Enterprise Messaging Infrastructure

Vitria Interface Engine

Enrollment Service Redesign

Service Integration Design Document



**Department of Veterans Affairs**

**April 2016**

**Version 0.05**

Revision History

| Version No. | Description/Comments | Author | Delivery Date |
| --- | --- | --- | --- |
| 0.01 | Initial document creation and review |  | 09/23/2015 |
| 0.02 | Peer review and updated for delivery to VA |  | 12/04/15 |
| 0.03 | Update contents and diagrams |  | 03/01/2016 |
| 0.04 | Updated content and message flow diagrams |  | 03/05/2016 |
| 0.05 | Tech writer review |  | 03/30/2016 |

Table of Contents

[1.0 Introduction 1](#_Toc447058739)

[1.1 Purpose 1](#_Toc447058740)

[1.2 Scope 1](#_Toc447058741)

[1.3 Audience 2](#_Toc447058742)

[1.4 References 2](#_Toc447058743)

[2.0 Interface Requirement 2](#_Toc447058744)

[2.1 Business Unit 2](#_Toc447058745)

[2.2 Service Level Agreement Metrics 3](#_Toc447058746)

[2.3 Message Type Metrics 4](#_Toc447058747)

[2.4 Logical System Overview 5](#_Toc447058748)

[2.5 Logical Deployment Overview 7](#_Toc447058749)

[3.0 Nominal ESR Message Flow 8](#_Toc447058750)

[4.0 ESR Design 11](#_Toc447058751)

[4.1 Architecture Deviations 11](#_Toc447058752)

[4.2 Pattern 12](#_Toc447058753)

[4.3 Protocol 12](#_Toc447058754)

[4.4 Message Routing 13](#_Toc447058755)

[4.5 Transformation 14](#_Toc447058756)

[4.5.1 Protocol Transformation 14](#_Toc447058757)

[4.5.2 Data Transformation 14](#_Toc447058758)

[5.0 Implementation Details 15](#_Toc447058759)

[5.1 ESR VistA Interface 15](#_Toc447058760)

[5.1.1 ESRVistAReceiver 15](#_Toc447058761)

[5.1.2 ESRVistASender 16](#_Toc447058762)

[5.2 ESR 17](#_Toc447058763)

[5.2.1 ESRReceiver 18](#_Toc447058764)

[5.2.2 ESRSender 19](#_Toc447058765)

[5.2.3 ESRRouters 20](#_Toc447058766)

[5.3 VBA VistA Interface 21](#_Toc447058767)

[5.3.1 ESRVBAReceiver 22](#_Toc447058768)

[5.3.2 ESRVBASender 23](#_Toc447058769)

[5.4 Error Handling For ESR/VistA/VBA Message Delivery 23](#_Toc447058770)

[5.5 Project Configuration File 24](#_Toc447058771)

[5.6 Queue Details 25](#_Toc447058772)

[6.0 Timeline 27](#_Toc447058773)

[7.0 Acronyms 27](#_Toc447058774)

[8.0 Architecture Design Decisions 30](#_Toc447058775)

[Appendix A. Messages 31](#_Toc447058776)

[Appendix B. Message Mapping 32](#_Toc447058777)

[Appendix C. IBM PMR’s 33](#_Toc447058778)

[Appendix D. Approval Signature 34](#_Toc447058779)

List of Tables

[Table 1 – Software Interfaces 2](#_Toc447295444)

[Table 2 – ES Business Unit 2](#_Toc447295445)

[Table 3 - Vista Business Unit 3](#_Toc447295446)

[Table 4 -- VBA Business Unit 3](#_Toc447295447)

[Table 5 -- SLA Metrics 3](#_Toc447295448)

[Table 6 -- Message Type Metrics 4](#_Toc447295449)

[Table 7-- VistA to eMI Interface 12](#_Toc447295450)

[Table 8 -- eMI Interface to ESR 13](#_Toc447295451)

[Table 9 -- VBA to eMI Interface 13](#_Toc447295452)

[Table 10 -- ESR Queue Routing Table (Sample) 13](#_Toc447295453)

[Table 11 -- ESR to VistA and VBA Routing Table (Sample) 14](#_Toc447295454)

[Table 12 -- Protocol Transformation in ESR 15](#_Toc447295455)

[Table 13 -- JMS Header Properties 19](#_Toc447295456)

[Table 14 -- JMS Header Application Properties 19](#_Toc447295457)

[Table 15 – JMS Header Properties 20](#_Toc447295458)

[Table 16 – JMS Properties 21](#_Toc447295459)

[Table 17 – Project Configurable Parameters eMI ESR 25](#_Toc447295460)

[Table 18 -- Project Configurable Parameters eMI ESRVistAInterface 25](#_Toc447295461)

[Table 19 -- Project Configurable Parameters eMI ESRVBAInterface 26](#_Toc447295462)

[Table 20 – Queues 26](#_Toc447295463)

[Table 21 – Acronyms 29](#_Toc447295464)

List of Figures

[Figure 1 – ESR eMI Logical System Overview 5](#_Toc447295701)

[Figure 2 – ESR eMI Deployment Overview 7](#_Toc447295702)

[Figure 3 – VistA to ESR Sequence 8](#_Toc447295703)

[Figure 4 – VistA to ESR Sequence 9](#_Toc447295704)

[Figure 5 – VBA to ESR Sequence 10](#_Toc447295705)

[Figure 6 – ESR to VBA Sequence 11](#_Toc447295706)

[Figure 7– VistA Receiver Flow 15](#_Toc447295707)

[Figure 8 – HL7 Input Sub-Flow 16](#_Toc447295708)

[Figure 9 – VistA Sender Flow 17](#_Toc447295709)

[Figure 10 – HL7 Output Sub-Flow 18](#_Toc447295710)

[Figure 11 – ESR Receiver Flow 19](#_Toc447295711)

[Figure 12 – ESR Sender Flow 20](#_Toc447295712)

[Figure 13 – Identify VistA Routing Information Message Flow 22](#_Toc447295713)

[Figure 14 – Identify ESR Routing Information Message Flow 22](#_Toc447295714)

[Figure 15 – VBA Receiver Flow 23](#_Toc447295715)

[Figure 16 – VBA Sender Flow 24](#_Toc447295716)

[Figure 17 ESR Implementation Timeline 28](#_Toc447295717)

Introduction

The Enrollment Service Redesign (ESR) Enterprise Messaging Infrastructure (eMI) application enables the exchange of patient enrollment queries and responses between the Veterans Health Information System and Technology Architecture (VistA) system and the ESR application. These queries and responses include, but are not limited to, information related to patient demographics, eligibility, means test, income, insurance, and enrollment eligibility. The ESR application also facilitates the exchange of Health Level 7 (HL7) messages between the Veterans Benefits Administration (VBA) and ESR application.

Purpose

The purpose of this document is to describe the interface specifications of the eMI ESR interface to the VistA systems, the ESR system, and the VBA system. This document also provides the detailed design of the eMI ESR message flow and includes protocol, transformation, broker patterns, happy path, and exception details.

Scope

The scope of this document is to describe the message flow between the VistA systems, the ESR system, and the VBA system. This document serves as both requirement and design specifications for the ESR message flow.

