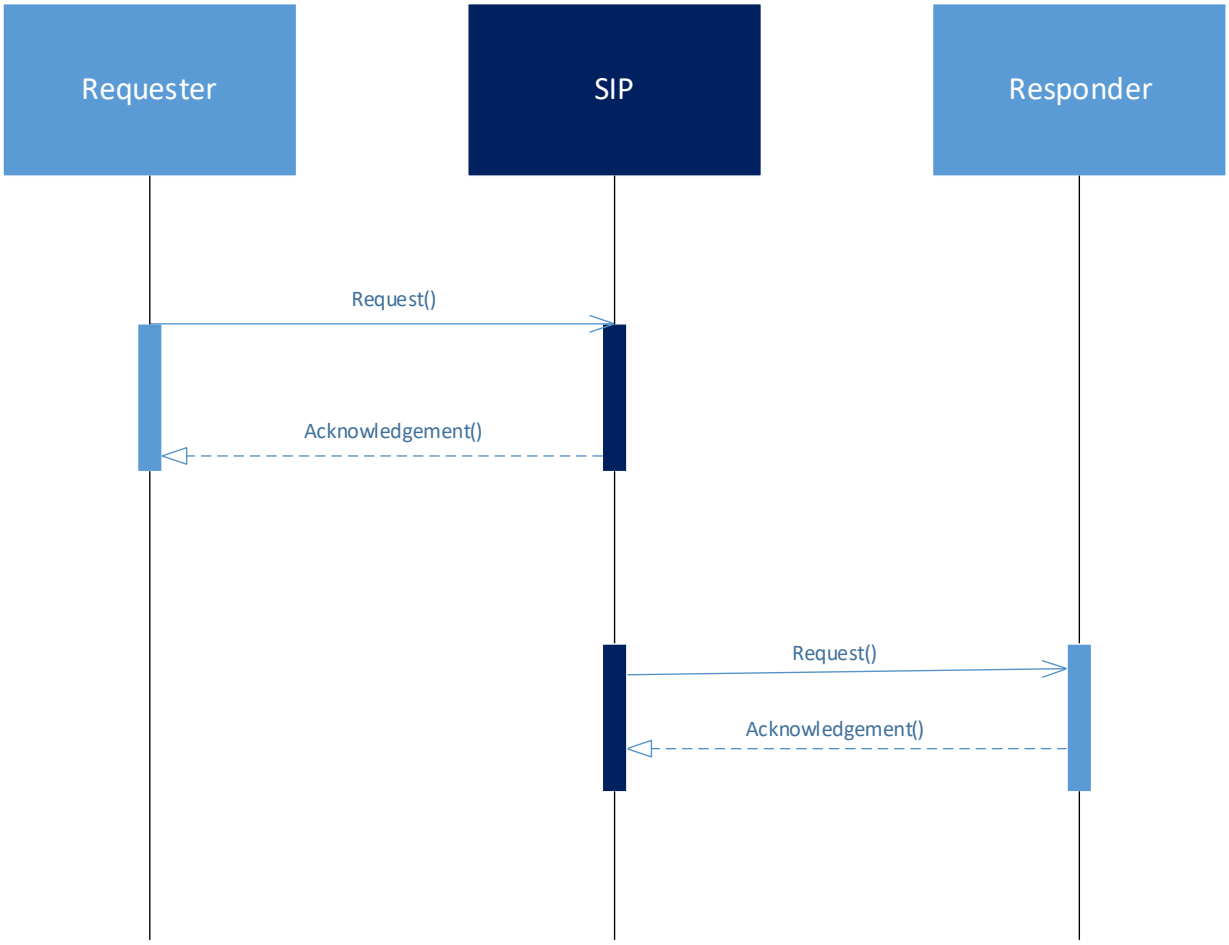
4.1.2 Asynchronous Request / Response

Use an asynchronous request / response pattern when a system requests another to perform a transaction that cannot be completed within a few seconds or where an immediate response is not required. For example, a request to manually review a prior authorization may be sent asynchronously as it will take longer than a few seconds to complete a request.

During the message exchange, there must be both a request and response message. However, the response will only indicate that the message has been successfully received. The action request is then processed by the responder without further interaction with the requestor. If the publishing system needs to know the status of any actions made, a separate interface must be used.

The following diagram depicts an asynchronous request / response pattern.

Exhibit 9: Asynchronous Request / Response Pattern



4.1.3 Publish / Subscribe

Use the publish / subscribe model for interfaces with any of the following characteristics:

1. Information sent to multiple receivers
2. Information that does not require immediate processing

3. Messages from event-driven application architectures

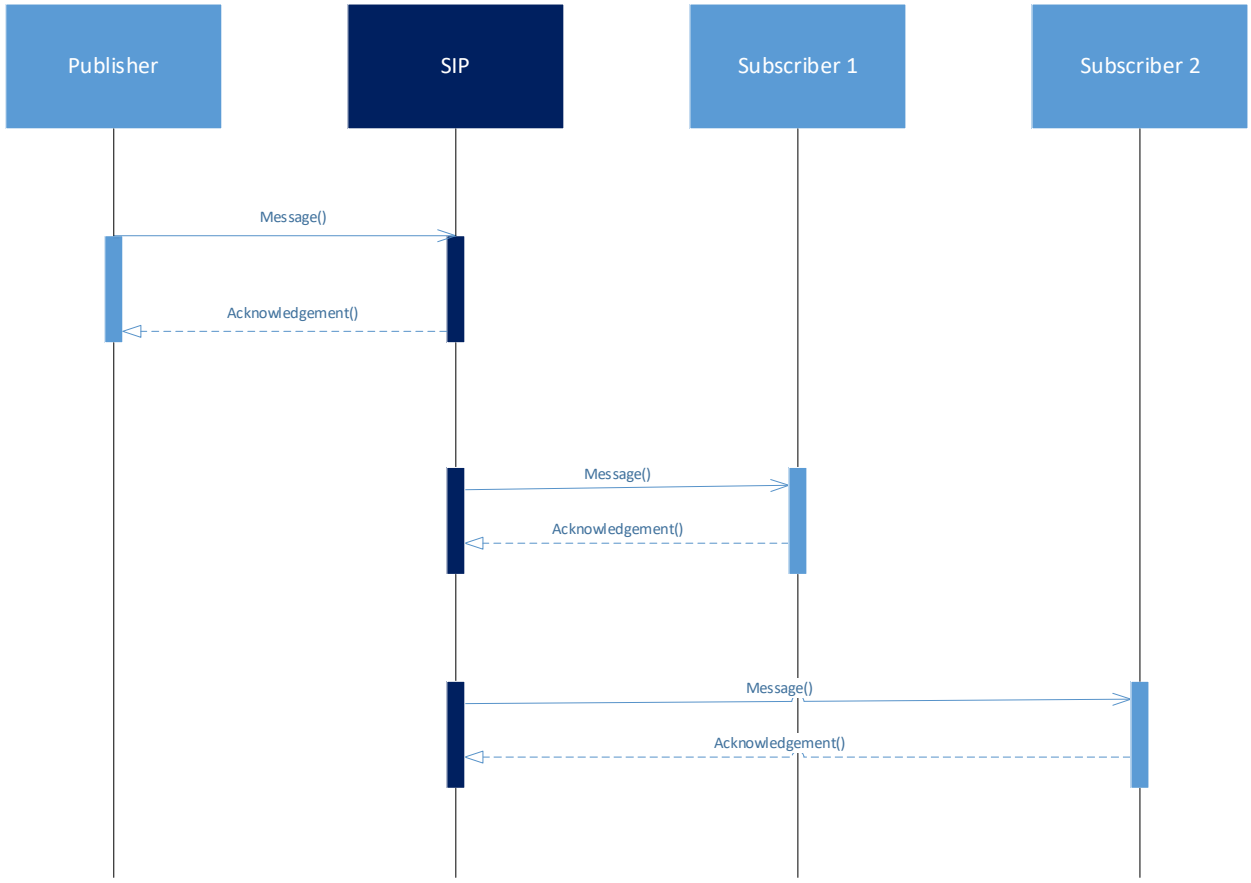
The publish / subscribe framework calls for the sending system to send out only one message, regardless of which systems need to receive. The message is sent to an external integration layer, typically an Enterprise Service Bus (ESB), which forwards the messages to all subscribing systems. The publish / subscribe model is always asynchronous.

When using the publish / subscribe model, the sending system has no information about which systems receive the message and when they receive and process it. There is no coordination between systems for completing any work resulting from the sending message. Messages may be queued but are usually designed to process data within a few minutes of receipt.

The publish / subscribe model is more complex than the request / response model. The request / response exchange requires the integration layer to receive a single message and forward it to another system. The publish / subscribe model requires that the integration layer build and maintain a repository of messages and the systems that subscribe to them. The exchange begins when a message publisher sends the message to integration layer. The integration layer receives the message and based on configuration, routes the message to the subscribers.

The integration layer must provide guaranteed delivery of messages. It must also identify and track errors and provide notifications to automated monitoring tools.

Exhibit 10: Publish / Subscribe Pattern



4.1.4 Determine Which Pattern to Use

The following table provides use cases for determining the appropriate SOA pattern to implement.

Exhibit 11: Selecting SOA Pattern Based on Use Cases

Use Cases	Use This Pattern	Examples
Retrieve data from another system Request another system to perform an action that is 1) completed quickly and 2) requires the sending system track the status of completion	Synchronous Request / Response	Request a provider's mailing address Request another system to adjudicate a claim
Request another system to perform an action Send data or request a single system to perform an action that takes longer than a few seconds and the sender does not care about the status of an action	Asynchronous Request / Response	Request a prior authorization review Request a manual review of provider enrollment forms
Any information that must be sent to two or more systems	Publish / Subscribe.	Notification that a new provider is enrolled in the program. The sender does not know which downstream systems get the information

4.2 File Transfer Pattern

File transfers are used to exchange large amounts of data that result in a very long processing time to complete.

