**Inbound ePrescribing**

System Design Document



**June 2016**

Version 1.07

Department of Veterans Affairs

Office of Information and Technology (OI&T)

.

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 06/17/2016 | 1.07 | TW edits throughout | L. Morahan |
| 05/24/2016 | 1.07 | Technical Updates | S. Butt |
| 04/04/2016 | 1.06 | Technical Updates DAS, Change Healthcare related | J. Wandling, S. Butt |
| 03/02/2016 | 1.05 | Technical Updates VistA OP related | J. Wandling, S. Butt, B. Fisher |
| 02/17/2016 | 1.04 | VistA OP design elements updated in Section 6 | J. Wandling, S. Butt, B. Fisher |
| 02/08/2016 | 1.03 | AERB Constraints Updates | J. Wandling, S. Butt |
| 01/15/2016 | 1.02 | Technical Updates | J. Wandling, S. Butt |
| 01/03/2016 | 1.01 | Updated Architecture Diagram Section 4 | J. Wandling |
| 12/31/2015 | 1.0 | Final revisions and edits | J. Wandling, S. Butt, B. Fisher, M. Murph, H. Trenkov |
| 12/2015 | 0.2 | Technical Updates: Sections 1, 2, 3, 4, 5, 6, 7 and 8 | J. Wandling, S. Butt, B. Fisher, M. Murph, H. Trenkov |
| 11/2015 | 0.1 | Initial Version Development | J. Wandling, S. Butt, B. Fisher, M. Murph, H. Trenkov |

ProPath SDD Template 2.10 – June 2015

Artifact Rationale

The System Design Document (SDD) is a dual-use document that provides the conceptual design as well as the as-built design. This document will be updated as the product is built, to reflect the as-built product.

When to Complete Each Section of the SDD

| Section | Completed On or Before PMAS Phase | Rationale |
| --- | --- | --- |
| 1 – Introduction | MS 0 Review; updated thereafter | Conceptual design should inform evaluation of investments |
| 2 – Background | MS 0 Review; updated thereafter | Conceptual design should inform evaluation of investments |
| 3 – Conceptual Design | MS 0 Review; updated thereafter | Conceptual design should inform evaluation of investments |
| 4 – System Architecture | MS 0 Review; updated thereafter | Conceptual design should inform evaluation of investments |
| 5 – Data Design | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |
| 6 – Detailed Design | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |
| 7 – External System Interface Design | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |
| 8 – Human Machine Interfaces | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |
| Attachments | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |

A product’s system design should be defined conceptually prior to the allocation of personnel and resources that occur at project initiation. This gives the enterprise an opportunity to evaluate IT investments before project teams are stood up and funding is allocated. Sections 1- 4 which discuss the high level design should be completed prior to MS 0. All sections should be completed and updated before MS 1. Projects will need to address all SDD approval constraints prior to the MS 2 review. In addition, the SDD should reflect the as-built product going into the MS 2 review.

Table of Contents

[1. Introduction 1](#_Toc454381252)

[1.1. Scope 1](#_Toc454381253)

[1.1.1. Inbound ePrescribing Scope Inclusions 1](#_Toc454381254)

[1.1.2. Inbound ePrescribing Scope Exclusions 2](#_Toc454381255)

[1.2. User Profiles 2](#_Toc454381256)

[2. Background 3](#_Toc454381257)

[2.1. Overview of the System 3](#_Toc454381258)

[2.1.1. Business Benefits 4](#_Toc454381259)

[2.2. Overview of the Business Process 5](#_Toc454381260)

[2.2.1. Overview of Key Terms 5](#_Toc454381261)

[2.2.2. Overview of the Inbound eRx Process Flow 5](#_Toc454381262)

[2.3. Overview of the Significant Requirements 9](#_Toc454381263)

[2.3.1. Functional Requirements 9](#_Toc454381264)

[2.3.2. Functional Workload/Performance Requirements 10](#_Toc454381265)

[2.3.3. Operational Requirements 11](#_Toc454381266)

[2.3.4. Pivotal Technical Requirements 13](#_Toc454381267)

[2.3.5. Security and Privacy Requirements 15](#_Toc454381268)

[2.3.6. System Criticality and High Availability Requirements 16](#_Toc454381269)

[2.3.7. Single Sign-On Requirements 16](#_Toc454381270)

[2.3.8. Use of Enterprise Portals 16](#_Toc454381271)

[3. Conceptual Design 17](#_Toc454381272)

[3.1. Conceptual Application Design 17](#_Toc454381273)

[3.1.1. Application Context 17](#_Toc454381274)

[3.1.2. High-Level Application Design 18](#_Toc454381275)

[3.1.3. VistA Outpatient Pharmacy Interface/Enhancements 18](#_Toc454381276)

[3.1.4. Application Locations 28](#_Toc454381277)

[3.2. Conceptual Data Design 29](#_Toc454381278)

[3.2.1. Pharma Data Formats 30](#_Toc454381279)

[3.2.2. High Level eRx Processing Hub Logic (Example Transaction) 30](#_Toc454381280)

[3.2.3. Project Conceptual Data Model 32](#_Toc454381281)

[3.2.4. Database Information 33](#_Toc454381282)

[3.2.5. User Interface Data Mapping 33](#_Toc454381283)

[3.3. Conceptual Infrastructure Design 40](#_Toc454381284)

[3.3.1. System Criticality and High Availability 41](#_Toc454381285)

[3.3.2. Special Technology 41](#_Toc454381286)

[3.3.3. Technology Locations 44](#_Toc454381287)

[3.3.4. Conceptual Infrastructure Diagram 45](#_Toc454381288)

[4. System Architecture 46](#_Toc454381289)

[4.1. Hardware Architecture 49](#_Toc454381290)

[4.2. Software Architecture 49](#_Toc454381291)

[4.2.1. Components 50](#_Toc454381292)

[4.2.2. VistA OP Components 58](#_Toc454381293)

[4.3. Network Architecture 60](#_Toc454381294)

[4.4. Service Oriented Architecture / ESS 61](#_Toc454381295)

[4.5. Enterprise Architecture 62](#_Toc454381296)

[5. Data Design 62](#_Toc454381297)

[5.1. DBMS Files 63](#_Toc454381298)

[5.2. DBMS Files – eRx Processing Hub 63](#_Toc454381299)

[5.3. Physical Database Organization 63](#_Toc454381300)

[5.4. Physical Data Structures and Files 63](#_Toc454381301)

[5.4.1. NCPDP XML Data Analysis 64](#_Toc454381302)

[5.4.2. Query Patterns 65](#_Toc454381303)

[5.4.3. Indexing 65](#_Toc454381304)

[5.4.4. DBMS Files – VistA OP 65](#_Toc454381305)

[5.5. Non-DBMS Files 66](#_Toc454381306)

[5.6. Data View 66](#_Toc454381307)

[5.6.1. eRx Processing Hub Tier 66](#_Toc454381308)

[5.6.2. VistA OP eRx Holding Queue 74](#_Toc454381309)

[6. Detailed Design 76](#_Toc454381310)

[6.1. Hardware Detailed Design 76](#_Toc454381311)

[6.2. Software Detailed Design 78](#_Toc454381312)

[6.2.1. Conceptual Design 78](#_Toc454381313)

[6.2.2. Specific Requirements 82](#_Toc454381314)

[6.3. Network Detailed Design 122](#_Toc454381315)

[6.4. Security and Privacy 123](#_Toc454381316)

[6.4.1. Security 123](#_Toc454381317)

[6.4.2. Privacy 125](#_Toc454381318)

[6.5. Service Oriented Architecture / ESS Detailed Design 125](#_Toc454381319)

[6.5.1. Service Description for MVI 125](#_Toc454381320)

[6.5.2. Service Description for E&E 125](#_Toc454381321)

[6.5.3. Service Description for HDR 125](#_Toc454381322)

[6.5.4. Service Design for PatientEligibility 125](#_Toc454381323)

[6.5.5. Service Design for PrescriptionProviders 129](#_Toc454381324)

[6.5.6. Service Design for PrescriptionDrugs 134](#_Toc454381325)

[6.5.7. Service Design for PharmacyTransfers 138](#_Toc454381326)

[6.5.8. Service Design for VistAErxTransmissions 142](#_Toc454381327)

[6.5.9. Service Design for VistAErxStatuses 146](#_Toc454381328)

[6.5.10. Service Design for ClinicalExchangePharmacyInformation 150](#_Toc454381329)

[7. External System Interface Design 155](#_Toc454381330)

[7.1. Interface Architecture 155](#_Toc454381331)

[7.2. Interface Detailed Design 155](#_Toc454381332)

[7.2.1. Connectivity Methods 155](#_Toc454381333)

[7.2.2. SSL Certificate 155](#_Toc454381334)

[7.2.3. Timeout Limits 155](#_Toc454381335)

[7.2.4. Two-way Mutual Certification Authentication 155](#_Toc454381336)

[7.2.5. Network Communications Header 156](#_Toc454381337)

[7.2.6. eRx XML Wrapper 156](#_Toc454381338)

[8. Human-Machine Interface 159](#_Toc454381339)

[8.1. Interface Design Rules 159](#_Toc454381340)

[8.2. Inputs 159](#_Toc454381341)

[8.3. Outputs 160](#_Toc454381342)

[8.4. Navigation Hierarchy 160](#_Toc454381343)

[8.4.1. Vista OP Screen 160](#_Toc454381344)

[9. Attachment A – Approval Signatures 161](#_Toc454381345)

[A.1. Relationship to Other Documents and Plans 162](#_Toc454381346)

[A.2. Acronyms and Abbreviations 162](#_Toc454381347)

[A.3. Identification of Technology and Standards 164](#_Toc454381348)

[A.4. Constraining Policies, Directives and Procedures 166](#_Toc454381349)

[A.5. Requirements Traceability Matrix 166](#_Toc454381350)

[A.6. Packaging and Installation 167](#_Toc454381351)

[A.7. Design Metrics 167](#_Toc454381352)

# Introduction

Pharmacy applications and systems are some of the oldest technologies in the Veterans Health Information Systems and Technology Architecture (VistA) system. The Veterans Health Administration (VHA) has identified system limitations and cumbersome, inconsistent pharmacy processes as a weakness in its ability to provide efficient pharmaceutical services across the VHA continuum. The United States Department of Veterans Affairs (VA) approved the Pharmacy Reengineering (PRE) project to address several fundamental problems with the current system. The objective of the overall PRE project was to facilitate the improvement of pharmacy operations, customer service, and patient safety for the VHA.

Veterans Health Administration (VHA), Patient Care Services (PCS) and Pharmacy Benefits Management (PBM) has requested a new capability as part of the Pharmacy Reengineering (PRE) program to receive inbound electronic prescriptions (e-prescriptions or eRxs) from an external provider (e.g., a doctor not associated with the Department of Veterans Affairs [VA], medical staff at a Department of Defense [DoD] military treatment facility, etc.). They also seek to have the ability to transfer prescriptions electronically between pharmacies, both VA to VA, as well as VA to non-VA (ideally). Once received, these prescriptions will then be fed into the existing Veterans Health Information Systems and Technology Architecture (VistA) Outpatient Pharmacy (OP) for processing and dispensing.

## Scope

This document focuses on the design of Inbound ePrescribing and enhancements to the VistA Outpatient Pharmacy (OP) module.

The purpose of this document is to describe in sufficient detail how the inbound ePrescribing application and VistA modules are to be modified to support Inbound eRx. This System Design Document (SDD) translates the Requirement Specifications into a document from which the developers can create the actual system. It identifies the top-level system architecture, and identifies hardware, software, communication, and interface components relevant to the affected application and modules.

The high-level scope of this project (as well as what is not included in the scope) is indicated below, with further detail outlined in the corresponding Inbound ePrescribing Requirements Specification Document (RSD).

### Inbound ePrescribing Scope Inclusions

The Inbound ePrescribing effort scope includes:

* Receiving and processing inbound eRxs, where “inbound” refers to the ordering of medication or medical related supplies for a VA patient by a non-VA provider to be filled at a VA pharmacy.
* Electronically receiving and processing outpatient prescriptions only (includes prescriptions created for a VA patient upon discharge from a non-VA hospital to be filled on an outpatient basis by a VA pharmacy).
* Receiving and processing inbound eRxs from non-VA providers for controlled substances.
* Receiving and processing inbound eRxs from non-VA providers that currently prescribe medications and medical related supplies for Civilian Health and Medical Program of the VA (CHAMPVA) beneficiaries and which are currently handled by the Medications by Mail (MbM)[[1]](#footnote-2) program.
* Electronically receiving and processing cancellations and changes for previously received eRxs from non-VA providers.
* Electronically sending refill requests to non-VA providers who previously sent eRxs to VA pharmacies, receiving responses from the non-VA providers, and processing those responses (i.e., eRx refill requests initiated from VA Pharmacy to a non-VA provider).
* Electronically receiving and processing refill requests from non-VA providers who previously sent eRxs to VA pharmacies.
* Electronically receiving and processing medication history[[2]](#footnote-3) requests initiated by any entity (e.g., non-VA provider or VA provider, Personal Health Record [PHR] system, or Electronic Medical Record [EMR] system, etc.) to a VA Pharmacy, as well as electronically sending medication history requests from a VA Pharmacy to any entity that supports receiving and processing medication history requests.
* Sending outbound electronic notifications from a VA pharmacy (that received an inbound eRx) to the non-VA provider that originally sent the eRx.
* Electronic transfers of prescriptions to VA pharmacies from VA pharmacies.

### Inbound ePrescribing Scope Exclusions

The Inbound ePrescribing effort scope does not include:

* VA providers generating Rxs at one VAMC location to be electronically transmitted to and processed by (filled, dispensed, etc.) a different VAMC location’s pharmacy.
* Initiating outbound eRxs (generation of an eRx by a VA provider to be filled at a non-VA pharmacy).
* Electronic receipt and processing of any VA and non-VA inpatient medication orders.
* Electronic receipt and processing of any VA and non-VA orders for Durable Medical Equipment (DME), such as wheel chairs.
* Electronic receipt and processing of eRx refill requests from a VA patient’s non-VA PHR system.

## User Profiles

The users of inbound ePrescribing application have been categorized into primary and secondary users. Users can easily request prescriptions, medical supplies, or controlled substances to be electronically transmitted from non-VA providers and pharmacies to VA pharmacies for processing.

The table below lists the primary and secondary users and their responsibilities.

Table : User Characteristics

| Type of User | Description | Responsibilities |
| --- | --- | --- |
| Primary Users | VA Pharmacists,  VA Pharmacy Technicians/Assistants  Change Healthcare | Can view inbound eRxs received from an external provider  Can manually validate an inbound eRx  Can transfer existing eRxs to another VA pharmacy and receive electronically transferred eRxs from another VA pharmacy  Can manage, fill, print labels, dispense, etc. an inbound eRx  Can send prescriptions and medication history requests |
| Secondary Users | VA Pharmacy Managers,  PBM personnel | Viewing reports on inbound eRx |
| Secondary Users | Non-VA Providers | Send/transmit inbound eRx and associated requests to VA  View inbound eRx notifications from VA  Respond to eRx requests (e.g., a refill request) regarding inbound eRxs from VA |
| Secondary Users | VA Administrators  Support Staff | Full Control  Turn on/off receiving inbound eRxs |

# Background

## Overview of the System

The Inbound ePrescribing software solution aims to meet the business needs identified in the Inbound ePrescribing Business Requirements Document (BRD) titled Pharmacy Reengineering Inbound ePrescribing Work Effort Unique Identifying #20140401. To meet those needs the Inbound ePrescribing system will:

* Allow external providers to select VA pharmacies that support inbound electronic prescriptions.
* Receive electronic prescriptions from non-VA providers to be processed by a VA Pharmacy.
* Allow VA pharmacy staff to view and interact with received inbound electronic prescriptions after they have gone through an automatic validation process by the system.
* Allow VA pharmacy staff to process (manually validate, manage, dispense) inbound electronic prescriptions.
* Provide tracking and auditing functions for the inbound electronic prescription capability.
* Provide VA pharmacies with the capability to transfer eRxs.
* Help standardize terminology for all medications at the VA by implementing and using the NCPDP SCRIPT Standard.
* Reduce time from when an Rx is written or created by a non-VA provider to when it is entered into or received by VistA OP.
* Increase satisfaction of Veterans/VA beneficiaries, VA pharmacy personnel, non-VA providers.

Additional detail around the system is also available in the RSD for this effort, titled Inbound ePrescribing Requirements Specification Document Draft V2.1. Further elaboration of system functionality with VA SMEs will also identify more granular functionality for delivery of Inbound ePrescribing capabilities.

### Business Benefits

A typical scenario of what happens today is a Veteran visits a non-VA provider specialist who hand-writes an Rx for a medication and gives it to the Veteran. Veterans then physically visits their VA pharmacy and submit the paper Rx to be processed and dispensed by that facility to optimize their VA benefits that offer them low, or no, copays. The VA pharmacy dispenses the Rx and the Veteran picks it up. Issues may arise in this scenario such as the VA pharmacy staff misinterpretation of the provider’s hand-written Rx, which could lead to the wrong medication being dispensed, a dosing error, or other patient safety issues. It is also time consuming and inconvenient for Veterans to physically visit their VA pharmacy in order to submit the Rx, and inefficient for the VA pharmacy staff to manually input the Rx information into VistA OP so that it can be processed.

Additionally, today, VA pharmacies cannot accept transferred Rxs from other pharmacies nor can they transfer prescriptions to other pharmacies. This presents an inconvenience for Veterans who travel or move frequently in that they must visit a provider in order to get a new Rx so that they can take the new Rx to a different pharmacy for dispensing.

Providing the capability to receive and process Inbound eRxs from outside sources and providers will improve the effectiveness and efficiency with which health care benefits are delivered to VA beneficiaries. Some of the high-level business benefits provided by the Inbound ePrescribing system include:

* **Improved efficiency:** The Inbound ePrescribing system leads to more efficient use of VA pharmacy resources and non-VA provider resources. For example: fewer transcribing/translation errors and clear/error-free communications equates to time savings by not having to communicate back and forth regarding the content of a prescription.
* **Improved Veteran/beneficiary satisfaction:** The Inbound ePrescribing system makes an existing manual process easier, more efficient, and more effective through automation of the prescription process. This automation helps improve the satisfaction of the Veteran community by:
* Reducing the risk of losing paper Rxs
* Enabling more secure communication of Rx data

And most importantly:

* Providing timelier dispensing of Rxs prescribed by non-VA providers
* **Improved patient safety:** The Inbound ePrescribing not only improves on the manual process from an efficiency standpoint, but from a safety one as well. By automating the process the risk of interpreting hand-written or transcribed prescriptions is drastically reduced, while Veteran health-safety is improved.
* **Improved data accuracy:** The Inbound ePrescribing system provides enhanced functionality within VistA OP by improving the accuracy and potential use of the data it collects. By automating data transmission from providers to the VA (and between other pharmacies), the need for VA pharmacy personnel to manually input Rx data from non-VA providers is largely eliminated, reducing the chance for data to be entered incorrectly or missed.

## Overview of the Business Process

The Inbound ePrescribing system is intended to provide the ability for the VA to receive and process eRxs from external to VA providers. Inbound eRx prescription data encompasses incoming to the VA prescription messages (such as newRx and RefillRx), as well as incoming/outgoing status, error and verification messages as defined in the NCPDP Script 10.6 XML.

### Overview of Key Terms

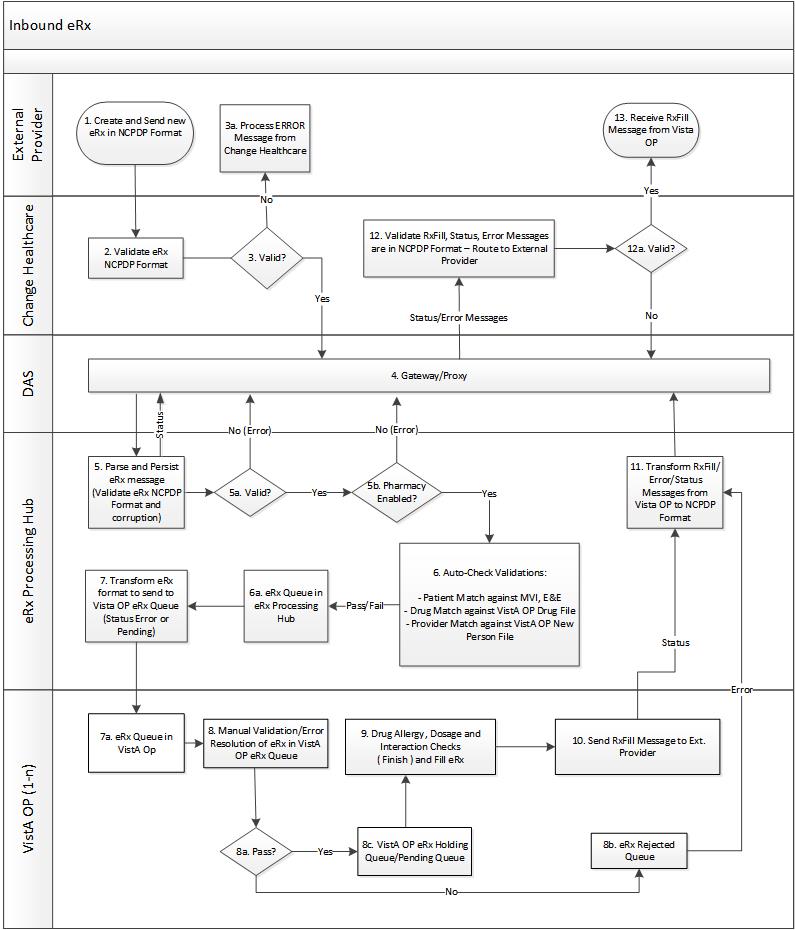
This section contains key terms used to describe the Inbound eRx process, as well as other aspects of the solution as depicted in this document. This is not the complete list of terms and is rather a sub-set of the most widely used ones:

* **Inbound eRx or Inbound ePrescribing** – refers to the end-to-end solution and process from external provider-to-fulfilled prescription
* **Clearinghouse** – refers to Change Healthcare’s pharmacy data exchange
* **Messaging Infrastructure** – refers to VA’s DAS infrastructure; leverages Oracle Weblogic JMS Queue
* **eRx Processing Hub** – refers to the main processing component of the Inbound eRx solution; leverages Pentaho Data Integration (PDI) and custom web services
* **eRx Processing Table** – refers to the data container in the eRx Processing Hub storing all inbound and outbound pharmacy messages and their statuses
* **eRx UI** – refers to the eRx web application providing eRx tracking, pharmacy enable/disable and reports
* **eRx Holding Queue** – the Inbound eRx staging file in VistA OP for processing auto-validated prescriptions from external providers

### Overview of the Inbound eRx Process Flow

A high-level overview of the Inbound eRx process for pharmacy data messages is outlined in Figure 1 below.

**Figure 1: Overall Inbound ePrescribing Process**



The Inbound eRx Process Flow depicts five (5) swim-lanes – one external to Inbound eRx (External Provider) and four (4) Inbound eRx processing tiers (Change Healthcare, DAS, eRx Processing Hub and VistA OP):

* (External to Inbound eRx) External Provider:
  + External to VA physicians who, with the use of a 3rd party Electronic Medical Records (EMR) software, issue a prescription for a Veteran
  + The EMR system is registered with Change Healthcare and is responsible for creating and sending the eRx in NCPDP XML format
  + The external physician is registered with Change Healthcare and their provider information (e.g., NPI number) is known and verified by Change Healthcare
* (Tier 1) Change Healthcare:
  + Serve as proxy for all messages between the external providers and the VA infrastructure (i.e., the DAS/eRx Processing Hub)
  + Support and validate NCPDP XML format and structure
* (Tier 2) DAS:
  + Serve as secured-layer gateway/proxy for all messages (in NCPDP XML) between Change Healthcare and the eRx Processing Hub
  + Current availability of DAS is limited in scope to proxy/gateway functionality
    - DAS will handle: Inbound NCPDP XML from Change Healthcare to DAS Oracle WebLogic JMS Queue. Outbound NCPDP XML from DAS to Change Healthcare
* (Tier 3) eRx Processing Hub
  + Receive, persist, validate, manipulate, construct/send NCPDP XMLs
  + Validate designated pharmacy from NCPDP XML and match to VistA OP instance.
  + Perform auto-validation and matching (including patient, enrollment/registration, provider, drug)
  + Transform prescription data and send to VistA eRx Holding Queue format
  + Maintain processing statuses and perform error/exception handling
  + Provide admin UI to track, enable/disable transmission, validate eRx and run reports
* (Tier 4) VistA OP (1-n)
  + Provide UI for eRx Users (manual steps, review)
  + Process eRx Holding Queue transactions (including drug-to-drug, drug-to-allergy, PAs, DEA rules, dosage, drug dispensation, order processing, prescription file and fulfillment)
  + Once the eRx is validated it will be processed into the PENDING OUTPATIENT ORDERS (#52.41) file
  + Perform error handling and status updating in eRx Holding Queue (and the eRx Processing Hub processing statuses)

The Inbound eRx processing flow is sequential in nature as depicted in Figure 1 (above):

* Step 1: The Inbound eRx process flow begins with the External Provider, using their EMR system, creates and sends to Change Healthcare eRx message data in NCPDP XML format
* Step 2: In Change Healthcare (Tier 1), the eRx message is validated against the NCPDP Format to ensure that message is in valid construct without any corruption
* Step 3: If the message is valid, Change Healthcare (Tier 1) routes the message to the VA infrastructure via DAS (Tier 2) for further processing
  + Step 3a: If the eRx message is invalid, Error message is sent back to the external provider (as per the NCPDP specifications) without sending the message to the VA
* Step 4: DAS (Tier 2) receives the message and forwards it to the eRx Processing Hub (Tier 3)
* Step 5: The eRx Processing Hub (Tier 3) validates the NCPDP XML format to ensure that the message is in valid construct without any corruption, and “shreds” the XML into its data elements, structure and stores the message; the message is recorded in a Transaction/Processing table which tracks the processing status of the message, as well as the coordinates auto-validations (step 6) and the synchronization (steps 7 and 11) with VistA OP instance (Tier 4)
* Step 6: The eRx Processing Hub (Tier 3) performs patient, provider and drug auto-validations – once all auto-validations are completed and the eRx is ready to be sent to VistA OP (step 7), the Transaction/Processing table prescription record in the eRx Processing Hub (Tier 3) is updated to reflect (a) successful auto-validation, (b) failed auto-validation (i.e., patient match was unsuccessful), or (c) partial auto-validation (i.e., patient match was successful, but other auto-validation(s) failed); furthermore, this steps identifies the respective VistA OP instance the message needs to be sent to
  + The VA pharmacy needs to be enabled to accept Inbound eRx or an error message will be sent to the External Provider – for simplicity purposes, this uncommon scenario is not explicitly depicted in the Inbound eRx process flow
* Step 7: The eRx Processing Hub (Tier 3) constructs the eRx data into the format of the eRx Holding Queue of the respective VistA OP (1-n) instance (Tier 4)
* Step 8: In the respective VistA OP instance (tier 4), pharmacy personnel performs manual validation and eRx Processing Hub (tier 3) auto-validation error resolution of the eRx (e.g., patient match, eligibility, drug dosage match, prior authorization received)
  + Any change to the status of the eRx in VistA OP (Tier 4) is synchronized with the Transaction/Processing table in the eRx Processing Hub (Tier 3); errors (and some successes) initiate NCPDP messages in the eRx Processing Hub (Tier 3) to be sent back to the External Provider
* Step 9: Upon successful patient and drug validation, additional auto-validation checks in VistA OP (Tier 4) occur (e.g., drug-to-drug, drug-to-allergy validations); successful and unsuccessful auto-validations update the eRx Holding Queue transaction status
  + Any change to the status of the eRx in VistA OP (Tier 4) is synchronized with the Transaction/Processing table in the eRx Processing Hub (Tier 3); errors (and some successes) initiate NCPDP messages in the eRx Processing Hub (Tier 3) to be sent back to the External Provider
* Step 10: Once all the validations are completed successfully, the prescription is fulfilled in VistA OP (Tier 4) based on the existing fulfillment routines
* Step 11: Upon success, VistA OP (Tier 4) updates the eRx Processing Hub (Tier 3) and triggers a message update to the External Provider
* Step 12: In the eRx Processing Hub (Tier 3), success (and error) status updates from VistA OP (Tier 4) trigger message(s) to the External Provider; those messages (in NCPDP XML format) are sent to DAS (Tier 2) for delivery to Change Healthcare (Tier 1) and ultimately to the External Provider
* Step 13: Change Healthcare (Tier 1) validates the received message against the NCPDP Format to ensure that the message is in valid construct without any corruption, and sends it to the External Provider
* Step 14: The Inbound eRx process flow ends with the External Provider receiving the message update from the VA

Technical details of the Inbound eRx processing architecture, data flow, exception/error handling, monitoring and logging, user interfaces, and reporting are included in Sections 3, 4, 5, 6, 7 and 8 of this document.

## Overview of the Significant Requirements

### Functional Requirements

The functional business requirements identified correspond to the original business requirements, as documented in the corresponding RSD for this effort*.* For purposes of design and architecture, this set of requirements expands and elaborates upon the original requirements, and provides sequential detail:

* From the initiation of an ePrescription inbound to a VA Pharmacy
* To the receipt of the ePrescription within the VA Pharmacy
* To the validation and processing of the ePrescription within the VA Pharmacy
* To the dispensing of the eRx to the Veteran Patient.

In support of this, new requirements were added to fill gaps where additional functionality was needed within the inbound ePrescribing system. These requirements were developed hierarchically, starting at the top with:

* An **EPIC** overarching major requirement
  + To a **User Story** who, what, and why requirement
    - To the **Tasks** that support that user story requirement.
      * **Business Acceptance Criteria** needed to support that requirement

Using the NCPDP script standard 10.6, along with analogous system functionality found in the Defense Health Agency’s (DHA) Composite Health Care System (CHCS), as guides, the functional requirements detailed in the RSD overview the major functional needs regarding:

* Patient, drug and external provider validation;
* Script routing through the new system; and,
* Interface/coordination with any existing functionality within VistA OP.

Note that some areas may require additional elaboration (e.g. requirements regarding the transfer of eRxs from external pharmacies), but the RSD extensively covers the baseline process of a new inbound eRx from an external provider, as well as the majority of the alternative flows that were identified.

**Requirements Specification Document:** [**Inbound ePrescribing v1.0 RSD**](http://DNS  DNS       .DNS   /projects/pre/PRE_Inb_eRx/Shared%20Documents/Requirement/Inb_eRx_v1-0_RSD.docx)

Business Requirements Document:

[**Inbound ePrescribing v1.0 BRD**](http://DNS  DNS       .DNS   /projects/pre/PRE_Inb_eRx/Shared%20Documents/Requirement/Inb_eRx_v1-0_BRD.doc)

### Functional Workload/Performance Requirements

The following Inbound ePrescribing functional workload/performance requirements are detailed in the corresponding RSD for this effort.