The eMI ESR application receives HL7 messages from the VistA application and sends HL7 messages to the ESR. The VistA system sends the enrollment-related data and queries as HL7 messages, either as individual messages with message headers (MSH) or as batch messages with batch headers (BHS), to the ESR application. The ESR application processes the request and sends application acknowledgements and/or query responses back to the VistA system. The ESR application may also transmit updates, such as a patient’s address change or changes to a patient’s financial data, to the VistA system.

The scope items include, but are not limited to:

1. Message Types
2. Validation
3. Ports and Protocol
4. Transformation
5. Error Handling
6. Routing
7. Security

The details of the above are described in the following sections. The common architectural specifications, such as security, logging, exception handling, etc. that apply to all message flows, are defined in eMI Software Design Document (SDD).

The definitions of the HL7 messages that are generated by VistA, ESR, or VBA are not in the scope of this document. Details of the IBM Enterprise Service Bus architecture (eMI), physical addresses, and system specifications are not in the scope of this document.

Audience

This document provides implementation details for project owners and serves as a blueprint for managers, architects, developers, and testers building the system. It is assumed that the reader has a moderate knowledge of Message Broker and HL7.

References

eMI SDD is accessible on the VA eMI SharePoint site.

Interface Requirement

Table 1 lists the software interfaces that are implemented.

Table 1 – Software Interfaces

| Application | Interface |
| --- | --- |
| ESR 🡪 eMI Enterprise(ENT) at Austin Information Technology Center (AITC) | Utilizes Java Message Service (JMS). eMI pulls JMS messages from ESR outbound messages. |
| eMI Regional Data Center (RDC) 🡪 VistA | Utilizes Socket to push the HL7 messages from eMI to VistA |
| VistA 🡪 eMI RDC | Utilizes Socket to push the HL7 messages to eMI |
| eMI ENT (AITC) 🡪 ESR | Utilizes JMS to push the HL7 messages from eMI to ESR |
| VBA 🡪 eMI ENT (AITC) | Utilizes Socket to push the HL7 messages to eMI |
| eMI ENT (AITC) 🡪 VBA | Utilizes Socket to push the HL7 messages from eMI to VBA |

Business Unit

Data moves between the VistA and the ESR systems, and between the ESR system and the VBA. Table 2, Table 3 and Table 4 list the point of contact (POC) information for those systems.

Table 2 – ES Business Unit

|  |  |  |  |
| --- | --- | --- | --- |
| ES Business Unit | | | |
| POC | Role | Email | Phone |
|  | Business Owner |  |  |
|  | Business Owner |  |  |
|  | Business Owner/VHA rep for CDI |  |  |
|  | Stake Holder |  |  |
|  | Stake Holder |  |  |
|  | Stake Holder |  |  |
|  | Stake Holder |  |  |
|  | Technical Leads |  |  |
|  | Test Lead |  |  |
|  | Program Manager for Enrollment |  |  |
|  | ASD Group |  |  |

Table 3 - Vista Business Unit

|  |  |  |  |
| --- | --- | --- | --- |
| VistA Business Unit | | | |
| POC | Role | Email | Phone |
|  | Technical Leads - Vista |  |  |
|  | Technical Leads - Vista |  |  |
|  | Technical Leads - Vista |  |  |

Table 4 -- VBA Business Unit

|  |  |  |  |
| --- | --- | --- | --- |
| VBA Business Unit | | | |
| POC | Role | Email | Phone |
|  | Technical Leads - VBA |  |  |
|  | Technical Leads - VBA |  | 5+ |

Service Level Agreement Metrics

Table 5 lists the Service Level Agreement (SLA) metrics for the ESR message flow that are expected to be met.

Table 5 -- SLA Metrics

| SLA Type | SLA Data | SLA PROJECTED (130%) |
| --- | --- | --- |
| Number of messages/day | 2014004 | 2618205 |
| Average Message size | 1.2 MB |  |
| Data Type | HL 7 v 2.4 |  |
| Throughput | 23 messages per second |  |

Message Type Metrics

Table 6 lists the message metrics based on message types.

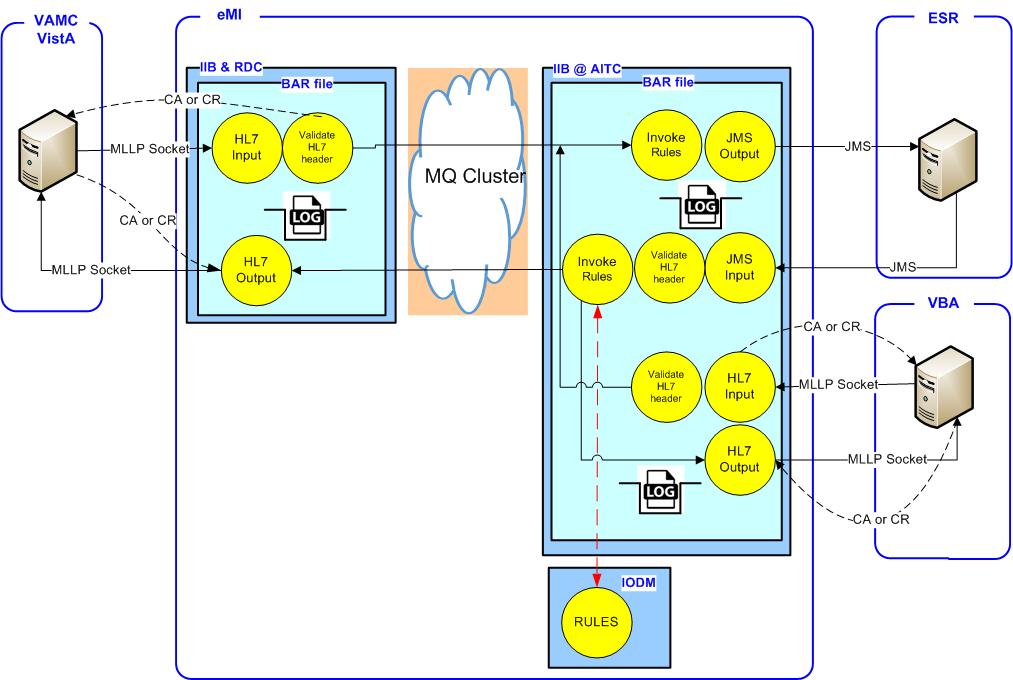
Table 6 -- Message Type Metrics

|  |  |  |
| --- | --- | --- |
| HL7 Message Type | Estimated Messages | Estimated Messages Projected (130%) |
| ORUZ11-S | 1024373 | 1331685 |
| ORFZ07-E | 709 | 922 |
| QRYZ11-M | 1856 | 2413 |
| ORUZ07-E | 290592 | 377770 |
| QRYZ07-S | 842 | 1095 |
| ORUZ05-S | 305071 | 396592 |
| ORFZ11-S | 4833 | 6283 |
| ORUZ04-S | 168 | 218 |
| ORFZ11-E | 1947 | 2531 |
| ORFZ10-S | 158188 | 205644 |
| QRYZ10-E | 157955 | 205342 |
| ORUZ10-S | 54547 | 70911 |
| QRYZ11-E | 5029 | 6538 |
| ORUZ11-E | 7894 | 10262 |

Logical System Overview

Figure 1 shows the high-level ESR/VBA/eMI logical system overview.