File transfers must be defined and built to use the integration layer. The need for reusable interfaces, minimizing dependencies between systems and use of an integration layer is as important for the file transfer pattern as it is for SOA patterns.

File transfers must include the following:

1. Reusable interfaces. For example, a system sending provider data to multiple systems must send a single file that will be used by all other systems

2. Message notification. A message notification is sent, using the publish / subscribe model, to the System Integration Platform (SIP), when the file has been successfully transferred. The SIP will forward the message to subscribers. Modules receiving the file will subscribe to the notification.

3. Use of the integration services which includes a managed file transfer (MFT) application available to all systems

4.3 Application Programming Interfaces

The modular approach to the MES requires that applications interact to perform business processing. Each MES application must provide an API for other applications to use when exchanging information. To limit proliferation of interfaces that use different APIs for the same data, the Agency expects systems to use a common API for exchanging similar data between source system and any other system.

Once the model of the message exchange is determined, the details of the application programming interfaces (APIs) must be defined. The interface pattern defines the interaction at a high level. The API defines the details of the exchange and the content of the messages.

4.3.1 API Frameworks

APIs will be web services that prioritize the use of common data structures and SOA patterns for data exchanges. The SOA model requires consistency across the various data exchanges throughout the MES. All interfaces will use versioning for both the message exchange and the message content. This subsection describes the API design required for the AMMP, including common APIs throughout the MES, security and privacy, and reuse. This section includes:

- Request / Response using a common and versioned API
- Publish / Subscribe using a common and versioned API
- Publishing API specifications

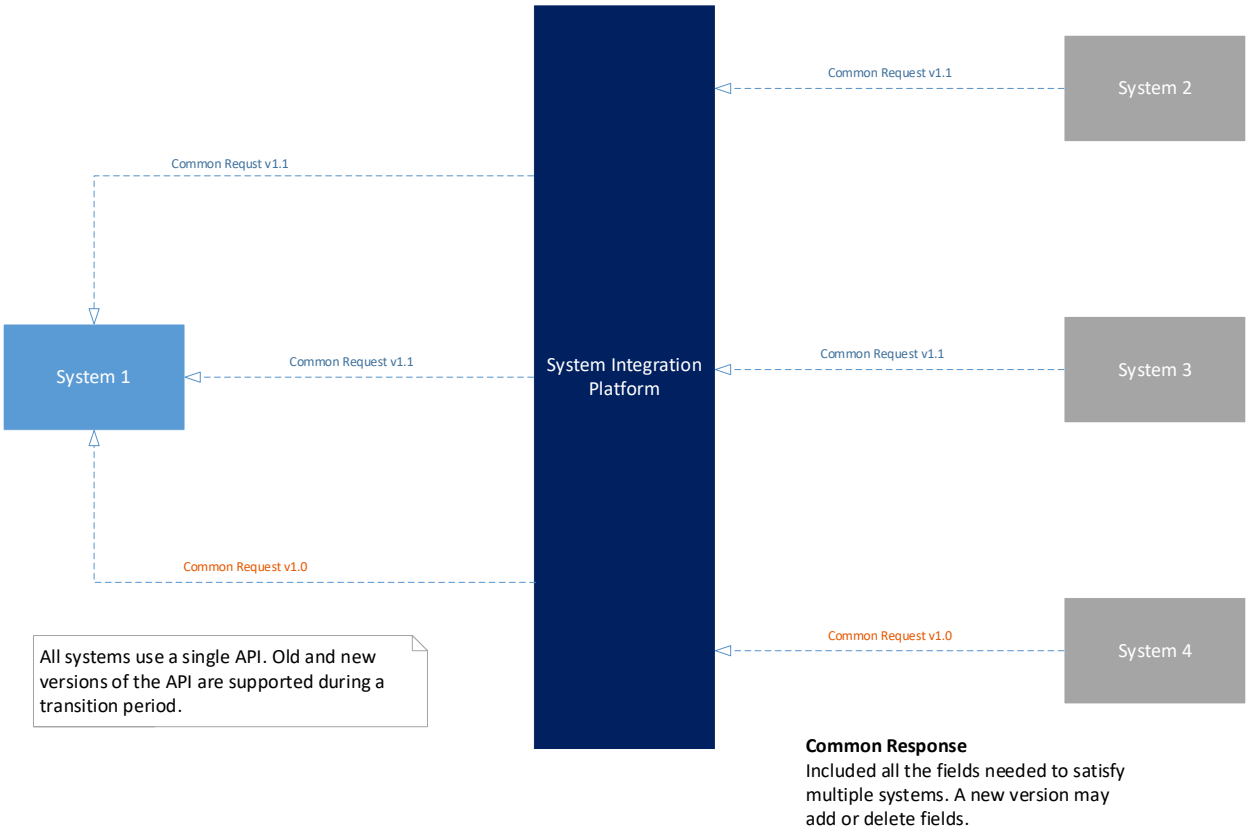
4.3.1.1 Request / Response Using a Common and Version API

Over time, APIs will change to support evolving business needs. To reduce the need for all systems to be updated at the same time, each system must support multiple versions of APIs during a transition period.

Exhibit 12 represents systems that use versioning for a common API. In this example, the systems are implementing application changes to use a new version. Systems 2 and 3 have already updated and use version 1.1. System 4 has not yet been updated and will continue to use version 1.0 for a short period of time. System 1 must support both queries for a period until all systems' changes are implemented. While some situations may require a simultaneous upgrade to a new version, the ability to migrate independently must exist through versioning of all APIs.

All APIs will be versioned, and applications must be configurable to send or receive using different versions of interfaces.

Exhibit 12: Support for Multiple Versions of a Common API

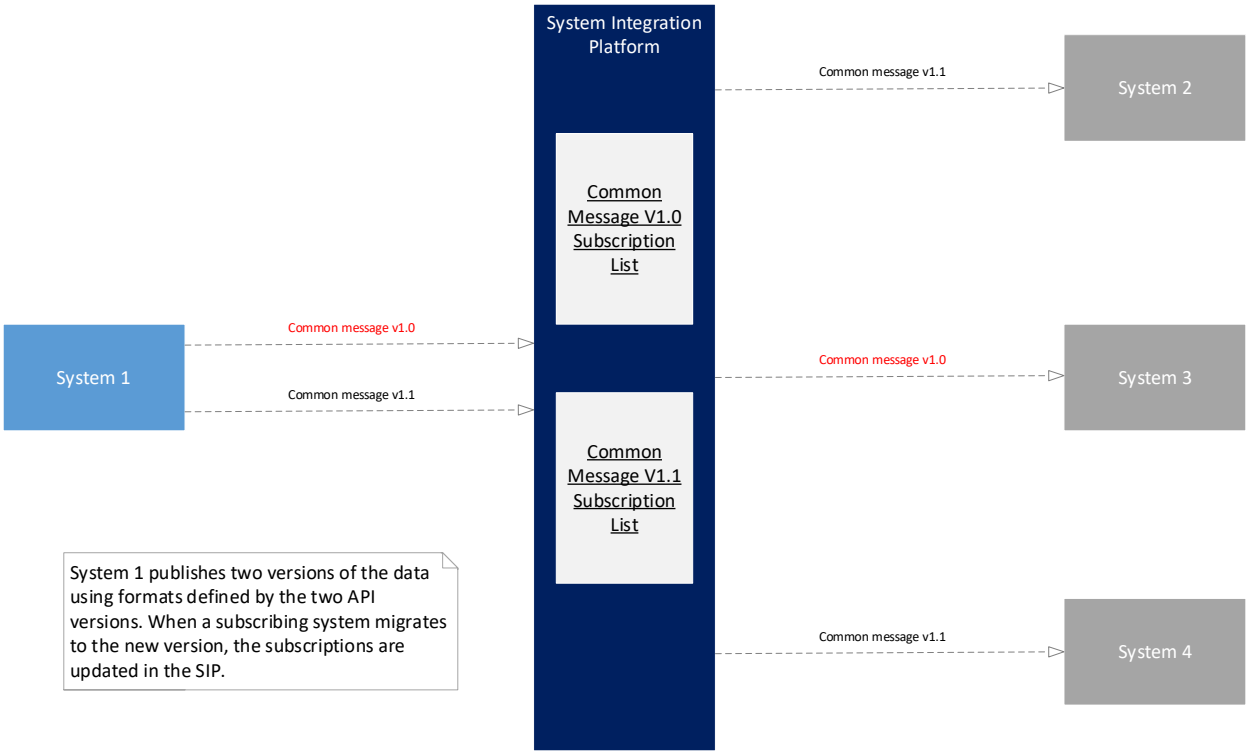


4.3.1.2 Publish / Subscribe Using a Common and Versioned API

API versioning is also required with the publish / subscribe model.

In Exhibit 13, System 1 has begun publishing a new version of message. During a transition period, it will publish both versions 1.0 and 1.1 so that the subscribing systems can change independently. When each system transitions, it will unsubscribe from version 1.0 and start a subscription to version 1.1. As with the request / response exchange, some scenarios may require all systems to change versions at the same time. The framework and capability to support different versions must be in place for every interface change.