Table : Functional Workload/Performance Requirements

| ID from BRD | Requirement |
| --- | --- |
| NONF2811 | The Inbound ePrescribing system shall provide the capability to measure the reporting requirements for Responsiveness, Capacity, and Availability. |
| NONF2812 | The actual system performance data shall be displayed to the Information Technology (IT) Performance Dashboard for customers and IT staff to view. |
| New | The Inbound ePrescribing system shall provide operating capacity100% of the time, 24 hours/day, 365 days/year to support inbound electronic prescription transactions |
| New | The Inbound ePrescribing system response times and page load times shall be consistent with VistA OP standards. |
| New | The Inbound ePrescribing system shall support the different National Council for Prescription Drug Programs (NCPDP) SCRIPT transaction types. |
| New | The Inbound ePrescribing system shall support new, refilled, and partially-filled prescriptions. |
| New | The Inbound ePrescribing system shall provide the capability to process current and future inbound electronic prescription transactions |
| New | The Inbound ePrescribing system shall provide the capability to capture the inbound electronic prescription transaction speed. |
| New | The Inbound ePrescribing system shall provide the capability to capture data retrieval times from the Holding Queues. |
| New | The Inbound ePrescribing system shall provide the capability to capture the dates and times of the inbound electronic prescription. |
| New | The Inbound ePrescribing system shall provide the capability to capture page refresh times. |
| New | The Inbound ePrescribing system shall provide the capability to capture page load times. |
| New | The Inbound ePrescribing system shall provide the capability to display performance data using different report formats. |
| New | The Inbound ePrescribing system shall provide the VA user with the ability to view performance summary data. |
| New | The Inbound ePrescribing system shall provide the VA user with the ability to view performance data in details by category. |
| New | The Inbound ePrescribing system shall provide the VA user with the ability to select tabular and graphical data views of the performance data. |
| New | The Inbound ePrescribing system shall provide the capability to capture message response times. |
| New | The Inbound ePrescribing system shall support the existing VistA OP users. Current Vista OP user count is 12,300 users (8,000 pharmacists and 4,300 pharmacy technicians) |
| New | The system is expected to support 1,000 concurrent users |
| New | Response time:  Running queries: five seconds or less  Running reports: five seconds or less  Synchronous web service response: 27 seconds or less  Transmit eRx to VistA OP: 5 minutes or less |
| New | Usage peak times: Monday through Friday, 8:00 a.m. Eastern Time – 9:00 p.m. Eastern Time |
| New | Raw single eRx message data size is estimated to be 100 KB |
| New | Total raw message data storage is estimated to be (100 KB \*4M)/106 = 400GB per year |

### Operational Requirements

The following Inbound ePrescribing operational requirements are detailed in the corresponding RSD/BRD for this effort.

Table : Operational Requirements

| ID from BRD | Requirement |
| --- | --- |
| NONF3297 | The Inbound ePrescribing system response times and page load times shall be consistent with VistA Outpatient Pharmacy (OP) standards. |
| NONF1780 | Maintenance, including maintenance of externally developed software incorporated into the VistA OP application, must be scheduled during off peak hours or in conjunction with relevant maintenance schedules. The business owner should provide specific requirements for establishing system maintenance windows when planned service disruptions can occur in support of periodic maintenance. |
| NONF1608 | The Inbound ePrescribing system shall disseminate unscheduled system outages or other events that impact the response time to the user community within 30 minutes of the occurrence. |
| NONF1609 | The Inbound ePrescribing system shall provide a real-time monitoring solution to report agreed/identified critical system performance parameters agreed/identified critical system performance parameters. |
| NONF2820 | Critical business performance parameters shall be captured and identified to support metric reporting, the performance meters are displayed to the Office of Information and Technology (OI&T) Performance Dashboard, and the system solution shall comply with all Service Level Agreements (SLAs). |
| NONF2811 | The Inbound ePrescribing system implementation includes Disaster Recovery (DR) procedures and 24/7 support consistent with organizationally established expectations relative to system availability. |
| NONF3314 | The Inbound ePrescribing solution provides a contingency plan that includes procedures for disable and enables receiving inbound electronic prescriptions. |
| NONF3213 | The Inbound ePrescribing system shall provide data protection measures, such as back-up intervals and redundancy that is consistent with systems categorized as mission critical (12 hour restoration, 2 hour recover point objective). |
| NONF1610 | Notification of scheduled maintenance periods that require the service to be offline or that may degrade system performance shall be disseminated to the business user community a minimum of 48 hours prior to the scheduled event. |
|  | The Inbound ePrescribing solution allows functional requirements to be integrated into the VistA Outpatient Pharmacy (OP) system without changes to the hardware. |
| NONF2229 | A monitoring process shall be provided to ensure that data is accurate and up-to-date and provides accurate alerts for malfunctions while minimizing false alarms. |
|  | Maintenance, including maintenance of externally developed software incorporated into the Inbound ePrescribing utilities use system (contingency/failover, etc.) redundancy to ensure that scheduled maintenance does not cause system down time. |
| NONF1615 | The Inbound ePrescribing solution provides a back-up and data recovery process for when the system is brought off-line for maintenance or technical issues/problems off-line. |
| NONF3085 | Technical Help Desk support for the application shall be provided for VA users to obtain assistance with receiving and processing inbound eRxs, and sending and receiving eRx transfers. |

### Pivotal Technical Requirements

Table : Technical Requirements

| ID from BRD/RSD | Requirement |
| --- | --- |
| EPIC 20 | Auto-validation in Inbound eRx is based on precise determination of pass/fail criteria – i.e., match or no-match; Patient identity match (MVI match) needs to pass successfully (i.e., exact match) before any further auto-checks (e.g., provider, registration, drug) can be performed; Patient mandatory attributes for matching (NCPDP required fields) are: first name, last name, DOB and gender; eRx Processing Hub will perform incrementally narrowing criteria until an exact (i.e., one) match or no-match occurs |
| OWNR170 | Provide the ability to express all content using nationally recognized reference and authoritative terminology standards (e.g., Logical Observation Identifiers, Names, and Codes [LOINC], Systematized Nomenclature of Medicine Clinical Terms [SNOMED CT], RxNORM, etc.). |
| OWNR13409 | Provide the ability for external providers to specify the VA pharmacy to which they can send an electronic prescription. |
| OWNR13410 | Provide the ability for a VA Pharmacy to initiate receiving inbound electronic prescriptions (i.e., turn “on”) and the ability to stop receiving inbound electronic prescriptions (i.e., turn “off”). |
| OWNR13411 | Provide the ability for a VA Pharmacy to receive and respond to an electronically transmitted new prescription for medication (including controlled substances) in accordance with the most current version supported of the NCPDP SCRIPT Standard (10.6 at minimum) from an external provider. |
| OWNR13412 | Provide the ability for a VA Pharmacy to receive and respond to an electronically transmitted new prescription for expendable medical supplies (other than durable medical equipment) in accordance with the most current version supported of the NCPDP SCRIPT Standard (10.6 at minimum) from an external provider. |
| New Prior Authorization 1 | In compliance with the NCPDP SCRIPT 10.6 standardized transaction framework for electronic prior authorization (ePA), provide the ability for the VA inbound ePrescribing system to indicate if an eRx requires prior authorization, and if the required prior authorization has been approved or not approved by the payer (i.e., insurance company) on the inbound eRx. |
| New Prior Authorization 2 | In compliance with the NCPDP SCRIPT 10.6 standardized transaction framework for electronic prior authorization (ePA), provide the ability for the VA inbound ePrescribing system to notify the VA pharmacist when prior authorization is approved or not approved for an eRx that has already been received and is sitting in a VistA Holding Queue waiting for confirmation that eRx is approved or not approved. |
| OWNR13413 | Provide the ability for a VA Pharmacy to send an electronically transmitted prescription change request in accordance with the most current version supported of the NCPDP SCRIPT Standard (10.6 at minimum) to an external provider and receive an associated response in relation to a previously received eRx from this provider. |
| OWNR13414 | Provide the ability for a VA Pharmacy to receive and respond to an electronically transmitted prescription cancellation request in accordance with the most current version supported of the NCPDP SCRIPT Standard (10.6 at minimum) from an external provider in relation to a previously received eRx from this provider. |
| OWNR13415 | Provide the ability for a VA Pharmacy to send an electronically transmitted prescription refill request in accordance with the most current version supported of the NCPDP SCRIPT Standard (10.6 at minimum) to an external provider and receive an associated response for a previously received eRx from this provider. |
| OWNR13416 | Provide the ability for a VA Pharmacy to receive and respond to an electronically transmitted prescription refill request in accordance with the most current version supported of the NCPDP SCRIPT Standard (10.6 at minimum) from an external provider in relation to a previously received eRx from this provider. |
| OWNR13417 | Provide the ability for a VA Pharmacy to receive and respond to an electronically transmitted medication history request in accordance with the most current version supported of the NCPDP SCRIPT Standard (10.6 at minimum) from an external entity. |
| OWNR13418 | Provide the ability for a VA Pharmacy to send an electronically transmitted medication history request in accordance with the most current version supported of the NCPDP SCRIPT Standard (10.6 at minimum) to an external entity and receive an associated response. |
| OWNR13419 | Provide the ability for a VA Pharmacy to send electronic notifications in accordance with the most current version supported of the NCPDP SCRIPT Standard (10.6 at minimum) to an external provider that inbound eRx requests and responses received from it by VA were successfully received or resulted in error. |
| OWNR13420 | Provide the ability for a VA Pharmacy to receive electronic notifications in accordance with the most current version supported of the NCPDP SCRIPT Standard (10.6 at minimum) from an external provider indicating that inbound eRx requests and responses sent to it from VA were successfully received or resulted in error. |
| OWNR13421 | Provide the ability for VA pharmacy staff to interact with received inbound eRxs after the system has automatically validated the patient specified on the eRx is a known/identifiable individual to VA and the individual is able to receive pharmacy benefits rendered by VA. |
| OWNR13422 | Provide the ability for VA pharmacy staff to interact with received inbound eRxs after the system has automatically validated the external provider that sent the eRx is a valid provider authorized to write prescriptions and/or provide services to the specified VA patient. |
| OWNR13423 | Provide the ability for VA pharmacy staff to interact with received inbound eRxs after the system has automatically validated the medication or medical supply specified on the eRx is recognized by VA and allowed to be prescribed to a VA patient. |
| OWNR13426 | Provide the ability for VA pharmacy staff to manually validate an inbound eRx (to include identifying valid patient, provider, and medication or medical supply) before the eRx can be further processed for dispensing. |
| OWNR13427 | Provide the ability for VA pharmacy staff to perform existing VistA OP functions on an eRx once it is automatically received into the VistA OP queue. |
| OWNR13430 | Provide the ability for VA pharmacies to track and account for the number of inbound eRxs received and processed in order to accommodate any future needs for reporting, auditing, or fulfilling contractual obligations, (e.g., for paying intermediaries). |
| NEED3358 | Provide VA pharmacies with the eRx transfer capability. |
| OWNR13433 | Provide the ability to electronically receive a transferred Rx for medication (including controlled substances) or medical supplies in accordance with the most current, appropriate NCPDP Standard(s) from an external pharmacy so that the VA pharmacy may dispense the prescription.  \*Note: This is currently out of scope. |

### Security and Privacy Requirements

The Federal Information Processing Standard 199 (FIPS 199), Standards for Security Categorization of Federal Information and Information Systems, defines the security categories, security objectives, and impact levels to which National Institute of Standards and Technology (NIST) Special Publication (SP) 800-60 Volume 1 Revision 1, maps information types. A FIPS 199 analysis was completed for the Inbound ePrescribing system, and it has been determined that the security categorization for this system is a MODERATE in accordance with FIPS 199.

This classification of MODERATE ensures that data is only received by the persons and applications that it is intended for; that data is not subject to unauthorized or accidental alterations; and that the system’s resources are available when needed. This classification also means that each end-user with direct or indirect access must have a valid clearance and be on a need-to-know basis for all information within the Inbound ePrescribing system (as well as any shared or networked systems). It should also be noted that personnel with system administration rights and roles will be required to undergo an elevated background investigation to fulfill their duties.

In terms of privacy requirements, the Inbound ePrescribing system will adhere to all that are required by the VA. The requisite Privacy Impact Assessment (PIA) System of Records Notice (SORN) will cover the system’s collection and maintenance of individually identifiable information. This, along with other documentation related to security and privacy, will be continually updated to ensure compliance with the latest VA guidance. An overview of the current security and privacy requirements that the system adheres to are outlined in the following table, and detailed to a much a greater degree in the Inbound ePrescribing RSD in the section titled “Security Specifications”.

Table : Security and Privacy Requirements

| ID | Requirement |
| --- | --- |
| 1 | All VA security requirements will be adhered to. Based on Federal Information Processing Standard (FIPS) 199 and National Institute of Standards and Technology (NIST) SP 800-60, recommended Security Categorization is Moderate. |
| 2 | The Security Categorization will drive the initial set of minimal security controls required for the information system. Minimum security control requirements are addressed in NIST SP 800-53 and VA Handbook 6500, Appendix D (BRD). |
| 3 | All VA Privacy requirements will be adhered to. Efforts that involve the collection and maintenance of individually identifiable information must be covered by a Privacy Act system of records notice. |

### System Criticality and High Availability Requirements

The Inbound eRx is considered a mission critical system and requires high availability. It follows the same standard procedures used for all critical VistA and VistA associated systems.

The system shall be available 24/7, with exceptions made for required planned and well communicated system maintenance during off-peak times.

### Single Sign-On Requirements

The Inbound eRx Admin web user interface requires users to authenticate using Personal Identification Verification (PIV) card and Active Directory. This requirement is met using the VA standard IAM SSOi implementation to allow seamless authentication between systems.

### Use of Enterprise Portals

The use of Enterprise Portals is not a current requirement of Inbound eRx as it does not currently expose suitable Enterprise Portal content.

# Conceptual Design

## Conceptual Application Design

This section provides the conceptual application design of the Inbound eRx system. The Inbound eRx design includes a newly designed Java Enterprise Edition application working together with a VistA Outpatient Pharmacy (OP) module enhanced and modified to handle electronic prescriptions and related NCPDP messages.

### Application Context

Figure 2 depicts the Inbound eRx application and the external systems that it interacts with.

Figure : Inbound eRx Application Context Diagram

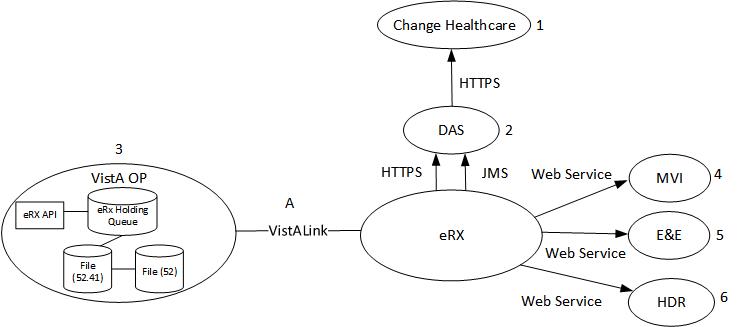


Table (Grouping): Application Context Description

| ID | Name | Description | Interface Name | Interface System |
| --- | --- | --- | --- | --- |
| 2 | DAS | Acts as a gateway between Change Healthcare and Inbound ePrescribing (eRx) | HTTPS, JMS | Inbound eRx, Change Healthcare |
| 3 | VistA OP | Allows VistA to receive prescription information from eRx | VistALink (TBD) | Inbound eRx |
| 4 | MVI | Allows a system like Inbound eRx to confirm patient identity. | Web Service | Inbound eRx |
| 5 | E&E | Allows a system like Inbound eRx to confirm patient’s enrollment and eligibility. | Web Service | Inbound eRx |
| 6 | HDR | Allows a system like Inbound eRx to request patient’s medication history. | Web Service | Inbound eRx |

Interfaces External to OI&T

| ID | Name | Related Object | Input Messages | Output Messages | External Party |
| --- | --- | --- | --- | --- | --- |
| 1 | Change Healthcare | Allows system like Inbound ePrescribing (eRx) to receive prescription data. | Web Services | Inbound eRx | Change Healthcare |

Interfaces Internal to OI&T

| ID | Name | Related Object | Input Messages | Output Messages | External Party |
| --- | --- | --- | --- | --- | --- |
| A | VistALink | Inbound eRx | N/A | Messages to VistA for updating prescription data. | N/A |

Externally Shared Data Stores

| ID | Name | Data Stored | Owner | Access |
| --- | --- | --- | --- | --- |
| N/A |  |  |  |  |

### High-Level Application Design

### VistA Outpatient Pharmacy Interface/Enhancements

Currently, VistA OP is not able to accept outside prescriptions electronically, due to the lack of software and RPC’s to directly handle electronically transmitted prescriptions. There are, however, API’s and other software processes in place that can be leveraged to assist in accepting and processing electronic prescriptions.

Within the current functionality, a prescription can be entered into CPRS, at which time the prescription is stored in the ORDER file (#100). Once the order is signed, the prescription information is sent to the PENDING OUTPATIENT ORDERS file (#52.41). After the Prescription has been finished, the prescription is logged into the PRESCRIPTION file (#52), and the entry in the PENDING OUTPATIENT ORDERS file is removed (deleted).

A provider name (located in NEW PERSON file (#200)) is required in order to facilitate this process, which is not readily available within VistA in the appropriate format for completing the above actions electronically. For example, providers outside of the VA do not have entries in the NEW PERSON file, an Electronic signature code, or a populated VA# or DEA value, which is required for certain medications.

The Outpatient Pharmacy package uses similar functionality when processing a new order from the option PSO LM BACKDOOR ORDERS. Using this process, entries are placed into the ORDER (#100) and PRESCRIPTION (#52) files. The order information itself is actually sent through PS EVSEND OR protocol. This is the means by which the order is created in the Order file. The key driving mechanism for the creation of the prescription and order entry information is an Outpatient Pharmacy API, named EN^PSON52. Sample HL7 sent through PS EVSEND OR:

GOLDDEV 16d1>ZW MSG

MSG(1)="MSH|^~\&|PHARMACY|500|||||ORM"

MSG(2)="PID|||100808||ONEHUNDRED,INPATIENT"

MSG(3)="PV1||O|||||||||||||||||"

MSG(4)="ORC|SN||404584^PS||CM||||20151208115942-0500|10000000226^FISHER,BRADLEY

W||10000000224^PROVIDER,ONE|||20151208|W^Written^99ORN^^^"

MSG(5)="RXO|^^^5^ACETAMINOPHEN TAB^99PSP"

MSG(6)="RXE|325\T\MG\T\1\T\TABLET^BID^^20151208^20161208^^^325MG^|1338.6642^ACET

AMINOPHEN 325MG TAB^99NDF^5591^ACETAMINOPHEN 325MG TAB^99PSD|||^^^20^MG^99PSU|^^

^63^TAB^99PSF||||60||11||10000000224|501379|||||||30|||325|^^^20^MG^99PSU"

MSG(7)="NTE|21||TAKE ONE TABLET BY MOUTH TWICE A DAY"

MSG(8)="RXR|^^^1^ORAL (BY MOUTH)^99PSR"

MSG(9)="ZRX||W|N|W|10000000226^USER NAME/CLERK NAME^99NP|"

MSG(10)="ZSC||||||||"

MSG(11)="ZCL|1|1|"

MSG(12)="ZCL|1|2|"

MSG(13)="ZCL|1|3|"

MSG(14)="ZCL|1|4|"

MSG(15)="ZCL|1|5|"

MSG(16)="ZCL|1|6|"

MSG(17)="ZCL|1|7|"

MSG(18)="ZCL|1|8|"

A new eRx Holding Queue will be created that will allow a pharmacist to view all inbound electronic prescriptions, rectify errors, review information, and ultimately, ‘submit’ or ‘approve’ the electronic prescription. New RPC’s will be created within the Outpatient Pharmacy package to accept incoming HL7 messages containing all the needed elements for a prescription from a non-VA medical facility. Using the inbound HL7 message, a new entry will be placed in the eRx Holding Queue file.

This queue will use List Manager for user interaction. The eRx Holding Queue will show electronic prescriptions in a list format, with extended options being available to see all of the details about the prescriptions. Additional extended options will be created to allow the pharmacist to fix issues with the prescription, such as provider, drug, and patient identification errors.

After all automatic and manual checks are complete, a Pharmacist will need to choose to ‘Submit’ or ‘Approve’ the electronic prescription from the eRx Holding Queue. At this point, there will be enough information about the prescription, and the eRx/prescription will be sent to the PENDING OUTPATIENT ORDERS file (#52.41). Once the eRx has been sent to the PENDING OUTPATIENT ORDERS file, a user will be able to ‘finish’ the prescription the same way as it is done today. This will leverage existing process, and reduce the need for duplicating logic. The PSO LMOE FINISH functionality will be modified to include eRx specific information.

Routine PSONEW1 is the driver for the entry of new orders through PSO LM BACKDOOR ORDERS. This routine prompts the user for a series of items. These items are: Patient Status, Drug, ICD diagnosis codes for the order (only prompted if user holds the provider key), Dosing, Patient Instructions, Days Supply, Quantity, label copies, # of Refills, Provider, Clinic, Mail/Window information, Remarks, Issue Date, Fill Date, and Clerk code. This routine calls several other routines, and requires user interaction to answer the prompts that are displayed.

Much of this data will already be available through the eRx Holding Queue, but may need to be verified against the logic within VistA before a submitted eRxs can be fully processed. Depending on where it makes the most sense, underlying logic within each of these prompting sequences will be leveraged to ensure the same checks are being done on eRx entries, as are being done for ‘New Orders’ within PSO LM BACKDOOR ORDERS. This may mean that the checking is done via verification options within the eRx Holding Queue. If the eRx Holding Queue is not the ideal place for such checks, the verification will occur through a series of ‘new’ prompts once the pharmacist chooses to ‘submit’ or ‘approve’ the eRx for completion. The eRx processing functionality will aim to keep the number of user interactive prompts to a bare minimum, and cannot be guaranteed to be able to remove user interaction entirely. There may be issues with dosing or other fields that may need to be resolved through user prompts after the ‘submit’/’approve’ process. These would be final steps to ensure the data is appropriate.

During the entry of a new order within PSO LM BACKDOOR orders, checks are performed based on the drug that has been selected. Once the order has been added to the PENDING OUTPATIENT ORDERS file it will follow the existing order checks that are built into the PSO LMOE FINISH functionality. Should additional order checks be needed prior to submitting the order to the PENDING OUTPATIENT ORDERS file, the logic associated with the eRx queue will need to be enhanced to tap into the Outpatient pharmacy order checks, much like CPRS does today.

Some new processes/RPCs will need to be created to potentially ‘bypass’ the order signing process, or make the order signing process happen automatically. The signing process requires an Electronic Signature code to be in place in the NEW PERSON file (#200) in order to complete the signature action.

Once the order is in the PENDING OUTPATIENT ORDERS file, pharmacists can then use existing processes (PSO LMOE FINISH/PSO VR) to finish and verify the medication. Due to the use of the existing file structures, minimal change will be needed in these areas to facilitate the release/finishing of a medication.

**Inbound eRx System**

Figure 3 provides a high level application diagram for the Inbound eRx system. The circles/ovals within the Inbound eRx oval represent the business services that comprise the Inbound eRx application Service-Oriented Architecture approach.

**Figure 3: Inbound eRx High Level Application Diagram**

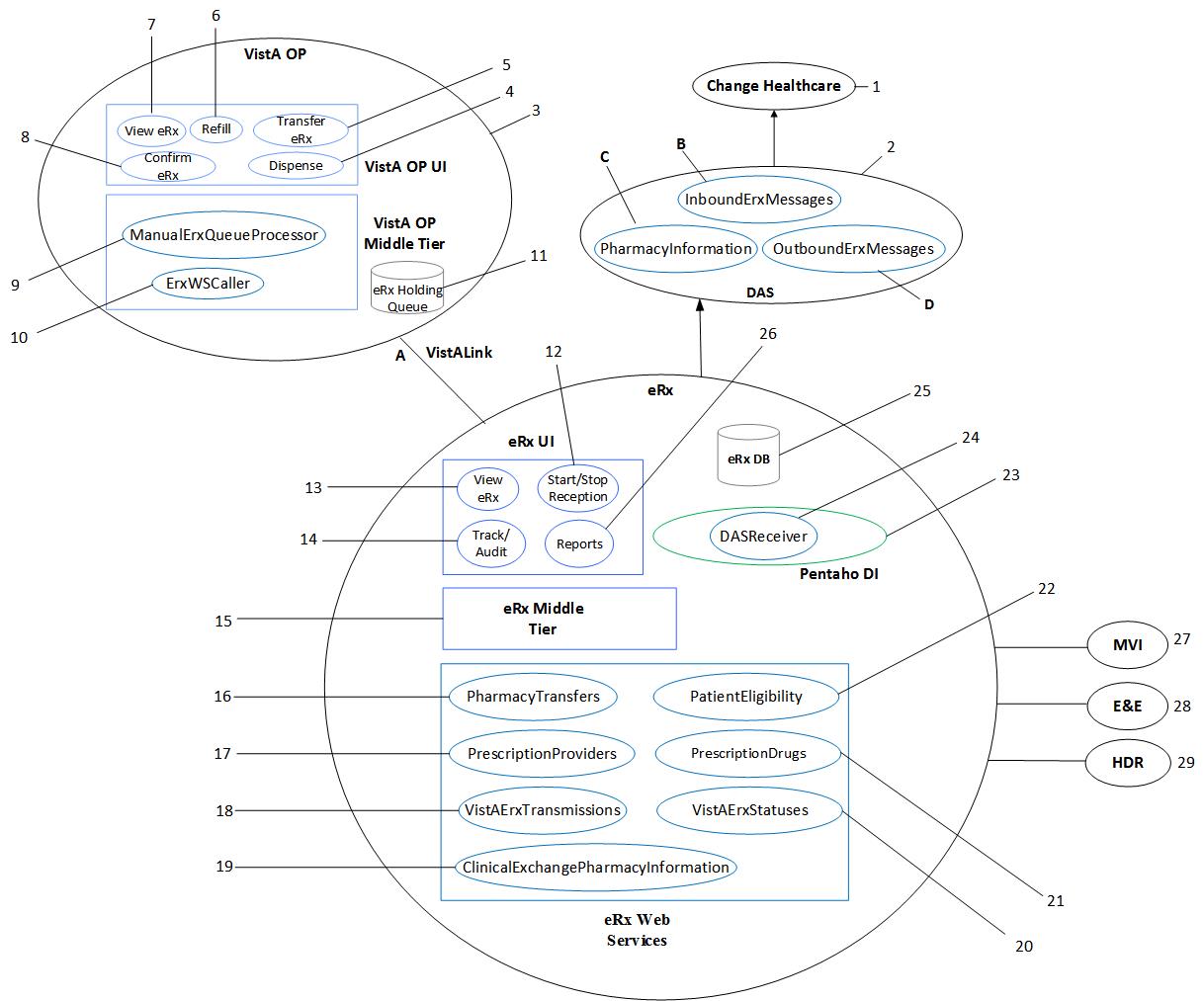


Table : Objects in the High Level Application Design

| ID | Name | Description | Service or Legacy Code | External Interface Name | External Interface ID | Internal Interface Name | Internal Interface ID | SDP Sections 1&2 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | VistA Link | It is used to authorize user in VistA and to send authenticated RPC requests to VistA. | Service | VistALink, the interface mechanism between VistA and Inbound eRx | A | Internal calls to this service are performed via direct Java method calls. | N/A | N/A |
| B | InboundErxMessages | It is service which responds to Incoming Prescription messages and it is to be developed by and hosted at DAS. | Service | Change Healthcare | N/A | N/A | N/A | N/A |
| C | PharmacyInformation | It is a proxy in DAS for sending pharmacy information updates. | Service | Change Healthcare | N/A | N/A | ClinicalExchangePharmacyInformation | N/A |
| D | OutboundErxMessages | It is a proxy in DAS for sending prescription updates. | Service | Change Healthcare | N/A | N/A | VistAErxStatuses | N/A |
| 4 | Dispense | This functionality allows dispensing of an eRx. It is part of VistA OP UI. | N/A | N/A | N/A | Internal calls to this service are via MUMPs routines, which are invoked when this service is selected from VistA OP UI options. | N/A | N/A |
| 5 | Transfer eRx | This functionality allows transferring of an eRx.  It is part of VistA OP UI. | N/A | N/A | N/A | Internal calls to this service are via MUMPs routines, which are invoked when this service is selected from VistA OP UI options. | N/A | N/A |
| 6 | Refill | This functionality allows refilling an existing prescription. It is part of VistA OP UI. | N/A | N/A | N/A | Internal calls to this service are via MUMPs routines, which are invoked when this service is selected from VistA OP UI options. | N/A | N/A |
| 7 | View eRx | This functionality allows viewing an eRx from the VistA eRx Holding Queue. It is part of VistA OP UI. | N/A | N/A | N/A | Internal calls to this service are via MUMPs routines, which are invoked when this service is selected from VistA OP UI options. | N/A | N/A |
| 8 | Confirm eRx | This functionality allows confirming an eRx. It is part of VistA OP UI. | N/A | N/A | N/A | Internal calls to this service are via MUMPs routines, which are invoked when this service is selected from VistA OP UI options. | N/A | N/A |
| 9 | ManualErxQueueProcessor | It enables the processing of the prescriptions from the eRx Holding Queue. It is Internal to VistA OP. | Service | N/A | N/A | Internal calls to this service are via MUMPs routines, which are invoked when manual validation step of eRx Prescription data starts. | N/A | N/A |
| 10 | ErxWSCaller | It sends the prescription status to the eRx middle by calling VistAErxStatuses web service. | Service | N/A | N/A | Internal calls to this service are via MUMPs routines, which are invoked when Prescriptions are manually processed and statues are generated and are passed to this. | N/A | N/A |
| 11 | eRx Holding Queue | A temporary storage file which will hold pending eRxs which were transmitted from eRx processing hub. | Service | N/A | N/A | Internal calls to this service are via MUMPs routines, which are invoked when manual validation step of eRx Prescription data starts. | N/A | N/A |
| 12 | Start/Stop Reception | This is part of eRx UI. It allows a VA pharmacy to start/stop eRx transmission from Change Healthcare. | Service | N/A | N/A | Internal calls to this service are performed via direct Java method calls. | N/A | N/A |
| 13 | View eRx | This service allows viewing eRx on eRx GUI. | Service | N/A | N/A | Internal calls to this service are performed via direct Java method calls. | N/A | N/A |
| 14 | Track/  Audit | This service allows to track/audit an eRx. | Service | N/A | N/A | Internal calls to this service are performed via direct Java method calls. | N/A | N/A |
| 16 | PharmacyTransfers | This is a RESTful Web Service. It converts the data into XML format and sends to Pentaho DI. | Service | N/A | N/A | It is called by VistA with prescription pharmacy transfer requests. | N/A | N/A |
| 17 | PrescriptionProviders | This is a RESTful Web Service. It validates the provider from the new eRxs. | Service | N/A | N/A | It is called by Pentaho DI Job. | N/A | N/A |
| 18 | VistAErxTransmissions | This is a RESTful Web Service. It transmits the eRxs to VistA OP via RPC call. | Service | N/A | N/A | It is called by Pentaho DI Job. | N/A | N/A |
| 19 | ClinicalExchangePharmacyInformation | This is a RESTful Web Service. It is used to send pharmacy data to be updated in Change Healthcare. | Service | N/A | N/A | It is called by Pentaho DI Job. | N/A | N/A |
| 20 | VistAErxStatuses | This is a RESTful Web Service. It gets the status of a prescription while it goes through processing. | Service | N/A | N/A | It is called by VistA OP ErxWSCaller. | N/A | N/A |
| 21 | PrescriptionDrugs | This is a RESTful Web Service. It checks drug information in local VistA OP via an RPC call. | Service | N/A | N/A | It is called by Pentaho DI Job. | N/A | N/A |
| 22 | PatientEligibility | This is a RESTful Web Service. It searches for patient and their eligibility | Service | N/A | N/A | It is called by Pentaho DI Job. | N/A | N/A |
| 24 | DASReceiver | This is a job inside Pentaho DI which picks up new eRxs messages from DAS JMS queue. | Service | N/A | N/A | N/A | N/A | N/A |
| 26 | Reports | This service allows running of reports on eRx. | Service | N/A | N/A | Internal calls to this service are performed via direct Java method calls. | N/A | N/A |