Figure 1 – ESR eMI Logical System Overview



Messages from VistA:

1. The VistA initiates the transfer of an HL7 message via Socket.
2. The eMI validates the MSH or BHS and sends an Accept Acknowledgement [Commit Acknowledgement (CA)/Commit Rejection (CR)] back to the VistA.
3. The eMI forwards the message from the RDC to the ENT (AITC) enterprise system.

Messages from VBA:

1. The VBA initiates the transfer of an HL7 message via Socket.
2. The eMI validates the MSH or BHS and sends an Accept Acknowledgement (CA/CR) back to the VBA.

Processing of messages from VistA and VBA at eMI ENT (AITC):

1. The eMI system at ENT (AITC) identifies the JMS queue name based on the message type.
2. The eMI publishes the received HL7 message to the identified JMS queue on the ESR system.
3. The eMI ENT (AITC) subscribes to the JMS queues on the ESR system.
4. The eMI validates the MSH/BHS of the subscribed HL7 message.
5. The eMI system identifies the routing information, RDC, and the VistA Internet Protocol (IP) and port or the VBA IP and port.

Messages for VistA:

1. The eMI forwards the message from the ENT (AITC) enterprise system to the RDC.
2. The eMI RDC sends the received HL7 message to the VistA, and receives an Accept Acknowledgement (CA/CR) from VistA.
3. The VistA initiates the transfer of an HL7 message via Socket.
4. The eMI validates the MSH and sends an Accept Acknowledgement (CA/CR) back to the VistA.

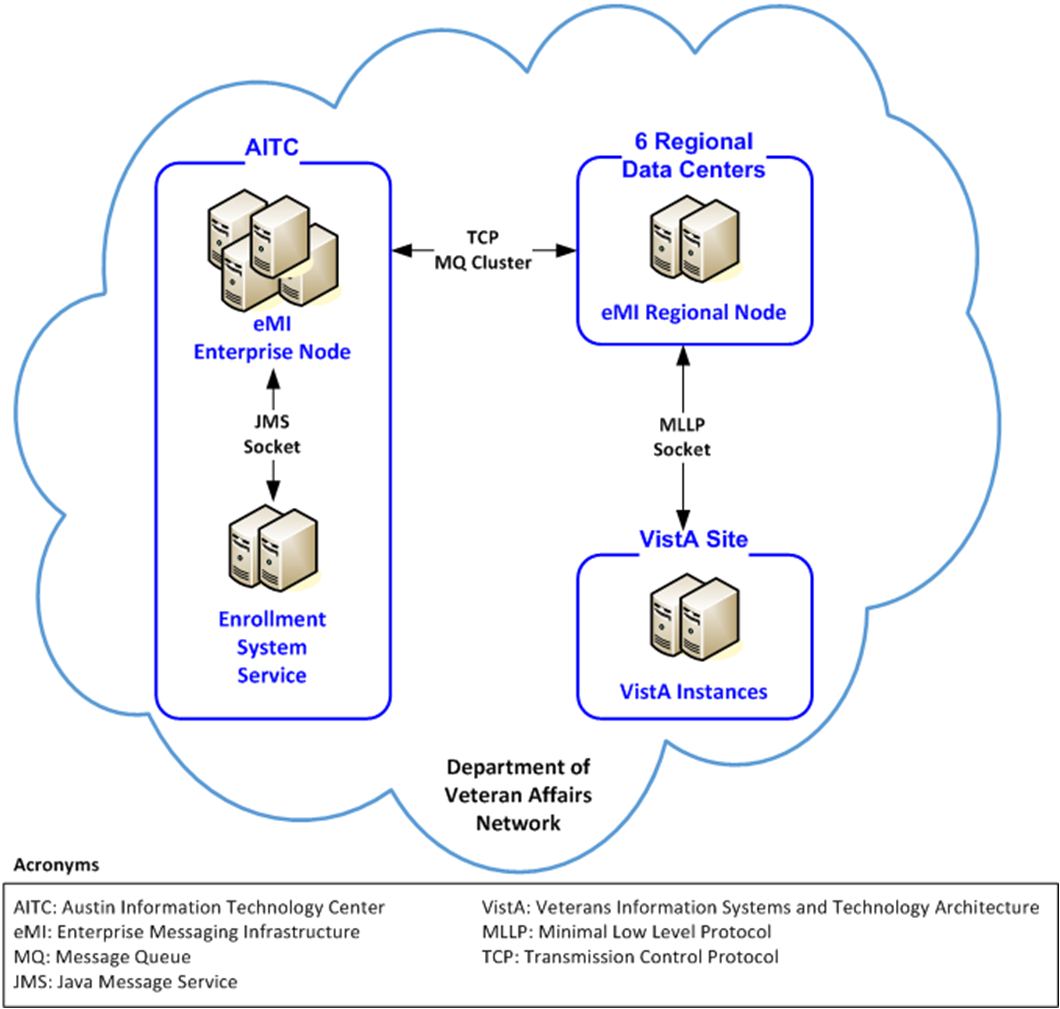
Messages for VBA:

1. The eMI ENT (AITC) sends the received HL7 message to the VBA, and receives an Accept Acknowledgement (CA/CR) from VBA.

Logical Deployment Overview

The eMI message broker hosts the message flow that listens on a configurable port for HL7 messages from the VA VistA over Transmission Control Protocol (TCP)/Minimal Lower Layer Protocol (MLLP), and on a configurable port for HL7 messages from VBA over TCP/MLLP. The eMI message broker then routes these messages to the ESR systems. Figure 2 shows the boundaries, gateway, and locations of the sending and receiving systems.

Figure 2 – ESR eMI Deployment Overview



Nominal ESR Message Flow

Figure 3 shows the message flow between the VistA and the ESR systems using the eMI.

Figure 3 – VistA to ESR Sequence



1. The VistA initiates the transfer of an HL7 message over TCP/MLLP. Protocol.
2. The eMI validates the MSH.
3. The eMI returns a CA if it does not encounter an error during validation; otherwise, it returns a CR.
4. The eMI RDC pushes the event to the eMI ENT (AITC). This push occurs as part of the Message Queue (MQ) cluster.
5. The eMI uses Operational Decision Management (ODM) to identify the JMS queue.
6. The eMI creates the JMS message.
7. The eMI publishes the message to the ESR to the JMS queue identified in Step 5.

Error Flow: For any type of exception that the message flow encounters, the eMI creates a monitoring event with exception details. If the ESR is unavailable, then eMI will keep retrying until it is able to deliver the message.

Figure 4 shows the message flow between the ESR and the VistA systems using the eMI.

Figure 4 – VistA to ESR Sequence



1. The ESR system makes JMS messages available on the JMS queues. The eMI system subscribes to these JMS queues.
2. The eMI translates the JMS message to an HL7 message and then validates the MSH/BHS segment of the HL7 message.
3. The eMI identifies the routing information; it identifies the VistA instance to which the message is to be delivered and with which RDC queue the VistA instance is associated.
4. The eMI ENT (AITC) pushes the event to the eMI RDC. This push occurs as part of the MQ cluster.
5. The eMI delivers the HL7 message to the VistA via Socket.
6. The eMI receives a CA/CR from the VistA.