Exhibit 13: Publish / Subscribe with a Common API and Versioning



4.3.1.3 Publishing API Specifications

To use a web service, the requester must know the full specification of how to call the API, including the location of the service, what messages are accepted, and the structure of the data exchanged. To support the exchanging web service, the organization providing the service must provide a specification that includes the following:

- A description of how to call the service
  - For SOAP services, the description is published as a Web Services Description Language (WSDL) file
  - For REST services, the description is an OpenAPI document
- A schema that defines the structure and content of the data exchanged
- Examples of the request and success response messages
- A list of error response codes and error response messages

4.3.2 Use of Custom APIs

Exhibit 14 depicts a system interfacing with three other systems. While each of the three systems exchange almost identical information, they use different APIs for each system. The responses, listed to the right of the requesting systems, show that each response contains similar provider details. This exchange may be considered modular, but the custom APIs result in tightly coupled systems.

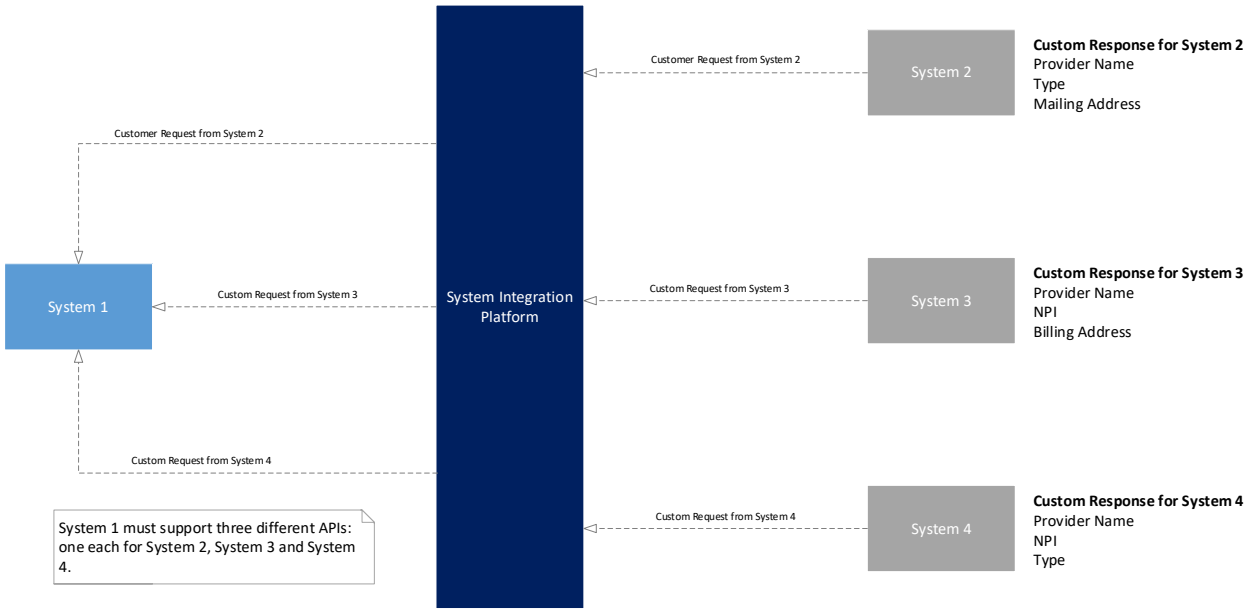
These systems are considered tightly coupled because:

- The interfaces are customized for the specific needs of each system
- Changing the internal operation of one system may require a change to the other
- Replacing either system will require a brand-new interface

This approach does not benefit from reuse, requiring more work to maintain three APIs rather than one.

The use of custom requests and responses for similar information must be avoided.

Exhibit 14: Custom API for Each Module



4.3.3 API Security and Privacy

This section will refer to PHI, PII, and other information such as banking and financial information as “sensitive”. Data Management processes will define which data is sensitive.

The primary pattern for MES interfaces is sending the same data to each system, and the receiving system decides which data is used and stored. When sensitive data is included in the response, additional protections are needed. Each interface must restrict exchanging sensitive data to only the systems authorized to receive the data.

Web service requests must include credentials from the requesting system and the receiving system must validate the credentials. Responses must only be returned when the credentials validate authorization for the request.

4.3.3.1 Request / Response API Security and Privacy

The following are patterns that restrict delivery of sensitive information in Request / Response interfaces.

The responding system provides a single service and only returns sensitive data for requests from authorized systems. Unauthorized systems receive responses without the sensitive data. Implementation details are internal to the application.

The responding systems provide more than one interface. One interface is used for non-sensitive information, and another provides the sensitive information to authorized requestors. The service providing sensitive information may provide only the sensitive information or it may include all the non-sensitive information as well. The first approach requires more than one service call from the requestor.

Systems include sensitive information in all responses and the sensitive information is removed by the SIP when delivered to systems that are not authorized to receive it.

All three patterns have advantages and disadvantages. The Agency will choose one or more patterns in the future for use in the MES.

4.3.3.2 Publish / Subscribe API Security and Privacy

The following are patterns that restrict delivery of sensitive information in Publish / Subscribe interfaces.

Systems sending data publish multiple messages. One message includes all information not classified as sensitive. Another message is published with sensitive information. Subscribing to messages with sensitive information is restricted to authorized systems.

Systems publish a single message and sensitive information is removed by the SIP when the subscriber is not authorized to receive it.

Both patterns have advantages and disadvantages. The Agency will choose one or more patterns in the future for use in the MES.

4.3.4 Interface Standards

Interfaces must meet multiple standards. The specific standards will be defined in the specifications for each interface.

Interfaces will require compliance with one or more of the following standards:

- W3C SOAP Version 1.2
- XML Technologies including XML (v1.0), XML Namespaces (v1.1), XML Schema (v1.1)
- Representational state transfer (REST)
- JSON (JavaScript Object Notation)
- FHIR® – Fast Healthcare Interoperability Resources (hl7.org/fhir)
- Office of the National Coordinator for Health Information Technology (ONC) Interoperability Standards Advisory (ISA)

4.3.5 API Architectural Standards

APIs will be used by different organizations; The Agency retains authority to approve or reject any API used to exchange information between modules.

The following apply to Request / Response and Publish / Subscribe APIs:

- APIs must be designed for reuse and a common API will be used for exchanging similar information. APIs must be extensible to enable modifications without rewriting the API
- APIs will use versioning. Each version requires publishing new API specifications. This version must be identified in WSDL files, OpenAPI documents, and schemas definitions
- Systems must support using multiple versions of an API at the same time to accommodate other systems transitioning from the old and new versions at different times
- The organization providing a web service is responsible for publishing the WSDL, OpenAPI, and message schemas.
- The organization publishing a message is responsible for publishing an ICD with the message specification to a repository defined by the Agency
- When using a Request / Response model, the requestor must authenticate with credentials
- Systems receiving requests must validate credentials and authorization for the request. Responses are restricted to authorized requestors.
- Some requests will include user ID or group ID. When required, the user ID or group ID must be used to determine if the user is authorized to receive the information requested. This pattern will be most common in services submitted to and from application portals. This authorization check is in addition to the authorization for accessing an interface
- Web service specifications must include descriptions of requests, responses, and error responses. These specifications are included in the WSDL for SOAP web services. For REST services, the specifications are in the OpenAPI document
- Within each system, integration endpoints must be configurable including such items as: Uniform Resource Locator (URL), Request Name, Request Version, ports, and binding. In a requesting system, these items define the location of the web service to be called. In the responding system, they specify how the web service will listen for requests
- In WSDL, Extensible Markup Language (XML) Schema Definition (XSD), and OpenAPI documents, URLs, namespaces, schema references and other data values referring to host names and addresses must use fully qualified domain names that are resolvable and accessible to all organizations using the API
- SOAP web services must use WSDL and XSD files to define the API. The files must:
- Define and reuse complex types for data structures used more than once
- Define the values for MinOccurs and MaxOccurs when values are other than 1
- Include type definitions
- More requirements to be defined in future versions of the TRA document
- REST services must use the OpenAPI Specification to define the schema of the data content. Data schemas must follow the standards described in the Information Architecture section

- Publish / subscribe messages must be defined using XSDs
- Standards for JavaScript Object Notation (JSON) schemas are currently evolving. The Agency expects OpenAPI v3.1, currently under development to be the standard. Until a standard is approved and widely available, the Agency may request an XSD representing the XML equivalent to provide full definition of the schema
- Web services must be stateless
- Systems sending and receiving data must be able to process multiple transactions in parallel for each interface. The number of simultaneous transactions should be configurable. The contractor is responsible to determine the configuration based on the Key Performance Indicators (KPIs) and Service Level Agreements (SLAs)

4.4 Standards

4.4.1 Architecture, Analysis and Design Standards

Exhibit 15 provides a listing of generally accepted standards and specifications for the planning, analysis, and design of the State Medicaid Enterprise architecture.

Note: A third party maintains the internet address for a source, which may change over time.