Internal Data Stores

| ID | Name | Data Stored | Steward | Access |
| --- | --- | --- | --- | --- |
| 4 | eRx Database | The eRx DB persists all the information needed by the Inbound eRx application. | *Intermediate Service layer classes perform the CRUD operations needed by the application.* | *The Inbound eRx application performs entity create, read, update and delete operations on the eRx database.* |

### Application Locations

Table : Application Locations

| Application Component | Description | Location at Which Component is Run | Type |
| --- | --- | --- | --- |
| View, confirm, validate eRx, manage, fill, print labels, dispense, etc. an inbound eRxSend/transmit inbound eRx. | AITC – Inbound eRx server | VA Pharmacists, VA Pharmacy Technicians/Assistants, VA Pharmacy Managers will be allowed to access these Inbound eRx application’s services. | View, confirm, validate eRx, manage, fill, print labels, dispense, etc. an inbound eRxSend/transmit inbound eRx. |
| Business Services | Supports presentation logic, business logic, and data CRUD services for information in the Inbound eRx application. | AITC – Inbound eRx server | Presentation Logic/ Business Logic/Data Logic (via Hibernate) |
| VistA Link Service | Supports the Inbound eRx application interface to VistA. | AITC – Inbound eRx server | Interface Code |

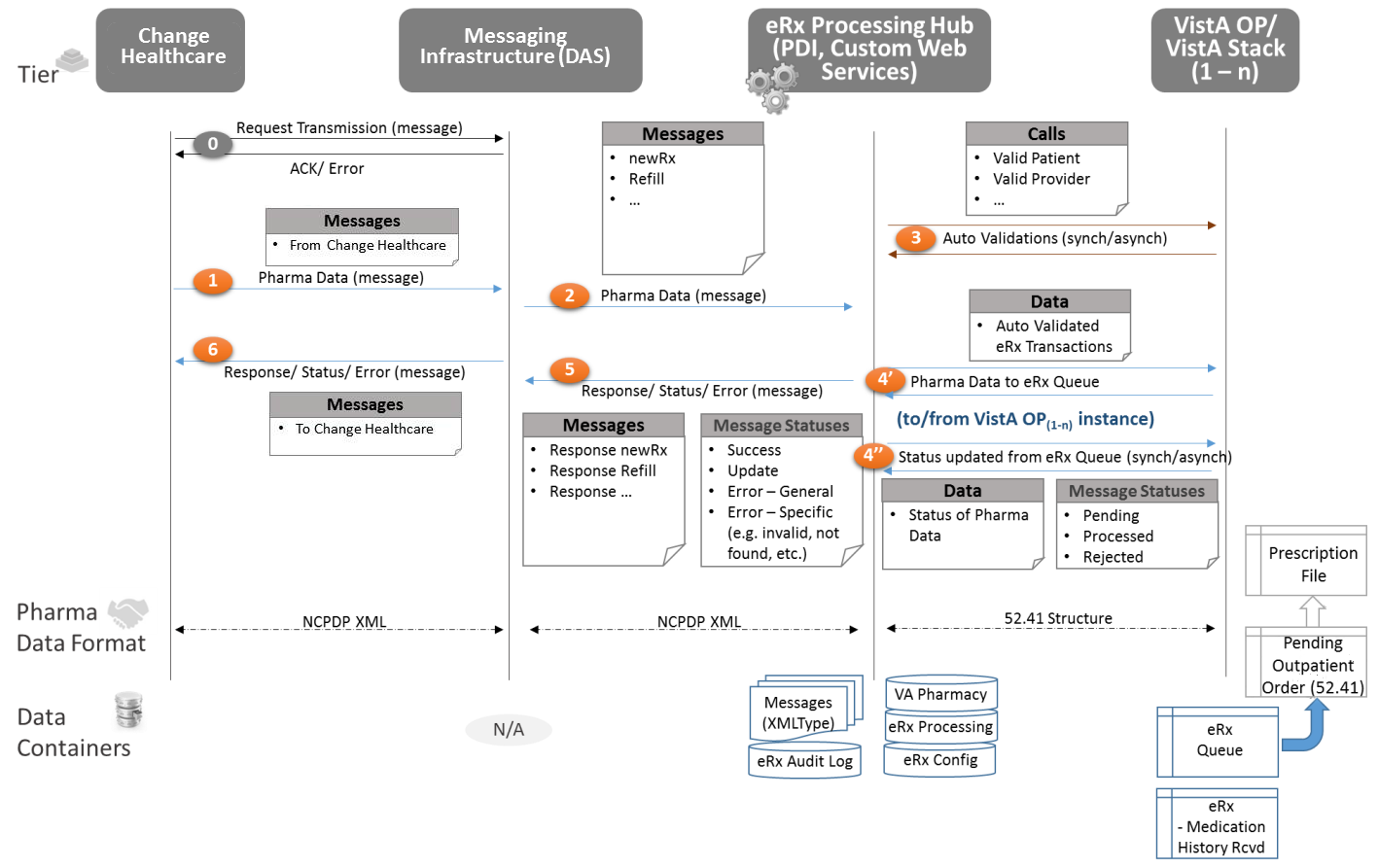
Table : Application Users

| Application Component | Location | User |
| --- | --- | --- |
| View, confirm, validate eRx, manage, fill, print labels, dispense, etc. an inbound eRxSend/transmit inbound eRx. | AITC – Inbound eRx server | VA Pharmacists, VA Pharmacy Technicians/Assistants, VA Pharmacy Managers will be allowed to access these Inbound eRx application’s services. |

## Conceptual Data Design

The conceptual data design presented in this section includes eRx Database design concepts and functionality, as well as conceptual details related to VistA OP files involved in the end-to-end Inbound eRx solution. Figure 4 depicts the conceptual data design and the data flows between the four (4) architectural tiers – Change Healthcare (external to VA), Messaging Infrastructure (DAS), eRx Processing Hub and VistA OP/VistA Stack.

Figure : Conceptual Data Flows and Designs



The main communication messages include:

* Step 0 – “handshake” messages between Change Healthcare and DAS.
* Step 1 – Pharma Data messages containing eRx message from Change Healthcare in NCPDP XML format; Change Healthcare supported messages are pre-defined and it is anticipated that new messages are not going to be added often.
* Step 2 – the eRx Processing Hub “listens” to the configured DAS Message Queue and receives the eRx messages; structural validation is performed on the eRx message; messages are of different types (e.g., newRx, Refill).
* Step 3 – the eRx Processing Hub coordinates auto-validations of the received eRx message (e.g., MVI call to identify and validate the patient).
* Step 4’ – validated eRx transaction is sent to the staging area in VistA OP (the eRx Holding Queue).
* Step 4’’ –eRx Processing Hub data is synchronized with the status of the transmitted eRx transactions in the eRx Holding Queue in VistA OP; the eRx processing logic in VistA OP is responsible for updating the status of each eRx transaction in the eRx Holding Queue.
* Step 5 – a response message is constructed and sent to DAS for transportation to Change Healthcare; response messages are of type corresponding to the incoming eRx messages (i.e., Response newRx, Response Refill, etc.); each response message is associated with a message status such as Success, Update, Error (general or specific – invalid, not found, or other).
* Step 6 – DAS receives the message in the eRx Message Queue and message is sent to Change Healthcare.

***Note:*** The step sequence described above is representative. Depending on the type of message and its content, some of the steps may be repeated multiple times (e.g., different match criteria are sent for patient identification and validation) or skipped entirely (e.g., message structure validation error occurs).

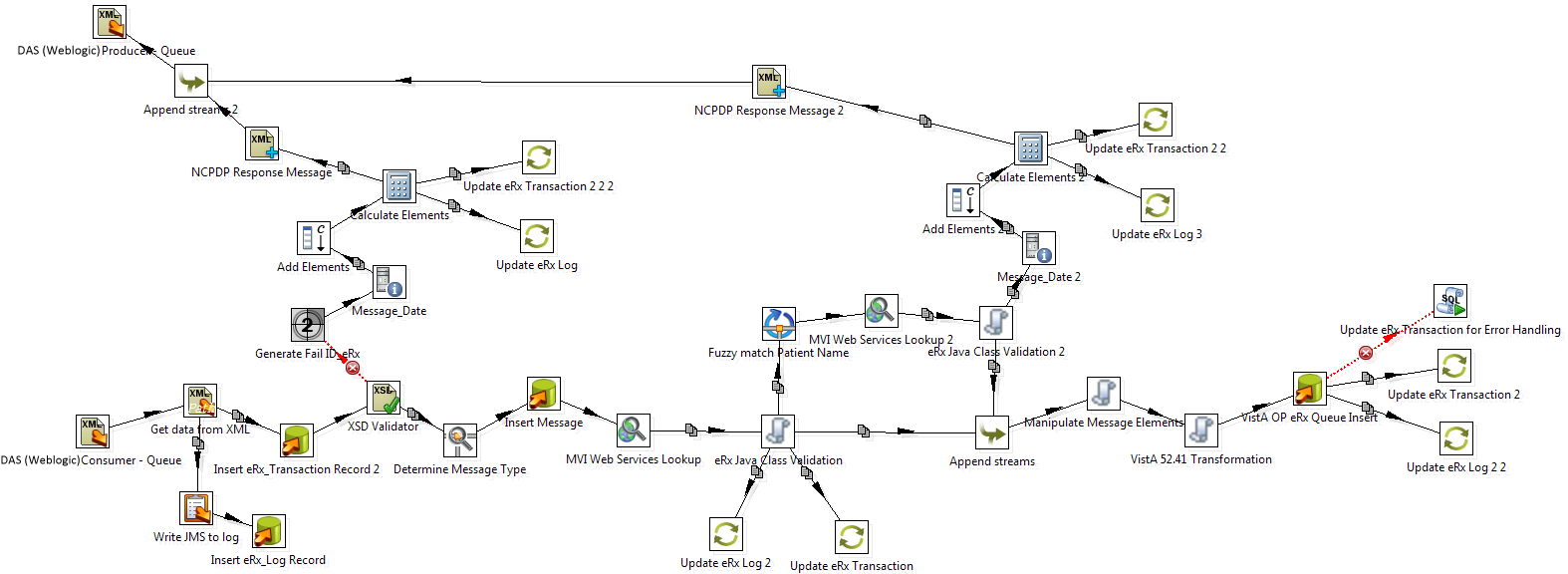
### Pharma Data Formats

* The data format between the architectural Tiers Change Healthcare and Messaging Infrastructure (DAS) is based on NCPDP XML messages as supported by Change Healthcare.
* The data format between the architectural Tiers Messaging Infrastructure (DAS) and eRx Processing Hub (Pentaho DI) is based on NCPDP XML messages as sent by Change Healthcare; no XML message manipulation is anticipated in the Messaging Infrastructure (DAS) tier.
* The data format between the architectural Tiers eRx Processing Hub (Pentaho DI) and VistA OP/ VistA Stack is synchronous (i.e., not message based); auto validation calls are in the format expected by the respective APIs/RPCs or other interfaces (e.g., patient MVI validation); Pharma data is “sent” and status data is “received” to/from the eRx Holding Queue in VistA OP (the staging area) via SOA data connection.

### High Level eRx Processing Hub Logic (Example Transaction)

This section depicts the high level eRx Processing Hub logic for newRx message transaction with Pentaho Data Integration (Figure 5). Please note that this logic is for illustrative purposes and the actual processing logic will likely be more complex.

Figure : eRx Message Processing Logic – Example



#### Integration with DAS (Oracle WebLogic JMS Queue)

VA’s DAS is a messaging and queuing middleware provider, which can be leveraged via the Java Message Service (JMS) specification. In this manner, Pentaho Data Integration exposes Oracle WebLogic JMS Queue. Consumer and Producer transformation steps allowing access to produce and consume messages through Oracle WebLogic. Oracle WebLogic provides this functionality through Oracle’s implementation of JMS.

With JMS (and as such, also with Oracle WebLogic JMS) it is possible to process messages in two basic methods:

* Queues - A message can be sent to a specified WebLogic JMS Queue and can be received by exactly one consumer. If there are no consumers available at the time the message is sent, it will be kept until a consumer is available that can process the message. This is also known as a point-to-point model.
* Topics - In WebLogic JMS a Topic implements publish and subscribe semantics. When a message is published it goes to all the subscribers who are interested - zero to many subscribers will receive a copy of the message. Only subscribers who had an active subscription at the time the broker receives the message will get a copy of the message. This is also known as a publish/subscribe model.

The eRx Processing Hub leverages the **Queues method** to interface with DAS using Oracle WebLogic JMS Queue.

#### High Level Processing Flow (for illustrative purposes)

* Message from Change Healthcare arrives in the DAS and is then forwarded to the JMS Queue
* Message is being read with the XML Get Data Utility; the JMS log is being read and the eRx\_Log and eRx\_Transaction records are created
* The structure of the message is validated with the XSD Validator Utility
  + If Error, NCPDP XML message is constructed, transaction and log tables are updated, and a response message is sent via DAS to Change Healthcare
  + If success, message type is determined (e.g., newRx) and raw message is stored in the respective XMLMessage table
* MVI validation is performed using the web services validation utility
  + If Error, for illustrative purposes, we have depicted a scenario where we do not reject the message; instead, a fuzzy match is performed based on “like” names where traditional lookup matches do not work and Levenshtein matching algorithm is used instead; if error, response message is constructed and sent via DAS to Change Healthcare
  + If success, the eRx transaction is (in this simplified example) considered auto-validated and ready to be sent to VistA OP eRx Holding Queue
* Message is manipulated and transformed to match the VistA OP expected structure; message is transferred to the VistA OP via an RPC.
  + For simplicity purposes, if case or error, the transaction status is updated allowing a separate (not depicted here) error handling process to identify and process the failed transaction again

As stated earlier, the purpose of this section is to outline a high level example of a design approach to processing eRx messages. As with all ETL transformations, the design can be finalized during a pilot phase and after extensive testing.

### Project Conceptual Data Model

Figure 4 Conceptual Data Flows and Designs (above) depicts the main data containers in each architectural tier. Please see [Section 5.4.1](#_eRx_Processing_Hub) for ERD Diagram.

* Messaging Infrastructure (DAS):
  + Data Containers: N/A
  + *Context:* The DAS SOA Suite is middleware software with no data storage other than message buffering.
* eRx Processing Hub (Pentaho DI):
  + Data Containers:
    - XML Messages – stores eRx inbound and outbound messages as XMLType (one table per message type).
    - eRx\_Transaction – stores processing data for managing the lifecycle of each eRx message.
    - VA\_Pharmacy – stores VA Pharmacy information.
    - eRx\_Config – stores eRx Processing Hub configuration and translation data.
    - eRx\_Log – stores audit log data and status changes for each eRx transaction.
  + *Context:* Data containers are stored in an Oracle 11g database.
* VistA OP:
  + Data Containers:
    - eRx – Holding Queue VistA OP – staging area for the validated by the *eRx Processing Hub* eRx transaction data; has a similar structure as the Pending Outpatient Order (#52.41) file + the eRx status flag(s) and “link(s)” with the eRx Transaction Hub records.
    - eRx – Medication History Rcvd – staging area to receive patient Medication History information.
  + *Context:* VistA OP eRx functionality, which is developed as part of this initiative, processes the received eRx transactions similarly as the existing VistA OP functionality to perform drug-to-drug, drug-to-allergy, DEA special handling and other validations; as part of the processing, the eRx transaction is loaded into the PENDING OUTPATIENT ORDERS (52.41) and PRESCRIPTION file (#52).

Section 5 of this document depicts further technical details about the data model.

### Database Information

Table : Database Inventory

| Database Name | Description | Type | Steward |
| --- | --- | --- | --- |
| Inbound eRx | Oracle 11g Relational Database used to store, process and archive eRx data and application metadata | Create | Inbound eRx |
| eRx Holding Queue VistA OP | New data file with similar to 52.41 structure for staging eRx transactions for processing within VistA OP | Create | VistA OP |

### User Interface Data Mapping

This section describes and defines the information that will be available for users of the product to be able to enter data into the database or to retrieve information from the database, if applicable.

Because the Inbound eRx system is at a preliminary design stage, some design details are not yet available and will be addressed in this document prior to final review. Additionally, it is expected that the information provided may need to be changed during later design stages.

#### Application Screen Interface

This section describes the eRx screen interfaces.

##### Login Screen

Figure 6 depicts the eRx login screen. If the authentication and authorization is successful, the user is taken to the home screen with a welcome message (shown in home screen).

Figure : eRx GUI Login Screen

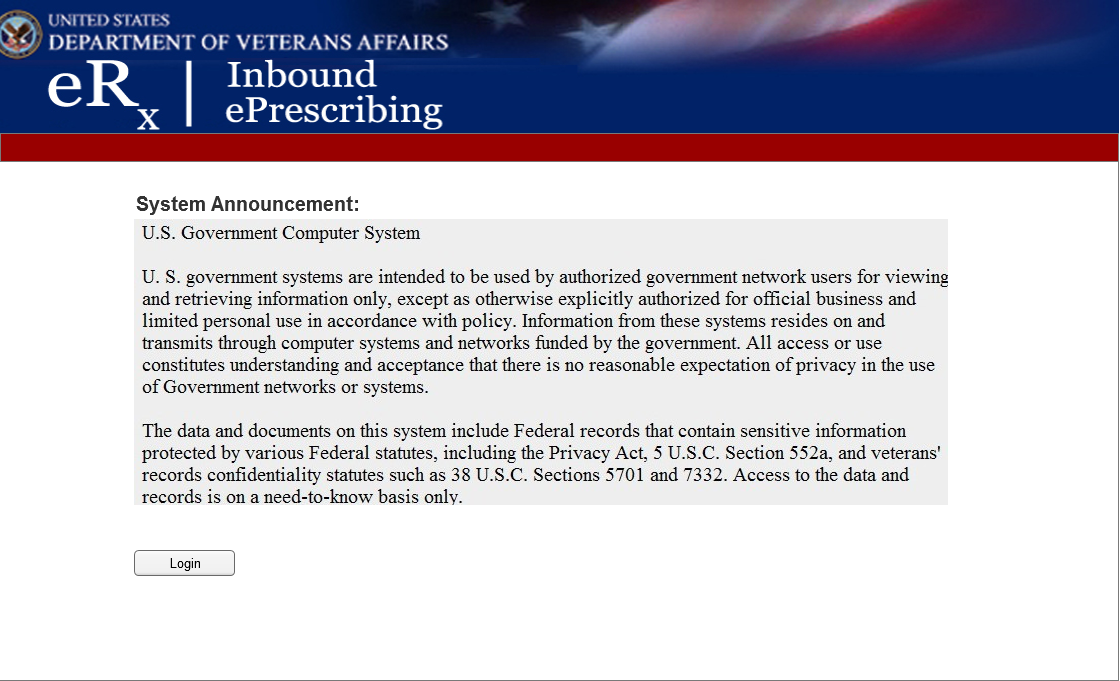


Table : Screen Description

| Graphical User Interface (GUI) Field | Table (Database Table that field connects to) | Field (Field in Table that the GUI field connects to) | Comments |
| --- | --- | --- | --- |
| No fields defined on this screen. | N/A | N/A | N/A |

##### Home Screen

Figure 7 shows the eRx application home screen after the user is authenticated successfully. It shows a welcome message.

Figure : eRx Home Screen



Table : Screen Description

| Graphical User Interface (GUI) Field | Table (Database Table that field connects to) | Field (Field in Table that the GUI field connects to) | Comments |
| --- | --- | --- | --- |
| No fields defined on this screen. | N/A | N/A | N/A |

##### View eRx Screen

Figure 8 shows the eRx application View eRx Screen. It allows authorized users to view an incoming eRx. Table 12 describes it.

Figure : eRx View eRx Screen

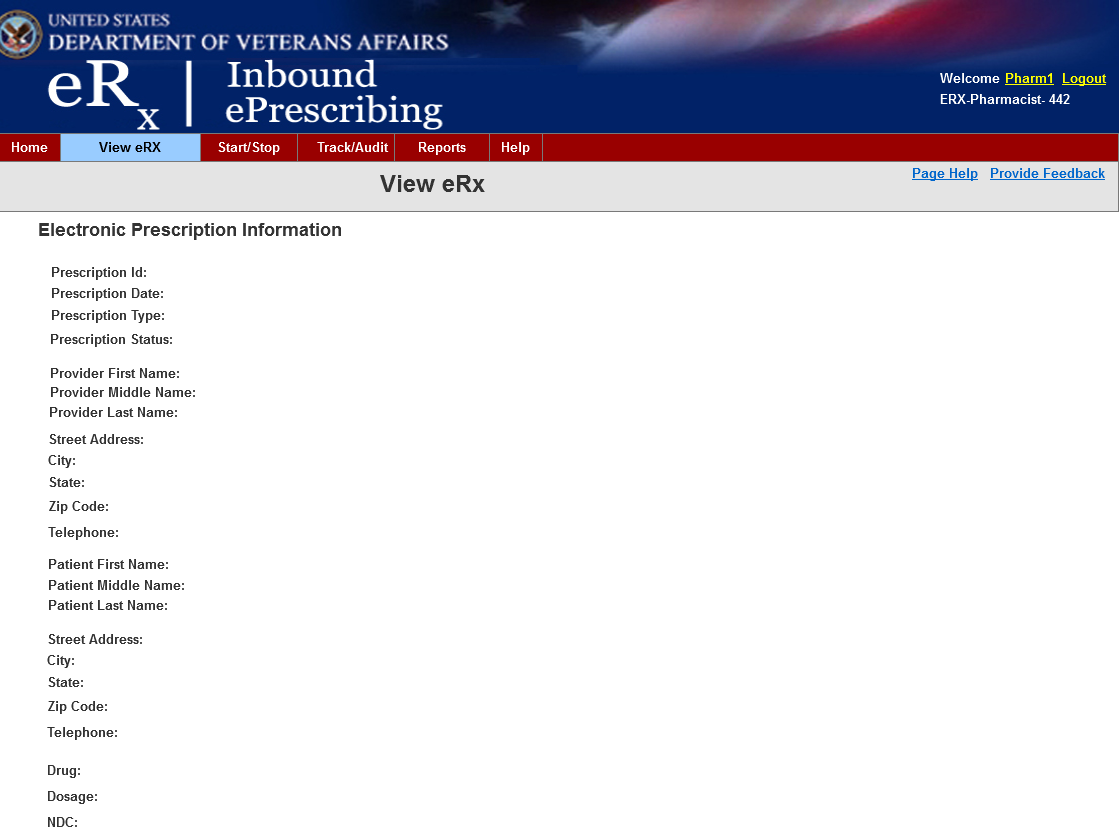


Table : Screen Description

| Graphical User Interface (GUI) Field | Table (Database Table that field connects to) | Field (Field in Table that the GUI field connects to) | Comments |
| --- | --- | --- | --- |
| Prescription Id | newRxInbound | MessageID | Unique message id of prescription |
| Prescription Date | newRxInbound | XMLType NCPDP element | Date prescription written |
| Prescription Type | newRxInbound | XMLType NCPDP element | Type of prescription |
| Prescription Status | newRxInbound | XMLType NCPDP element | Status of prescription |
| Provider First Name | newRxInbound | XMLType NCPDP element | First name of Prescribing provider. |
| Provider Middle Name | newRxInbound | XMLType NCPDP element | Middle name of Prescribing provider |
| Provider Last Name | newRxInbound | XMLType NCPDP element | Last name of Prescribing provider |
| Street Address | newRxInbound | XMLType NCPDP element | Street address of Prescribing provider |
| City | newRxInbound | XMLType NCPDP element | City of Prescribing provider |
| State | newRxInbound | XMLType NCPDP element | State of Prescribing provider |
| Zip Code | newRxInbound | XMLType NCPDP element | Zip Code of Prescribing provider |
| Telephone | newRxInbound | XMLType NCPDP element | Telephone of Prescribing provider |
| Drug | newRxInbound | XMLType NCPDP element | Drug prescribed |
| Dosage | newRxInbound | XMLType NCPDP element | Drug dosage |
| NDC | newRxInbound | XMLType NCPDP element | National Drug Code |
| Patient First Name | newRxInbound | XMLType NCPDP element | First name of patient. |
| Patient Middle Name | newRxInbound | XMLType NCPDP element | Middle name of patient |
| Patient Last Name | newRxInbound | XMLType NCPDP element | Last name of patient |
| Street Address | newRxInbound | XMLType NCPDP element | Street address of patient |
| City | newRxInbound | XMLType NCPDP element | City of patient |
| State | newRxInbound | XMLType NCPDP element | State of patient |
| Zip Code | newRxInbound | XMLType NCPDP element | Zip Code of patient |
| Telephone | newRxInbound | XMLType NCPDP element | Telephone of patient |
| Drug | newRxInbound | XMLType NCPDP element | Drug prescribed |
| Dosage | newRxInbound | XMLType NCPDP element | Drug dosage |
| NDC | newRxInbound | XMLType NCPDP element | National Drug Code |

##### Start/Stop Screen

Figure 9 shows the eRx application start/stop screen. It allows the eRx application to stop receiving prescriptions targeted for delivery to a particular pharmacy. This screen will only be visible to users authorized to enable/disable pharmacies from receiving eRx’s. Table 13 describes it.

Figure : eRx Start/Stop Screen



Table : Screen Description

| Graphical User Interface (GUI) Field | Table (Database Table that field connects to) | Field (Field in Table that the GUI field connects to) | Comments |
| --- | --- | --- | --- |
| Pharmacy | VAPharmacy | Name | Pharmacy to start/stop eRx |
| Action | VAPharmacy | C\_Active | Action to take: Start/Stop |
| Reason | VAPharmacy | Code | The reason the eRx reception is stopped/started. |

##### Track/Audit Screen

Figure 10 depicts eRx application track/audit screen. It allows the pharmacist or pharmacy technician to track a prescription or see an audit trail of a prescription. Table 14 describes it.

Figure : eRx Track/Audit Screen



Table : Screen Description

| Graphical User Interface (GUI) Field | Table (Database Table that field connects to) | Field (Field in Table that the GUI field connects to) | Comments |
| --- | --- | --- | --- |
| Search Criteria | eRx\_Transaction | Multiple fields will be queried that are yet to be defined. | Allows changing the criteria used for the search. |
| Prescription Status | eRx\_Transaction | C\_TransactionStatus | The status of the prescription targeted for the search. |

##### Help Screen

The help screen is a general purpose screen which provides help topics and production support information.

Figure : Help Screen

Table : Screen Description

| Graphical User Interface (GUI) Field | Table (Database Table that field connects to) | Field (Field in Table that the GUI field connects to) | Comments |
| --- | --- | --- | --- |
| N/A | N/A | N/A | N/A |

#### Application Report Interface

This section describes and defines the reports that will be available in the user interface.

##### Reports Screen

Figure 12 shows the eRx application reports screen. It allows the pharmacist or pharmacy technician to access eRx reports such as Daily, Weekly or Monthly transaction metrics. Table 16 describes it.

Figure : Reports Screen



Table : Report Screen Description

| Report Column | Data Source |
| --- | --- |
| Report Type | Not currently defined. |

#### Unmapped Data Element

In this section describe any database element that was not mapped to a screen and the reason the data element(s) was not mapped. This section may be skipped if there is no User Interface involved in the project, such a building a service offering etc.

## Conceptual Infrastructure Design

The Conceptual Infrastructure Design is a high-level overview of the infrastructure that will be used to support the Inbound eRx application. Primary emphasis is on the environments that will be required and the locations at which they will be installed. The Conceptual Infrastructure Design becomes more detailed at later stages as more information is collected regarding the system, and the infrastructure requirements (i.e., capacity requirements) are better known.

Because the Inbound eRx system is at a preliminary design stage, it is expected that the information provided may need to be changed during later design stages or increments.

### System Criticality and High Availability

The Inbound eRx project is part of the overall PRE program and will not have separate system criticality and high availability requirements. As such, Inbound eRx will follow the same criteria that the rest of the PRE applications use.

The Pentaho Data Integration (component of the eRx Processing Hub) scalability and high-availability considerations are depicted in section 3.3.2 Special Technology below.

The Inbound eRx project will follow VA guidance on contingency planning and disaster recovery. It will rely on AITC Disaster Recovery Services. Detailed Disaster Recovery specifications can be found in the Inbound ePrescribing RSD, Section 2.4, Disaster Recovery Specifications.

In line with the requirements established by Pharmacy Benefits Management (PBM) in the product’s Requirements Specification Document (RSD), the system receives “essential support” as defined in the AITC Disaster Recovery Services.

### Special Technology

This section contains the special technology requirements for the *eRx Processing Hub*.

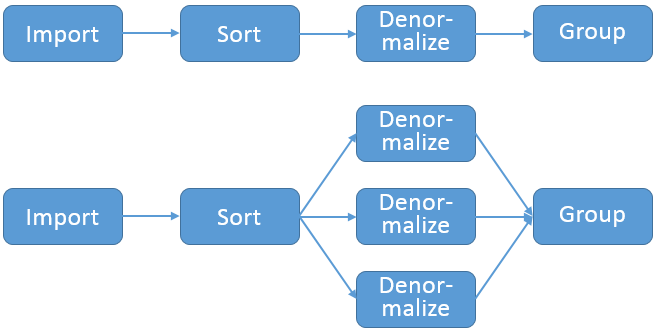
Table : Special Technology Requirements

| Special Technology | Description | Notional Location | TRM Status |
| --- | --- | --- | --- |
| Pentaho Data Integration\* | Comprehensive ETL platform with the ability to efficiently manipulate XML messages, out-of-the-box integration with Oracle Weblogic JMS | Co-located as the eRx Processing Hub components | TRM Approved, version 6.x (current version) |

#### PDI Multi-threaded Architecture

PDI’s streaming engine architecture provides the ability to work with extremely large data volumes, and provides Enterprise-class performance and scalability with a broad range of deployment options including dedicated, clustered, and/or cloud-based ETL servers. The architecture allows both vertical and horizontal scaling. The engine executes tasks in parallel and across multiple CPUs on a single machine as well as across multiple servers via clustering and partitioning. Figure 13 depicts a multi-thread example of a single step.