Figure 5 shows the message flow between the VBA and the ESR systems using the eMI.

Figure 5 – VBA to ESR Sequence



1. The VBA initiates the transfer of an HL7 message over HTTP.
2. The eMI validates the MSH.
3. The eMI returns a CA if it does not encounter an error during validation; otherwise, it returns a CR.
4. The eMI RDC pushes the event to the eMI ENT (AITC); this push occurs as part of the MQ cluster.
5. The eMI uses ODM to identify the JMS queue.
6. The eMI creates the JMS message.
7. The eMI publishes the message to the ESR to the JMS queue identified in Step 5.

Figure 6 shows the message flow between the ESR and the VBA systems using the eMI.

Figure 6 – ESR to VBA Sequence



1. The ESR system makes JMS messages available on the JMS queues. The eMI system subscribes to these JMS queues.
2. The eMI translates the JMS message to an HL7 message and then validates the MSH segment of the HL7 message.
3. The eMI identifies the routing information; it identifies the VBA instance to which the message is to be delivered and to the queue of the ENT (AITC).
4. The eMI ENT (AITC) pushes the event to the queue for delivery to the VBA.
5. The eMI delivers the HL7 message to the VBA via Socket.
6. The eMI receives a CA/CR from the VBA.

ESR Design

The following sections address ESR design aspects.

Architecture Deviations

The ESR message flow uses an HL7-to-HL7 Data Format Description Language (DFDL) pattern. Instead of using the HL7DFDLInput node, the ESR message flow uses the source of the HL7DFDLInput node as a sub-flow; similarly, instead of using HL7DFDLOutput node, the ESR message flow uses the source of the HL7DFDLOutput node as a sub-flow. Modifications ensure that the message flow meets VA-specific requirements for validation. There is no known impact on the performance or functionality.

Pattern

The ESR message flow uses modified instance of a HL7-to-HL7 DFDL pattern. This pattern is modified to process Single and Batch HL7 messages on HL7DFDLInput node accepting messages from MLLP socket and JMS queues. The following options are specified in the tool configuration due to the requirements of the interface requirements.

* No sequencing
* No journaling
* Unchecked canonical feed, report remainders, check duplicates
* Publish to queue

Protocol

The ESR integration service uses the protocols described in Table 7, Table 8, Table 9 to interface with the sending and receiving systems.

Table 7-- VistA to eMI Interface

| VistA to eMI Interface | |
| --- | --- |
| Protocol: | MLLP over TCP/IP |
| Message Type: | HL7 |
| VistA hostname: | VistA system |
| eMI hostname: | RDC Load balancer |

Table 8 -- eMI Interface to ESR

| EMI Interface to ESR | |
| --- | --- |
| Protocol: | JMS |
| Message Type: | JMS Message with HL7 as message body |
| eMI hostname: | ENT (AITC) Message brokers |
| ESR hostname: | ESR Weblogic instance |

Table 9 -- VBA to eMI Interface

| VBA to eMI Interface | |
| --- | --- |
| Protocol: | MLLP over TCP/IP |
| Message Type: | HL7 |
| VBA hostname: | VBA system |
| eMI hostname: | ENT (AITC) Load balancer |

Message Routing

Content-based routing is utilized for routing messages to the ESR queues. Depending upon the value of the Message Type (MSH-09), the messages are routed to various JMS destination queues to the ESR application. Table 10 provides routing queues for the ESR application.

Table 10 -- ESR Queue Routing Table (Sample)

| Application | Message Type | Event Type | JMS Queue |
| --- | --- | --- | --- |
| ESR | ORU | Z07 | Edb.jms.queue.DistributedInboundDataQueue |
| ORU | Z05 | Edb.jms.queue.DistributedInboundDataQueue |
| ORU | Z04 | Edb.jms.queue.DistributedInboundDataQueue |
| ORU | Z11 | Edb.jms.queue.DistributedInboundDataQueue |
| ORU | Z12 | Edb.jms.queue.DistributedInboundDataQueue |
| ORU | Z13 | Edb.jms.queue.DistributedInboundDataQueue |
| ORU | Z09 | Edb.jms.queue.DistributedInboundDataQueue |
| MFK | ZEG | Edb.jms.queue.DistributedInboundAckQueue |
| QRY | Z10 | Edb.jms.queue.DistributedInboundQueryQueue |
| QRY | Z11 | Edb.jms.queue.DistributedInboundQueryQueue |
| ORF | Z07 | Edb.jms.queue.DistributedInboundSolicitedQueue |
| ORF | Z11 | Edb.jms.queue.DistributedInboundSolicitedQueue |
| ACK | MSA type is 'AA' | Edb.jms.queue.DistributedInboundAckQueue |
| ACK | MSA type is 'AE' | Edb.jms.queue.DistributedInboundErrorQueue |
| ACK | MSA type is 'AR' | Edb.jms.queue.DistributedInboundErrorQueue |

Content-based routing is utilized for routing messages from the ESR system to various VistA instances and VBA system. Depending upon the value of the receiving facility (MSH-06) and receiving application (MSH-05), the messages are first routed to various local queues or a queue on the RDC. The RDCs use the hostname and port information to then route the messages to the given VistA instance. Table 11 provides the sample mapping of the routing RDC queues and VistA hostname and ports. Once development is completed, this table will be updated.

Table 11 -- ESR to VistA and VBA Routing Table (Sample)

| Application | Receiving Facility | Receiving Application | RDC Queue | VistA Host and Port |
| --- | --- | --- | --- | --- |
| ESR | 660 | VISTA | VIE.ESR.ESRTOVISTA.RDC1.TAR.RDC.C |  |
| VISTA | VIE.ESR.ESRTOVISTA.RDC1.TAR.RDC.C |  |
| 565 | VISTA | VIE.ESR.ESRTOVISTA.RDC1.TAR.RDC.C |  |
| VISTA | VIE.ESR.ESRTOVISTA.RDC1.TAR.RDC.C |  |
| 200 | VBA | VIE.ESR.ESRTOVBA.TAR.ENT.L |  |

Transformation

The following sections document the various transformations for the ESR message flow.

### Protocol Transformation

Table 12 lists the various protocol transformations occurring in the ESR message flow.

Table 12 -- Protocol Transformation in ESR

| Application | Protocol |
| --- | --- |
| VistA 🡪 eMI RDC | Vista utilizes HL7/MLLP protocol to send data to RDC |
| eMI RDC 🡪 eMI Enterprise | MQ Secure Sockets Layer (SSL) channel protocol is used to transfer HL7 data securely to enterprise |
| eMI Enterprise 🡪 ESR | Utilizes JMS to push the HL7 messages from eMI to ESR |
| ESR 🡪 eMI Enterprise | Utilizes JMS to pull the HL7 messages to eMI from EST |
| eMI Enterprise 🡪 VBA | eMI utilizes HL7/MLLP protocol to send data to VBA |
| eMI Enterprise 🡪 VistA | eMI utilizes HL7/MLLP protocol to send data to VistA |

### Data Transformation

Data transformation is not applicable to the ESR message flow.