Exhibit 15: Architecture, Analysis and Design Standards

Standard Name	Objective	Source
Unified Modeling Language (UML) Profiles	This standard addresses business specific needs and technologies. The profiles include: • Platform Independent Model (PIM) • Platform Specific Model (PSM) • Consolidated Omnibus Budget Reconciliation Act (COBRA) Component Model (CCM) • Enterprise Application Integration (EAI) • Enterprise Distributed Object Computing (EDOC) • Modeling Quality of Service (QoS) and Fault Tolerance Characteristics and Mechanisms • Schedule ability, Performance and Time	www.uml.org/#UMLProfiles

Standard Name	Objective	Source
	• System on a Chip (SoC) • Systems Engineering (SysML) • Testing Profile	
Meta-Object Facility (MOF)	This standard provides an environment where models can export from one application, import into another, transport across a network, store in a repository and then stakeholders can retrieve and render it into different formats. (We do not restrict these functions to structural models, or to models defined in UML.)	www.omg.org/mof/
Model Driven Architecture (MDA)	This standard unifies development from a PIM to a PSM. Object Management Group (OMG) MOF-enabled transformations are the basis of this standard.	www.omg.org/mof/
Business Process Definition Metamodel (BPDM)	This standard provides the ability to model business process with standard language and metadata.	www.omg.org/
UML Enterprise Distributed Object Computing (EDOC)	This standard simplifies development of component-based systems using a modeling framework in UML. There are seven specifications within EDOC: • Enterprise Collaboration Architecture (ECA) • Metamodel and UML Profile for Java • Flow Composition Model (FCM) • UML Profile for Patterns • UML Profile for ECA • UML Profile for Meta Object Facility • UML Profile for Relationships	www.omg.org/

Standard Name	Objective	Source
Web Ontology Language (OWL-S)	Applications that process content of information rather than presenting information to humans use this standard. It facilitates machine interpretability of web content.	www.w3.org/TR/owl-features/
Web Service Definition Language (WSDL)	WSDL is an Extensible Markup Language (XML) format that describes services as endpoints. It abstractly describes the operations and messages bound by concrete protocols.	www.w3.org/TR/wsdl
Universal Business Language (UBL)	UBL is a normative set of XML schema design rules and naming conventions that coincide with Electronic Business XML (ebXML) Core Components Technical Specifications.	www.oasis-open.org/
Web Services (WS)-Composite Application Models (WS-CAF)*	WS-CAF defines a generic and open framework for applications containing multiple services.	www.oasis-open.org/
Web Application and Compound Document*	This standard is under development and addresses client-side web applications. The standard will focus on: • Hosting environments • Declarative versus script web applications • User interface controls • Parsing data over the network	www.oasis-open.org/
Web Services Modeling Ontology (WSMO)*	WSMO describes aspects of a Semantic Web with four main elements: • Ontology's for terminology • Intention goals • Web service descriptions • Mediators	www.w3.org/

Standard Name	Objective	Source
National Human Service Interoperability Architecture (NHSIA)*	For a number of years, the Health and Human Services (HHS) Administration for Children and Families (ACF) has been working with others to create a conceptual Human Services Information Architecture (HSIA) to produce a technology framework and standards that would result in shared components and shared services among state human service program systems. The goal is to create an environment and infrastructure that would match state and federal data for ACF and the Johns Hopkins University to leverage past development of various federal and state programs, including MITA, National Information Exchange Model (NIEM), Global Reference Architecture (GRA), Service-Oriented Architecture (SOA), and cloud computing.	www.acf.hhs.gov

***Note:** This item is not an official standard or standards organization.

4.4.2 Service Interoperability Standards

There are several standards groups involved in web service security and service interoperability including, but not limited to, the Institute of Electrical and Electronic Engineers (IEEE), the National Institute of Standards and Technology (NIST), Organization for the Advancement of Structured Information Standards (OASIS), World Wide Web Consortium (W3C), and the OASIS Web Services Interoperability Organization (WS-I). The following two (2) tables provide information pertaining to these standards groups. Exhibit 16 provides a listing of generally accepted Service Interoperability Standards that the State Medicaid Agency (SMA) can use.

Note: A third party maintains the internet address for a resource, which may change over time.

Exhibit 16: Service Interoperability Standards

Standard Name	Objective	Source
Extensible Markup Language (XML)	XML is a simple, flexible text format derived from Standard Generalized Markup Language (SGML). This standard provides a variety of associated standards, such as: • Associating Schemas • XQuery	www.w3.org

Standard Name	Objective	Source
	• Efficient XML Interchange • Extensible Stylesheet Language (XSL) Transformations (XSLT) – document transformation and presentation, XSL Formatting Objects (XSL-FO) and eXtended Memory Specification (XMS) Path (XPath) Language	
Simple Object Access Protocol (SOAP) with attachments-Message Transmission Optimization Mechanism (MTOM)	This is a protocol for the exchange of information. It does not define application semantics, but a simple mechanism for expressing application semantics. SOAP with attachments allows a message to contain attachments and provides rules for Uniform Resource Identifier (URI) references.	www.w3.org
Universal Description, Discovery, and Integration (UDDI)	UDDI is a platform independent extensible markup language registry. Originally, proposed as a core web service standard, it interrogates SOAP messages to provide WSDL protocol bindings and message formats.	www.oasis-open.org
Hypertext Transfer Protocol (HTTP) and Hypertext Transfer Protocol – Secure (HTTPS)	These standards are networking protocol for distributed, collaborative, hypermedia information systems. The Internet Engineering Task Force (IETF) and World Wide Web Consortium. (W3C) develop and coordinate these standards. HTTP is a request-response protocol for client-server models – web browsers.	www.ietf.org www.w3.org

Standard Name	Objective	Source
	HTTPS combines HTTP with Secure Sockets Layer (SSL) / Transport Layer Security (TLS) protocol to provide encrypted communications and secure identification.	
Web Service Definition Language (WSDL)	This is a messaging standard in XML format for describing network services as endpoints. The messages are either document-oriented or procedure-oriented information. The messages bind to the concrete network protocol.	www.w3.org
Electronic Business XML (ebXML) Registry	This standard provides interoperable registries and repositories with an interface that enables submission, query, and retrieval.	www.oasis-open.org
Web Services (WS)-Policy	The WS-Policy provides a general-purpose model and corresponding syntax to describe and communicate the policies of a web service. WS-Policy defines a base set of constructs used and extended by other web services specifications to describe a broad range of service requirements, preferences, and capabilities.	www.w3.org
WS-Agreement	Standards are at varying levels of maturity. Some standards are ready for use today, some are emerging, and others are in a stage referred to as “incubating.” The term incubating describes a standard that is developing convergence and may require three (3) to five (5) years before finalization and adoption.	Grid Resource Allocation Agreement Protocol (GRAAP) Working Group https://2020.standict.eu/standards-watch/grid-resource-allocation-agreement-protocol-working-group-graap-wg

Standard Name	Objective	Source
WS-Addressing	This is a key element in the definition of a complete process flow. Middleware and service-delivery companies have an interest in this standard because it is one of the key elements for adding more resource definition information to the URI points. It currently consists of three major pieces: • Core • SOAP binding • WSDL binding with WSDL 2.0	www.w3.org
WS-Reliability	WS-Reliability is a SOAP-based protocol for exchanging SOAP messages with guaranteed delivery, no duplicates, and guaranteed message ordering. WS-Reliability is SOAP message header extensions and is independent of the underlying protocol. It includes a binding to HTTP. The focus is on Business-to-Business (B2B) reliable message delivery. The specification borrows from previous work in messaging (e.g., ebXML) and transport and applies to WS services.	www.w3.org
Defense Advanced Research Projects Agency (DARPA) Agent Markup Language (DAML-S)	DAML-S is a semantic markup language; however, Web Ontology Language (OWL-S) supersedes it.	www.w3.org/Submission/OWL-S/

Standard Name	Objective	Source
Structured Query Language (SQL)	A database computer declarative language designed for managing data in relational database management systems.	www.ansi.org
XML Schema	Other developers who are building their own special-purpose application use these sets of standard application elements. Standard XML Applications consist of the following: • XSL • XSLT • XSL-FO • XML Schema	www.w3.org
Service Level Arrangement Language (SLAng)	SLAng records a common understanding about services, priorities, responsibilities, and other contractual items. The SLAng contains segments for address, service definitions, performance, problem management, customer duties, warranties, disaster recovery (DR), and agreement termination. Specific examples include Web Service Level Agreement Language for Collaborations (WSLA+), Cloud Computing, and Backbone Internet providers.	ieeexplore.ieee.org/document/1204317/

Standard Name	Objective	Source
Web Services Distributed Management (WSDM)	WSDM is a web service standard for managing and monitoring the status of other services. It contains two specifications: • Management Using Web Services (MUWS) defines a basic set of manageability capabilities. • Management of Web Services (MOWS) defines how to manage web services as resources.	www.oasis-open.org
WS-Reliable Messaging (WSRM)	A protocol that allows reliably delivery of SOAP messages to distributed applications.	www.oasis-open.org
Information Technology (IT) Infrastructure Library (ITIL) – IT Service Management Capabilities Level	This is an IT management standardization effort to understand and compare the IT resource utilization and addressing in order to improve the effectiveness and efficiency of the infrastructure used.	https://www.itlibrary.org/
Distributed Management Task Force (DMTF)	DMTF worked on infrastructure management and has developed a series of standards that are gaining acceptance in the system management industry segment.	www.dmtf.org

Standard Name	Objective	Source
Common Information Model (CIM)	CIM is an object-oriented model that describes the conceptual framework for describing management data. CIM messages are in XML format and over HTTP. CIM messages are well-defined request or response data packets used to exchange information between CIM products.	www.dmtf.org
Representational State Transfer (REST) Architecture - Web Services*	A RESTful web service (also called a RESTful web API) is a simple web service implemented using Hypertext Transfer Protocol (HTTP) and the principles of REST. The REST Web is the subset of the World Wide Web (WWW) (based on HTTP) in which agents provide uniform interface semantics – essentially create, retrieve, update and delete – rather than arbitrary or application-specific interfaces, and manipulate resources only by the exchange of representations. Furthermore, the REST interactions are "stateless" in the sense that the meaning of a message does not depend on the state of the conversation.	www.w3.org/TR/ws-arch

4.4.3 Security and Privacy Standards

Exhibit 17 provides a simple listing of generally accepted Security and Privacy Standards that the State Medicaid Enterprise can use.

Note: A third party maintains the internet address for a resource, which may change over time.