Figure : Example Flow with Multiple Threads for a Single Step (Row Denormalizer)



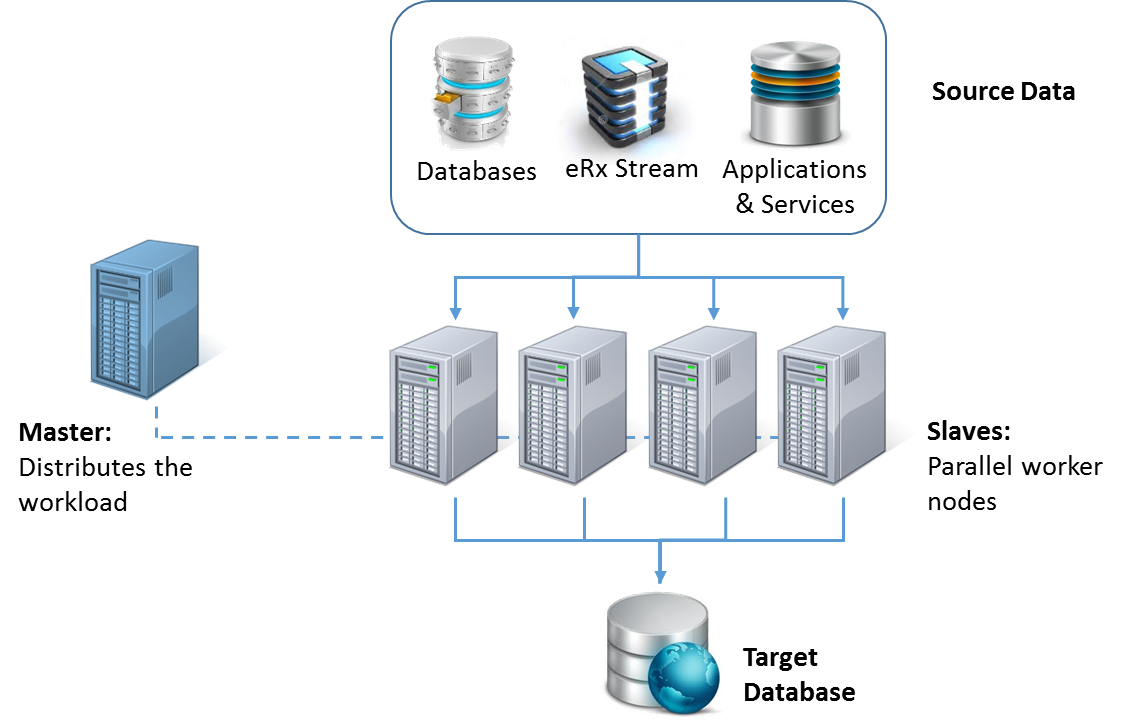
#### Transformation Processing Engine

PDI’s transformation processing engine starts and executes all steps within a transformation in parallel (multi-threaded) allowing maximum usage of available CPU resources. Done by default this allows processing of an unlimited number of rows and columns in a streaming fashion. Furthermore, the engine is 100% metadata driven (no code generation) resulting in reduced deployment complexity. PDI also provides different processing engines that can be used to influence thread priority or limit execution to a single thread which is useful for parallel performance tuning of large transformations. Additional tuning options include the ability to configure streaming buffer sizes, reduce internal data type conversions (lazy conversion), leverage high performance non-blocking I/O (NIO) for read large blocks at a time and parallel reading of files, and support for multiple step copies to allowing optimization of Java Virtual Machine multi-thread usage.

#### Clustering and Partitioning

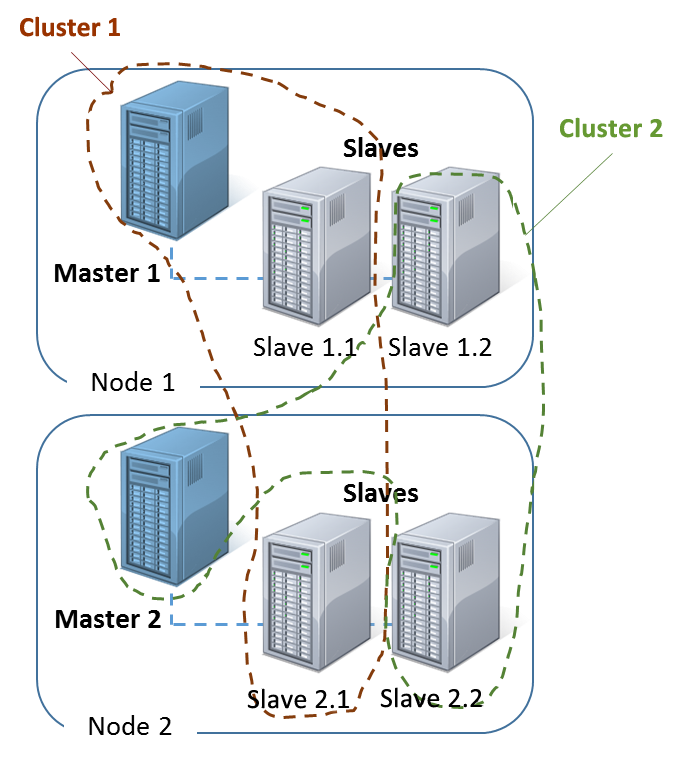
PDI provides advanced clustering and partitioning capabilities that allow Inbound eRx to scale up the processing of incoming external prescriptions. PDI clusters are built for increasing performance and throughput of data transformations; in particular they are built to perform classic “divide and conquer” processing of data sets in parallel. PDI clusters have a strong master/slave topology (Figure 14). There is one master in cluster, but there can be many slaves. This cluster scheme can be used to distribute the processing workload in parallel appropriately across these multiple systems. Transformations are broken into master/slaves topology and deployed to all servers in a cluster – where each server in the cluster is running a PDI engine to listen, receive, execute and monitor transformations. It is also possible to define dynamic clusters where the Slave servers are only known at run-time.

Figure : Clustering in Pentaho Data Integration



PDI clusters are completely software based and can be deployed as physical servers or as virtual servers. As described above, PDI clusters consist of a master service and one or more slave services. A master service can be configured to be dynamic, allowing new slave services to self-register with the master service without interrupting transformations currently running on the cluster.

Each master and slave service is implemented by a light-weight web service called Carte. Furthermore, multiple slave services can run on the same server as the master (master server) or on separate servers (slave servers). Figure 15 depicts a contextual design of fail-over architecture based on two (2) nodes, running two (2) clusters with each cluster having a master service (Master 1 and Master 2, respectively) and two (2) slave services (Slave 1.1 and Slave 1.2 for Master 1, and Slave 2.1 and Slave 2.2 for Master 2). Note that, this depiction is for illustrative purposes and that the PDI component will be implemented within the overall eRx Processing Hub fail-over architecture.

**Figure 15: Conceptual Fail-Over PDI Cluster Architecture**

### Technology Locations

This section will be completed during a future increment.

Table : Technology Location Details

| Technology Component  Production 1 | | Location | | Usage | |
| --- | --- | --- | --- | --- | --- |
| Workstations | |  | |  | |
| Special Hardware | |  | |  | |
| Interface Processors | |  | |  | |
| Legacy Mainframe | |  | |  | |
| Legacy Application Server | |  | |  | |
| Legacy Databases | |  | |  | |
| Other | |  | |  | |
| Technology Component  Production 2 | | Location | | Usage | |
| <copy from Prod 1 set, or enter new ones as appropriate> | |  | |  | |
| Technology Component  Certification | | Location | | Usage | |
|  | |  | |  | |
| Technology Component  Education | | Location | | Usage | |
|  | |  | |  | |
| Technology Component  Test | Location | | Usage | |
|  |  | |  | |
| Technology Component  Development | Location | | Usage | |
|  |  | |  | |

### Conceptual Infrastructure Diagram

#### Location of Environments and External Interfaces

Figure 16 is a conceptual environment diagram.

**Figure 16: Conceptual Networks and Environments**



#### Conceptual Production String Diagram

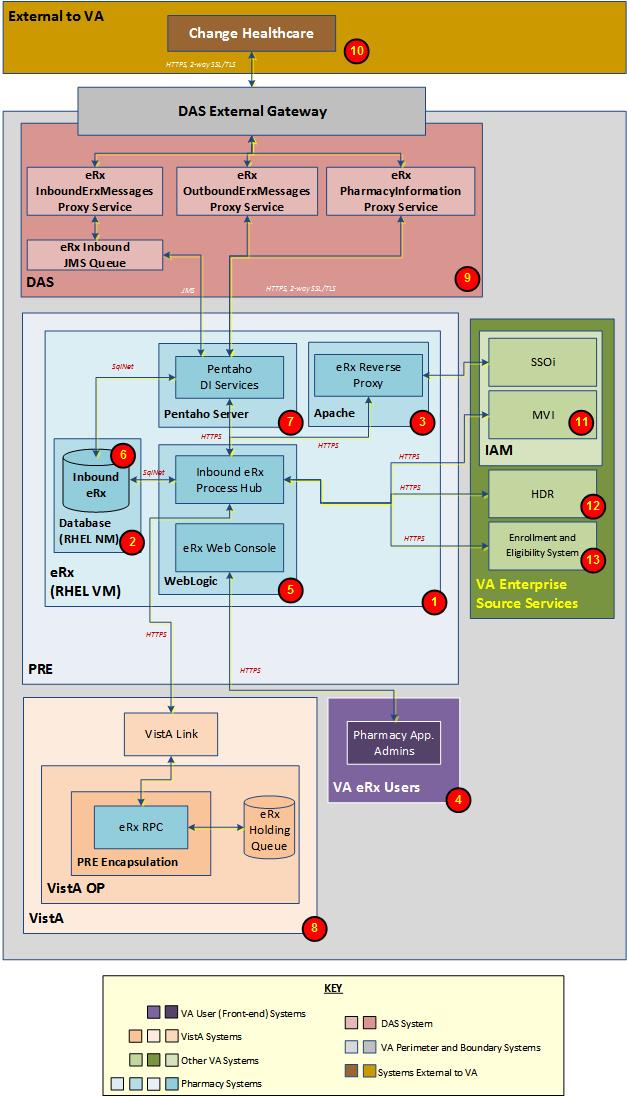
Figure 17 shows the conceptual Production String.

Figure : Conceptual Production String DiagramThis image depicts the conceptual Production String

# System Architecture

The system architecture is comprised of the hardware, software, and communication architectures. The hardware architecture describes the physical components needed in the system and their relationship to one another. The software architecture describes the software needed to support the system and what hardware component each software component will exist on. The communication architecture describes the connections needed between the hardware components.

Figure : Inbound eRx System Architecture



## Hardware Architecture

The table below references the Figure 18 System Architecture from above and further describes the Hardware Components.

| ID | System Component | Comments |
| --- | --- | --- |
| 1 | Redhat Enterprise Linux VM | Apache HTTP, WebLogic 12c, Pentaho Data Integration |
| 2 | Redhat Enterprise Linux VM | Oracle Database 11g |

## Software Architecture

This section describes the overall system software and organization, building upon the High Level Application Design. The architecture for Inbound eRx is defined as a multi-tiered system consisting of the following logical layers:

* Presentation Layer – Contains user interface components
* Service Layer – Contains business logic components
* Domain Layer – Contains data and database interfacing components
* Interface Layer – Contains interface components, to systems external to the Inbound eRx application

The table below references the Figure 18 System Architecture from above and further describes the Software Components.

| ID | System Component | Software | Function |
| --- | --- | --- | --- |
| 3 | Web Server | Apache HTTP 2.2 | Reverse proxy, load balancing and WebLogic high availability clustering support, static content |
| 4 | Client Workstation/Laptop | Internet Explorer 11 or greater | User Interface |
| 5 | Application Server | WebLogic 12.1 | Web Service Orchestration, eRx Initial Validation, RESTful Services supporting UI |
| 6 | Database Server | Oracle 11g | Storage of Inbound eRx messages and application metadata |
| 7 | Data Integration Server | Pentaho Data Integration 6.0 | XML Extraction, Transformation and Loading |
| 8 | VistA Outpatient Pharmacy (OP) | VistA/MUMPS | Existing VistA OP module extended to support Inbound eRx |
| 9 | Data Access Services (DAS) | Oracle Weblogic | Provides access to VA shared services |
| 10 | Change Healthcare eRx Network | Externally hosted (non-VA) | Verifies and transmits eRx transactions to/from external provider EHR systems and the Inbound eRx system |
| 11 | Master Veteran Index | Existing VA system | Provides patient lookup service for incoming eRx message validation |
| 12 | Health Data Repository (HDR) | Existing VA system | Provides list of all prescribed medications for individual to fulfill eRx medication history requests |
| 13 | Enrollment and Eligibility System | Existing VA system | Provides patient enrollment and eligibility status for incoming eRx message validation |

### Components

Inbound eRx interacts with Change Healthcare using DAS as gateway.

DAS hosts the following proxy services for Inbound eRx:

**InboundErxMessages**

Proxy for incoming

**PharmacyInformation**

Proxy for outgoing updates to Change Healthcare.

**OutboundErxMessages**

Proxy for outgoing messages to Change Healthcare.

**Inbound eRx Middle Tier Components**

The Inbound eRx middle tier/ processing hub consists of the following components and services:

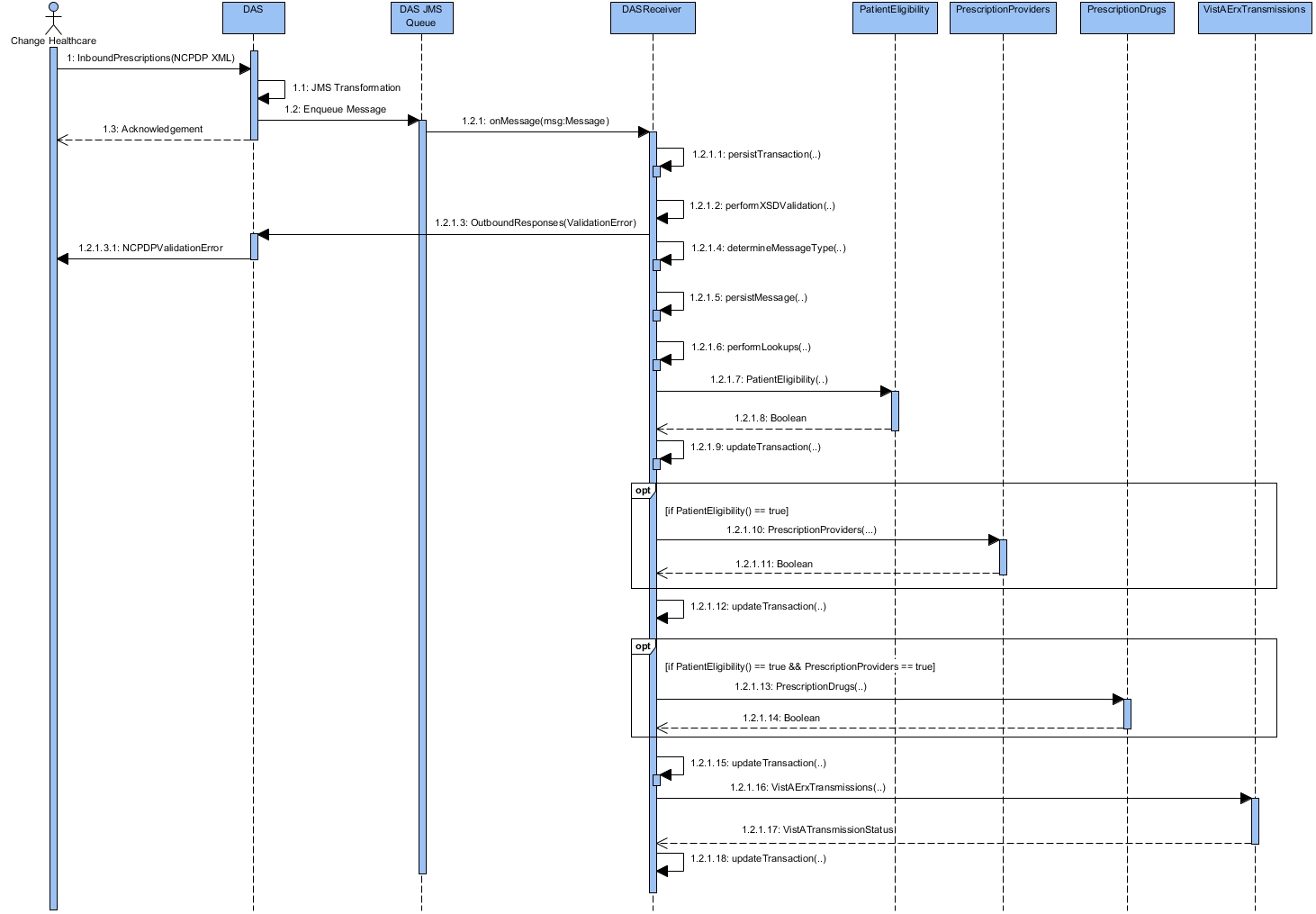
**DASReceiver**

It is a job created inside Pentaho DI when the application starts.

**Steps**

1. It picks up messages from DAS message queue and forwards them to Pentaho processing hub, where Pentaho parses the message, stores the prescription into its own XML type table and inserts a record into eRx\_Transaction table.
2. Pentaho DI then calls PatientEligibility, PrescriptionProviders PrescriptionDrugs for each prescription message being processed and updates the eRx\_Transaction table in the process.
3. Finally Pentaho DI calls VistAErxTransmission to transmit that prescription to VistA. Figure 19 shows the sequence diagram for DASReceiver.

Figure Inbound eRx – Incoming eRx Flow



**PatientEligibility**

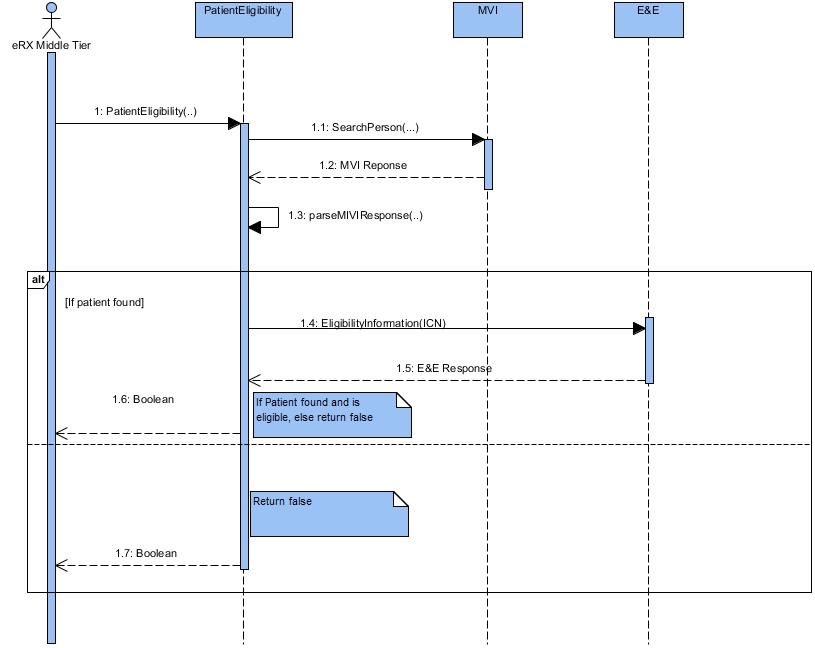
* 1. **Service Description**

This service is called by Pentaho DI and is implemented as a RESTful web service. It is a provider service. Figure 20 shows the sequence diagram for this service.

* 1. **Steps**
     1. It receives patient information Name (FirstName, Middle, Last Name), DOB, Gender, Address and Phone number.
     2. It calls the MVI RESTful web service by passing patient information.
     3. If the patient is not found, it stops further processing, and returns a Boolean flag of false.
     4. If the patient is found, it calls the E&E (Enrollment and Eligibility) web service passing the ICN returned by the MVI web service response and gets the E&E web service response back with the enrollment and eligibility information of the patient and returns a Boolean flag of true.
     5. It returns a Boolean flag of either true or false.

| **Parameter** | **Type** |
| --- | --- |
| patientFirstName | String |
| patientMiddleName | String |
| patientLastName | String |
| gender | String |
| address | String |
| telephone | String |
| dateofBirth | String |

Figure Inbound eRx – PatientEligibility Check Flow



**PrescriptionProviders**

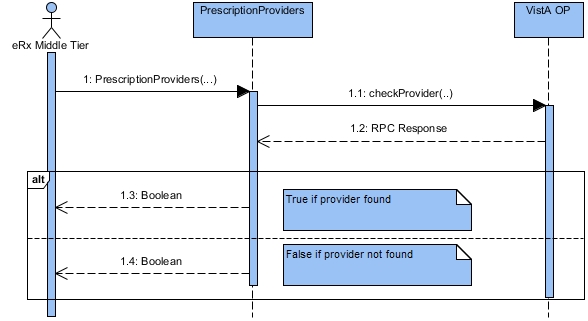
* 1. **Service Description**

This service is called by Pentaho DI and is implemented as a RESTful web service. It is a provider service.It is called even if the PatientEligibility web service returns false in the response as per requirements. Figure 21 shows the sequence diagram for this service.

* 1. **Steps**
     1. It receives provider information i.e., ID, Name, Address and Phone number
     2. It does a RPC call to VistA, passing in the provider information and gets a response back.
     3. If provider is found, it returns a Boolean flag of true.
     4. If provider is not found, it returns a Boolean flag of false.
     5. It returns a Boolean flag of either true or false

|  |  |
| --- | --- |
| **Parameter** | **Type** |
| providerID | String |
| providerName | String |
| proivderAddress | String |
| telephone | String |

Figure Inbound eRx – Prescription Provider Check Flow



**PrescriptionDrugs**

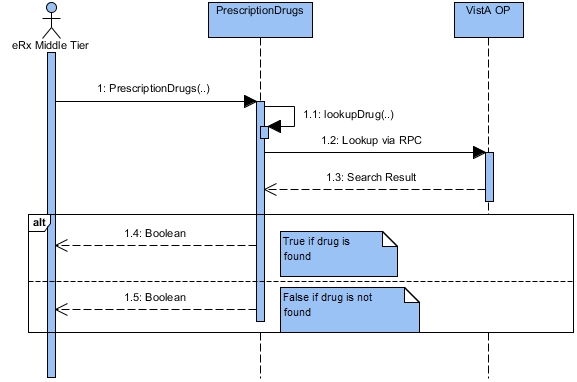
* 1. **Service Description**

This service is called by Pentaho DI and is implemented as a RESTful web service. It is a provider service. It is called even if the PatientEligibility and PrescriptionProviders web service return a false flag in their responses as per requirements. Figure 22 shows the sequence diagram for this service.

* 1. **Steps**
     1. It receives drug information i.e., drugName, ndc, drugClass, generic, ingredients.
     2. It searches local VistA OP for the drug with given information via RPC call and if the drug is found, it will return a Boolean flag of true.
     3. If drug is not found, it will return a Boolean flag of false.
     4. It returns a Boolean flag of either true or false.

| **Parameter** | **Type** |
| --- | --- |
| drugName | String |
| ndc | String |
| drugClass | String |
| generic | String |
| ingredients | String |

Figure Inbound eRx – Prescription Drug Check Flow



**PharmacyTransfers**

* 1. **Service Description**

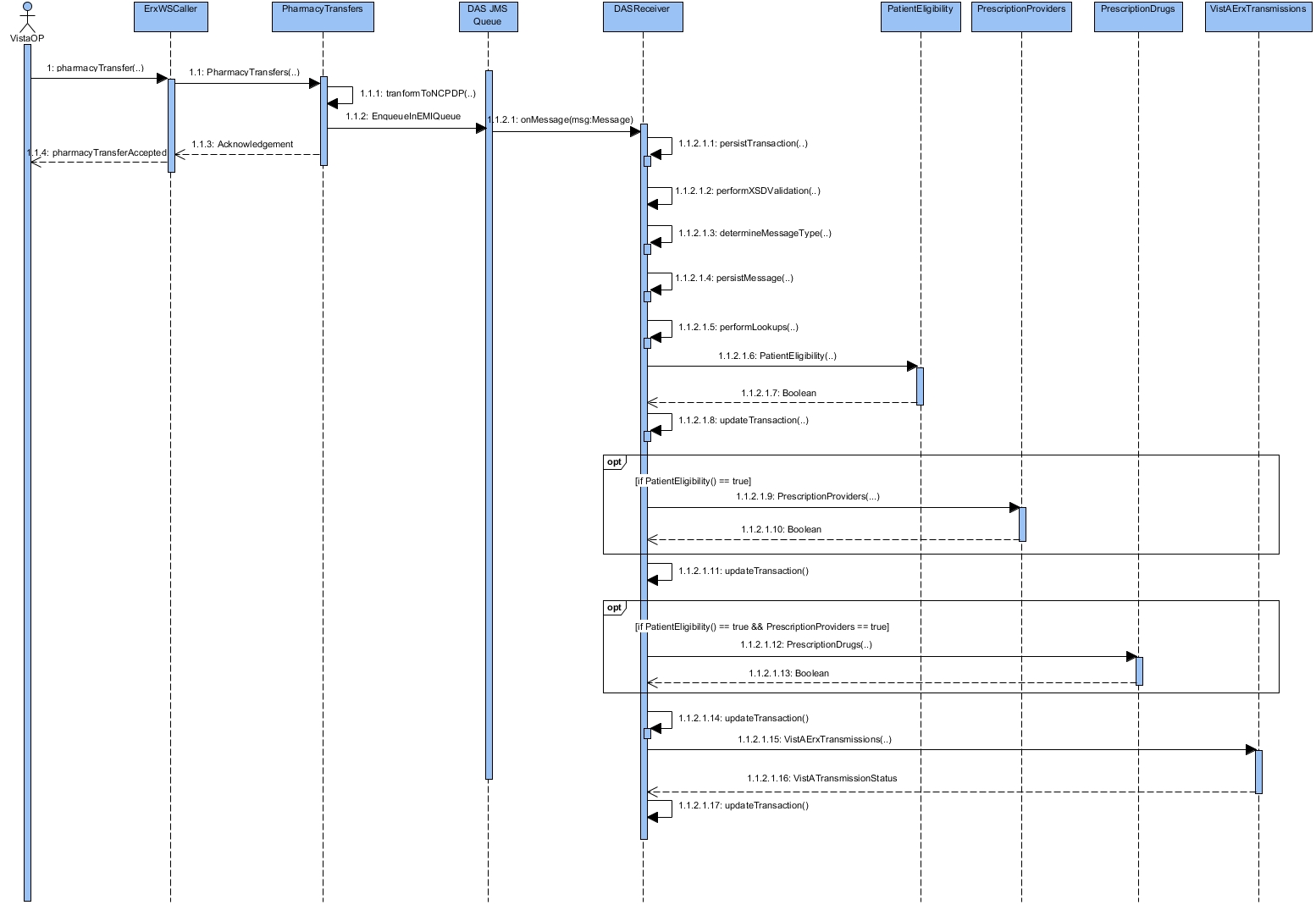
This service is called by VistA OP and is implemented as a RESTful web service. It is a processing service. Figure 23 shows the sequence diagram for this service.

**Steps**

* + 1. It receives prescription details, which is to be transferred to another pharmacy, via VistA OP.
    2. It builds the XML conforming to NCPDP prescription transfers and puts it into DAS JMS Queue where DASReceiver picks it up and sends to Pentaho middle tier for processing which finally calls VistAErxTransmissions to transfer it.
    3. It returns success in the response; if all processing is successful otherwise it returns an error.

| **Parameter** | **Type** |
| --- | --- |
| pharmacyID | String |
| prescriptonID | String |
| prescriptionDetails | String |
| patientID | String |
| drugID | String |
| prescriptionDate | String |

**Figure 23 Inbound eRx – Pharmacy Transfer Flow**



**VistAErxTransmissions**

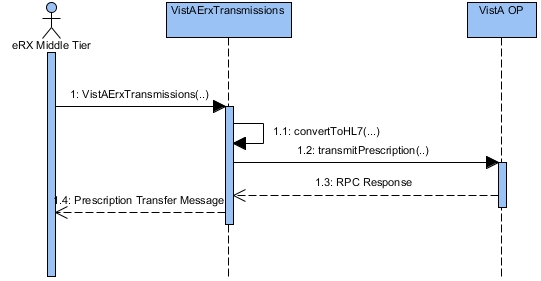
1. **Service Description**

This service is called by Pentaho DI and is implemented as a RESTful web service. It is a processing service. It is called only after PatientEligibility, PrescriptionProviders and PrescriptionDrugs have been called and returned a response. Figure 24 shows the sequence diagram for this service.

1. **Steps**
   * 1. It receives prescription information (shown below) to be transferred to VistA OP.
     2. It translates/converts the data to HL7 for VistA OP with status indicating success or error.
     3. It transmits to VistA OP via RPC call and gets a response back.
     4. It returns success in the response; if all processing is successful otherwise it returns an error.

| **Parameter** | **Type** |
| --- | --- |
| pharmacyName | String |
| pharmacyID | String |
| address | String |
| city | String |
| state | String |
| zipcode | String |
| telephone | String |
| patientID | String |
| patientName | String |
| patientAddress | String |
| patientTelephone | String |
| drugID | String |
| dosageDetails | String |
| prescriptionDate | String |
| prescriptonID | String |
| prescriptionDetails | String |

Figure Inbound eRx – VistAErxTransmission Flow



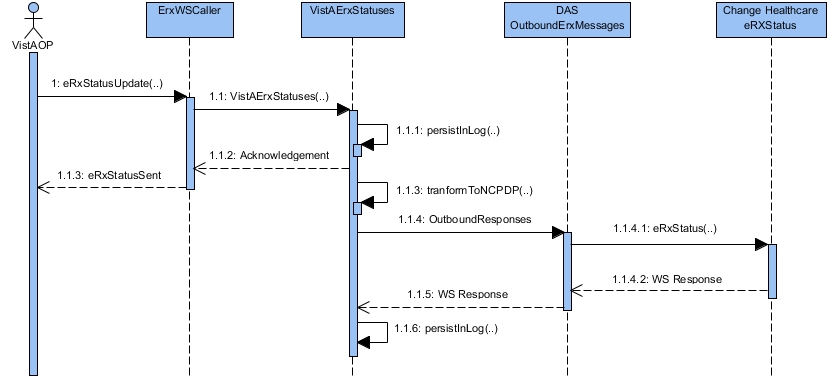
**VistAErxStatuses**

1. **Service Description**

This service is called by VistA OP after a prescription has either been dispensed, updated, cancelled or it causes an error while processing in VistA OP. It sends the data to Change Healthcare and is implemented as a RESTful web service. It is a processing service. Figure 25 shows the sequence diagram for this service.

1. **Steps**
2. It is called by the VistA OP passing the data for a particular prescription after the processing from VistA is finished and status of that particular prescription needs to be sent outbound to the provider.
3. It stores the data into eRX\_Log table.
4. It translates the data into NCPDP format xml.
5. It transmits the data to Change Healthcare via DAS OutboundErxMessages and gets a response back.
6. It returns success in the response; if all processing is successful otherwise it returns an error.