Implementation Details

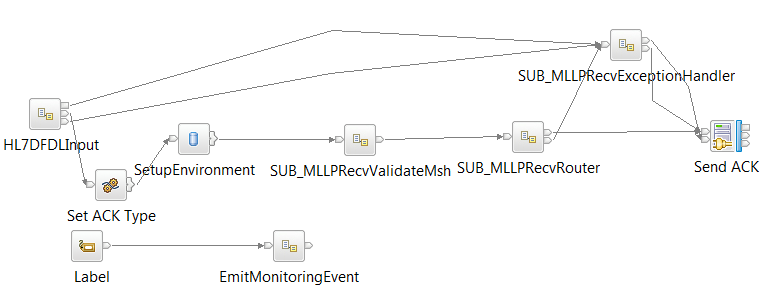
For the ESR message flow, there are three components. One component runs on the RDC, which receives and sends messages to and from VistA. The second component runs on the enterprise system, which sends and receives messages to and from the ESR system. The third component runs on the enterprise system, which sends and receives the messages to and from the VBA system.

ESR VistA Interface

This component gets deployed at all RDCs. It has a VistAReceiver that receives an HL7 message from VistA using MLLP over TCP/IP, and a VistASender that receives a message from eMI ENT (AITC) and sends it to VistA using MLLP over TCP/IP.

### ESRVistAReceiver

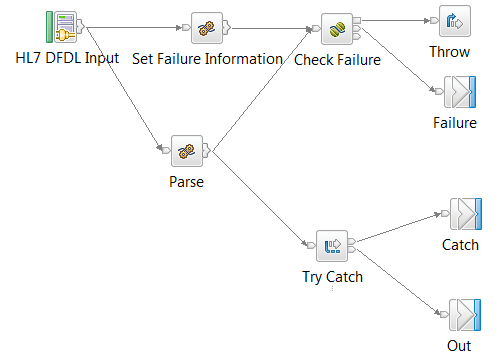
Figure 7– VistA Receiver Flow



The message flow receives HL7 messages from VistA using MLLP over TCP/IP. The flow performs basic validation of the MSH/BHS segment and sends an acknowledgement (CA/CR) back to VistA. The flow translates the received message to an MQ DFDL HL7 message and publishes it to the enterprise ESR queue. As Figure 7 shows, the Receiver Flow of the HL7-to-HL7 DFDL pattern is updated in the following manner to support the interface requirements.

* Update ‘Receiver Flow’ of the HL7-to-HL7 DFDL pattern to:
* Update the MSH/BHS field validation
* Update HL7v251DFDLLibrary Schemas to support Batch Messages (BHS)
* Update to support return of Accept Acknowledgement as a CA and/or CR to VistA
* Replace HL7DFDLInput node with HL7DFDLInput message sub-flow to suppress validation of HL7 segments in received HL7 messages against the HL7v251 message definition set (Figure 8 shows the HL7DFDLInput message sub-flow)

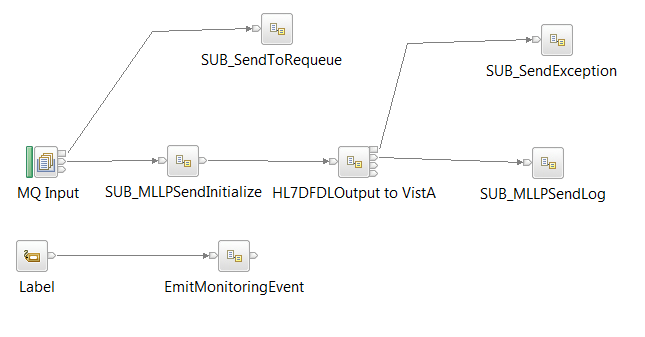
Figure 8 – HL7 Input Sub-Flow



If there is an unexpected exception, then the message flow creates a monitoring event with exception details. eMI will save error message on error queue with exception details.

### ESRVistASender

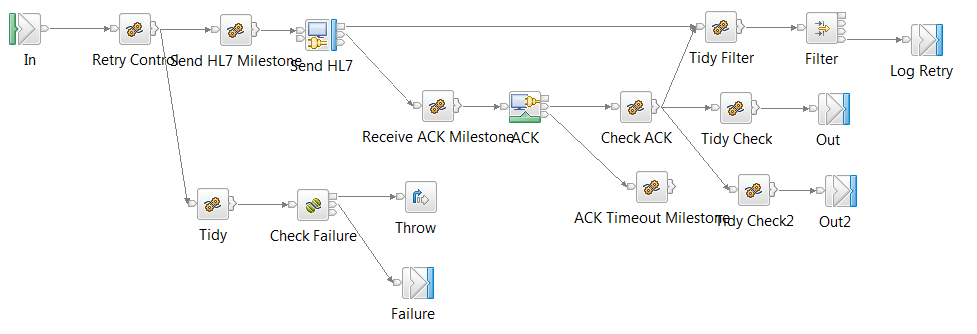
Figure 9 – VistA Sender Flow



The message flow receives the message from the eMI ENT (AITC) and it is delivered to the respective VistA instance. The message payload is an HL7 message that is received from the ESR system along with the host and port information of the VistA instance. As Figure 9 shows, the Sender Flow of the HL7-to-HL7 DFDL pattern is updated in the following manner to support the interface requirements.

* Update ‘Sender Flow’ of the HL7-to-HL7 DFDL pattern to:
* Dynamically set the hostname and port for the output node
* Update to log milestone information
* Replace HL7DFDLOutput node with HL7DFDLOutput message sub-flow to avoid treating the Accept Acknowledgement of type CR, Commit Error (CE), Application Reject (AR), and Application Error (AE) as exceptions and to prevent retries (Figure 10 shows the HL7DFDLOutput message sub-flow).

Figure 10 – HL7 Output Sub-Flow



* Update to keep retrying to send messages when VistA is unavailable

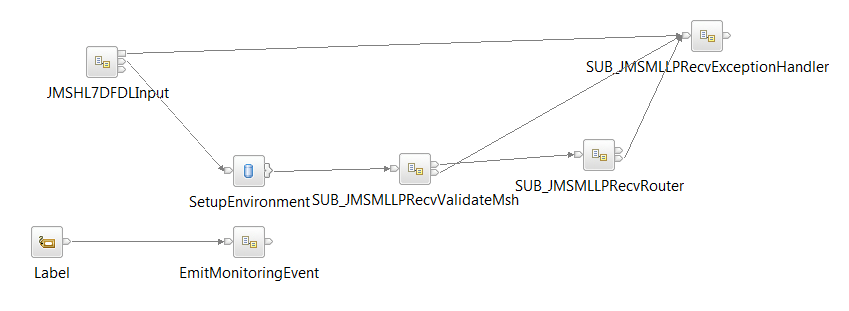
If there is an unexpected exception, then the message flow creates a monitoring event with exception details. eMI will save error message on error queue with exception details.