Exhibit 17: Security and Privacy Standards

Standard Name	Objective	Source
Federal Enterprise Architecture Security and Privacy Profile (FEA SPP) *	FEA SPP is a scalable and repeatable methodology for addressing information security and privacy from a business-centric perspective. The documentation is at a high level. It does not replace other security and privacy standards but seeks to work across the enterprise.	https://csrc.nist.gov/CSRC/media/Events/ISPAB-SEPTEMBER-2004-MEETING/documents/Sept2004-Fed-Enterprise-Architecture-Security.pdf
National Institute of Standards and Technology (NIST) Initiatives	NIST has a variety of initiatives to address IT standards. Some of these initiatives include: • Computer Security • Cloud Computing • Biometrics • Data and Informatics • Health IT • Information Delivery	www.nist.gov/information-technology-portal.cfm
HIPAA Security and Privacy Rule*	The HIPAA Privacy Rule establishes national standards to protect health information. It requires specific safeguards, establishes personal health information, and sets limits and conditions on the disclosure of information.	www.hhs.gov/ocr/privacy/hipaa/administrative/privacyrule/index.html
WS-Security – WS-I Security Profile	The standard enhances the SOAP messaging to provide message integrity and confidentiality. This supports a variety of security models and encryption technologies. It provides a general approach of associating a security token allowing support for multiple token formats. It describes how to encode binary security tokens and describes the tokens associated with a message.	www.oasis-open.org
Liberty Alliance – Federated Approach*	Federated network identity is the key to reducing the friction between the need to share, the desire for autonomy, and the need for clear identity without centralized control. A federated network identity model will ensure that appropriate parties use critical private information. Liberty Identity Federation Framework (ID-FF) offers a viable approach for implementing such as single sign-on and federated identities.	www.projectliberty.org

Standard Name	Objective	Source
Security Assertion Markup Language (SAML)	SAML defines a framework for exchanging security information between online business partners. SAML defines a common Extensible Markup Language (XML) framework for exchanging assertions between entities in order to define, enhance, and maintain a standard XML-based framework for creating and exchanging authentication and authorization information. SAML requires agreements between source and destination sites about information such as Uniform Resource Locators (URLs), source and destination IDs, certification and keys, and other information in the form of metadata. This standard captures the metadata in a standard format as attributes used by SAML entities. The entities define Identity Providers, Service Providers, Attribute Authorities, Attribute Consumers, Authorization Decision Authorities, and Affiliate Members.	Organization for the Advancement of Structured Information Standards (OASIS) Security Services Technical Committee www.oasis-open.org
Enterprise Privacy Authorization Language (EPAL) – W3C	EPAL goes beyond an application and lays out a standard to protect customers' and citizens' private information enterprise wide. Customer and citizen information should be private and secure based on a global enterprise-wide privacy policy. The enterprise privacy policy defines a set of rules where each rule can allow a set of data users to perform an action in a set of actions on a category in a set of categories for any purpose(s).	www.oasis-open.org
WS-Trust Model	This standard takes the Liberty Alliance Trust Guidance reviewed by a broader, more inclusive community. Most concepts are the same as the earlier Liberty Alliance Trust Guidelines.	www.oasis-open.org
eAuthentication and use of services Object Management Group (OMG) initiative	The OMG initiative is an additional security team with FEA SPP. This team is considering extending and adding additional security and privacy services. The United States Department of Agriculture (USDA) has established an eAuthentication setup. OASIS tested and proved the E-Gov eAuthentication initiative using WS-* standards.	www.idmanagement.gov/

Standard Name	Objective	Source
Public Key Infrastructure (PKI)	This standard describes how communities share policies and authorization schemes based on sharing attributes known as proxy credentials. It enables entity A to grant entity B the authorization right with other entities as if it were entity A. This profile allows limited proxy by providing a framework for carrying policies in Proxy Certificates. X.509 Public Key Infrastructure (X.509) started in 1988. Since that time, several Requests for Comments (RFC) exist for the X.509 standard specifying formats for public key certificates certificate revocation lists, attribute certificates, and certification path validation algorithm. RFC 3820 is the most popular.	www.ietf.org
Health Security	International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) 17799:2000 “Information Technology Code of Practice For Information Security Management”. Standards and Certification Criteria for Electronic Health Records (EHRs). Metadata Standards To Support Nationwide Electronic Health Information Exchange.	www.iso.org/iso/catalogue_detail?csnumber=33441 www.gpo.gov/fdsys/pkg/FR-2011-08-09/pdf/2011-20219.pdf
Unified Modeling Language (UML)sec and Security Engineering Profiles	UMLsec is an extension to the Unified Modelling Language for integrating security related information in UML specifications. This information can be used for model-based security engineering. Most security information is added using stereotypes and covers many security properties including secure information flow, confidentiality and access control. Using an attacker model these properties can be checked on a model level.	www.omg.org
Security and Privacy Data Content Labeling and XML Access Authorization*	Oracle Labeling Security has strong appeal, and there is extensive background information on distributed labeling (e.g., the work at Cornell by Andrew Meyers, et al). This is necessary for cross-line of business security and privacy control.	https://www.oracle.com/database/technologies/security/label-security.html

Standard Name	Objective	Source
Consumer Health Informatics (CHI) Initiatives	CHI is a Kaiser Foundation Model that assists in minimizing the gap between patients and health resources. HITECH and other initiatives have grown from this model. There are a variety of sources for standards: • Electronic Health Records Systems (EHR-S) • Nationwide Health Information Network (NwHIN)	https://www.gao.gov/products/t-aimd-96-134
Minimum Acceptable Risk Standards for Exchanges (MARS-E) 2.2	Framework defines the framework established by the Department for managing the security and privacy of systems deployed to administer the health insurance purchasing aspects of the Affordable Care Act. The scope covers all systems operated by ACA Administering Entities (namely, state Medicaid Agency, state CHIP, state BHP, or an Exchange). Volume II of the MARS-E document suite provides guidance to Administering Entities and their contractors regarding the minimum-level security controls and privacy controls that must be implemented to protect information and information systems for which CMS has oversight responsibility.	https://www.cms.gov/files/document/mars-e-v2-2-vol-1final-signed08032021-1.pdf
Internal Revenue Service (IRS) Publication 1075	This publication provides guidance to ensure the policies, practices, controls, and safeguards employed by recipient agencies, agents, contractors, or sub-contractors adequately protect the confidentiality of FTI.	https://www.irs.gov/pub/irs-pdf/p1075.pdf
Social Security Administration (SSA) Technical System Security Requirements (TSSR)	Electronic information exchange security requirements and procedures for state and local agencies exchanging electronic information with the Social Security Administration	https://medicaid.alabama.gov/documents/2.0_Newsroom/2.4_Procurement/2.4.1_Consolidated/2.4.1.3_2020/2.4.1.3_2020_MEVV_Procurement_Library.pdf

***Note:** This item is not an official standard or standards organization.

4.4.4 Business Enabling Technologies

Exhibit 18 provides a listing of generally accepted standards and specifications for process management involving definition, improvement, and innovation of business processes that drive the Medicaid Enterprise.

Note: A third party maintains the internet address for a resource, which may change over time.

Exhibit 18: Business Enabling Technologies

Standard Name	Objective	Source
Business Process Model and Notation (BPMN) previously known as Business Process Modeling Notation Business Motivation Model (BMM)	The computer industry consolidated all Business Process Model activities under Object Management Group (OMG). The BPMN is a standard for business process modeling that provides a graphical notation for specifying business processes. The BMM specification provides a scheme for developing, communicating, and managing business plans; while BPMN provides a formal mechanism that maps business process to appropriate execution format.	https://www.omg.org/bpmn/ About the Business Motivation Model Specification Version 1.3 (omg.org)
Extensible Markup Language (XML) Forms (XForms)	XForms is an XML application that integrates into other markup languages. XForms gathers and processes XML data using an architecture that separates presentation, purpose, and content. XForms accommodates form component reuse, fosters strong data type validation, eliminates unnecessary roundtrips to the server, and offers device independence.	www.w3.org/TR/xforms11/
Rule Markup Language (RuleML) Initiative	This is an international non-profit organization covering all aspects of web rules and their interoperation. There are Structure and Technical Groups that focus on RuleML specifications, tool, and application development.	Cover Pages: Rule Markup Language (RuleML)
Customer Relationship Management (CRM) Extended Relationship Management (xRM)*	xRM is the principle and practice of applying CRM and is a standardized interchangeable relationship for services.	https://en.wikipedia.org/wiki/Customer_relationship_management

***Note:** This item is not an official standard or standards organization.

4.4.5 Data and Information Standards

Exhibit 19 provides a listing of generally accepted data and information standards and specifications for validation of content and structure.