Figure Inbound eRx – eRx Status Transmission Flow



**ClinicalExchangePharmacyInformation**

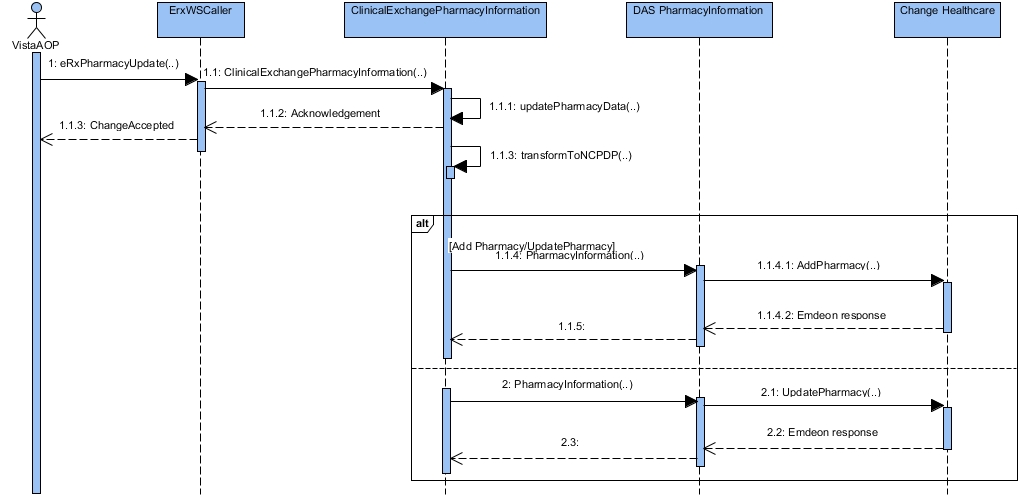
* 1. **Service Description**

This service can be called by VistA OP and is implemented as a RESTful web service. It is a processing service. It is used to send updates to Change Healthcare regarding updating a pharmacy in the Change Healthcare system. Figure 26 shows the sequence diagram for this service.

* 1. **Steps**
     1. It gets the pharmacy information i.e. pharmacyName, pharmacyID, address, city, state, zip code, telephone to be update/added from VistA OP.
     2. It updates the pharmacy information in the VA Pharmacy table in eRx.
     3. It builds the message from the data in NCPDP format and transmits to Change Healthcare via DAS PharmacyInformation and gets a response back.
     4. It returns success in the response, if all processing is successful, otherwise it returns an error.

| **Parameter** | **Type** |
| --- | --- |
| pharmacyName | String |
| pharmacyID | String |
| address | String |
| city | String |
| state | String |
| zipcode | String |
| telephone | String |

Figure : Inbound eRx – Pharmacy Information Update Transmission Flow



### VistA OP Components

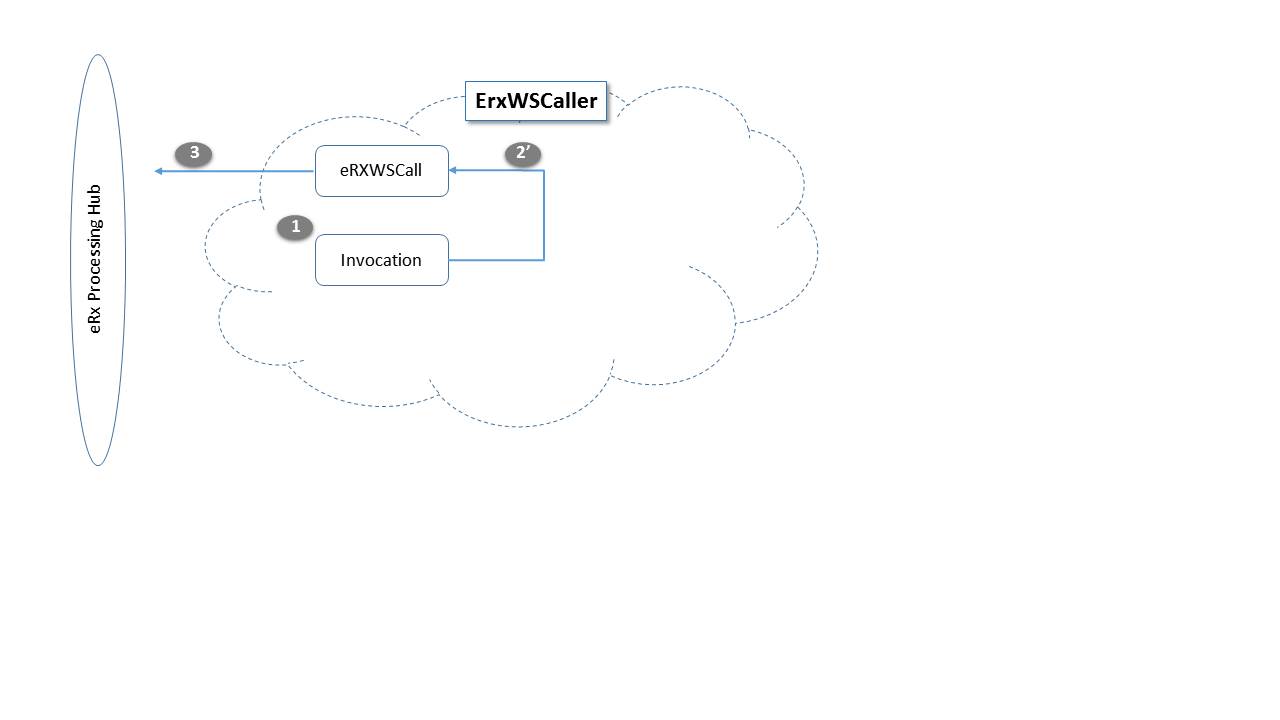
**ErxWSCaller**

1. **Service Description**

It is a MUMPs routine for calling web services in eRx e.g., for updating pharmacy information in Change Healthcare and for updating the status in eRx middle tier during the processing of prescriptions in VistA OP. Upon status change in the eRx Holding Queue, the prescription’s status is changed accordingly in the transaction/processing table in the eRx Processing Hub. It is used as wrapper for other routines to call the VistAErxStatuses web service in eRx Processing Hub. Figure 27 shows the flow for this service.

1. **Steps**
2. Call the VistAErxStatuses – the web service passing in the prescription details
3. Update Status in the eRx Processing Hub – the status of the prescription (C\_TransactionStatus) is updated to status = [the specific status – e.g., fulfilled, rejected, etc.]; no additional services are invoked

**Figure 27: ErxWSCaller**



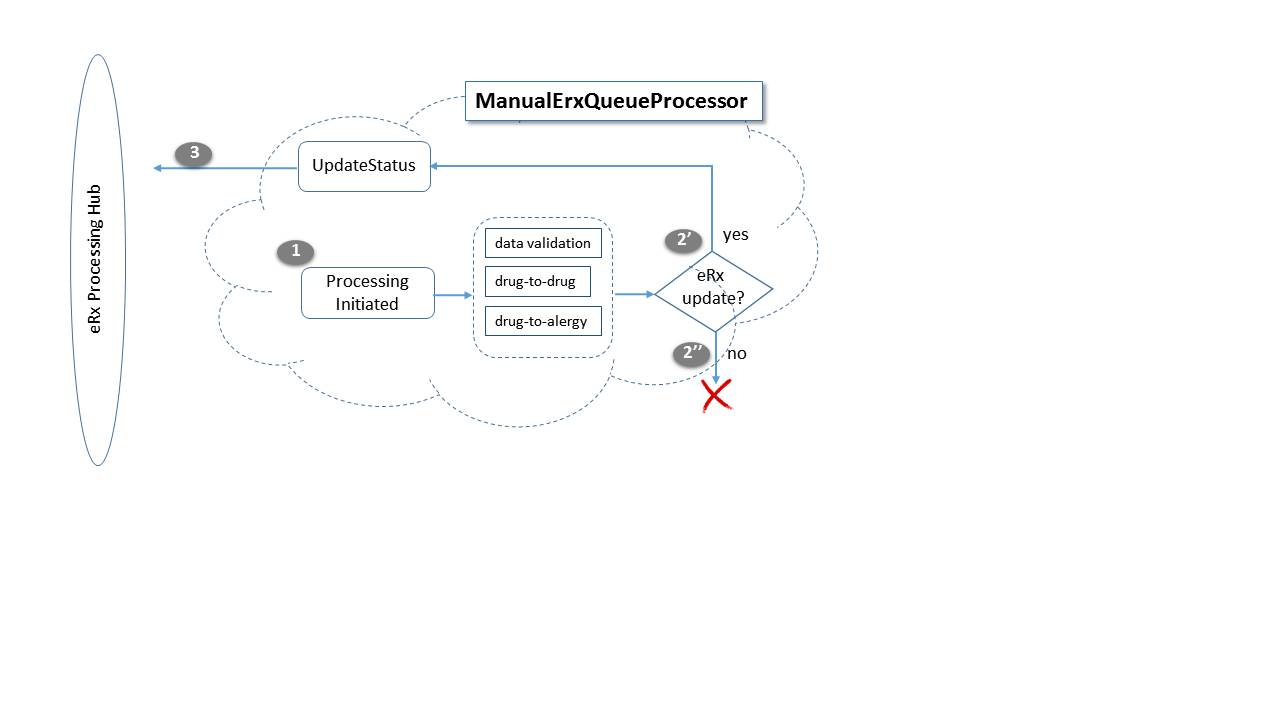
**ManualErxQueueProcessor**

1. **Service Description**

It is a MUMPs routine triggered from the manual validation step for initiating the existing VistA OP processing logic - prescription validation, approval and fulfillment processing; sub-routines include patient/provider/drug/data validation, drug-to-drug check, drug-to-allergy check and fulfillment; eRx Processing Hub status update is invoked, as well. Figure 28 shows the flow for this service.

1. **Steps**
2. Process Prescription Data – upon manual validation, prescriptions are processed in VistA OP; the main functions invoked are validation, approval and fulfillment processing; sub-routines include patient/ provider/ drug/ data validation, drug-to-drug check, drug-to-allergy check, and fulfillment
3. Update Status in the eRx Processing Hub – the status of the prescription (C\_TransactionStatus) is updated to status = [the specific status – e.g., fulfilled, rejected, etc.]; no additional services are invoked

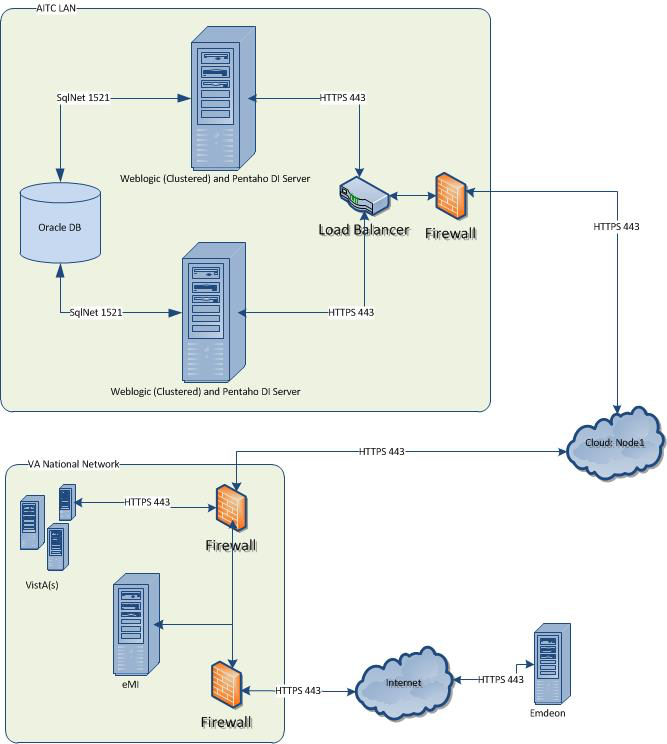
Figure : eRx Processing Hub (Manual)



## Network Architecture

The communications architecture for eRx will use a combination of wide area networks (WAN) coupled with Local Area Networks (LAN) as depicted in Figure 29. The LANs will use a star topology with transportation communication protocol and internet protocol (TCP/IP). The LAN will use one gigabit or greater switches where required. All nodes within the confines of the Austin Information Technology Center (AITC) will be connected to the AITC LAN. A firewall is used to provide security and connect each LAN to the cloud. The requirements for each firewall are defined by the system administrator for the respective LANs. The cloud is used to connect the two LANs resulting in a WAN. Access to the eRx is accomplished via a standalone laptop or workstation connected to the cloud using the appropriate browser.

**Figure 29: Network Architecture**



## Service Oriented Architecture / ESS

Use of Enterprise Shared Services is prevalent in the design when noting the use of DAS, MVI, HDR and the Enrollment web services. Service-Oriented Architecture (SOA) is also implemented with a design consideration being the potential sharing of Inbound eRx services in the Enterprise. Web services are designed, when practical, to be of potential use to other systems in the Enterprise

## Enterprise Architecture

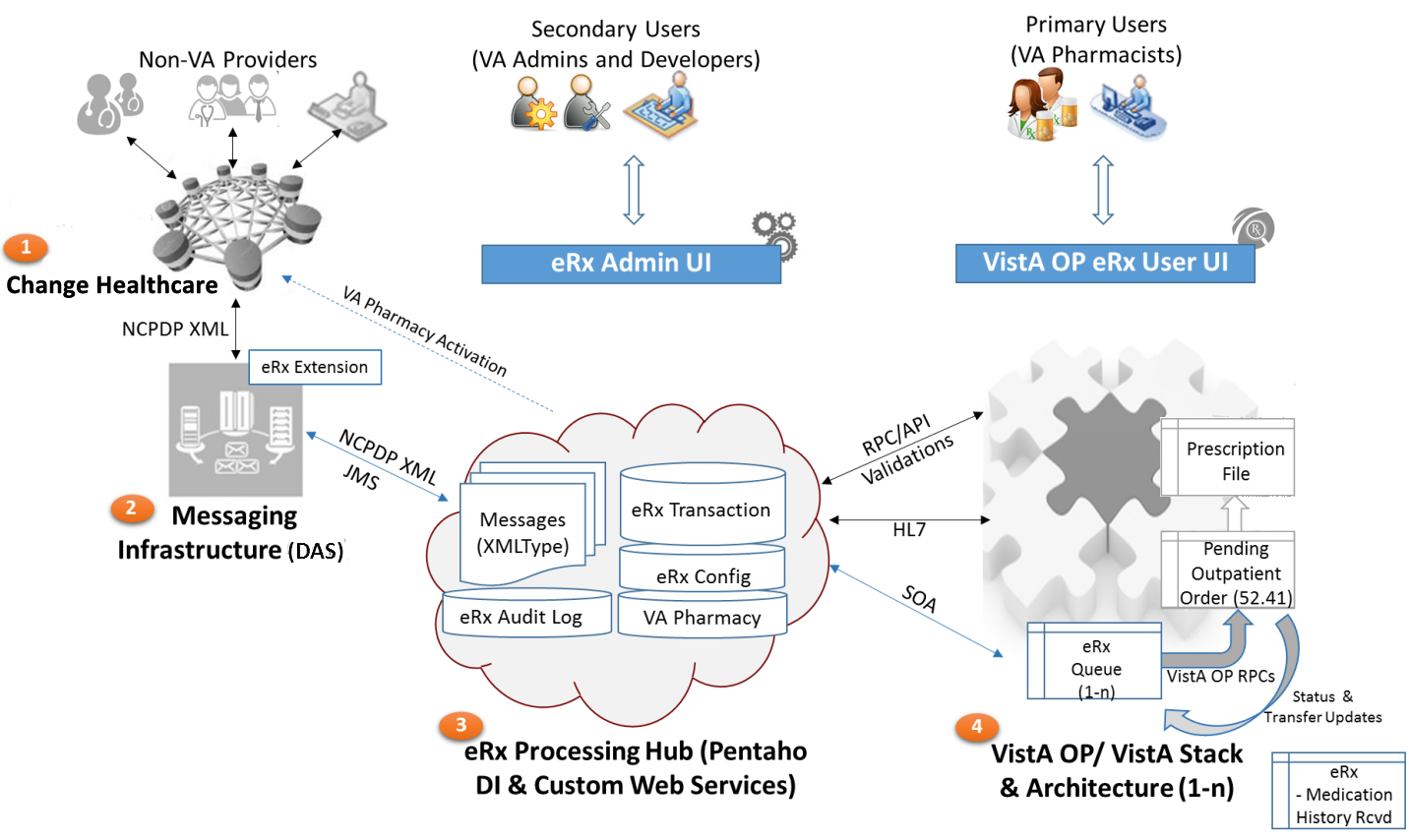
Inbound eRx will target Java SE 8 and WebLogic 12c, Pentaho Data Integration and Oracle Database 11g. The major components planned include Java, Spring, and Hibernate, along with various other VA Technical Reference Model (TRM) approved libraries. A server software list with links to the TRM approvals is available in [Section 6](#_Detailed_Design).

Inbound eRx will follow the recent ProPath Software Assurance (SwA) process of statically scanning the application code using HP Fortify SCA to uncover potential security and privacy issues. In addition to flagging potential security vulnerabilities, Fortify has the ability to recognized potential “leaks” of PII and PHI in application source code. One example being the potential logging of PHI and PII to unsecured application log files.

# Data Design

The Inbound eRx Architecture is composed of four (4) processing tiers Figure 30: Change Healthcare, Messaging Infrastructure (DAS), eRx Processing Hub (Pentaho DI) and VistA OP/ VistA Stack.

Figure : Conceptual Data Flow



*Change Healthcare* tier is external to the Inbound eRx architecture and is not required to be covered in this document. The *Messaging Infrastructure (DAS)* tier is middleware software with no data storage other than message buffering. The focus of the design is the data model of the remaining two (2) tiers – *eRx Processing Hub (Pentaho DI)* and *VistA OP*.

## DBMS Files

## DBMS Files – eRx Processing Hub

This section depicts the database design for the *eRx Processing Hub* and discusses the physical database organization, the physical data structures and files, the use of indexing as it relates to performance of the database, querying and accessing of the data, and the estimated database size and update frequency.

## Physical Database Organization

A database is a collection of tables. These tables, however, must ultimately be deployed on storage devices and server technology consistent with Oracle 11g - the Relational Database Management System (RDBMS) chosen for deployment.

Oracle RDBMS technology maintains a clear separation between the database (viewed as a set of tables) and the actual physical structures that implement those tables. The following sections outline some of the considerations that eventually drive the low-level physical design for deployment of the database portion of the eRx Processing Hub, noting that these implementation considerations are properly placed in the purview of a DBA. The DBA also has the responsibility to tune the database to improve performance after actual transactional and data-volume metrics are analyzed.

## Physical Data Structures and Files

While the set of physical tables presented in Section 5 of the document represent a relational database design, those models do not constitute a complete physical design of the database. This representation can be used to generate the basic American National Standards Institute (ANSI) SQL Data Definition Language (DDL) that would create the tables, keys, and constraints. That basic DDL would not include specific instructions about actual physical storage disks and servers. For the RDBMS selected for deployment (Oracle 11g), detailed design work is needed in order for precise physical representations to be included in the DDL code base. That is, low-level physical representation is the one area of RDBMS technology that is not vendor neutral.

Using Oracle 11g, some of these low-level design decisions would include:

* Tablespace organization
* Physical file organization

Oracle allows a DBA to organize tables, as well as constructs such as indexes, rollback segments, and temporary work areas, into a set of tablespaces. A tablespace is a parameterized set of one or more physical files in which database tables and other database constructs are stored. One tablespace, called SYSTEM always exists and an Inbound eRx tablespace will be needed.

Design considerations for selecting the set of tablespaces to create and table parameters include:

* The level or degree of stability or volatility of the data for the tables within the tablespace;
* The transaction volume expected on the tables;
* The projected growth rate of the data in the tables;
* The degree of concurrency with which multiple users might access the tables.

General design principles suggest that it is desirable to keep like “things” together and different “things” apart to minimize contention in an Optimal Flexible Architecture (OFA), which is a standard for Oracle database instances. However, in order to make sound decisions with respect to OFA, detailed sizing analysis work must be accomplished. The following high level analysis has been performed:

* VA has estimated that ~1.4M new eRx prescriptions and ~2.6M refills from external providers will be processed annually by Inbound eRx (total of 4M eRx transactions);
* For example, newRx message contains 238 and refill has 395 nodes; we are estimated average message size is 100KB (including response messages);
* Total raw message data size is estimated to be (100 KB \*4M)/106 = 400GB;
* We estimate Oracle 11g compression will reduce the message size 3x (i.e., 67% size reduction) – total annual growth of estimated 132GB for storing eRx raw message data;
* The eRx Processing Hub metadata size is a linear function of the raw data message size; we anticipate it to be 5-10% - expected metadata annual growth of ~6.6 – 13.2GB;
* Based on the above assessment, it is estimated that the annual eRx Processing Hub data growth will not exceed 145GB; higher compression methods (e.g., OLTP, warehouse and archive) can reduce the total annual data further.

In Oracle 11g, each tablespace is a collection of one or more files. Design decisions must be made about how many files are needed and where they are to be located. A file size of 30GB may be appropriate for Inbound eRx. Furthermore, tablespace encryption of AES256 (or as per the VA security policy) is anticipated to be required to protect PHI/PII data.

Along with the files required for the tablespaces, file space is needed to support redo logs, archive logs, and other database artifacts that allow the RDBMS optimally. Furthermore, the DBA must also consider the nature of the physical hardware disk and server technology. As the differing hardware configurations at different deployment levels are defined, the physical database design can be finalized.

### NCPDP XML Data Analysis

Inbound eRx is primarily processing NCPDP XML messages received from and send to Change Healthcare. XML is being used in a variety of ways – e.g., sometimes XML is constructed from relational data sources, so it is relatively structured, sometimes it is used in the Extract, Transform, Load (ETL) scenario, which is also very structured, sometimes it is used for storing free-form documents that do not have a pre-defined structure. This diversity in XML data leads to significantly different retrieval patterns. The data-centric processes usually have a fixed set of queries, whereas the document-centric users issue more ad-hoc queries.

Since XML usage falls in a broad spectrum, there is no one-size-fits-all storage to offer the best performance and flexibility. Oracle 11g offers different storage models for handling XML data via XMLType – primarily Binary and Object Relational. XMLType is an abstract data type that provides different storage and indexing models to best fit the eRx data and use cases. As an abstract data type, the Inbound eRx applications and database queries gain in flexibility leveraging the same interface for all XMLType operations.

The VA Inbound eRx use case is data-centric and falls in the ETL scenario where XML is used as a staging area for producing relational values from XML, as well as generating XML from relational data. Furthermore, the NCPDP XML data is highly structured and conforms to an XML schema.

As such, the Inbound eRx XMLs are best stored utilizing XMLType Object Relational storage model (or Structured Storage). The data model of Inbound eRx is designed accordingly.

The structured storage is an entity-relationship decomposition of the XML. With Structured Storage, the XMLType data is stored relationally which results in significant performance advantages matching the performance of relational tables. It also provides relational-like schema evolution capability which will allow VA to accommodate changes as they occur in the future.

### Query Patterns

Another important consideration in designing the data model and indexing for Inbound eRx is the query patterns. The VA use case for eRx data is a multi-root hierarchy query pattern. The NCPDP messages are uniquely coded by the MessageID and (for most messages) the RelatedMessageID. It is anticipated that the search queries will be predominantly based on the MessageID and RelatedMessageID. The Object Relational storage model gives the VA the best performance as it performs relational-style lookups starting from any storage table to the parent/child tables.

### Indexing

One of the key topics in database design, especially with respect to SQL-based relational databases, is data integrity; however, database performance is an equally important topic especially for transactional systems such as Inbound eRx.

One of the most common techniques for increasing performance in RDBMS-based database systems is indexing. Assuming an Oracle 11g implementation, the Inbound eRx Processing Hub data model will use primarily B-tree indexing with Structured Storage. This gives VA the best performance for data-centric transactions as the metadata (i.e. tags) is pulled out into column level, and hence queries can do a metadata lookup, which is extremely fast. Furthermore, Oracle supports XMLIndex and other types of indexes with Structured Storage which will be considered as performance tuning tools when the queries are known ahead of time and the list of Xpaths queried is known.

### DBMS Files – VistA OP

This section contains special data design considerations for the *VistA OP eRx Holding Queue*. These considerations are specific to the VistA OP Intersystem’s Cache database.

The VistA’s Cache database is a key-value database engine optimized for high-throughput transaction processing. As such it is in the class of "schema-less", "schema-free," or NoSQL databases. Internally, it stores data in multidimensional hierarchical sparse arrays (also known as key-value nodes, sub-trees, or associative memory). Each array may have up to a predefined number of subscripts, or dimensions. A scalar can be thought of as an array element with zero subscripts. Nodes with varying numbers of subscripts (including one node with no subscripts) can freely co-exist in the same array.

Inbound eRx will create a new eRx Holding Queue file to hold the staging data of auto-validated eRx transactions. Detailed data sizing task is yet to be completed, but the following analysis has been performed:

* The eRx Holding Queue in VistA OP will need to house similar size data as the Messages table in the eRx Processing Hub;
* This is data that is otherwise entered manually in VistA when the paper eRx is processed as per the current process;
* We anticipate the annual data growth to be in-line with the current VistA OP annual growth projections for the paper-based eRx processing.
* Furthermore, we anticipate that performance tuning will be performed with similar considerations as the Pending Outpatient Orders (52.41).

## Non-DBMS Files

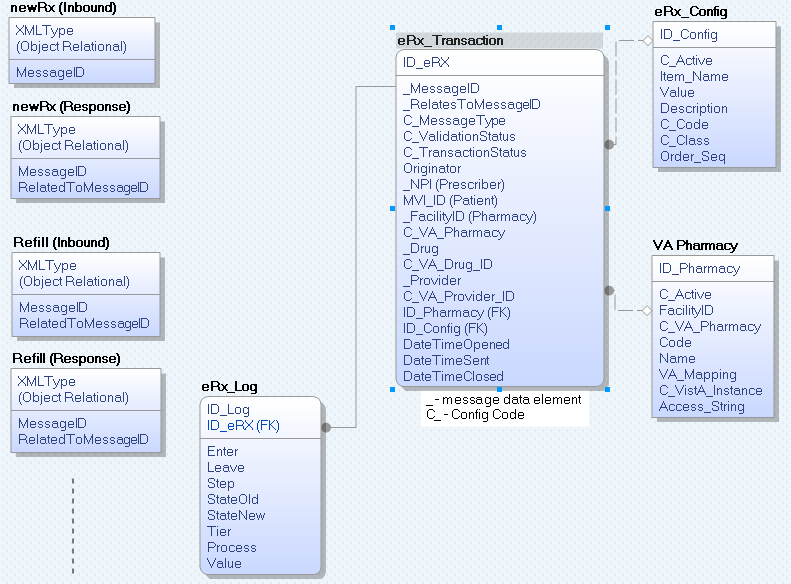
All data within Inbound eRx is saved in the eRx Processing Hub and VistA OP data models, as described in Sections 5.1 and 5.2.

## Data View

### eRx Processing Hub Tier

Figure 31 depicts the Entity Relationship Diagram (ERD) of the eRx Processing Hub Data Model.

Figure : eRx Processing Hub ERD



The ERD diagram depicts the following tables:

* eRx\_Transaction
* eRx\_Config
* VA\_Pharmacy
* eRx\_Log
* XML Message tables – newRx & Response newRx, Refill & Response Refill, etc. – one pair for each supported by Change Healthcare message type

**eRx\_Transaction**

The eRx\_Transaction stores all the necessary data elements to receive, auto-validate and coordinate the processing of the incoming eRx messages. Each transaction has an auto-generated ID\_eRx primary key, as well as is coded with the values of \_MessageID and \_RelatedMessageID (if exists) message element. This enables an XQuery to be constructed with the respective XML Message tables. The table below depicts the fields of the eRx\_Transaction table:

Table : eRx\_Transaction Table Fields

| **#** | **Name** | **Type** | **Sample Values** | **Notes** |
| --- | --- | --- | --- | --- |
| 1 | ID\_eRx | Number(10) | 1001 | Auto generated |
| 2 | \_MessageID | Char(255) | abc2121212 | MessageID from the incoming eRx message |
| 3 | \_RelatedMessageID | Char(255) | xyz2121200 | RelatedMessageID from the incoming eRx message |
| 4 | C\_Message\_Type | Number(6) | 10 | Describes the type of message (e.g., newRx, Refill, Response Refill); reference values are depicted in the eRx\_Config table |
| 5 | C\_ValidationStatus | Number(6) | 8 | Current status of the auto-validations (e.g., pending, validated, non-validated, error); reference values are depicted in the eRx\_Config table |
| 6 | C\_TransactionStatus | Number(6) | 4 | Current status of the transaction (e.g., pending, sent, processed, error, failed); reference values are depicted in the eRx\_Config table |
| 7 | \_NPI | Number(10) | 1122334455 | Prescriber NPI number; from the incoming eRx message |
| 8 | MVI\_ID | Number(10) | 1345 | MVI ID of the VA patient |
| 9 | \_FacilityID | Char(255) | C7899 | Pharmacy ID; from the incoming eRx message |
| 10 | C\_VA\_Pharmacy | Number(10) | 10 | VA’s internal PharmacyID based on the \_FacilityID |
| 11 | ID\_Pharmacy | Number(10) | 14 | VA Pharmacy unique identifier |
| 12 | ID\_Config | Number(10) | 78 | VA Facility access/config string |
| 13 | \_ProductCode | Char(255) | 123456 | The drug and drug qualifier codes |
| 14 | C\_VA\_Product\_ID | Number(10) | 998877 | VA’s product code ID |
| 15 | \_Strength | Char(255) | C25301 | The prescribed strength |
| 16 | C\_VA\_Strength\_ID | Number(10) | 4567 | VA’s strength code ID |
| 17 | \_Potency | Char(255) | 99 | The prescribed potency of the drug |
| 18 | C\_VA\_Potency\_ID | Number(10) | 10 | VA’s potency code ID |
| 19 | \_DoseQuantity | Char(255) | 1 | The prescribed quantity |
| 20 | C\_VA\_Qty\_ID | Number(10) | 1 | VA’s quantity code ID |
| 21 | DateTimeOpened | datetime |  | When the transaction was created |
| 22 | DateTimeSent | datetime |  | When the transaction was sent to VistA OP |
| 23 | DateTimeClosed | datetime |  | When the transaction was completed in VistA OP |

**Note:** Some data elements are represented in an abbreviated form and multiple fields may be defined in the database.

**eRx\_Config**

The eRx\_Transaction stores all the necessary configuration values for the eRx Processing Hub. Each transaction has an auto-generated ID\_Config primary key. The table below depicts the fields of the eRx\_Config table:

Table : eRx\_Config Table Fields

| **#** | **Name** | **Type** | **Sample Values** | **Notes** |
| --- | --- | --- | --- | --- |
| 1 | ID\_Config | Number(10) | 55 | Auto generated |
| 2 | C\_Active | Number(1) | 1 | Indicates that the value is active (1) or inactive (0) |
| 3 | Item\_Name | varchar(20) | C\_Message\_Type | Multiple rows exist |
| 4 | Value | varchar (255) | newRx | Value |
| 5 | Description | varchar (255) | 8 | Description |
| 6 | C\_Code | Number(10) | 4 | The numeric code |
| 7 | C\_Class | Number(6) | 3 [=message] |  |
| 8 | Order\_Seq | Number(6) | 10 | Display sequence number |

Note: As the requirements, user stories and technical designs are refined, additional data elements may need to be added.

The eRx\_Config table depicts the configuration values for the item\_names - C\_Message\_Type, C\_ValidationStatus, C\_TransactionStatus, C\_Class, Step, StateOld, StateNew, Tier, Process, C\_VistA\_Instance, communication strings, and others required for the operations of the eRx Processing Hub and VistA OP.