ESR

This component is deployed at the eMI ENT (AITC) and has three sub flows: ESRReceiver, ESRSender, and ESRRouter. The ESRReceiver flow receives JMS messages from the ESR system. The JMS messages contain the HL7 message as the JMS body. The ESRSender subscribes to the queue for HL7 messages that are sent by various VistA sites at the RDCs and VBA. The ESRSender then translates the messages to JMS messages with an HL7 message as the JMS body and delivers them to ESR over JMS. The ESRRouter routes the received messages from the ESR to the RDC to be delivered to VistA and to the queue so that they can be delivered to the VBA.

### ESRReceiver

Figure 11 – ESR Receiver Flow



The message flow receives HL7 messages from the ESR system over JMS (Figure 11). The flow performs a basic validation of the MSH/BHS segment and translates the received message to an MQ DFDL HL7 message and publishes it to the queue.

Table 13 and Table 14 provides header properties for JMS messages

Table 13 -- JMS Header Properties

| Property | Value |
| --- | --- |
| JMSDeliveryMode | Persistent |
| JMSExpiration | 0 |
| JMSPriority | 4 |
| JMSTimestamp | Current time |
| JMSType | String |
| JMSRedelivered | False |

Table 14 -- JMS Header Application Properties

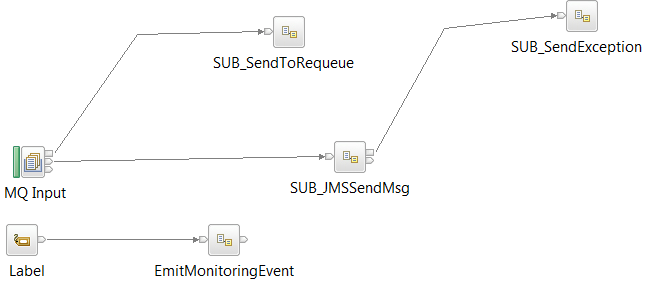
| Property | Value |
| --- | --- |
| Message\_Id | MSH-10 |
| Message\_Type | MSH-09 |
| Receiving\_Site | MSH-03 |
| Receiving\_Application | MSH-04 |
| JMS\_BEA\_RedeliveryLimit | 20 |
| JMSXDeliveryCount | 1 |

A similar flow is defined for each of the following JMS queues.

* OutboundDataQueue
* OutboundAckQueue
* OutboundSolicitedQueue
* OutboundQueryQueue

### ESRSender

Figure 12 – ESR Sender Flow



The message flow subscribes to messages intended to be delivered to the ESR. It first identifies the JMS queue destination and then prepares the JMS messages to be sent to the ESR. When routing information is not identified or routing rules are not defined, the message flow directs messages to error queue with exception details. If the ESR is unavailable, then sender flow will keep retrying until (MaxRetryCount) is e c ed with maximum value. If MaxRetryCount is e c ed, then the message is saved to error queue.

A JMS message has three parts: a header, properties, and a body. The ESR expects each of the three parts of a message to adhere to the following format. The provider populates most of the header properties. Table 15 lists the JMS header properties.

Table 15 – JMS Header Properties

| Property | Value |
| --- | --- |
| JMSDeliveryMode | Persistent |
| JMSExpiration | 0 |
| JMSPriority | 4 |
| JMSTimestamp | Current time |
| JMSType | String |
| JMSRedelivered | False |

Properties are application-specific. Table 16 lists the remaining JMS properties.

Table 16 – JMS Properties

| Property | Value |
| --- | --- |
| Message\_Id | MSH-10 |
| Message\_Type | MSH-09 |
| Sending\_Application | MSH-03 |
| Sending\_Site | MSH-04 |

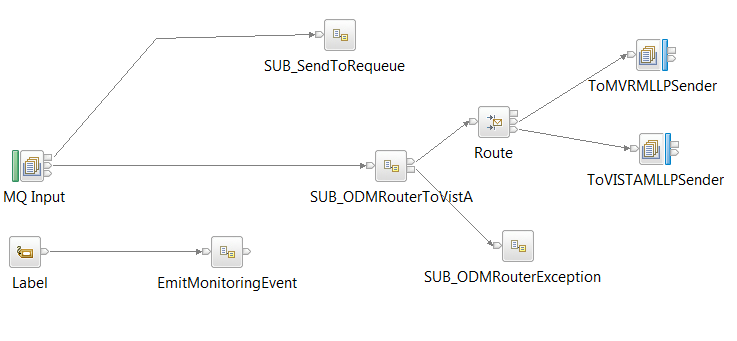
The body of the message is of type byte as an HL7 message.

If there is an unexpected exception, then the message flow creates a monitoring event with exception details. eMI will save error message on error queue with exception details.

### ESRRouters

There are two message flows: RouterToVistAVBA and RouterToESR. The RouterToVistAVBA flow determines the destination queue name, hostname, and IP for the VistA instance or VBA instance on the receiving facility (MSH-06) and receiving application (MSH-05). The routing rules are defined in the decision table in the ODM. The flow uses the decision service to connect to the ODM to retrieve and execute the rules. It also caches them in the memory. Figure 13 shows the message flow that is used to load the configuration.

Figure 13 – Identify VistA Routing Information Message Flow



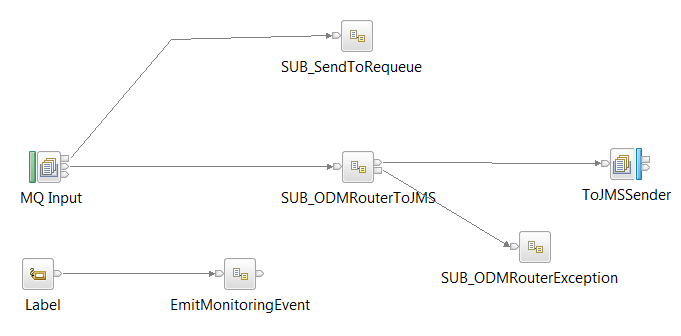
The decision table, Table 12, maps the receiving facility (MSH-06) and receiving application (MSH-05) to the destination queue, host, and port for VistA or VBA. The ODM defines and maintains this decision table.

If there is an unexpected exception, then the message flow creates a monitoring event with exception details. eMI will save error message on error queue with exception details.

Similarly, the RouterToESR flow determines the destination JMS queue information based on the message type (MSH-09), as described by the decision table, Table 10. Figure 14 shows how the message flow is used to load the configuration.

If there is an unexpected exception, then the message flow creates a monitoring event with exception details. eMI will save error message on error queue with exception details.

Figure 14 – Identify ESR Routing Information Message Flow

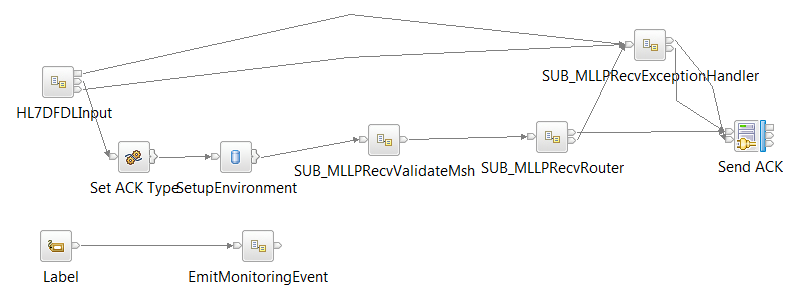


VBA VistA Interface

This component gets deployed at the eMI ENT (AITC). It has an VBA Receiver that receives an HL7 message from VBA using MLLP over TCP/IP and an VBA Sender that receives a message from eMI ENT (AITC) and then sends it to VBA using MLLP over TCP/IP.