Exhibit 19: Data and Information Standards

Standard Name	Objective	Source
Accredited Standards Committee X12 (ASC X12)	ASC X12, chartered by the American National Standards Institute, develops, and maintains Electronic Data Interchange (EDI) and Context Inspired Component Architecture (CICA) standards along with Extensible Markup Language (XML) schemas that drive business processes globally.	www.x12.org
Current Procedure Terminology (CPT)	The American Medical Association is the source for official Current Procedural Terminology (CPT) - the most widely accepted medical nomenclature used to report medical procedures and services under public and private health insurance programs.	www.ama-assn.org
Current Dental Terminology (CDT)	CDT is a code set with descriptive terms developed and updated by the American Dental Association (ADA) for reporting dental services and procedures to dental benefits plans.	www.ada.org/
Digital Imaging Communications in Medicine (DICOM)	DICOM standards enable stakeholders to retrieve images and associated diagnostic information, transfer them from various manufacturers' devices and medical workstations.	https://www.dicomstandard.org/

Standard Name	Objective	Source
Fast Healthcare Interoperability Resources (FHIR)	FHIR Release 4.0.1 provides the first set of normative FHIR resources. A subset of FHIR resources is normative, and future changes on those resources marked normative will be backward compatible. These resources define the content and structure of core health data, which developers to build standardized applications.	http://www.fhir.org
Health Level 7 (HL7)	Health Level Seven (HL7) International is the global authority on standards for interoperability of health information technology with members in over 55 countries.	www.hl7.org
International Classification of Diseases (ICD)	The International Statistical Classification of Diseases and Related Health Problems (most commonly known by the abbreviation ICD) is a medical classification that provides codes to classify diseases and a wide variety of signs, symptoms, abnormal findings, complaints, social circumstances, and external causes of injury or disease.	www.who.int
National Council for Prescription Drug Programs (NCPDP)	National Council for Prescription Drug Programs (NCPDP) standards applies to ordering drugs from retail pharmacies. They standardize information between health care providers and pharmacies.	www.ncpdp.org/
National Information Exchange Model (NIEM)	This is a national program supported by the Federal Government that provides a community of users, tools, common terminology, governance, methodologies, and support that enables enterprise-wide information exchange.	www.niem.gov/

Standard Name	Objective	Source
Public Health Information Network (PHIN)	This agency provides various standards and measure definitions including Syndromic Surveillance messaging, EHR Meaningful Use, and Immunization Messaging.	www.cdc.gov/phin/
Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT)	This is the most comprehensive set of multilingual clinical health care terminology. Its aim is to improve patient care through the development of standardized clinical terminology regardless of language.	www.ihtsdo.org/snomed-ct/
Unified Medical Language System (UMLS)	This is a set of files and software collections from health and biomedical vocabularies and standards to enable interoperability between systems.	www.nlm.nih.gov/research/umls/quickstart.html
United States Core Data for Interoperability (USCDI)	The USCDI is a standardized set of health data classes and component data elements for nationwide, interoperable health information exchange. CMS required that payers share the USCDI data they maintain with patients via the Patient Access API, and with other payers via the Payer-to-Payer Data Exchange.	https://www.healthit.gov/isa/united-states-core-data-interoperability-uscdi

4.5 User Interface (UI)

The Agency does not expect the User Interface from different contractors and Commercial-Off-The-Shelf (COTS) software to be identical. However, User Interfaces must meet the following requirements:

1. Have configurable settings for:

    a. Field labels for business data. Example: selecting one of the following labels for use throughout the system: “Recipient,” “Client,” or “Member” to represent an individual covered by the Agency health care services

    b. Images to provide similar graphics in each application

    c. Colors to implement a similar look for each application

2. Must be web browser-based and commonly accessable to end-user devices
3. Shall be compatible with multiple standard browsers in accordance with the Agency's ISO Standards, including but not limited to:
    a. Google Chrome
    b. Microsoft Chromium Edge (Commonly referred to as Edge)
    c. Apple Safari
    d. The contractor must provide any specific browser configurations required
4. Have a responsive design to support a variety of devices and screen sizes. Unless specified in the Request for Proposal (RFP), this does not mean that a mobile app is needed
5. Comply with Americans With Disabilities Act (ADA) Section 508
6. Include page level help
7. Display field level help without taking the user away from the current page, when a user hovers over a field or label
8. Cache the following file types in the browser and reuse throughout the session or until the non-cached file changes: style sheets, images, and JavaScript and other scripts
9. Use consistent design and layout throughout the interface
10. Ensure server errors displayed in the UI, either from the application or the web server, contain user friendly descriptions meaningful to business users with instructions on what the user should do.
11. Ensure browser scripting errors, displayed in the UI, contain user friendly descriptions meaningful to business users with instructions on what the user should do
12. Recover from errors without requiring users to delete local cached files and/or cookies
13. Include capabilities that can be turned on to capture and log response time measured at the browser. These capabilities must report the information to a server-based data collection system
14. Ensure URLs for browser applications, in all environments other than dedicated development environments, use fully qualified domain names (FQDN), resolvable through public Domain Name System (DNS) servers
15. Ensure the FQDNs, used in URLs for hosts, map to public, routable, Internet Protocol (IP) addresses. Addresses in the Private Address Space defined by Request for Comment (RFC) 1918 are not permitted

User interfaces must maintain compliance with the following standards:

1. Hypertext Markup Language (HTML) Living Standard specification from Web Hypertext Application Technology Working Group (WHATWG)
2. Most recent major version of Hypertext Markup Language (HTML), currently HTML5 specification from W3C

3. Most recent major version of Cascading Style Sheets (CSS), currently CSS3 module specifications from W3C

4. The annual ECMAScript standard published by Ecma International (Applies to JavaScript)

The contractor is responsible for all integrity scans of the code and reporting of the results to the Agency's Chief Information Security Officer (CISO). The contractor is also responsible for resolutions according to any federal or state guidelines.

4.6 Fault Tolerance

Systems must catch all errors and return meaningful errors for a user to report. Exception handling should be implemented using constructs such as "Try…Catch" at each appropriate level in the code, ensuring the resulting error is presented to the appropriate layer for error messaging and logging. Applications must recover without unhandled exceptions, application crashes, or other failures that terminate processing.

Applications, developed and purchased, must be sufficiently fault tolerant to minimize the need for any user or process to repeat entry or processing due to an application or data error.

User Applications

1. If a user enters invalid data in a field, the application must clearly indicate which field has invalid data and provide information to the user about the problem. Message content must describe the rule that has not been met. For example, display "Name must contain only alphabetic characters" rather than "Invalid entry".

2. Application must retain user input until a response is returned to a user. The application must respond in a way that the user can submit the information again without reentering the data.

3. Applications must catch errors in a way to present meaningful information to a user or an interface. Server-level errors should use a common error page and display text indicating the action a user must take. This includes providing an error code and a human-readable description to report.

Interfaces

1. Each interface specification must define how errors are handled by the interface. Each system must implement error handling consistently across all applicable interfaces

2. Interfaces encountering data validation errors must use pre-defined codes or messages to indicate the data that failed validation

3. Interface specifications must define which system is responsible for resolution of each error

4. Interfaces must return information identifying the transaction that produced an error

All Application Code

Applications must use a facility to report errors to external tools for automated monitoring. The application must be able to report to multiple external applications to support both system-level and enterprise-level monitoring.

4.7 Performance

The application must process the first transaction received after a restart at the same speed all other transactions are processed. Any first transaction performance penalty, such as compilation or cache initialization, is not permitted.

Applications must include configurable performance measurement capability. Configuration will determine the level of detail for the performance measurements.

Applications must include configurable capability to generate log messages when errors, performance problems, and other events occur. Logged messages must include date and time, and indicate which process generated the message.

The Agency recommends building applications for performance through optimized frameworks, caching, load balancing, multi-threading, and automated scaling.

Additional requirements are in the Outcomes and Performance section.

4.8 COTS Software

The Agency expects all systems to interface with COTS software for services that are leveraged by multiple modules in the MES (leveraged services). Systems may also make use of other COTS tools to support its own processes, provided that the capabilities are not available in existing leveraged services.

All systems using leveraged services or dedicated COTS software must meet the following requirements to ensure flexibility and minimize rework for updates to the COTS software's interfaces or changing to another COTS software altogether.

1. Each system must have internal functionality, independent of the COTS software's API, that manages communication communication to the COTS software's API
2. Systems must support versioning of interfaces with the COTS software's APIs
3. Systems must use authentication when communicating with a COTS software's API
4. When required by the Agency in either the TRA or interface specification, the systems will pass information about the end user initiating the request. This information may be used by the COTS software to validate authorization to specific services and data

The following leveraged services are expected to be provided via COTS tools and will also satisfy the outcomes listed by EA-f: MMIS Concept of Operations:

1. Identity Management – A solution that creates, modifies, disables, and deletes user accounts and their profiles across the MES.
    a. Tools TBD
2. Health and Performance Monitoring – The process of monitoring system health and stability and presenting system health metrics for management of the MES.
    a. Tools TBD
3. MES Portal – A login and registration functionality and initial landing page for the MES.
    a. Tools TBD

4. Centralized Service Desk Management Tool – A single view into all service requests across the MES.
    a. Tools TBD

4.9 Testing

All applications must be unit tested prior to release to system test environment. Unit testing must include validation that each interface, application, data transformation and workflow are functional.

Each contractor must create their own data for unit testing. Contractors hosting a web service must publish example transactions with each specification. Contractors should share test data with each other when possible.

Unit testing must include unit testing of interface components. Each contractor is responsible for building any tools, service, automation, or other facilities to test their interfaces. The unit tests must validate that the interfaces can create and receive data per the interface specification, including processing of errors.

Each contractor is responsible for performing all system and user acceptance testing.

Contractors must conduct performance testing throughout the development cycle. Testing must identify and resolve performance problems prior System Integration Testing.

4.10 Implementing Reuse in the Application Architecture

Contractors for the Medicaid Enterprise Systems will utilize methods and patterns to ensure the re-usability of inputs, outputs, and integration points for business services. This will be accomplished by way of APIs, reports, transmission standards, etc. that allow for data elements to flow unimpeded throughout the system to its destination for consumption and use by end users.

5 Technology Architecture

This section provides a blueprint for hosting of the MES, including permitted hosting mechanisms, containerization strategies, etc. This section will also cover hosting and infrastructure performance management and messaging, monitoring, and logging of problems. It will cover requirements related to Business Continuity and Disaster Recovery.

5.1 Platform and Hosting

The preferred hosting platform for MES systems is a public or private cloud. The Agency expects that contractors hosting in a cloud environment will have more infrastructure flexibility, scaling, and upgrade capabilities than hosting in a private data center.

Cloud hosting models can be classified into one of three categories: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS)

- SaaS applications are deployed in a cloud environment for multiple customers. The Agency expects that SaaS solutions will require the least amount of time during design, development, and implementation (DDI) to deploy applications and environments.

- PaaS and IaaS solutions are the basis for building a cloud application using a combination of services offered by a cloud hosting vendor. While this approach provides more flexibility, the

Agency expects a PaaS or IaaS solution will require building dedicated environments during DDI, which will require a longer implementation period and higher cost to the Agency.