The following lists the values for the C\_ValidationStatus – depicting the auto-validation status of a prescription processed in the eRx Processing Hub:

Table : eRx Processing Hub C\_Validation Status Values

| **#** | **Item\_Name** | **C\_Code** | **C\_Class** | **Value** | **Description** |
| --- | --- | --- | --- | --- | --- |
| 1 | C\_ValidationStatus | 0 | 0 | Not-started | Auto-validation hasn’t started yet |
| 2 | C\_ValidationStatus | 1 | 10 | PatientStarted | Patient-related data validation in-progress |
| 3 | C\_ValidationStatus | 2 | 10 | ProviderStarted | Provider-related data validation in-progress |
| 4 | C\_ValidationStatus | 4 | 10 | DrugStarted | Drug-related data validation in-progress |
| 5 | C\_ValidationStatus | 8 |  | [reserved] | Reserved for future use |
| 6 | C\_ValidationStatus | 16 |  | [reserved] | Reserved for future use |
| 7 | C\_ValidationStatus | 32 |  | [reserved] | Reserved for future use |
| 8 | C\_ValidationStatus | 64 |  | [reserved] | Reserved for future use |
| 9 | C\_ValidationStatus | 128 | 12 | PatientNotValidated | Patient-data is not validated |
| 10 | C\_ValidationStatus | 256 | 12 | ProviderNotValidated | Provider-data is not validated |
| 11 | C\_ValidationStatus | 512 | 12 | DrugNotValidated | Drug-data is not validated |
| 12 | C\_ValidationStatus | 1024 | 14 | PatientRejected | Patient is validated, but is ineligible for benefits |
| 13 | C\_ValidationStatus | 2048 |  | [reserved] | Reserved for future use |
| 14 | C\_ValidationStatus | 4096 |  | [reserved] | Reserved for future use |
| 15 | C\_ValidationStatus | 8192 |  | [reserved] | Reserved for future use |
| 16 | C\_ValidationStatus | 16384 | 16 | PatientValidated | Patient data is validated |
| 17 | C\_ValidationStatus | 32768 | 16 | ProviderValidated | Provider data is validated |
| 18 | C\_ValidationStatus | 65536 | 16 | DrugValidated | Provider data is validated |

As an illustration, C\_ValidationStatus = 7 (1+2+4) means that patient (#2), provider (#3) and drug (#4) data auto-validation is in progress, and C\_ValidationStatus = 17152 (256+512+16384) means that provider data is not validated (#10), drug data is not validated (#11) and patient data is validated (#16).

The following lists the values for the C\_TransactionStatus – depicting the processing status of a prescription processed in the VistA OP:

Table : C\_TransactionStatus in VistA OP

| **#** | **Item\_Name** | **C\_Code** | **C\_Class** | **Value** | **Description** |
| --- | --- | --- | --- | --- | --- |
| 1 | C\_TransactionStatus | 0 | 0 | Not-started | Auto-validation hasn’t started yet |
| 2 | C\_TransactionStatus | 1 | 10 | ValidationStarted | Auto-validation in progress |
| 3 | C\_TransactionStatus | 2 | 10 | ValidationCompleted | Auto-validation completed |
| 4 | C\_TransactionStatus | 4 |  | [reserved] | Reserved for future use |
| 5 | C\_TransactionStatus | 8 |  | [reserved] | Reserved for future use |
| 6 | C\_TransactionStatus | 16 | 10 | TransmittedOP | Transmitted to OP eRx Holding Queue |
| 7 | C\_TransactionStatus | 32 |  | [reserved] | Reserved for future use |
| 8 | C\_TransactionStatus | 64 | 10 | TransmittedOPRetry | OP Transmission retry required |
| 9 | C\_TransactionStatus | 128 | 10 | ProcessingOPStarted | OP processing in-progress |
| 10 | C\_TransactionStatus | 256 |  | [reserved] | Reserved for future use |
| 11 | C\_TransactionStatus | 512 |  | [reserved] | Reserved for future use |
| 12 | C\_TransactionStatus | 1024 | 14 | ProcessingOPFailed | OP processing failed permanently |
| 13 | C\_TransactionStatus | 2048 |  | [reserved] | Reserved for future use |
| 14 | C\_TransactionStatus | 4096 | 12 | ProcessingOPActionPending | OP processing failed and action is needed to restart |
| 15 | C\_TransactionStatus | 8192 |  | [reserved] | Reserved for future use |
| 16 | C\_TransactionStatus | 16384 | 16 | ProcessingOPCompleted | OP processing completed successfully |
| 17 | C\_TransactionStatus | 32768 |  | [reserved] | Reserved for future use |
| 18 | C\_TransactionStatus | 65536 |  | [reserved] | Reserved for future use |

As an illustration, C\_TransactionStatus = 5120 (1024 + 4096) means that OP processing failed (#12) and action is pending to restart (#14).

The following lists the values for the C\_Class – depicting the internal to the eRx Processing Hub designators:

Table : eRx Processing Hub Designators C\_Class Values

| **#** | **Item\_Name** | **C\_Code** | **C\_Class** | **Value** | **Description** |
| --- | --- | --- | --- | --- | --- |
| 1 | C\_Class | 0 | 0 | Configuration | configuration |
| 2 | C\_Class | 1 | 0 | State | processing state |
| 3 | C\_Class | 2 | 0 | [reserved] | Reserved for future use |
| 4 | C\_Class | 3 | 0 | Message | Message type |
| 5 | C\_Class | 4 | 0 | Log | Log and audit |
| 6 | C\_Class | 5 | 0 | Tier | Processing Tier |
| 7 | C\_Class | 6 | 0 | [reserved] | Reserved for future use |
| 8 | C\_Class | 7 | 0 | Process | Process identifier |
| 9 | C\_Class | 8 | 0 | [reserved] | Reserved for future use |
| 10 | C\_Class | 9 | 0 | [reserved] | Reserved for future use |
| 11 | C\_Class | 10 | 0 | Initial | Initial stage of processing |
| 12 | C\_Class | 11 | 0 | [reserved] | Reserved for future use |
| 13 | C\_Class | 12 | 0 | Unsuccess | Processing resulted in unsuccessful outcome |
| 14 | C\_Class | 13 | 0 | [reserved] | Reserved for future use |
| 15 | C\_Class | 14 | 0 | Violation | Processing resulted in violation outcome |
| 16 | C\_Class | 15 | 0 | [reserved] | Reserved for future use |
| 17 | C\_Class | 16 | 0 | Success | Processing resulted in successful outcome |
| 18 | C\_Class | 17 | 0 | [reserved] | Reserved for future use |

**VA\_Pharmacy**

The VA\_Pharmacy stores VA Pharmacy data and the corresponding VistA instance. Each transaction has an auto-generated ID\_Pharmacy primary key. The following depicts the fields of the VA\_Pharmacy table:

Table : VA\_Pharmacy Table Fields

| **#** | **Name** | **Type** | **Sample Values** | **Notes** |
| --- | --- | --- | --- | --- |
| 1 | ID\_Pharmacy | Number(10) | 55 | Auto generated |
| 2 | C\_Active | Number(1) | 1 | Indicates that the value is active (1) or inactive (0) |
| 3 | \_FacilityID | Char(255) | C7899 | From the eRx message |
| 4 | C\_VA\_Prarmacy | Number(10) | 14 | VA Internal Pharmacy Code |
| 5 | Code | varchar(20) | DC501 | Extended pharmacy code |
| 6 | Name | varchar(20) | Facility501 | Pharmacy Name |
| 7 | VA\_Mapping | varchar(20) | HQ | Mapping to a VA Pharmacy Group |
| 8 | C\_VistA\_Instance | Number(6) | 11 | VistA OP instance |
| 9 | Access\_String | varchar(1024) | https://VistAregion1.DNS :6789 | String to access the VistA instance |

Note: As the requirements, user stories and technical designs are refined, additional data elements may need to be added.

**eRx\_Log**

The eRx\_Log stores audit log data and status changes for each eRx transaction. Each transaction has an auto-generated ID\_Log primary key and ID\_eRx foreign key. The following below depicts the fields of the eRx\_Log table:

Table : eRx\_Log Table Fields

| **#** | **Name** | **Type** | **Sample Values** | **Notes** |
| --- | --- | --- | --- | --- |
| 1 | ID\_Log | Number(10) | 345111 | Auto generated |
| 2 | ID\_eRx | Number(10) | 1001 | Indicates that the value is active (1) or inactive (0) |
| 3 | Enter | Datetime |  | Start datetime of the transaction change |
| 4 | Leave | Datetime |  | End datetime of the transaction change |
| 5 | Step | varchar(20) | C\_Validation\_Type | Captures the process step |
| 6 | StateOld | varchar(20) | 2 (=pending) | Captures the old state |
| 7 | StateNew | varchar(20) | 4 (=validated) | Captures the new state |
| 8 | Tier | varchar(20) | 2 (=eRx) | Captures the tier that reported the status change |
| 9 | Process | varchar(20) | Auto validation | Captures the process |
| 10 | Value | varchar(255) | Validated | The value that changed |

Note: As the requirements, user stories and technical designs are refined, additional data elements may need to be added.

**XML Message Tables**

XML Message tables store the NCPDP XML messages as XMLType Object Relational types. Each (Change Healthcare) message is stored in a separate table – i.e., newRx, Response newRx, Refill, Response Refill, etc. are all separate XMLType tables.

Although for production purposes a more comprehensive statement is anticipated, the pseudo code for creating an XMLType table can be depicted as follows:

SQL> **CREATE** **TABLE** newRx **OF** XMLType

XMLTYPE store **AS** OBJECT RELATIONAL

XMLSCHEMA "http://localhost/xsd/newRx\_EMD\_NCPDP\_SCRIPT\_10\_6.xsd"

ELEMENT "MessageID";

**TABLE** created.

SQL> **SELECT** dbms\_metadata.get\_ddl('TABLE','newRx')

2 **FROM** dual;

### VistA OP eRx Holding Queue

This section depicts the structure of the VistA OP eRx Holding Queue file.

The following data elements are expected to be required in the VistA OP eRx Holding Queue:

eRx Queue File (#tbd)

.01 - NCPDP or Change Healthcare message id/number

.02 - Related or Parent message ID

.03 - Message Date/Time

.04 - Patient

.05 - Hospital Location

.06 - Pharmacy (pointer to the pharmacy system file?)

.07 - Message Type (Newrx, rxchangerequest, cancelrx, rxHistoryrequest)

.08 - External Order Number

  1 - eRx Order status (pending/approved/processed etc)

1.1 - Provider Status (auto validation)

1.11 - Provider status (Manual validation)

1.2 - Drug Status (auto validation)

1.21 - Drug Status (manual validation)

1.3 - Patient Status (auto validation)

1.31 - Patient status (manual Validation)

  2 - External Provider Information

.01 - Name

.02 - Address

.03 - Telephone

.04 - DEA #

.05 - Additional info 1

.06 - Additional info 2

2.1 - VA (matched) provider (pointer)

  3 - Order Type (similar to order type in 52.41)

  4 - Quantity Timing sub-file (similar to 52.41)

  5 - Effective Date

  6 - Nature of Order (This may not be available in the same format)

  7 - Provider Comments

  8 - External Drug/Supply

  9 - Matched Drug/Supply (VA)

10 - QTY

11 - # of Refills

12 - LOGIN Date

13 - Medication Route

14 - Service Connected

15 - Order Checks

16 - Pickup Routing

17 - Rx refill request (if needed)

18 - Reason Order Created

19 - Previous Order # (may not be applicable for us)

20 - Comments

21 - Dispensing Instructions

22 - Priority

23 - Possible Filldate

30 - Days Supply

31 - Flag

32 - Patient Instructions

33 - Patient Instructions Flag

34 - Expanded Patient Instructions

35 - Signature Status (maybe not need this, or will it get signed through CPRS)

36 - Directions for use (if applicable)

37 - Diagnosis

101 - eRx Allergies (multiple)

      .01 - Allergy Name

      .02 - Other info 1

      .03 - Other info 2

201 - Errors

      .01 - Error ID

      .02 - Error type (Drug/Provider/patient/allergy etc.)

      .03 - Status

      .04 - Resolved by

      .05 - Date/time resolved

**Data Transformation**

As the requirements and technical designs are finalized, detailed field-by-field mapping and transformation rules will be defined for each Change Healthcare message and the VistA OP eRx Holding Queue. This mapping and transformation is not practical to be completed at this stage of the project, but will be critical component of the Inbound eRx implementation.

**eRx - Medication History Rcvd**

The Medication History requirements, both VA requests from external provider and external provider requesting from VA, are currently in the process of review of privacy considerations and finalization. Once approved, this data file will be finalized.

# Detailed Design

## Hardware Detailed Design

Table : Development/SQA Detailed VM Requirements

| **VM** | **RAM (GB)** | **Space\* (GB)** | **CPUs** | **OS** | **VM Description/Use/DNS Required** |
| --- | --- | --- | --- | --- | --- |
| 1 | 16 | 300 | 4 | RHEL 6.X | DEV 1 DB Server running Oracle |
| 2 | 16 | 300 | 4 | RHEL 6.X | DEV 2 DB Server running Oracle |
| 3 | 16 | 300 | 4 | RHEL 6.X | SQA 1 DB Server running Oracle |
| 4 | 16 | 300 | 4 | RHEL 6.X | SQA 2 DB Server running Oracle |
| 5 | 16 | 300 | 4 | RHEL 6.X | DEV1 AP Server running Apache/WebLogic |
| 6 | 16 | 300 | 4 | RHEL 6.X | DEV 2 AP Server running Apache/WebLogic |
| 7 | 16 | 300 | 4 | REHL 6.X | SQA 1 AP Server running Apache/WebLogic |
| 8 | 16 | 300 | 4 | REHL 6.X | SQA 2 AP Server running Apache/WebLogic |
| Total | 128 | 2400 | 32 | 8 |  |

Table : Staging Detailed VM Requirements

| **VM** | **RAM (GB)** | **Space\* (GB)** | **CPUs** | **OS** | **VM Description/Use/DNS Required** |
| --- | --- | --- | --- | --- | --- |
| 1 | 16 | 800 | 4 | RHEL 6.X | STAGING DB Server running Oracle |
| 2 | 16 | 300 | 4 | RHEL 6.X | STAGING Application Server running Apache/WebLogic |
| 3 | 16 | 300 | 4 | RHEL 6.X | STAGING Application Server running Apache/WebLogic |
| Total | 48 | 1400 | 16 | 3 |  |

Table : Pre-Production Detailed VM Requirements

| **VM** | **RAM (GB)** | **Space\* (GB)** | **CPUs** | **OS** | **VM Description/Use/DNS Required** |
| --- | --- | --- | --- | --- | --- |
| 1 | 16 | 1300 | 4 | RHEL 6.X | PRE-PRODUCTION DB Server running Oracle |
| 2 | 16 | 300 | 4 | RHEL 6.X | PRE-PRODUCTION Application Server running Apache/WebLogic |
| 3 | 16 | 300 | 4 | RHEL 6.X | PRE-PRODUCTION Application Server running Apache/WebLogic |
| Total | 48 | 1900 | 12 | 3 |  |

Table : Production Detailed VM Requirements

| **VM** | **RAM (GB)** | **Space\* (GB)** | **CPUs** | **OS** | **VM Description/Use/DNS Required** |
| --- | --- | --- | --- | --- | --- |
| 1 | 16 | 1300 | 4 | RHEL 6.X | PRODUCTION DB Server running Oracle |
| 2 | 16 | 300 | 4 | RHEL 6.X | PRODUCTION Application Server running Apache/WebLogic |
| 3 | 16 | 300 | 4 | RHEL 6.X | PRODUCTION Application Server running Apache/WebLogic |
| Total | 48 | 1900 | 12 | 3 |  |

Table : Server Software List

| **#** | **Product Name** | **Version** | **TRM Approved (Y/N)** | **Description/Use/TRM link** |
| --- | --- | --- | --- | --- |
| 1 | Apache HTTP Server | 2.2 | Y | http://www.DNS /TRM/ToolPage.asp?tid=5009# |
| 2 | WebLogic Server | 12.1 | Y | http://www.DNS /TRM/ToolPage.asp?tid=7# |
| 3 | Oracle Database | 11.2 | Y | http://www.DNS /TRM/ToolPage.asp?tid=9# |
| 4 | Pentaho Data Integration | 6.0 | Y | http://www.DNS /TRM/ToolPage.asp?tid=6653# |

## Software Detailed Design

Because the Inbound eRx system is at a preliminary design stage, some design details are not yet available and will be addressed in this document prior to final review. Additionally, it is expected that the information provided may need to be changed during later design stages.

### Conceptual Design

Please refer to [Section 3](#_Conceptual_Design) Conceptual Design.

#### Product Perspective

This subsection of the SDD should put the product into perspective with other related products. If the product is independent and completely self-contained, it should be stated here. If the SDD defines a product that is a component of a larger system, then this subsection should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software.

A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful.

Sections of the Requirements Specification Document (RSD) can be referenced in the subsections, if applicable.

##### User Interfaces

Please refer to [Section 4.2](#_System_Architecture) Software Architecture.

##### Hardware Interfaces

This subsection should specify the logical characteristics of each interface between the software product and the hardware components of the system. This includes configuration characteristics (for example, hardware platform or mainframe versus personal computer). It also covers matters such as what devices the system will support, how they will be supported, and protocols. Examples include scanners, pen driven devices, and radio frequency devices.

Recommendation: Create a block diagram showing the hardware interfaces.

##### Software Interfaces

This subsection should specify the use of other required software products (e.g., VA Kernel, VA FileMan, Windows NT); and interfaces with other applications or other systems such as commercial off-the-shelf (COTS) or national databases. Specify the application interfaces (e.g., the linkage between an accounts receivable system and a general ledger system and a COTS software package that will be interfaced using an existing interface). This section should provide the following information for each required software product:

* Name
* Version number
* Discussion of the purpose of the interfacing software as related to this software product
* Definition of the interface in terms of message content and format (e.g., Health Level Seven [HL7], electronic data interchange).

##### Communications Interfaces

This subsection should specify the various interfaces to communications such as local network protocols, e-mail, Transmission Control Protocol (TCP), modems.

Recommendation: Create a block diagram showing the communications interfaces.

##### Memory Constraints

This subsection should specify any applicable characteristics and limits on memory or partition size.

##### Special Operations

This subsection should specify the special operations required by the user such as backup, recovery, and archiving operations.

This section should also include any operations for external devices or COTS systems.

#### Product Features

This subsection should provide a summary of the major features of the software.

For example, an SDD for an accounting program might use this section to address customer account maintenance, customer statement, and invoice preparation without mentioning the vast amount of detail that each of those features requires.

Note: For clarity, remember these items when creating this section of the SDD:

* The features should be organized in a way that makes the list of features understandable to the customer or to anyone else reading the document for the first time.
* Textual or graphical methods can be used to show the different features and their relationships.
* Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables.

#### User Characteristics

This subsection should describe the general characteristics of the intended users of the product, including experience and technical expertise. It should not be used to state specific requirements but rather should provide the reasons why certain specific requirements are specified in the RSD.

#### Dependencies and Constraints

The following list details the dependencies and constraints assumed by the Inbound eRx project:

* The solution must comply with the appropriate business architecture.
* The application developers must refactor components into partitioned logical layers (i.e. presentation, business, logical, data access) with each layer containing functionality specifically related to that layer.
* The application layers must expose interface components that promote loose coupling between layers.
* The application developers must refactor components so that business logic and data management processes are fully decoupled from each other.
* The application user interfaces must follow enterprise common UI templates and style guidelines.
* The application must store data on enterprise servers and not on end-user devices or workstations.
* Inbound eRx development team must develop unit tests for application functions and publicly exposed methods.
* The application must implement procedures for communicating and resolving unhandled exceptions.
* The application shall scale horizontally and operate on a series of loosely coupled commodity platforms.
* The application must scale out based on the need.
* Application business logic must exist statelessly (i.e., session information is not stored within the business logic layer).
* Application user interfaces must comply with Section 508.
* The application must comply with VA Enterprise Architecture published data standards (HL7, NCPDP).
* Inbound eRx must identify and leverage authoritative information sources for data retrieval and manipulation.
* Information captured by the Application must syntactically and semantically harmonize with the VA Enterprise CDM.
* The application must operate optimally using information from the authoritative source or receive permission for caching data locally.
* The team must configure system/and server platforms used by the application using standard system images published in the current VA Release Architecture.
* The team must publish relational and object oriented databases utilized by the solution in the current VA Release Architecture.
* The solution design must operate on the standard OIT defined virtual environments.
* The team must base application production capacity requirements on workload analysis, simulated workload benchmark tests, or application performance models.
* The team must base application storage capacity requirements on detailed capacity analysis and/or models.
* The team must design the solution to operate within the current VA Local Area Network (LAN) and Wide Area Network (WAN) network configurations.
* The deployment environment must meet the performance and downtime monitoring requirements of the solution.
* The team and data center must develop and provision a disaster recovery plan.
* All critical infrastructure components (including data) must be located at multiple physical locations.
* The application backup and restore solution must meet data recovery requirements [Recovery Point Objectives (RPO) and Recovery Time Objectives (RTO)].
* The application UIs must exist as browser based UIs and roll and scroll in the VistA.
* The application must adhere to all applicable information security rules contained within the VA Policy Handbook (6500).
* If the application is deployed externally, it must follow all guidelines for using commercial partners.
* The application must establish secure access paths for accessing the application and application data.
* The solution must document specific reasons for all limited, external access to data, including the need to know along with security, privacy and other legal restrictions.
* The solution must implement appropriate controls that prevent unwarranted disclosure of sensitive, Personally Identifiable Information (PII), or Protected Health Information (PHI).
* The team must base all system interfaces (both external and internal) implemented by the solution on open standards such as SOAP, REST, JMS, MQ, HTTPS and standard message formats such as HL7and NCPDP.
* The solution must utilize the Enterprise IAM service for authentication and authorization.
* The solution must access available enterprise information through services.
* The VA TRM must identify all products and standards used by this solution as permissible for usage.

### Specific Requirements

#### Database Repository

The Database Repository section in the RSD can be referenced in this section.

If a logical database design is a part of the system, it should be listed here. Logical database design should specify the logical requirements for any information that is to be placed into a database. This may include:

* Types of information used by various functions
* Frequency of use
* Accessing capabilities
* Data entities and their relationships
* Integrity constraints
* Data retention requirements.

Recommendation: Create a block diagram showing the databases and where the data resides.

#### System Features

Describe the system features, functional requirements, sub-requirements, etc. which can be organized in an outline format that matches the RSD. Specific formatting and organization of the paragraphs (i.e., section numbering) is left to the discretion of the author and is dependent on the level of detail essential to fully describe the design. Some designs may only require two levels; others may require multiple levels. The information necessary to define the items or to specify modifications to the items affected by the functionality being designed should be provided in the appropriate design element tables. Where feasible, instead of duplicating the RSD, it can be referenced via a link, to avoid unnecessary duplication. The key goal is to provide traceability to requirements.