### ESRVBAReceiver

Figure 15 – VBA Receiver Flow



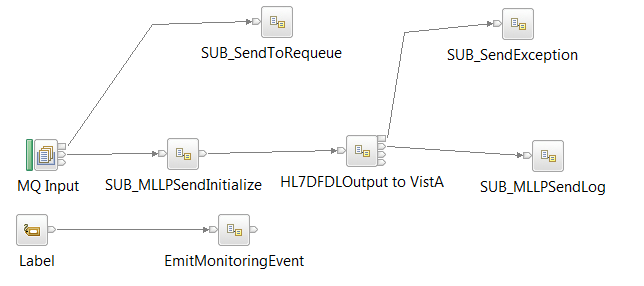
The message flow receives HL7 messages from VistA using MLLP over TCP/IP. The flow performs basic validation of the MSH segment and sends an Accept Acknowledgement (CA/CR) back to VistA. The flow translates the received message to an MQ DFDL HL7 message and publishes it to the enterprise ESR queue. As Figure 15 shows, the receiver flow of the HL7-to-HL7 DFDL pattern is updated in the following manner to support the interface requirements.

* Update ‘Receiver Flow’ of the HL7-to-HL7 DFDL pattern to:
* Update the MSH/BHS field validation
* Update to support return of Accept Acknowledgement as a CA and/or CR to the VBA
* Update to log milestone information
* Replace HL7DFDLInput node with HL7DFDLInput message sub-flow to suppress validation of HL7 segments in received HL7 messages against the HL7v251 message definition set (Figure 8 shows the HL7DFDLInput message sub-flow)

If there is an unexpected exception, then the message flow creates a monitoring event with exception details. eMI will save error message on error queue with exception details.

### ESRVBASender

Figure 16 – VBA Sender Flow



The message flow receives the message from the eMI ENT (AITC) and delivers it to the VBA. The message payload is an HL7 message that is received from the ESR system along with the host and port information of the VBA instance. As Figure 16 shows, the receiver flow of the HL7-to-HL7 DFDL pattern is updated in the following manner to support the interface requirements.

* Update ‘Sender Flow’ of the HL7-to-HL7 DFDL pattern to:
* Dynamically set the hostname and port for the output node
* Update to log milestone information
* Replace HL7DFDLOutput node with HL7DFDLOutput message sub-flow to avoid treating the Accept Acknowledgement of type CR, CE, AR, and AE as exceptions and prevent retries (Figure 10 shows the HL7DFDLOutput message sub-flow)
* Update to keep retrying to send messages when VBA is unavailable
* Update to handle generating an alert email to an administrator when the flow receives an Accept Acknowledgement (CR or CE)

If there is an unexpected exception, then the message flow creates a monitoring event with exception details. eMI will save error message on error queue with exception details

Error Handling For ESR/VistA/VBA Message Delivery

If VistA, VBA, or ESR is unavailable, then the flow uses the re-queuing mechanism. Undelivered messages would be published to the retry queue and would be retried for delivery at a configurable interval until MaxRetryCount is e c ed or message is delivered successfully.

Project Configuration File

Table 17, Table 18 and Table 19 lists the project configuration file details that are either environment-specific or control the flow of messages.

Table 17 – Project Configurable Parameters eMI ESR

| Property | Default Value | Purpose |
| --- | --- | --- |
| router.RouterToESR#RetryWaitTime | 3000 | Wait time in milliseconds between each retry to identify routing information for ESR messages |
| router.RouterToVistAVBA#RetryWaitTime | 3000 | Wait time in milliseconds between each retry to identify routing information for VistA or VBA messages |
| requeue.Requeue#delay | 30 | Wait time in seconds between each retry for all the re-queued messages |
| requeue.Requeue#retryCount | 15 | Number of retries |
| sender.ESRSender#RetryWaitTime | 3000 | Wait time in milliseconds between each retry to publish message to ESR |

Table 18 -- Project Configurable Parameters eMI ESRVistAInterface

| Property | Default Value | Purpose |
| --- | --- | --- |
| healthcare.ESRVistAReceiver#HL7DFDLInput.connectionDetails | localhost | Port on which eMI would listen for messages coming from VA |
| sender.VistASender#MQ input.queueName | VIE.ESR.ESRTOVISTA.RDC1.TAR.RDC.C | Queue to which ESRVistAInterface subscribes to send messages to VistA |
| sender.VistASender#RetryWaitTime | 3000 | Wait time in milliseconds between each retry to publish message to VistA |
| requeue.Requeue#delay | 30 | Wait time in seconds between each retry for all the re-queued messages |
| requeue.Requeue#retryCount | 15 | Number of retries |

Table 19 -- Project Configurable Parameters eMI ESRVBAInterface

| Property | Default Value | Purpose |
| --- | --- | --- |
| healthcare.ESRVBAReceiver#HL7DFDLInput.connectionDetails | localhost | Port on which eMI would listen for messages coming from VA |
| sender.VBASender#RetryWaitTime | 3000 | Wait time in milliseconds between each retry to publish message to VBA |
| requeue.Requeue#delay | 30 | Wait time in seconds between each retry for all the re-queued messages |
| requeue.Requeue#retryCount | 15 | Number of retries |

Queue Details

Table 20 lists the queues that the ESR message flows use.