Contractor-managed data centers are considered by the Agency to be on-premise hosting. On-premise hosting also includes solutions run in a shared data center with any combination of shared and dedicated resources. On-premise hosting should be avoided, as technical enhancements and technology refreshes are constrained by capital expense. The capital expense constrains flexibility and timeliness of changes to the hosting environments.

Cloud-based SaaS solutions are the Agency 's preferred hosting framework. Cloud-based PaaS and IaaS solutions are potential alternatives.

5.2 Environments

The hosting and application architecture for any MES application must support the full range of System Development Life Cycle (SDLC) activities. These include development, user acceptance and integration testing, performance testing, and production. These activities traditionally are performed in dedicated environments. The Agency does not require an environment for each activity, but the environments must be architected to meet the requirements for the activity.

The Agency is not prescriptive about how the activities are separated. One approach is to dedicate a full hardware and software infrastructure to an environment. Alternately, some modules may be able to leverage tools across environments. The activities may need dedicated user interfaces, business applications, and databases to test different versions of code and test data. Some components, such as infrastructure, COTS tools, and integration applications, can be leveraged with a design that delivers the required separation of activities between environments.

When leveraging single instances of tools to support multiple environments, a plan to update and test the tools must be part of the architecture. For example, a COTS tool for a service bus may provide integration capabilities for activities in multiple environments. Upgrading the software in both environments may interfere with testing. Appropriate testing in the integration environment is needed before changes are moved to other environments.

The following table presents the activity, a description, and the need for the capability over the lifetime of the program.

Exhibit 20: Activities That May Require Separate Environments

Activity	Activity Description	Project Phase
Development	Used by staff to create code and configurations and unit test to validate proper results prior to integration testing.	DDI and Operations
System Integration Test	Each system in the MES is integrated and tested. This includes the flow of data between systems, contractors, and organizations. It is heavily dependent on the integration layer.	DDI and Operations

Activity	Activity Description	Project Phase
User Acceptance Test	Users execute tests to validate correct functionality and configuration.	DDI and Operations
Conversion Testing	Build or configure tools to convert data from an existing system into data for a new system	DDI
Training	Define in each RFP. Training cannot impact changes or testing for any other activity. This activity occurs in the later phase of DDI and periodically during Operations.	DDI and Operations
Performance Test	Staff conducts tests to measure performance of the system or system components. This activity occurs throughout DDI and periodically during Operations.	DDI and Operations
Operational Readiness Test	Validate that the technical components needed for discrete business functions and non-functional technical requirements.	DDI
Parallel Policy Test	Prior to implementation, run data in both old and new systems to validate the outputs match in both systems for scenarios in which business policy does not change.	DDI
Production	Live use following completion of testing.	DDI and Operations

5.3 Environment Requirements

Disaster Recovery requirements included in the tables in the following subsections are illustrative to show the relative priority of the recovery. The requirements for each system will be defined in the RFP.

5.3.1 Development and Unit Testing

Exhibit 21: Development and Unit Test Requirements

Characteristic	Requirement
Resources	Contractor determines sizing to meet development and unit test requirements
Single Sign-On (SSO)	Not used
PHI and PII	Not permitted
Points of Access	Contractor
Integration	None
File Transfer	Capability in place
Job scheduling, COTS tools, System, and application monitoring	As needed to support development and configuration testing of the module provider
Estimated DR Recovery Requirement	Three (3) days

5.3.2 System Integration Test (SIT)

Exhibit 22: System Integration Test Requirements

Characteristic	Requirement
Resources	25% of production capacity. Resources must exist to fully test system to system integration, COTS tool configuration, and networks
Connection to other systems and contractors	All
SSO	Required
PHI and PII	Not permitted
Points of Access	Agency, Contractors, Virtual Private Network (VPN)
Integration	Full features to support integration testing. All integration capabilities must support changes without impacting any other environment
Job scheduling, COTS tools, System, and application monitoring	Dedicated, full features to support integration testing
Estimated DR Recovery Requirement	Three (3) days

5.3.3 User Acceptance Test (UAT)

Exhibit 23: User Acceptance Test Requirements

Characteristic	Requirement
Resources	100% of production capacity. Resources must be able to scale to support testing of production level data volumes.

Characteristic	Requirement
SSO	Required
PHI and PII	Permitted. Adequate controls required to protect PHI and PII
Points of Access	Any access point that may be used in production
Integration	Full features, including support for System-to-System Testing and end-to-end (E2E) Testing. All integration capabilities must not be impacted by any activity other than User Acceptance Testing (UAT)
Job scheduling, COTS tools, System, and application monitoring	Dedicated, full features to integration needed for E2E testing of business functions
Estimated DR Recovery Requirement	Three (3) days

5.3.4 Conversion Test

Exhibit 24: Conversion Test Requirements

Characteristic	Requirement
Resources	25% - 100% of production data capacity. Resources must scale to support production scale conversion tests
SSO	Not required
PHI and PII	Permitted. Adequate controls to protect PHI and PII are required
Points of Access	Contractor sites
Integration	Define in each RFP
Job scheduling, COTS tools, System, and application monitoring	COTS tools as needed to support conversion
Estimated DR Recovery Requirement	Three (3) days

5.3.5 Parallel Policy Test

Exhibit 25: Parallel Policy Test Requirements

Characteristic	Requirement
Resources	50% of production capacity
SSO	Not required
PHI and PII	Permitted. Adequate controls to protect PHI and PII are required
Points of Access	Any access point that may be used in production
Integration	Define in each RFP
Job scheduling, COTS tools, System, and application monitoring	Dedicated, full features to integration needed for E2E testing of business functions

Characteristic	Requirement
Estimated DR Recovery Requirement	Three (3) days

5.3.6 Performance Test

Exhibit 26: Performance Test Requirements

Characteristic	Requirement
Resources	100% of production capacity
SSO	Not required
PHI and PII	Permitted. Adequate controls to protect PHI and PII are required
Points of Access	Any access point that may be used in production.
Integration	Full features. All integration capabilities must support changes without impacting any other environment
Job scheduling, COTS tools, System, and application monitoring	Dedicated, full features to support integration needed for E2E performance testing
Estimated DR Recovery Requirement	Three (3) days

5.3.7 Training

Exhibit 27: Training Requirements

Characteristic	Requirement
Resources	25% of production capacity.
Connection to other systems and contractors	Define in each RFP
SSO	Required
PHI and PII	Not permitted
Points of Access	Define in each RFP
Integration	As required to support full user experience during training
Job scheduling, COTS tools, System, and application monitoring	As required to support full user experience during training
Estimated DR Recovery Requirement	Five (5) days

5.3.8 Production

Exhibit 28: Production Requirements

Characteristic	Requirement
Resources	Full resources for production scale and meeting KPIs and SLAs
Connection to other systems and contractors	Defined in RFP for each Module
SSO	Not required
PHI and PII	Permitted. Adequate controls to protect PHI and PII are required
Points of Access	Defined in RFP for each Module
Integration	Full features
Job scheduling, COTS tools, System, and application monitoring	Full features
Required DR Recovery	Defined in RFP for each Module

5.4 Services Management

All systems that are part of the MES must be supported through Information Technology Service Management (ITSM) processes and tools. The supporting organization must use an ITSM framework, preferably ITIL V4.

The organization's service management process must include tools for release management and incident management that account for a multi-module system with multiple organizations supporting the system. Contractors are expected to work together on resolution and corrective action for incident management.

Release management should be automated to support changes in multiple environments with minimal impact on system availability. With multiple, independent modules in the MES, release management processes must support other systems changing at different times without synchronization. Release management must account for running multiple versions of services, interfaces, and other system components to support transition periods that may vary for each module.

Incident management tools must have capabilities to interface with tools run by other organizations. The interfaces must include bi-directional exchange of data.

The requirements defined for each module include both processes and documentation to demonstrate effective service management. The ITSM framework must be the basis for the processes and documentation.

Release management during the DDI will need to be coordinated by the contractors. All production release management will need to be handled through the CCB.

5.5 Leveraged Services

This section presents the architecture for services that are leveraged by multiple modules. All modules requiring capabilities available through leveraged services must utilize the leveraged services rather than other solutions. Exceptions to using the above services will require EAB approval.

The following leveraged services are expected to be provided via COTS tools:

1. Identity Management – A solution that creates, modifies, disables, and deletes user accounts and their profiles across the MES.

    a. Tools TBD

2. Health and Performance Monitoring – The process of monitoring system health and stability and presenting system health metrics for management of the MES.

    a. Tools TBD

3. MES Portal – A login and registration functionality and initial landing page for the MES.

    a. Tools TBD

4. Centralized Service Desk Management Tool – A single view into all service requests across the MES.

    a. Tools TBD

Requirements defined in any RFP supersede any conflicting requirements in this section.

5.5.1 Integration Services

Integration services include features for sending and receiving messages between systems. The features include processing both all query / response and publish / subscribe messages. Queries are forwarded to the appropriate system, and a response is returned to the system originating the query. The integration service manages a repository of published message definitions and the systems that subscribe to them. When a message is published, the integration services accept the message and then send copies to all the subscribed systems.

Integration services have been traditionally provided by a hosted COTS tool and called an Enterprise Service Bus. SaaS-based applications are also available, although many contractors use terms other than ESB to describe the functionality.