| EPICs | Requirements |
| --- | --- |
| **EPIC 4:** | Inbound ePrescriptions received from external providers must be expressed using nationally recognized reference and authoritative terminology standards |
| **4.1** | Change Healthcare validates external providers against NCPDP. The eRx Processing Hub will utilize NCPDP XML format for eRx validation |
| **4.2** | The eRx Processing Hub will validate incoming eRx against NCPDP XML format, it will perform auto validation on patient, provider and drug. The eRx Processing Hub will store the data in the eRx database tables as the eRx movies through the processing flow.  The eRx Processing Hub will transmit the eRx to VistA OP, where it will be store in the eRx Holding Queue. The pharmacist will perform manual validation. After all the checks have been passed, the eRx will be either Pending state - not yet dispensed or returned with appropriate message which will be sent back to the provider via eRx Processing Hub webservice. |
| **4.3** | The eRx Processing Hub will return the invalidate eRx to Change Healthcare via OutboundErxMessages proxy in DAS. |
| **4.4** | The eRx Processing Hub will return the invalidate eRx with failure reason in NCPDP format |
| **EPIC 5:** | Search for and identify the VA Pharmacy to send a Veteran patient’s ePrescription to (would be performed by an external provider/non-VA physician) |
| **5.1** | Change Healthcare will have the list of the VA pharmacies. Any addition or modification of pharmacy information will be initiated by the VistA OP via ErxWSCaller routine (see section 4.2.1). This routine will call ClinicalExchangePharmacyInformation webservice, which is hosted in the eRx middle tier. The pharmacy information will be then sent to Change Healthcare via PharmacyInformation proxy which is hosted at DAS |
| **5.2** | The eRx application will establish the VA Pharmacies standard naming convention format by coordinating with PBM |
| **5.3** | The location information of VA pharmacy will be sent to Change Healthcare via PharmacyInformation proxy in DAS |
| **EPIC 6:** | The system needs to be able to receive and hold (store) inbound eRxs before they are auto-checked |
| **6.1** | The incoming eRx will be held in a JMS queue hosted by DAS before they are picked up by the scheduled job in the eRx Processing Hub for the auto check process |
| **EPIC 7:** | The VA Pharmacy system must be able to “enable” or “disable” receipt of inbound ePrescriptions from external providers |
| **7.1** | The eRx middle tier will provide a UI to enable reception of inbound ePrescriptions from external providers |
| **7.2** | The eRx middle tier will provide a UI to disable reception of inbound ePrescriptions from external providers |
| **7.3** | The eRx Processing Hub will check if the VA pharmacy is enabled during the auto check process and if it’s disabled, it will send a message back to Change Healthcare |
| **7.4** | The eRx Processing Hub and the VistA OP will process any existing eRxs and will send its updates back to Change Healthcare, even if the pharmacy is disabled in the processing hub |
| **EPIC 8:** | All inbound ePrescription orders to VA Pharmacies for medication or expendable medical supplies must comply with NCPDP SCRIPT 10.6 standards |
| **8.1** | The eRx Processing Hub will validate all eRx against NCPDP SCRIPT 10.6 standard |
| **8.2** | The eRx Processing Hub will validate all eRx for medication against NCPDP SCRIPT 10.6 standard |
| **8.3** | The eRx Processing Hub will validate all eRx for expendable medical supplies against NCPDP SCRIPT 10.6 standard |
| **EPIC 9:** | The system needs to be able to receive and notify VA pharmacists that prior authorization has been approved or not approved by the payer for eRxs prescribed for non-formulary or restricted drugs. This requirement assumes compliance with NCPDP SCRIPT Standard for ePA transactions by external providers, payers, and VA pharmacies |
| **9.1** | Requirements in progress. This will be updated in the future. |
| **9.2** | Requirements in progress. This will be updated in the future. |
| **9.3** | Requirements in progress. This will be updated in the future. |
| **EPIC 10:** | All messages sent out from VA Pharmacies to external providers or received into VA Pharmacies from external providers must comply with NCPDP SCRIPT 10.6 standards |
| **10.1** | The eRx Processing Hub components will send the outgoing messages to Change Healthcare in NCPDP XML |
| **10.2** | The eRx Processing Hub components will send the outgoing messages to Change Healthcare in NCPDP XML |
| **10.3** | The eRx Processing Hub will communicate to Change Healthcare via Web Services following NCPDP XML |
| **EPIC 11:** | The system needs to electronically receive and send change requests to external providers based on a previously received eRx from that external provider |
| **11.1** | During the manual processing of entries in the eRx Holding Queue, there may be a need for a change request to be generated. A ‘Change Request’ action will be added within the List Manager interface that will allow the user to edit the eRx and submit/process the change request. The ‘Change Request’ option will identify what needs to be changed, and allow the pharmacist to initiate an electronic message back to the provider to approve the change. When a change request occurs, the status of the eRx entry will need to reflect that a change has been requested. The status will be updated to ‘Change Request’ or something equivalent.  The request for RxChange will be initiated by the VistA OP via ErxWSCaller routine by calling VistAErxStatuses webservice in the eRx middle tier. The eRx change request will be then sent to Change Healthcare via OutboundErxMessages proxy in DAS |
| **11.2** | Change Healthcare will send a system generated auto response to the eRx change request mentioned in 11.1 |
| **11.3** | The external provider will send Change Healthcare Rx change request. Change Healthcare will send the RxChangeRequest which will arrive in DAS JMS Queue. The eRx Processing Hub “listens” to the configured DAS Message Queue picks up the message and processes it. Finally it is sent to the staging area in VistA OP, from there it is ready to be processed in VistA OP. |
| **11.4** | Once the ‘updated’ eRx message is received, it will be utilized to ‘update’ the eRx in the eRx Holding Queue. This will be done by the eRx unique identifier. The pharmacist will be given the opportunity to review the changes and ‘accept’ or ‘reject’ those changes. If the response is a ‘denial’, the Pharmacist will have the ability to ‘reject’ or ‘accept’ that denial.  The eRx middle tier will send an auto acknowledge response for the RxChangeRequest coming from Change Healthcare. Once the Rx has been dispensed in VistA OP, another message will be sent to Change Healthcare via ErxWSCaller routine calling VistAErxStatuses webservice in the eRx middle tier. That webservice formats the data in NCPDP and sends the status update to Change Healthcare via DAS OutboundErxMessages proxy. |
| **EPIC 12:** | The system needs to be capable of electronically receiving and responding to cancellation requests for previously received eRxs from external providers. |
| **12.1** | The external provider will send Change Healthcare CancelRx request. Change Healthcare will send the CancelRx request which will arrive in DAS Queue. The eRx Processing Hub “listens” to the configured DAS Message Queue picks up the message and processes it. Finally it is sent to the staging area in VistA OP; from there it is ready to be processed in VistA OP.  The eRx middle tier will send an auto acknowledge response for the CancelRx coming from Change Healthcare. If the eRx is in still pending status or if the eRx has been filled and still has refills available, it will be cancelled by the Pharmacist. The VistA OP will initiate a message for cancelled eRx and that message will be sent to Change Healthcare via ErxWSCaller routine calling VistAErxStatuses webservice in the eRx middle tier. That webservice formats the data in NCPDP and sends the status update to Change Healthcare via DAS OutboundErxMessages proxy.  Much like the sending of a change request to an external provider, the system will monitor incoming messages to see if the unique eRx id matches an existing entry in the eRx Holding Queue. If it does, this will be considered an external change request. When received, the Pharmacist will be able to review the change request and approve those changes. This is dependent on the status of the eRx.  When an incoming change request occurs, the system will automatically generate an acknowledgement indicating the receipt of the message. Whether the Pharmacist accepts or rejects the change request, the system will respond with the appropriate information for the action taken.  When a cancellation request for an eRx that currently exists in the eRx Holding Queue is received, the system will evaluate the request versus the status of the eRx. If the eRx has not been completed, the system will update the eRx to a cancelled status. If there is a need for a Pharmacist to approve the cancellation, an option will be created to allow the Pharmacist to do so. |
| **EPIC 13:** | The system needs to be able to electronically send and receive refill requests to and from external providers, and to receive responses based on the refill request. |
| **13.1** | The PSO LM BACKDOOR orders option currently allows a user to ‘Refill’ an existing prescription. The PSO LM BACKDOOR orders functionality will be modified to check for an eRx ID in the prescription file. The eRx ID will be a new field added in order to associate the eRx identifier to the local prescription number. If the eRx ID exists, the system will take a different path than normal to generate and send a message to the external provider for refill approval. Once the response is received, the pharmacist will be able to use the eRx Holding Queue to accept the approval. The approval process will then update the prescription in the prescription file and allow the pharmacist to complete the refill processing the PSO LM BACKDOOR orders option.  If the intent is to utilize the eRx Holding Queue for all actions, an extended action will be included in the eRx Holding Queue that will allow the Pharmacist to initiate this message directly from the eRx Holding Queue instead of through the PSO LM BACKDOOR orders option. The functionality will be mostly the same; however the incoming response will need to be processed from the eRx Holding Queue rather than from the backdoor orders option.  The eRx refill request will be initiated by the VistA OP via ErxWSCaller routine by calling VistAErxStatuses webservice in the eRx middle tier. The eRx refill request will be then sent to Change Healthcare via OutboundErxMessages proxy in DAS.  Change Healthcare will send a system generated auto response to the eRx refill request mentioned above. |
| **13.2** | A refill response will be received just like any other incoming message from the provider. An acknowledgement will be sent for all messages. If the incoming message is a response to a refill request, the pharmacist will be given the ability to approve or submit the message for further processing within the outpatient pharmacy package. Upon approval, the logic associated with the eRx Holding Queue will leverage the refill process within the PSO LM BACKDOOR orders functionality to ‘refill’ the eRx. Linkage from the eRx Holding Queue and the Prescription file will be handled by a new field in the Prescription file that holds the eRx#.  All incoming messages will generate an acknowledgement message back to the provider via Change Healthcare.  The eRx middle tier will send an auto acknowledge response for the refill response coming from Change Healthcare. Once the eRx has been dispensed in VistA OP, another message will be sent to Change Healthcare via ErxWSCaller routine calling VistAErxStatuses webservice in the eRx middle tier. That webservice formats the data in NCPDP and sends the status update to Change Healthcare via DAS OutboundErxMessages proxy. |
| **EPIC 14:** | The system needs to send or receive medication history requests to and from external providers. |
| **14.1** | The external provider will send Change Healthcare medication history/ RxHistoryRequest. Change Healthcare will send the RxHistoryRequest which will arrive in DAS Queue. The eRx Processing Hub “listens” to the configured DAS Message Queue picks up the message and processes it.  It will call the HDR web service to get medication history/Rx History for the given patient and will send it to Change Healthcare in NCPDP 10.6 standard via OutboundErxMessages proxy. |
| **14.2** | The medication history/Rx history request will be initiated by the pharmacist in VistA OP via ErxWSCaller routine by calling VistAErxStatuses webservice in the eRx middle tier. The medication history request will be then sent to Change Healthcare via OutboundErxMessages proxy in DAS. |
| **EPIC 15:** | The system needs to send and receive internal transfer requests out from or into VA pharmacies. |
| **15.1** | Within the eRx Holding Queue, an option will be created so that the user may select an eRx, and choose to ‘transfer’ the eRx to another VA pharmacy. This logic will require that the pharmacist identify the new VA pharmacy to which the eRx will be transferred. The eRx to be transferred will be sent back out to the middleware, where the information will be communicated with Change Healthcare, as well as the external provider. The external provider will need to be updated as to where the eRx has been transferred to.  The eRx Holding Queue will accept any eRx, whether it is a new request, or a transfer from another VA facility. If the eRx is a transfer, it may need to hold an identifier indicating such as such as part of the message. VistA OP will call via ErxWSCaller routine PharmacyTransfers webservice which is hosted in the eRx middle tier.   * PharmacyTransfers web service receives prescription details, which is to be transferred to another pharmacy, via VistA OP. * It builds the XML conforming to NCPDP prescription transfers and puts it into DAS JMS Queue where DASReceiver picks it up and sends to Pentaho middle tier for processing which finally calls VistAErxTransmissions to transfer it. * It returns success in the response; if all processing is successful, otherwise it returns an error. |
| **15.2** | This is covered in 15.1 |
| **15.3** | During the eRx processing in 15.1, a check will be created to ensure the eRx to be transferred doesn’t contain any illegal or prohibited drugs (at both the federal and state level) |
| **EPIC 16:** | The system needs to send and receive transfer requests from external pharmacies or to and from DoD medical facilities. |
| **16.1** | This is currently out of scope. It may be addressed in the future. |
| **16.2** | This is currently out of scope. It may be addressed in the future. |
| **16.3** | This is currently out of scope. It may be addressed in the future. |
| **EPIC 17:** | The system needs to process inbound eRxs received from external providers separately from eRxs received from within VA pharmacy systems. |
| **17.1** | The Inbound eRx middle tier will have new database and tables to support the processing of eRxs. The VistA OP component for Inbound eRx processing will be new components built to support inbound eRxs and will try to avoid interfering with internal eRxs from within VA.  A new eRx Holding Queue in VistA OP will be created that will allow a pharmacist to view all inbound electronic prescriptions, rectify errors, review information, and ultimately, ‘submit’ or ‘approve’ the electronic prescription. New RPC’s will be created within the Outpatient Pharmacy package to accept incoming HL7 messages containing all the needed elements for a prescription from a non-VA medical facility. Using the inbound HL7 message, a new entry will be placed in the eRx Holding Queue file.  This queue will be created using List Manager. The eRx Holding Queue will show electronic prescriptions in a list format, with extended options being available to see all of the details about the prescriptions. Additional extended options will be created to allow the pharmacist to fix issues with the prescription, such as provider, drug, and patient identification errors.  After all automatic and manual checks are complete; a Pharmacist will need to choose to ‘Submit’ or ‘Approve’ the electronic prescription from the eRx Holding Queue. At this point, there will be enough information about the prescription to process the prescription into the PENDING OUTPATIENT ORDERS file (#52.41), much like the way it is done through CPRS today.  The VistA OP system will be able to provide 2 different ‘views’ for prescriptions coming from external providers, versus eRx’s coming from other VAMC’s. |
| **EPIC 18:** | VA pharmacy staff must determine that external providers were notified of status on their inbound eRxs at various intervals, from receipt to dispense. |
| **18.1** | * Once an Inbound eRx arrived in DAS, an auto acknowledgement of the message is sent to Change Healthcare. * The eRx Processing Hub “listens” to the configured DAS Message Queue and receives the eRx messages; structural validation is performed on the eRx message; messages are of different types (e.g., newRx, Refill). * The eRx Processing Hub coordinates auto-validations of the received eRx message (e.g., MVI call to identify and validate the patient). * Validated eRx transaction is sent to the staging area in VistA OP (the eRx Holding Queue). * Any change to the status of the eRx in VistA OP is synchronized with the Transaction/Processing table in the eRx Processing Hub. * VistA OP updates the eRx Processing Hub and triggers a message update to the External Provider (error, ready to dispense, dispensed etc.). * In the eRx Processing Hub, success (and error) status updates from VistA OP trigger message(s) to the External Provider; those messages (in NCPDP XML format) are sent to DAS for delivery to Change Healthcare and ultimately to the External Provider |
| **18.2** | *The eRx Processing Hub will transmit the status of eRxs being processed asynchronously at different intervals.* |
| **EPIC 19:** | The system needs to notify the VA Pharmacist when an auto-transmission was sent to the external provider that their eRx could not be filled due to a patient, provider, or drug validation error. |
| **19.1** | If an eRx has a validation error during the manual validation process in VistA OP.  VistA OP will call via ErxWSCaller routine VistAErxStatuses webservice which is hosted in the eRx middle tier. The MUMPS routine will notate the date and time of the status update. The VistAErxStatuses web service will store the status data in the eRX\_Log table. It will reply with a response to the ErxWSCaller routine. The web service will build Error message with details like error details, date and time according to NCPDP standard and will send to Change Healthcare via OutboundErxMessages proxy in DAS.  The user will not have to specify the date/time when an auto-transmission occurs based on a patient, provider, or drug validation error. During the auto validation process, if a validation error occurs on one of the above listed items, the eRx will be placed into an ‘error’ state, and will identify the item that validation failed on. An HL7 message will be sent according to the validation error, and information about that HL7 message will be linked to the eRx entry in the eRx Holding Queue. The pharmacist will be notified either through a mailman message, or a separate view within the eRx queue. The latter may be preferred, due to not knowing how much the users actually review their mail.  Rejections of eRxs can occur for many different reasons, including patient, provider, or drug validation errors. NCPDP messages sent back to the provider indicating that the eRx will not be processed will changed based on the type of error that occurs. This will allow VistA to communicate the appropriate meaningful message back to the provider via eRx middle tier, giving them information about what exactly needs to be resolved. |
| **19.2** | The eRx validation error message will be built on the Error message template provided by Change Healthcare. |
| **EPIC 20:** | Upon receipt of the ePrescription, the VA’s inbound ePrescription system must auto-match the patient specified on the eRx to a known/identifiable individual in VA who is confirmed to be eligible for VA pharmacy benefits. |
| **20.1** | The Processing Hub will attempt to match the incoming Patient to a patient in MVI with a web service call. Additionally, after a successful match in MVI, the Processing Hub will attempt to determine the eligibility status of the patient using the E&E (Enrollment and Eligibility) web service |
| **20.2** | Once an eRx is received in the eRx Holding Queue (error or pending status), it will be manually checked to determine if the eRx is for a patient that exists in the sites Patient file (#2). If the patient does not exist in the target system, the message will be rejected. An error message will be sent back to the provider indicating the system could properly match the patient. |
| **20.3** | The system will attempt to match a patient based on all patient information provided, so that alias’s will cause as little problem as possible. Text matching is difficult to do in general, however there are some programmatic things that can be done to further check the name, such as the ‘contains [‘ function within Mumps. |
| **20.4** | Once the patient has been matched, and meets all criteria, the entry will be placed into the eRx Holding Queue, with a status of ‘auto-passed’. This will be a field in the eRx Holding Queue that is a flag of true or false, identifying whether or not the auto-checks were successful. |
| **20.5** | If patient, provider and drug auto-match are successful the eRx will be placed into the eRx Holding Queue with a status of ‘Pending’. |
| **20.6** | In the event of an error, the eRx will be placed into the eRx Holding Queue with a status of ‘Error’. It would be wise to indicate an error ‘subtype’ for proper processing of messages back to the external provider. The subtype here could be something like ‘Patient Mismatch’. This will assist in driving the ‘template’ or message processing to be used for each unique error type. The patient info as included in the incoming prescription as well as the auto-matched patient info will be included in the eRx Holding Queue to facilitate manual validation by the Pharmacist. |
| **EPIC 21:** | Upon the receipt of an inbound ePrescription, the VA’s Pharmacy system auto-checks to verify external provider that sent the eRx is a valid provider, authorized to write prescriptions and/or provide services to the specified patient. If provider auto-matches required criteria, system advances eRx to Holding Queue – Pending for processing and dispensing. If any one of the required criteria for provider, patient, or drug fails, auto-checking, system suspends in eRx in an Error Holding Queue for manual intervention by the pharmacist for resolution or rejection. |
| **21.1** | Once the provider has been added to the new person file, the data on the incoming eRx will be used to validate NPI or License #, provider information in general, and the DEA# in the event the eRx is a controlled substance. |
| **21.2** | If all provider checks are successful, the eRx will be updated, identifying that the auto-checks were successful. |
| **21.3** | Once auto-checks are finished successfully, the eRx will be updated to a status of ‘Pending’. |
| **21.4** | If any of the auto-checks fail, the status of the eRx will be set to ‘Error’. An additional subtype will most likely be needed to identify different types of errors. In this case, “Provider Validation Error” could be the subtype. The Provider information as included in the incoming prescription as well as the auto-matched provider will be included in the eRx Holding Queue to facilitate manual validation by the Pharmacist. |
| **EPIC 22:** | Upon receipt of the inbound ePrescription, the VA’s Pharmacy system auto-checks to verify medication or medical supply specified on the eRx is recognized by the VA and allowed to be prescribed to a VA patient. If drug auto-matches required criteria, system advances eRx to Holding Queue – Pending for processing and filling. If any one of the required criteria for provider, patient or drug fails auto-checking, system suspends eRx in an Error Holding Queue for manual intervention by the pharmacist for resolution or rejection. |
| **22.1** | New eRx logic will be created to utilize the NDC and other drug identifiers to attempt to match the drug requested to the same or equivalent drug within Vista. In the event the drug cannot be matched, the pharmacist will be given the option to select the ‘appropriate’ drug. This will be done from a filtered list of drug within the drug file. Drug Validation steps:   * 1. Retrieve the local VistA OP instance for the Pharmacy to which the eRx is being routed. Using the Product code associated with the NDC from the prescription (DrugCodedType🡪 ProductCode) in the NCPDP message.      1. Check the product files in the local VistA instance:         1. Check files 50.67 (NDC/UPN file) to get the VA Product Name field (#5) which is a pointer to file 50.68      2. Using the pointer to 50.68, find the product in the file 50.68 (List of available drug products)      3. Get the IEN and GCNSEQ of the product from file 50.68      4. Get all products matched to the IEN from file 50.0   2. If no local drugs have a match based on IEN, locate local drug(s) that match based on GCNSEQNO   3. Single or multiple results may be returned   4. Use the one which is only marked for outpatient use. |
| **22.2** | The Processing Hub will perform a level of drug matching using a newly developed RPC call based on the logic in 22.1 prior to sending the message to the VistA OP eRx Holding Queue. |
| **22.3** | The ePrescribing system within VistA will utilize existing API’s for allergies to get a list of allergies for the patient associated with the eRx. The incoming eRx can optionally contain a list of drug allergies if supplied by the provider. If any allergies listed in the eRx are not contained in the known allergies within VistA, the Pharmacist will be presented with a warning message, and taken through steps to verify/review all allergies from both VistA and the eRx. Will there be a need for entering new allergies? |
| **22.4** | After auto-checking is successful, the auto-check/auto-pass flag will be set to true |
| **22.5** | Once all criteria are met within the drug validation process, the eRx will be set to a ‘Pending’ status within the eRx Holding Queue for further processing. |
| **22.6** | If any of the auto-checks fail, the status of the eRx will be set to ‘Error’. An additional subtype will most likely be needed to identify different types of errors. In this case, “Drug Validation Error” could be the subtype.The drug as included in the incoming prescription as well as the auto-matched drug will be included in the eRx Holding Queue to facilitate manual validation by the Pharmacist. |
| **EPIC 23:** | After eRx automatically validates, pharmacist needs to be able to perform additional verification of eRx in VistA OP eRx Holding Queue – Pending to confirm patient, provider and drug. Once reviewed, start processing the eRx to dispense to Veteran patient. |
| **23.1** | An option will be included within the List Manager interface that will allow the pharmacist to review all patient, provider, and drug information. Once the pharmacist is certain everything is correct, the pharmacist will be allowed to ‘confirm’ or ‘approve’ the eRx for further processing. |
| **23.2** | An option will be created within List manager that allows the Pharmacist to select an eRx. The eRx will then display the entire contents of the eRx (as needed for the pharmacists’ current task). The Pharmacist will then be able to ‘confirm’ validity of the information related to patient, provider and drug. |
| **EPIC 24:** | The pharmacist needs to manually confirm the patient information before the eRx is moved to the VistA OP Holding Queue – Pending for processing. |
| **24.1** | Within the eRx Holding Queue, an option will be created in List Manager that will allow the pharmacist to manually verify patient identity. The Pharmacist will also be able to view eligibility information currently available to VistA OP. This will be done by comparing incoming patient information with information from the PATIENT file (#2). |
| **EPIC 25:** | The pharmacist needs to manually confirm the external provider information before the eRx is moved to the VistA OP Holding Queue – pending for processing. |
| **25.1** | Within the eRx Holding Queue, an option will be created in List Manager that will allow the pharmacist to manually verify provider information. If the prescription is a controlled substance, the DEA# will need to be verified as well. |
| **EPIC 26:** | The pharmacist needs to manually confirm the drug information before the eRx is moved to the VistA OP Holding Queue – pending for processing. |
| **26.1** | Within the eRx Holding Queue, an option will be created in List Manager that will allow the pharmacist to manually verify the drug/supply information.  If there are any allergies within the Patient’s profile within Vista, those allergies will be checked to determine if any of the allergies are associated with the electronically prescribed drug. |
| **EPIC 27:** | VA pharmacist manually handles patient, provider, and drug errors in the Error Queue, and if resolved, moves the eRx to the Holding Queue – Pending for eRxs ready to process. If eRx cannot be resolved, the eRx is rejected. |
| **27.1** | Options will be created to allow the pharmacist to manually interact with the eRxs that failed the auto-checking logic. If the error can be resolved, the pharmacist will be given the tools to do so. This will include tools to assist in matching the patient, provider, and drug information in the event any of these items failed the auto-check. In the event the error cannot be resolved, the pharmacist will be able to choose the option to ‘reject’ the eRx. In the event rejection is chosen, a message will be sent back to the external provider indicating what errors/issues could not be resolved. |
| **27.2** | As part of the above process, the option to open and view the eRx will be included, similar to the ‘Select Order’ function within PSO LM BACKDOOR ORDERS. |
| **EPIC 28:** | Provide the ability for VA pharmacy staff to manually validate an inbound eRx (to include identifying valid patient, provider, and medication or medical supply) before the eRx can be further processed for dispensing. |
| **28.1** | New actions/options will be included with the eRx Holding Queue that will allow the pharmacist to manually match a patient, if auto-matching fails. |
| **28.2** | If the pharmacist is able to validate or manually match an eRx that was previously in an error state due to patient mismatch, the pharmacist will be able to ‘approve’ the matching criteria, promoting the eRx to ‘Pending’ for further processing. |
| **28.3** | In the event the patient matching error cannot be resolved, the pharmacist will have an action/option within the list manager interface to ‘reject’ the eRx. In the event of a rejection, an appropriate message will be generated back to the external provider indicating the patient mismatch type of error. |
| **28.4** | The current VistA process to add a patient will be utilized. |
| **EPIC 29:** | Provide the ability to VA pharmacy staff to manually validate an inbound eRx (to include identifying valid patient, provider, and medication or medical supply) before the eRx can be further processed for dispensing. |
| **29.1** | New actions/options will be included with the eRx Holding Queue that will allow the pharmacist to attempt to manually match a provider, if auto-matching fails. |
| **29.2** | If the pharmacist is able to validate or manually match an eRx that was previously in an error state due to provider mismatch, the pharmacist will be able to ‘approve’ the matching criteria, promoting the eRx to ‘Pending’ for further processing. |
| **29.3** | In the event the provider matching/validation error cannot be resolved, the pharmacist will have an action/option within the list manager interface to ‘reject’ the eRx. In the event of a rejection, an appropriate message will be generated back to the external provider indicating the patient mismatch type of error. |
| **29.4** | We will need to ensure that all of the required data elements are available for a provider before we can insert a new provider into the New Person file. |
| **EPIC 30:** | VA pharmacy staff must manually interact with drug errors in the VistA Error Queue to resolve or reject if error cannot be resolved. If drug error is resolved, eRx is moved to VistA OP Holding Queue – Pending for eRxs ready to process. |
| **30.1** | New actions/options will be included with the eRx Holding Queue that will allow the pharmacist to attempt to manually match a drug/supply, if auto-matching fails. |
| **30.2** | If the pharmacist is able to validate or manually match an eRx that was previously in an error state due to drug/supply mismatch, the pharmacist will be able to ‘approve’ the matching criteria, promoting the eRx to ‘Pending’ for further processing. |
| **30.3** | In the event the drug/supply matching/validation error cannot be resolved, the pharmacist will have an action/option within the list manager interface to ‘reject’ the eRx. In the event of a rejection, an appropriate message will be generated back to the external provider indicating the patient mismatch type of error. |
| **30.4** | Additional review actions will be included allowing the pharmacist to review the validity of the drug. This includes NDC number, GCNSEQNO, and must not be designated as inpatient. |
| **30.5** | Once all validation is complete, the pharmacist will be able to use the options to move the eRx into the ‘Pending’ state. |
| **30.6** | In the event the drug fails to meet the required criteria, the pharmacist will be allowed to ‘reject’ the eRx. This will generate a ‘rejected’ message back to the external provider, indicating the reason for the rejection. |
| **30.7** | The eRx Holding queue logic will tie into the existing outpatient pharmacy logic. Once the eRx is approved, the eRx will first go to the Pending Outpatient Orders File. Once the information has been filed across the Prescription, Order, and Pending outpatient orders files, the existing pharmacy logic will take over. This will allow the users to leverage all of the existing allergies, and interactions built into that system. |
| **30.8** | This can also be handled by the existing outpatient pharmacy logic (associated with PSO LM BACKDOOR ORDERS). |
| **30.9** | This will also be handled by the existing PSO logic. |
| **30.10** | In the event an allergy is sent to VistA OP that does not exist in VistA OP, an API will be created that will allow the user to add the allergy information to the allergies file. Note\* - this may create a need for an ICR (Integration Control Registration) with the Adverse Reactions package. |
| **EPIC 31:** | The VistA OP Holding Queue needs to support the ability to validate, edit, inquire, search, sort, remove, or reject, toggle between eRxs pending and eRxs with errors, and exit from the Holding Queue functionality. |
| **31.1** | New options/functions will be created to allow a pharmacist to process incoming eRxs from external providers. |
| **31.2** | Edit functionality will be included within the List manager screens that will allow the user to validate, revise, edit, and update eRxs in the eRx Holding Queue. |
| **31.3** | A ‘Select Order’ (naming may change depending on needs) function will be included in the List manager display that will allow the user to display all of the available details for the selected order. This should look similar to the PSO LM BACKDOOR orders ‘Select Order’ option, but with additional/different data elements involved. |
| **31.4** | An order filtering option will be included in the eRx Holding Queue functionality that will allow the user to filter the eRxs in the queue by the specified data points (Patient name, date of birth, date received or date range, provider name, VA Pharmacy name or location, drug name, controlled substance, non-controlled substance). Note\* - pharmacy name or location seems odd, since only orders for this pharmacy will be available, unless this is a UI in the middleware. |
| **31.5** | This sounds like the same as the previous item. |
| **31.6** | An action will be included within the eRx Holding Queue that will allow the user to reject/remove an eRx due to cancellation or other validation errors. |
| **31.7** | An outgoing HL7 message will be generated and sent back to the external provider if an eRx is rejected or removed. There will be a required ‘reason’ for rejected/removed eRxs, to communicate the reason back to the ordering (external) provider. |
| **31.8** | A purging mechanism could be implemented, cleaning out the eRx Holding Queue file based on a specified time frame. This would require a new field in the OUTPATIENT SITE file to indicate the number of days to keep items in the Holding Queue. If this method is utilized, it is very likely that these views of the eRx Holding Queue could leverage the exact same functionality listed in 31.4. |
| **31.9** | A list manager option called ‘Switch view’ will be included that will allow the user to select ‘SV’ (or something similar. Once the user indicates a ‘switch view’ action, the user will be given an option to choose from Pending or Error queue. (May want to include ‘complete or finished’? |
| **31.10** | The user will be able to exit the Holding Queue via the built in List manager functionality. If the user does not select and action at the main screen, they will be taken back to the previous menu option menu. |
| **EPIC 32:** | The pharmacist needs to be able to sort on one or a set of eRxs for priority processing, and also advance resolved error-status eRxs to the pending queue so they can be successfully processed. |
| **32.1** | An auto-generated eRx would be one that passed all of the auto-validations in the Processing Hub and was entered into the Vista OP eRx Holding Queue as Pending |
| **32.2** | A priority setting will be included in the eRx Holding Queue. As further definition occurs, the eRx Holding Queue will display the highest priority eRxs first. Reverse video can be used to highlight the higher priority eRxs if needed. This would probably be something to do only for the highest priority setting |
| **32.3** | This can be handled by including ‘error type’ as one of the search/sort mechanisms, and can leverage the logic associated with previous requirements. |
| **32.4** | An action will be included in list manager that will allow the user to select an eRx that has been ‘validated’ and has no current errors, and promote that eRx to a ‘Pending’ status. If there are any errors associated with the eRx when the promotion action is selected, the system will inform the user that the action cannot be taken, and they will be returned to their previous view. |
| **32.5** | This will require the eRx Holding queue to have some sort of protocol hanging off of the PSO LM BACKDOOR orders functionality. The protocol would send updates to the eRx Holding Queue once the eRx has been ‘Finished’ by the pharmacist. This is commonly done throughout vista (an example being the OR EVSEND PS protocol). |
| **32.6** | Using the PSO LM BACKDOOR orders current functionality, this will be handled automatically, as long as the provider information is entered into file 200 (new person), and it current/correct. |
| **EPIC 33:** | The system needs to track and audit inbound eRxs from external providers. |
| **33.1** | When the status of the eRx changes in Vista OP we will need to also update the status in the Processing Hub. This will allow the tracking and auditing of the eRxs to happen in the mid-tier Admin UI. |
| **33.2** | All eRx messages inbound and outbound will be stored in the Transactions table in the eRx Oracle database with a date and a timestamp. |
| **33.3** | The Transaction table will also allow reporting of failed and successful eRxs. |

#### Design Element Tables

The design element tables are provided for your convenience. Copy each table as many times as necessary to address multiple items within each section. Add rows and headings to the tables to provide any additional required information to define the item or to specify the modifications to the item. Numbering of the design element tables to align them underneath the applicable requirement or sub-requirement is recommended, but is left to the author’s discretion. For that reason they are not numbered in this template.

##### Routines (Entry Points)

This section is an illustration that is VistA specific. The authors are free to organize this information by technology, different templates, or optional sections depending on the task at hand.

Complete the table for each routine affected by the functionality being designed.

Table : Routines (Instructions)

| Routines | Instructions |
| --- | --- |
| Routine Name | List the routine affected by the functionality being designed. |
| Enhancement Category | Check the appropriate box: New, Modify, Delete, or No Change. |
| RTM | List the RSD item number within the SDD (i.e., If the RSD has a requirement of 3.3.1, add Support for a new API, then in this column list RSD Requirement 3.3.1) |
| Related Options | List options that directly call or are called by the routine. |
| Related Routines | List routines that directly call or are called by the routine. |
| Data Dictionary (DD) References | List files that reference the routine through input transforms, cross reference logic, etc. |
| Related Protocols | List protocols that reference or are referenced by the routine. |
| Related Integration Control Registrations (ICRs) | List proposed new ICRs and subscribed ICRs. Also, list any obscure Supported ICRs. |
| Data Passing | Check the appropriate box. Also a short description of what invokes the new/changed routine should be included in this section. An example of such a description would be a note that the new/changed routine will be invoked as part of a function call or it would be invoked through user menu-driven options, system protocols, HL7 Logical Links, etc. This section refers specifically to the change implemented with the design. |
| Input Attribute Name and Definition | List the Input Attributes passed into the new or changed routine logic. Each attribute should be defined. |
| Output Attribute Name and Definition | List the Output Attributes returned from the new or changed routine logic. Each attribute should be defined. |
| Current Logic | Define the current logic in the routine that the design will modify. If this is new code, enter “N/A”. |
| Modified Logic (Changes are in bold) | Define the logic in the routine that the design will implement. |