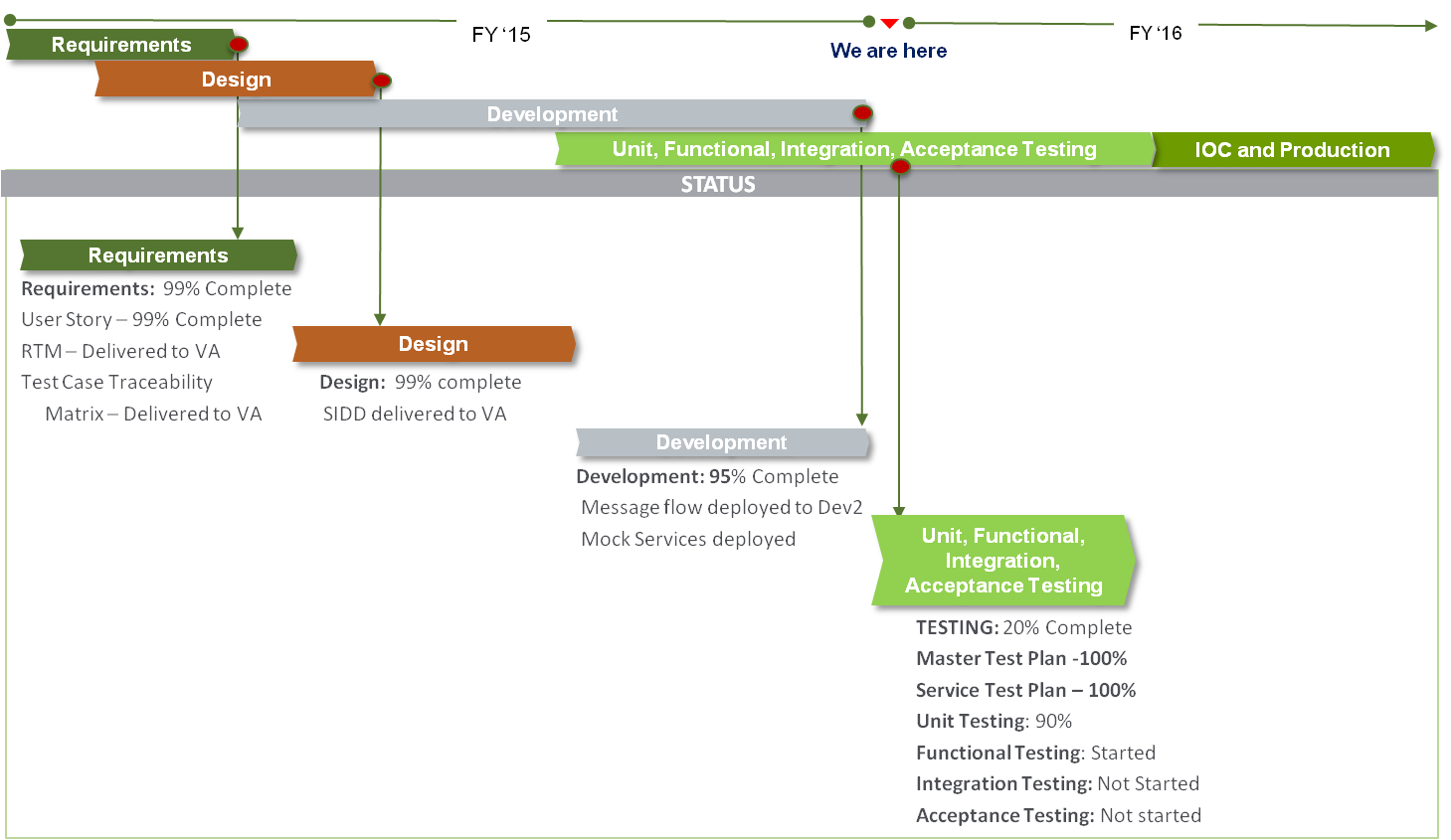
Table 20 – Queues

| Queue Name | Queue Location | Purpose |
| --- | --- | --- |
| VIE.ESR.VISTAANDVBATOESR.SRC.ENT.C | ENT (AITC) | The queue for messages that are coming from VistA |
| VIE.ESR.VISTAANDVBATOESR.SRC.ENT.C | ENT (AITC) | The queue for messages that are coming from VBA |
| VIE.ESR.ESRTOVISTAANDVBA.SRC.ENT.L | ENT (AITC) | The queue for messages that are coming from ESR |
| VIE.ESR.ESRRECEIVER.ERROR.ENT.L | ENT (AITC) | The error queue that stores the messages that was not able to process in the ESR receiver flow |
| VIE.ESR.VISTAANDVBATOESR.TAR.ENT.L | ENT (AITC) | The queue for messages to be sent to ESR |
| VIE.ESR.REQUEUE.RETRY.ENT.L | ENT (AITC) | The queue for messages to be retried |
| VIE.ESR.REQUEUE.MAXRETRIED.ENT.L | ENT (AITC) | The queue for messages that have exhausted max retries |
| VIE.ESR.REQUEUE.ERROR.ENT.L | ENT (AITC) | The queue for messages placed for retry that cannot be processed correctly |
| VIE.ESR.ESRTOVISTA.RDC1.TAR.RDC.C  …  VIE.ESR.ESRTOVISTA.RDC8.TAR.RDC.C | RDC1  ….  RDC8 | The queue for messages to be sent to RDC for VistA delivery |
| VIE.ESR.VISTARECEIVER.ERROR.RDC.L | RDC | The error queue that stores the messages that was not able to process in the VistA receiver flow |
| VIE.ESR.ESRTOVBA.TAR.ENT.L | ENT(AITC) | The queue for messages to be sent to VBA |
| VIE.ESR.VBARECEIVER.ERROR.ENT.L | RDC | The error queue that stores the messages that was not able to process in the VBA receiver flow |
| VIE.ESR.REQUEUE.RETRY.RDC.L | RDC | The queue for messages to be retried |
| VIE.ESR.REQUEUE.MAXRETRIED.RDC.L | RDC | The queue for messages that have exhausted max retries |
| VIE.ESR.REQUEUE.ERROR.RDC.L | RDC | The queue for messages placed for retry that cannot be processed correctly |

Timeline

Figure 17 shows the timeline for the ESR implementation.

Figure 17 ESR Implementation Timeline



Acronyms

Table 21 – Acronyms

| Acronym/Term | Definition |
| --- | --- |
| ACK | Acknowledgement |
| AE | Application Error |
| AITC | Austin Information Technology Center |
| AR | Application Rejection |
| BSH | Batch Header |
| CA | Commit Acknowledgement |
| CE | Commit Error |
| CR | Commit Rejection |
| DFDL | Data Format Description Language |
| eMI | Enterprise Messaging Infrastructure |
| ENT | Enterprise Broker at AITC |
| ESR | Enrollment Service Redesign |
| HL7 | Health Level Seven |
| HTTP | Hypertext Transfer Protocol |
| IFCAP | Integrated Funds Distribution, Control Point Activity, Accounting, and Procurement |
| IP | Internet Protocol |
| JMS | Java Message Service |
| kb | Kilobyte |
| MB | Megabyte |
| MLLP | Minimal Lower Layer Protocol |
| MQ | Message Queue |
| MSH | Message Header |
| VBA | Veterans Benefits Administration |
| ODM | Operational Decision Management |
| POC | Point of Contact |
| RDC | Regional Data Center |
| SLA | Service Level Agreement |
| SSL | Secure Sockets Layer |
| TBD | To Be Determined |
| TCP | Transmission Control Protocol |
| VIE | VistA Interface Engine |
| VistA | Veterans Health Information System and Technology Architecture |

Architecture Design Decisions

The ESR message flow has enhanced the standard the following patterns defined by IBM:

HL7v251DFDLLibrary is modified to process BHS messages

HCP HL7DFDL flow is modified to process JMS messages

1. Messages
   1. Message Type

HL7 v2.4 messages

* 1. Sample Messages

Not available

1. Message Mapping

None

1. IBM PMR’s

The following PMR’s are resolved and combined patch is provided for 9.0.0.2/9.0.0.3/9.0.0.4 IIB versions.

* **PMR Number:** 466,442,000 --JMS Output Node credentials issue
* **PMR Number:** 14,176,442,000 -- JMS OUTPUT Node, destination queue list issue
* **PMR Number:** 82,285,442,000 – JMS Output node caching queue information issue

1. Approval Signature

REVIEW DATE:

Signed:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Integrated Project Team (IPT) Chair Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Business Sponsor Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IT Program Manager Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Manager Date