The Agency prefers SaaS solutions over hosted COTS tools, provided all required features are available, including:

- Route service calls and responses
- Deliver publish / subscribe messages
- Provide capability to configure parameters of service end points for incoming messages
- Provide capability to configure destinations for message delivery
- Configure message subscriptions
- Provide guaranteed delivery of messages when required by business needs
- Measure duration of request / response process for all transactions
- Provide ability to exchange data with a variety of COTS tools and SaaS service delivery applications

- Provide ability to configure a series of processing activities (orchestration) for messages
- Provide ability to perform rules-based transformation of message data based on sender, receiver, and message type
- Provide ability to remove PHI data from messages using rules based on the sender, receiver, and message type

Transformation of data must occur outside of the interface services layer. The capability will exist for a limited number of exceptions.

The orchestration capabilities must not be used when the orchestration is only performed for the benefit of a single system. In these scenarios, the orchestration must occur with the benefitting system.

All systems must use the leveraged interface services for message exchange.

5.5.2 Communication / Correspondence Management

MES Technical services may include communication and / or correspondence management.

The following requirements apply to all email sent on behalf of the Agency, regardless of the capability provided by the technical services:

- The email return address must be in the medicaid.alabama.gov domain or another domain identified by the Agency. Use of other domains may be approved if the solution meets Agency requirements and is approved by the CCB as recommended by the EAB.
- Outgoing email must be sent through medicaid.alabama.gov email servers. Email may not be sent from outside of medicaid.alabama.gov domains with a medicaid.alabama.gov address in the “From:” field. Use of other domains may be approved if the solution meets Agency requirements and is approved by the CCB as recommended by the EAB.
- Email content should also minimize the likelihood that the email gets identified as spam.

Responsibility for successful delivery of email, without being identified as spam, belongs to the contractor.

5.6 COTS

This section provides guidance on use, selection, and implementation of COTS tools.

COTS tools must be evaluated and selected based on providing capabilities to the entire MES rather than a single module. Exceptions to this standard require approval of the CCB.

COTS tools selected for use in the MES must have the following capabilities:

1. All integration with the COTS tool must take place using APIs. This applies to both business and technical functions using the COTS tool and to interfaces initiated by the COTS tool

1. COTS tools must obtain credentials from the Agency Identity and Access Management (IAM) system. The SAML 2.O standard will be used, and the user ID and roles will be included in the encrypted SAML token. The COTS tool must use the roles to determine what functions and data the user may access. The IAM system will be a COTS or SaaS tool

2. The Agency prefers SaaS COTS solutions over non-SaaS cloud-based systems and contractor-hosted systems, provided the SaaS solution meets all requirements

3. Business applications will exchange information with the COTS tool using APIs. Applications must not use the COTS tool by transferring the user to a COTS User Interface (UI). The CCB may approve exceptions for complex processes using a COTS tool
4. Administrative functions to maintain and configure the COTS tool may use the COTS UI
5. Evaluation criteria for COTS tools must include the ability to be used by the MES enterprise or scale large enough to support the MES Enterprise
6. COTS tools must meet the requirements for performance measurement, monitoring, logging, and performance reporting defined in the TRA
7. COTS tools must meet all security compliance requirements
8. COTS tools must include licenses needed to meet all Agency requirements
9. COTS APIs should be simple and non-proprietary. This may require an additional API layer, as part of the COTS tool, that simplifies the API and hides the native complexity from other systems
10. COTS APIs must be secured as defined in the Application Reference Architecture
11. Outcome and performance requirements must be defined prior to selection of the COTS tool. The deployment and/or configuration of the tool must meet these requirements

COTS and Open-Source software must be maintained, patched, and updated to ensure functionality and security. Both COTS and Open-Source software must be supported via a maintenance agreement with the contractor providing the software or another organization qualified for production system support.

5.7 Network Configuration and Accessibility

5.7.1 Naming and Addressing

Any application, service, host, cloud function, or other interface accessed by a user or system must meet the following requirements:

1. URL endpoint must use an SSL certificate signed by a Certificate Authority. The Certificate Authority must be publicly accessible from the internet and not internal to a company or other organization.
2. The endpoint must use a Fully Qualified Domain Name (FQDN) in the URL. The FQDN must be resolvable through the public internet DNS and not DNS internal to a company or other organization.
3. Systems must use IP addresses outside of the Private Address Space defined in RFC 1918. The addresses used must be accessible from use locations and other system locations.
4. These requirements must be met for both incoming and outgoing connections.

A gateway and proxy server or other method of exposing a limited number of resource connections may be used to meet the requirements.

These requirements do not apply to any network connections used internally by a system that does not exchange information with another system or provide a user interface for users outside the contractor’s organization. They also do not apply to development environments that are accessed only by the contractor.

5.7.2 SMTP Email Servers

Simple Mail Transfer Protocol (SMTP) email servers must be configured to avoid designating email as spam or blacklisting the sender. The email servers must:

1. Use authentication for all client connections to the server for sending and receiving email.
2. Use encryption for client connections compliant with the Minimum Acceptable Risk Standards for Exchanges (MARS-E) security standards.
3. The From address for all outgoing mail must match the email server domain and address specified in DNS.
4. Include A, MX, PTR, SPF, DKIM, and DMARC DNS records for the mail server and mail server domain.

A SaaS-based email service must comply with these standards. For any type of email service, including self-hosted, it is the contractor's responsibility to ensure that emails are not identified as spam because of technical configuration, blacklisting, or use of a service with a reputation for sending spam.

5.7.3 Testing

The Agency requires that UAT systems have the same accessibility to end users as the production system, including access from the public internet. Accessibility to all testing, training, and other environments must be made available on the internet upon request from the Agency.

6 Outcomes and Performance

The MES effectiveness is based on the business outcomes during operations. Each system must measure and report on operational performance and identify ways to improve outcomes.

6.1 Outcomes Based Requirements

Each system will require measuring data for KPIs. The KPIs will be metrics that influence or define system outcomes.

A business outcome may be to increase satisfaction for providers calling an Agency help desk. Systems supporting this capability must measure the amount of time it takes to return a response through the User Interface to the help desk staff supporting the provider.

Another outcome may be to reduce the amount of time to process a provider enrollment application. The applications are best suited to measure start and end times and report the measurements.

6.2 Performance Management

Contractors must measure system performance for use in monitoring, analysis, and reporting. The following items must be measured across all systems:

- Duration of all service requests, measured at the server
- The duration of a request as experienced by the user between submitting a request and the response completely rendered in the browser
- Quantity of service requests and the time they are received
- Duration of automated and scheduled processes
- Number of concurrent users

- Peak transaction volumes
- Quantity of successful and unsuccessful transactions
- Service Response Times under Performance and Stress Loads
- Response times of service calls to external systems

Contractors must identify and measure additional KPIs that indicate the performance of system components to meet all state and federal requirements.

The Agency requires contractors to measure user experienced response time, despite a common perception that they have no control over what happens outside of their system. Contractors have control of the size of the request and response messages, both of which impact the speed of network delivery. Contractors also control the technical makeup of web pages and therefore how efficiently they are built for browser processing. Automated tools are available for measuring the desktop response time and returning information to a contractor data collection system. The Agency recognizes the complexity of systems and factors impacting performance. Contractors must identify, measure, and maintain performance of the components that they control and influence.

6.3 Monitoring

MES components require operational monitoring 24x7. Monitoring assesses whether the system is performing normally. Monitoring can also proactively identify a degradation in system performance and respond appropriately to the problem.

RFPs must specify specific monitoring requirements and ask contractors to identify other monitoring they will use to ensure systems meet KPIs.

Contractors must implement automated monitoring of all systems. Monitoring must assess, in real time or near real time, whether the system is operating normally.

6.4 Logging

Each system must have logging capabilities integrated into the application, technical services, and infrastructure. The system must be configurable, in real time, to specify the level events that are logged. The following capabilities are required:

- Logging processing errors
- Logging abnormal conditions detected via monitoring
- Logging periodic processing metrics
- Configurable automated actions when receiving a specific message
- Configurable escalation for log messages indicating abnormal processing to ensure timely action
- Configurable routing of specific log messages to a leveraged service used for consolidating events
- Accepting messages from an external source and triggering actions

The Agency recommends use of a COTS tool for logging, forwarding, and analysis rather than a custom-built tool.

6.5 Performance Reporting

Performance data must be reported on a regular basis. Contractors must include automated creation of performance reports in their solution. Use of dashboards is encouraged.

6.6 Recommendations

The Agency recommends configurable measurement capabilities integrated into applications and infrastructures.

The following recommendations support high levels of performance:

1. Use applications frameworks built for high performance.
2. Provide highly efficient database interfaces.
3. Implement data caching.
4. Design interfaces for high levels of performance.
5. Use load balancing for application and infrastructure components.
6. Provide automated scaling of resources for application and infrastructure.
7. Implement networks that support the performance requirements.
8. Make use of cloud-based tools to measure performance.
9. Measure performance frequently in the development and testing life cycle.

7 Security and Compliance

7.1 Privacy and Protection

Systems must protect PHI as defined by HIPAA. Systems must meet additional Federal and State requirements to protect information. The MARS-e and the NIST frameworks, upon which MARS-e is built, contain many of the security requirements.

7.2 Identity and Access Management

Each module of the MES must securely integrate with the MES Identity Management System. The Agency IAM will provide services to authenticate a user, pass identity information to business applications, and technical services to deliver single sign-on environment.

7.3 Information / Data Security

The following list presents a selection of high-level requirements. The list of security requirements is included with each RFP and include such activities as:

- Meet all State and federal regulations

- Perform security scanning of code and infrastructure
- Perform penetration testing on a regular basis
- Create and maintain a system security plan
- Develop a disaster recovery plan and conduct drills on a regular basis
- Maintain hardware, software, and infrastructure and ensure software versions are current
- Provide change and release management
- Provide notification of security incidents and breaches
- Conduct audits on a regular basis

Appendix A. Acronyms/Glossary

For a complete list of Acronyms and Glossary of Terms, please reference the AMMP Acronyms and Glossary.