Table (Grouping): Routines

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| Routine Name | PSOERX | | | | |
| Enhancement Category | New | Modify | | Delete | No Change |
| RTM | EPIC 6 | | | | |
| Related Options | PSO ERX QUEUE | | | | |
| Related Routines | Routines “Called By” | | Routines “Called” | | |
|  |  | | PSOERX\* | | |

| Routines | Activities | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data Dictionary (DD) References | PSO ERX HOLDING QUEUE (#tbd) | | | | | | | | |
| Related Protocols | PSO ERX MENU – This will be the extended action menu for list manager. | | | | | | | | |
| Related Integration Control Registrations (ICRs) |  | | | | | | | | |
| Data Passing | Input | Output Reference | | Both | | | Global Reference | | Local |
| Input Attribute Name and Definition | Name:  Definition: | | | | | | | | |
| Output Attribute Name and Definition | Name:  Definition: | | | | | | | | |
| Current Logic | | | | | | | | | |
| N/A | | | | | | | | | |
| Modified Logic (Changes are in bold) | | | | | | | | | |
| List manager interface that will display the contents of the eRx Holding Queue. | | | | | | | | | |
| Routines | Activities | | | | | | | | |
| Routine Name | PSOERX1..n | | | | | | | | |
| Enhancement Category | New | | Modify | | | Delete | | No Change | |
| RTM |  | | | | | | | | |
| Related Options | PSO ERX QUEUE | | | | | | | | |
| Related Routines | Routines “Called By” | | | | Routines “Called” | | | | |
|  |  | | | |  | | | | |
| Routines | Activities | | | | | | | | |
| Data Dictionary (DD) References | PSO ERX HOLDING QUEUE (#TBD) | | | | | | | | |
| Related Protocols |  | | | | | | | | |
| Related Integration Control Registrations (ICRs) |  | | | | | | | | |
| Data Passing | Input | Output Reference | | Both | | | Global Reference | | Local |
| Input Attribute Name and Definition | Name:  Definition: | | | | | | | | |
| Output Attribute Name and Definition | Name:  Definition: | | | | | | | | |
| Current Logic | | | | | | | | | |
| N/A | | | | | | | | | |
| Modified Logic (Changes are in bold) | | | | | | | | | |
| Utilities to ‘Edit’, ‘Change’, ‘Process’, ‘Accept’, ‘Verify Drug’, ‘Verify Provider’ and any other needed functions. This will require multiple PSOERX\* routines to account for all actions. Once the prescription has been validated, it will be moved to the PENDING OUTPATIENT ORDERS file (#52.41) for further processing. | | | | | | | | | |
| Routines | Activities | | | | | | | | |
| Routine Name | PSOHTTP? | | | | | | | | |
| Enhancement Category | New | | Modify | | | Delete | | No Change | |
| RTM |  | | | | | | | | |
| Related Options |  | | | | | | | | |
| Related Routines | Routines “Called By” | | | | Routines “Called” | | | | |
|  |  | | | |  | | | | |
| Routines | Activities | | | | | | | | |
| Data Dictionary (DD) References | PSO ERX HOLDING QUEUE (#tbd) | | | | | | | | |
| Related Protocols | tbd | | | | | | | | |
| Related Integration Control Registrations (ICRs) |  | | | | | | | | |
| Data Passing | Input | Output Reference | | Both | | | Global Reference | | Local |
| Input Attribute Name and Definition | Name: TBD  Definition: | | | | | | | | |
| Output Attribute Name and Definition | Name: tbd  Definition: | | | | | | | | |
| Current Logic | | | | | | | | | |
| N/A | | | | | | | | | |
| Modified Logic (Changes are in bold) | | | | | | | | | |
| This will function much like PSSHTTP, to call out of Vista and to a web service. The web service target will be different, and additional processing may need to occur.  PEPSPOST(DOCHAND,XML) ;  ; @DESC Sends an HTTP request to PEPS as a POST  ;  ; @DOCHAND Handle to XML document  ; @XML XML request as string  ;  ; @RETURNS A handle to response XML document  ; 1 for success, 0 for failure  ;  NEW PSS,PSSERR,$ETRAP,$ESTACK  ;  ; Set error trap  SET $ETRAP="DO ERROR^PSSHTTP"  ;  SET PSS("server")="PEPS"  SET PSS("webserviceName")="ORDER\_CHECKS"  SET PSS("path")="ordercheck"  ;  SET PSS("parameterName")="xmlRequest"  SET PSS("parameterValue")=XML  ;  ; Get instance of client REST request object  SET PSS("restObject")=$$GETREST^XOBWLIB(PSS("webserviceName"),PSS("server"))  IF $DATA(^TMP($JOB,"OUT","EXCEPTION"))>0 QUIT 0  ;  ; Insert XML as parameter  DO PSS("restObject").InsertFormData(PSS("parameterName"),PSS("parameter Value"))  IF $DATA(^TMP($JOB,"OUT","EXCEPTION"))>0 QUIT 0  ;  ; Execute HTTP Post method  SET PSS("postResult")=$$POST^XOBWLIB(PSS("restObject"),PSS("path"),.PSS  ERR)  IF $DATA(^TMP($JOB,"OUT","EXCEPTION"))>0 QUIT 0  ;  DO:PSS("postResult")  . SET PSS("result")=##class(gov.va.med.pre.ws.XMLHandler).getHandleToXm  lDoc(PSS("restObject").HttpResponse.Data, .DOCHAND)  . QUIT  ;  DO:'PSS("postResult")  . SET ^TMP($JOB,"OUT","EXCEPTION")="Unable to make http request."  . SET PSS("result")=0  . QUIT  ;  QUIT PSS("result")  ;;  ERROR ;  ; @DESC Handles error during request to PEPS via webservice.  ;  ; Depends on GLOBAL variable PSSERR to be set in previous call.  ;  ; @RETURNS Nothing. Value store in global.  ;  NEW ERRARRAY  ;  ; Get error object from Error Object Factory  IF $GET(PSSERR)="" SET PSSERR=$$EOFAC^XOBWLIB()  ; Store the error object in the error array  DO ERR2ARR^XOBWLIB(PSSERR,.ERRARRAY)  ;  ; Parse out the error text and store in global  SET ^TMP($JOB,"OUT","EXCEPTION")=$$GETTEXT(.ERRARRAY)  ;  ; Set ecode to empty to return to calling function  SET $ECODE=""  ;  QUIT  ;;  GETTEXT(ERRARRAY) ;  ; @DESC Gets the error text from the array  ;  ; @ERRARRAY Error array stores error in format defined by web service p  roduct.  ;  ; @RETURNS Error info as a single string  ;  NEW PSS  ;  ; Loop through the text subscript of error array and concatenate  SET PSS("errorText")=""  SET PSS("I")=""  FOR SET PSS("I")=$ORDER(ERRARRAY("text",PSS("I"))) QUIT:PSS("I")="" D  O  . SET PSS("errorText")=PSS("errorText")\_ERRARRAY("text",PSS("I"))  . QUIT  ;  QUIT PSS("errorText")  ;; | | | | | | | | | |
| Routines | Activities | | | | | | | | |
| Routine Name | PSOORRNW | | | | | | | | |
| Enhancement Category | New | | Modify | | | Delete | | No Change | |
| RTM |  | | | | | | | | |
| Related Options | PSO LM BACKDOOR ORDERS | | | | | | | | |
| Related Routines | Routines “Called By” | | | | Routines “Called” | | | | |
|  | ^PSOORFIN | | | |  | | | | |
| Routines | Activities | | | | | | | | |
| Data Dictionary (DD) References |  | | | | | | | | |
| Related Protocols |  | | | | | | | | |
| Related Integration Control Registrations (ICRs) |  | | | | | | | | |
| Data Passing | Input | Output Reference | | Both | | | Global Reference | | Local |
| Input Attribute Name and Definition | Name:  Definition: | | | | | | | | |
| Output Attribute Name and Definition | Name:  Definition: | | | | | | | | |
| Current Logic | | | | | | | | | |
| Renewal function for Prescriptions | | | | | | | | | |
| Modified Logic (Changes are in bold) | | | | | | | | | |
| This code will need to be modified to check whether or not this was an eRx. If this prescription was an eRx from an external provider, a message will need to be generated back to the provider to inform the provider that there has been a renewal on an eRx. This will be done by checking the new field in the prescription file that will hold the eRx unique ID. | | | | | | | | | |
| Routines | Activities | | | | | | | | |
| Routine Name | PSOERXA\* | | | | | | | | |
| Enhancement Category | New | | Modify | | | Delete | | No Change | |
| RTM |  | | | | | | | | |
| Related Options | N/A | | | | | | | | |
| Related Routines | Routines “Called By” | | | | Routines “Called” | | | | |
|  |  | | | |  | | | | |
| Routines | Activities | | | | | | | | |
| Data Dictionary (DD) References | NEW PERSON FILE #200  DRUG file (#50)  PATIENT file (#2) | | | | | | | | |
| Related Protocols |  | | | | | | | | |
| Related Integration Control Registrations (ICRs) | MPI calls? | | | | | | | | |
| Data Passing | Input | Output Reference | | Both | | | Global Reference | | Local |
| Input Attribute Name and Definition | Name:  Definition: | | | | | | | | |
| Output Attribute Name and Definition | Name:  Definition: | | | | | | | | |
| Current Logic | | | | | | | | | |
| N/A | | | | | | | | | |
| Modified Logic (Changes are in bold) | | | | | | | | | |
| These routines (PSOERXAn) will be created as needed, and will hold all RPC’s needed for the project. This includes, but is not limited to, Provider Validation and Drug Validation. Patient validation may also need to occur, as well as other validation items. | | | | | | | | | |
| Routines | Activities | | | | | | | | |
| Routine Name | PSOORFIN\* | | | | | | | | |
| Enhancement Category | New | | Modify | | | Delete | | No Change | |
| RTM |  | | | | | | | | |
| Related Options |  | | | | | | | | |
| Related Routines | Routines “Called By” | | | | Routines “Called” | | | | |
|  |  | | | |  | | | | |
| Routines | Activities | | | | | | | | |
| Data Dictionary (DD) References | ^PS(52.41 | | | | | | | | |
| Related Protocols |  | | | | | | | | |
| Related Integration Control Registrations (ICRs) |  | | | | | | | | |
| Data Passing | Input | Output Reference | | Both | | | Global Reference | | Local |
| Input Attribute Name and Definition | Name:  Definition: | | | | | | | | |
| Output Attribute Name and Definition | Name:  Definition: | | | | | | | | |
| Current Logic | | | | | | | | | |
| Allows the user to finish a prescription. | | | | | | | | | |
| Modified Logic (Changes are in bold) | | | | | | | | | |
| Will need to be modified to include the eRx unique ID within the prescription file entry, as well as any other applicable functionality for handling an eRx. | | | | | | | | | |
| Routines | Activities | | | | | | | | |
| Routine Name | PSORX1 | | | | | | | | |
| Enhancement Category | New | | Modify | | | Delete | | No Change | |
| RTM |  | | | | | | | | |
| Related Options | PSO LM BACKDOOR ORDERS | | | | | | | | |

| Related Routines | Routines “Called By” | | | | Routines “Called” | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | |  | | | | |
| Routines | Activities | | | | | | | | |
| Data Dictionary (DD) References | ^PSRX | | | | | | | | |
| Related Protocols |  | | | | | | | | |
| Related Integration Control Registrations (ICRs) |  | | | | | | | | |
| Data Passing | Input | Output Reference | | Both | | | Global Reference | | Local |
| Input Attribute Name and Definition | Name:  Definition: | | | | | | | | |
| Output Attribute Name and Definition | Name:  Definition: | | | | | | | | |
| Current Logic | | | | | | | | | |
| This is the main driver for PSO LM BACKDOOR ORDERS. | | | | | | | | | |
| Modified Logic (Changes are in bold) | | | | | | | | | |
| PSORX1 will be modified to check for pending/non-approved medications within the eRx Holding Queue. If pending eRx’s are found for the selected patient, the user will be informed of the pending prescriptions. | | | | | | | | | |
| Routines | Activities | | | | | | | | |
| Routine Name |  | | | | | | | | |
| Enhancement Category | New | | Modify | | | Delete | | No Change | |
| RTM |  | | | | | | | | |
| Related Options |  | | | | | | | | |
| Related Routines | Routines “Called By” | | | | Routines “Called” | | | | |
|  |  | | | |  | | | | |
| Routines | Activities | | | | | | | | |
| Data Dictionary (DD) References |  | | | | | | | | |
| Related Protocols |  | | | | | | | | |
| Related Integration Control Registrations (ICRs) |  | | | | | | | | |
| Data Passing | Input | Output Reference | | Both | | | Global Reference | | Local |
| Input Attribute Name and Definition | Name:  Definition: | | | | | | | | |
| Output Attribute Name and Definition | Name:  Definition: | | | | | | | | |
| Current Logic | | | | | | | | | |
|  | | | | | | | | | |
| Modified Logic (Changes are in bold) | | | | | | | | | |
|  | | | | | | | | | |

##### Templates

Complete Table 21 for each template affected by the functionality being designed. A short description of what change will be made to the templates should be included in this section.

Note: If preferred, copy and paste this section directly from VA FileMan DDs instead of using the tables.

Table : Templates (Instructions)

| Templates | Instructions |
| --- | --- |
| Template Name | Identify the template affected by the functionality being designed |
| Enhancement Category | Check the appropriate box: New, Modify, Delete, or No Change. |
| RSD Traceability | List the Requirement Specification Document (RSD) item number within the SDD (i.e., If the RSD has a requirement of 3.3.1, add Support for a new API, then this column should list RSD Requirement 3.3.1) |
| Template Type | Indicate the type of template identified (Sort, Input, or Print). |
| Related Options | List options that directly call or are called by the template. |
| Related Routines | List routines that directly call or are called by the template. |
| Data Dictionary (DD) References | List files/fields that reference the template(s) through input transforms, and cross reference logic. |
| Global References | List the ICRs for global references that are outside your namespace. |

Table : Templates

| Templates | Description | | | | |
| --- | --- | --- | --- | --- | --- |
| Template Name |  | | | | |
| Enhancement Category | New | Modify | | Delete | No Change |
| RSD |  | | | | |
| Template Type | Sort | Input | | Print | Other |
| Related Options |  | | | | |
| **Related Routines** | **Routines “Called By”** | | **Routines “Called”** | | |
|  |  | |  | | |
| Routines | Description | | | | |
| Data Dictionary (DD) References |  | | | | |
| Global References |  | | | | |

##### Bulletins

If the project develops or affects bulletins, then complete this section; if not then state that the section is not applicable and delete the tables and content of the section. Complete the table for each bulletin affected by the functionality being designed. A short description of what change will be made to the bulletins should be included in this section.

Note: If preferred, copy and paste this section directly from VA FileMan DDs instead of using the tables.

Table : Bulletins (Instructions)

|  |  |
| --- | --- |
| Bulletins | Instructions |
| Bulletin Name | List the specific bulletin affected by the functionality being designed. |
| Enhancement Category | Check the appropriate box: New, Modify, Delete, or No Change. |
| RTM | List the RSD item number within the SDD (i.e., If the RSD has a requirement of 3.3.1, add Support for a new API, then in this column list RSD Requirement 3.3.1). |
| Related Options | List options that directly send the bulletin. |
| Related Routines | List routines that directly send the bulletin. |
| Mail Subject | List the subject of the mail message, i.e., which bulletin this affects. |
| Mail Group | List the mail group (recipients) of the mail message. |
| Parameters | List necessary parameters. |
| Data Dictionary (DD) References | List files/fields that reference the bulletin(s) through input transforms, cross reference logic, etc. should be listed under Data Dictionary (DD) References. |

Table : Bulletins

| Bulletins | Description | | | | |
| --- | --- | --- | --- | --- | --- |
| Bulletin Name |  | | | | |
| Enhancement Category | New | Modify | | Delete | No Change |
| RTM |  | | | | |
| Related Routines | Routines “Called By” | | Routines “Called” | | |
|  |  | |  | | |
| Routines | Description | | | | |
| Mail Subject |  | | | | |
| Mail Group |  | | | | |
| Parameters |  | | | | |
| Data Dictionary (DD) References |  | | | | |

##### Data Entries Affected by the Design

Provide the following data for each field to be created, modified, or deleted or provide a “Before and After: Data Entries Affected by the Design.”

Identify the entries affected by the design. If a blanket change will be made to each entry affected, that change should be defined in this table.

Only changes that are unique to each record should be defined in the Unique Record(s) section (Section 6.2.2.3.5). Redundant information should not be entered into each chart in the Unique Record(s) section.

Table : Data Entries Affected by the Design

| Field Name | Current Value | New Value |
| --- | --- | --- |
|  |  |  |

##### Unique Record(s)

List the unique record ID(s) that will be affected by the changes implemented by the design. This is commonly done in the .01 field. The values defined in the Current Value and New Value columns should be the exact value of the data. For each unique record ID, copy this table and provide the information.

Table : Unique Record ID

| Field Name(s) | Current Value | New Value |
| --- | --- | --- |
|  |  |  |

##### File or Global Size Changes

Indicate the change to the size of the file or global as a result of the design implemented with this description. Global size changes tie back to the business requirements and RSD. Growth or reduction in the size of the global should be indicated in this section. If the file is static across all VistA systems, a blanket statement of how the change will affect the size of the global will suffice.

For example, “The National Procedure file is a new file and will require 8.7K of disk space to install.”

If a file is dynamic and its size may vary from VistA system to VistA system, the description should indicate the change in the file per record and the number of records that the site may anticipate. For example, if a field is being added to the patient file that will result in an increase of 7K per patient, the site can estimate the global growth based on the number of entries in that file.

Note: If the Capacity Planning analysis is available, then enter it here. If not, then use the Project Team projection.

Table : File or Global Size Changes

| File/Global Name(s) | Estimated Increase | Estimated Decrease |
| --- | --- | --- |
|  |  |  |

##### Mail Groups

Complete the table for each of the mail groups affected by the functionality being designed. A short description of what changes will be made to the affected mail groups should be included in this section.

Note: If preferred, this can be captured directly from VA FileMan DDs after the fact.

Table : Mail Groups (Instructions)

| Mail Groups | Instructions |
| --- | --- |
| Mail Group Name | List the name of the mail group being modified. The mail group name may include a domain name. |
| Enhancement Category | Check the appropriate box: New, Modify, Delete, or No Change. |
| Related Options | List options that directly reference the file. |
| Related Routines | List routines that reference the mail group. |
| Data Dictionary (DDs) References | List files that reference the mail group through input transforms, cross-reference logic, etc. |
| Related Protocols | List protocols that directly reference the mail group. |
| Mail Group Description | Describe the purpose for the mail group. |
| Self-Enrollment Allowed | Check the appropriate box either Yes or No. |
| Type | Check the appropriate box either Public or Private. |

Table : Mail Groups

| Mail Groups | Activities | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Mail Group Name |  | | | | | |
| Enhancement Category | New | | Modify | | Delete | No Change |
| Related Options |  | | | | | |
| Related Routines | Routines “Called By” | | | Routines “Called” | | |
|  |  | | |  | | |
| Mail Groups | Instructions | | | | | |
| Data Dictionary (DD) References |  | | | | | |
| Related Protocols |  | | | | | |
| Mail Group Description |  | | | | | |
| Self-Enrollment Allowed | Yes | No | | | | |
| Type | Public | Private | | | | |

##### Security Keys

This section lists the specific security keys affected by the functionality being designed. A short description of the changes that will be made to the security keys affected should be included in this section.

Note: If preferred, this can be captured directly from VA FileMan DDs after the fact.

Table : Security Keys (Instructions)

| Security Keys | Instructions |
| --- | --- |
| Security Key Name | List the specific name of the security key being modified. |
| Enhancement Category | Check the appropriate box: New, Modify, Delete, or No Change. |
| Related Options | List options that directly reference the security key. |
| Related Routines | List routines that reference the security key. |
| Data Passing | Check the appropriate box. Enter a short description of an event that would trigger the new/changed routine, for example, a note that the change to the security key will be referenced through user menu driven options, routines, etc. This section refers specifically to the change implemented with the design. |
| Security Key Description | List a brief description of the security key. |
| Subordinate Keys | List any subordinate keys. |
| Mutually Exclusive Keys | Enter the name of a key that may not be held jointly with this one. |
| Granting Condition Logic | Define the logic for the Granting Condition of the Security Key affected by the functionality being designed. |
| Current Logic | If the security key currently has a granting condition, define the current logic for that granting condition. If the security key did not exist before, indicate that there is currently no security key. |
| Modified Logic  (Changes are in bold) | Define the granting condition that the design will implement. If the security key is new to the field, define the logic here. |
| Hierarchical Precedence | Define which key is used if one key will take precedence over another key. |

Table : Security Keys

| Security Keys | Activities | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Security Key Name | PSOERX | | | | | | | | |
| Enhancement Category | New | | Modify | | | Delete | | No Change | |
| Related Options | PSO ERX QUEUE | | | | | | | | |
| Related Routines | Routines “Called By” | | | | Routines “Called” | | | | |
|  |  | | | |  | | | | |
| Security Keys | Activities | | | | | | | | |
| Data Passing | Input | Output | | Both | | | Global Reference | | Local Reference |
| Security Key Description | This is the security key required for a user to access the eRx Holding Queue. | | | | | | | | |
| Subordinate Keys |  | | | | | | | | |
| Mutually Exclusive Keys |  | | | | | | | | |
| Granting Condition Logic |  | | | | | | | | |
| Current Logic | | | | | | | | | |
|  | | | | | | | | | |
| Modified Logic (Changes are in bold) | | | | | | | | | |
|  | | | | | | | | | |
| Security Keys | Activities | | | | | | | | |
| Hierarchical Precedence |  | | | | | | | | |

##### Options

Complete the table for each of the options affected by the functionality being designed. A short description of the changes that will be made to the options affected should be included. Changes to the OPTION file (#19) are to be included, not the functionality of the option invoked.

Note: If preferred, this can be captured directly from VA FileMan DD after the fact.

Table : Options (Instructions)

| Options | Instructions |
| --- | --- |
| Option Name  (MENU TEXT field) | Enter the name of the option affected. |
| Enhancement Category | Check the appropriate box: New, Modify, Delete, or No Change |
| Associated Menu Options that will invoke this reference | List the menu type options on which the respective option is or will be contained. |
| Data Passing | Check the appropriate box. Also a short description of what invokes the new/changed routine should be included in this section. An example of such a description would be a note that the change to the option will be referenced through VA Mailman server messages, user selection of the option from the VA Kernel Menu Management system, etc. This section refers specifically to the change implemented with the design. |
| Menu Text Description | Enter the name of the option as it will be displayed to the user within the menu system. |
| Option Type | Specify the type of option |
| Option Definition | Provide all the information necessary to fully define the option. Include options that are included in the menu, if applicable. |
| Current Entry Action Logic | Define the current logic for the entry action of the option affected by the functionality being designed. If the entry action did not exist before, indicate that there currently is no entry action. |
| Modified Entry Action Logic (Changes are in bold) | Define the entry action that the design will implement. If the entry action is new to the field, define the logic here. |
| Current Exit Action Logic | Define the current logic for the exit action of the option affected by the functionality being designed. If the exit action did not exist before, indicate that there currently is no exit action. |
| Modified Exit Action Logic  (Changes are in bold) | Define the exit action that the design will implement. If the exit action is new to the field, define the logic here. |

Table : Options

| Options | Activities | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Option Name | PSO ERX | | | | | | | |
| Enhancement Category | New | | Modify | | Delete | | No Change | |
| Associated Menu Options that will invoke this reference |  | | | | | | | |
| Data Passing | Input | Output | | Both | | Global Reference | | Local Reference |
| Menu Text Description |  | | | | | | | |
| Option Type | Edit | | Print | | Menu | | Inquire | |
| Action | | Run Routine | | Other | |  | |
| Associated Routine |  | | | | | | | |
| Option Definition |  | | | | | | | |
| Current Entry Action Logic | | | | | | | | |
|  | | | | | | | | |
| Modified Entry Action Logic (Changes are in bold) | | | | | | | | |
|  | | | | | | | | |
| Current Exit Action Logic | | | | | | | | |
|  | | | | | | | | |
| Modified Exit Action Logic (Changes are in bold) | | | | | | | | |
|  | | | | | | | | |
| Options | Activities | | | | | | | |
| Option Name | PSO ERX QUEUE | | | | | | | |
| Enhancement Category | New | | Modify | | Delete | | No Change | |
| Associated Menu Options that will invoke this reference | PSO USER1 -> Rx (Prescriptions) | | | | | | | |
| Data Passing | Input | Output | | Both | | Global Reference | | Local Reference |
| Menu Text Description | Outpatient Pharmacy Electronic RX Holding Queue. | | | | | | | |
| Option Type | Edit | | Print | | Menu | | Inquire | |
| Action | | Run Routine | | Other | |  | |
| Associated Routine | PSOERX | | | | | | | |
| Option Definition |  | | | | | | | |
| Current Entry Action Logic | | | | | | | | |
| N/A | | | | | | | | |
| Modified Entry Action Logic (Changes are in bold) | | | | | | | | |
|  | | | | | | | | |
| Current Exit Action Logic | | | | | | | | |
| N/A | | | | | | | | |
| Modified Exit Action Logic (Changes are in bold) | | | | | | | | |
|  | | | | | | | | |
| Options | Activities | | | | | | | |
| Option Name |  | | | | | | | |
| Enhancement Category | New | | Modify | | Delete | | No Change | |
| Associated Menu Options that will invoke this reference |  | | | | | | | |
| Data Passing | Input | Output | | Both | | Global Reference | | Local Reference |
| Menu Text Description |  | | | | | | | |
| Option Type | Edit | | Print | | Menu | | Inquire | |
| Action | | Run Routine | | Other | |  | |
| Associated Routine |  | | | | | | | |
| Option Definition |  | | | | | | | |
| Current Entry Action Logic | | | | | | | | |
|  | | | | | | | | |
| Modified Entry Action Logic (Changes are in bold) | | | | | | | | |
|  | | | | | | | | |
| Current Exit Action Logic | | | | | | | | |
|  | | | | | | | | |
| Modified Exit Action Logic (Changes are in bold) | | | | | | | | |
|  | | | | | | | | |

##### Protocols

Complete the table for each of the protocols affected by the functionality being designed. A short description of the changes that will be made to the protocols affected should be included in this section. Changes to the PROTOCOL file (#101) are to be included, not the functionality of the protocol invoked.

Note: If preferred, this can be captured directly from VA FileMan DDs after the fact.

Table : Protocols (Instructions)

| Protocols | Instructions |
| --- | --- |
| Protocol Name | List the name of the protocol affected. |
| Enhancement Category | Check the appropriate box: New, Modify, Delete, or No Change. |
| Associated Protocols | List the ancestors of the protocol being designed, i.e., those protocols that contain the respective protocol as an item. |
| Data Passing | Check the appropriate box. An event that would trigger the new/changed protocol should be included in this section. An example would be a note that the change to the protocol will be referenced through the VA event driver, List Manager, user selection of a protocol from the VA Kernel Menu Management system. This section refers specifically to the change implemented with the design. |
| Item Text Description | Enter the protocol's text as it appears to the user on the menu or sub-header. |
| Protocol Type | Define the type of protocol to be executed |
| Associated Routine | List any associated routines affected by the protocol being designed. |
| Current Entry Action Logic | Define the current logic for the entry action of the protocol affected by the functionality being designed. If the entry action did not exist before, indicate that there currently is no entry action. |
| Modified Entry Action Logic  (Changes are in bold) | Define the entry action that the design will implement. If the entry action is new to the field, define the logic here. |
| Current Exit Action Logic | Define the current logic for the exit action of the protocol affected by the functionality being designed. If the exit action did not exist before, indicate that there currently is no exit action. |
| Modified Exit Action Logic  (Changes are in bold) | Define the exit action that the design will implement. If the exit action is new to the field, define the logic here. |

Table : Protocols

| Protocols | Activities | | | |
| --- | --- | --- | --- | --- |
| Protocol Name |  | | | |
| Enhancement Category | New | Modify | Delete | No Change |
| Associated Protocols |  | | | |
| Data Passing | Input  Output  Both  Global Reference  Local Reference | | | |
| Item Text Description |  | | | |
| Protocol Type | Action  Menu  Protocol  Protocol Menu  Limited Protocol  Extended Action  Dialog  Other | | | |
| Associated Routine |  | | | |

| Current Entry Action Logic |
| --- |
|  |
| Modified Entry Action Logic (Changes are in bold) |
|  |
| Current Exit Action Logic |
|  |
| Modified Exit Action Logic (Changes are in bold) |
|  |

##### Remote Procedure Call (RPC)

Complete the table for each RPC affected by the functionality being designed.

Note: If preferred, this can be captured directly from VA FileMan DDs after the fact.

Table : RPCs (Instructions)

| RPCs | Instructions |
| --- | --- |
| Name | List the specific name of the RPC affected. |
| TAG^RTN | List the tag (label) and routine. |
| Input Parameters | This field is used to identify an input parameter for the API. |
| Results Array | This field tells the RPC Broker how to process the resulting data from the call. |
| Description | Provide a brief description of the RPC affected. |

Table : RPCs

| RPCs | Activities | | |
| --- | --- | --- | --- |
| Name | PSO ERX PROVIDER VALIDATION | | |
| TAG^RTN | TBD | | |
| Input Parameters |  | | |
| Results Array | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| Description |  | | |
| RPCs | Activities | | |
| Name | PSO ERX PATIENT VALIDATION | | |
| TAG^RTN | TBD | | |
| Input Parameters |  | | |
| Results Array | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| Description |  | | |

| RPCs | Activities | | |
| --- | --- | --- | --- |
| Name | PSO ERX DRUG VALIDATION | | |
| TAG^RTN |  | | |
| Input Parameters |  | | |
| Results Array | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| Description |  | | |

##### Constants Defined in Interface

Provide the name and description.

Table : Constants Defined in Interface

| Name | Description |
| --- | --- |
|  |  |

##### Variables Defined in Interface

Provide the name, type, and description.

Table : Variables Defined in Interface

| Name | Type | Description |
| --- | --- | --- |
|  |  |  |

##### Types Defined in Interface

Provide the name, type, and description.

Table : Types Defined in Interface

| Name | Type | Description |
| --- | --- | --- |
|  |  |  |

##### GUI

List the GUI affected by the functionality being designed and include a short description of the changes made to the affected GUI. The headers in the following tables have names for the information outlined. There are a number of items in this section that would generally be global information and visible to all other aspects.

Table : GUI

| Unit Name | Description |
| --- | --- |
|  |  |

##### GUI Classes

Table : GUI Classes (Instructions)

| GUI Classes | Instructions |
| --- | --- |
| Class Name | List the name of the class affected. The headers in the following tables have names for the information outlined. Note that only the new properties and methods for a class are listed below. All ancestor properties and methods are still available and unchanged. |
| Derived From Class | List the class that this is derived from, its parent and any interfaces listed as part of this class. |
| Purpose | Describe the functionality that users can access from this class and related form, if any. |

Table : GUI Classes

| GUI Classes | Instructions |
| --- | --- |
| Class Name |  |
| Derived From Class |  |
| Purpose |  |

##### Current Form

Provide a screen capture or graphical representation of the current layout.

##### Modified Form

Provide a screen capture or graphical representation of the layout that the design will implement.

##### Components on Form

Table : Components on Form

| Name | Type | Description |
| --- | --- | --- |
|  |  |  |

##### Events

Table : Events

| Name | Type | Description |
| --- | --- | --- |
|  |  |  |

##### Methods

Table : Methods

| Method Name | Procedure/Function | Description |
| --- | --- | --- |
|  |  |  |

##### Special References

Include references that are not listed elsewhere.

| Special Reference Name | Type | Description |
| --- | --- | --- |
|  |  |  |

##### Class Events

Table : Class Events

| Name | Type | Description |
| --- | --- | --- |
|  |  |  |

##### Class Methods

Table : Class Methods

| Name | Procedure/Function | Description |
| --- | --- | --- |
|  |  |  |

##### Class Properties

Table : Class Properties

| Class Properties Name | Type | Visibility | Description |
| --- | --- | --- | --- |
|  |  |  |  |

##### Uses Clause

Use this section to provide a uses clause that lists the other units (code or form units) that this unit will use. This may be documented in the form of a Unified Modeling Language (UML) drawing.

##### Forms

This section lists the forms that will be affected or created by the functionality being designed. A short description of the change that will be made to the forms should be included.

Table : Forms (Instructions)

| Forms | Instructions |
| --- | --- |
| Form Name | List the name of the form affected by the functionality being designed. |
| Enhancement Category | Check the appropriate box: New, Modify, Delete, or No Change. |
| Form Functionality | Describe the form’s functionality and refer to the usage of the form. An example of such a description is “This form is used to enter patient demographic data.” |
| Current Form Layout | Define the current form layout that the design will modify. If this is a new form, enter “N/A”. |
| Modified Form Layout (Changes are in bold) | Define the form layout that the design will implement. |

Table : Forms

| Forms | Description | | | |
| --- | --- | --- | --- | --- |
| Form Name |  | | | |
| Enhancement Category | New | Modify | Delete | No Change |
| Form Functionality |  | | | |
| Current Form Layout | | | | |
|  | | | | |
| Modified Form Layout (Changes are in bold) | | | | |
|  | | | | |

##### Functions

The functions affected by the capabilities being designed should be listed in this section. A short description of what change will be made to the functions and/or new functions should be included.

Table : Forms (Instructions)

| Functions | Instructions |
| --- | --- |
| Function Name | List the specific function affected by the capability being designed. |
| Short Description | List a short description of the change that will be made to the functions and/or new functions. |
| Enhancement Category | Check the appropriate box: New, Modify, Delete, or No Change. |
| Related Options | List the options that directly call or are called by the function. |
| Related Routines | List the routines that directly call or are called by the function. |
| Data Dictionary (DD) References | List the files that reference the function through input transforms, cross reference logic, etc. |
| Related Protocols | List the protocols that reference or are referenced by the function. |
| Related Integration Control Registrations (ICRs) | List proposed new ICRs and subscribed ICRs. Also, list any obscure Supported ICRs. |
| Data Passing | Check the appropriate box. An event that would trigger the new/changed function should be included in this section. An example of such a description would be a note that the new/changed function will be invoked as part of a function call or it would be invoked through system protocols, HL7 Logical Links, etc. This section refers specifically to the change implemented with the design. |
| Input Attribute Name and Definition | List the input attributes passed into the new or changed function logic. Each attribute should be defined. |
| Output Attribute Name and Definition | List the output attributes returned from the new or changed function logic. Each attribute should be defined. |
| Current Logic | Define the current logic in the function that the design will modify. If this is new code, enter “N/A”. |
| Modified Logic (Changes are in bold) | Define the logic in the function that the design will implement. |

Table : Forms

| Function Name | Activities | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Short Description |  | | | | | | | | |
| Enhancement Category | New | | Modify | | Delete | | | No Change | |
| Related Options |  | | | | | | | | |
| Related Routines | Routines “Called By” | | | | | Routines “Called” | | | |
|  |  | | | | |  | | | |
| Function Name | Activities | | | | | | | | |
| Data Dictionary (DD) References |  | | | | | | | | |
| Related Protocols |  | | | | | | | | |
| Related Integration Control Registrations (ICRs) |  | | | | | | | | |
| Data Passing | Input | Output | | Both | | | Global Reference | | Local Reference |
| Input Attribute Name and Definition | Name: | | | | | | | | |
| Definition: | | | | | | | | |
| Output Attribute Name and Definition | Name: | | | | | | | | |
| Definition: | | | | | | | | |
| Current Logic | | | | | | | | | |
|  | | | | | | | | | |
| Modified Logic (Changes are in bold) | | | | | | | | | |
|  | | | | | | | | | |

##### Dialog

In this section list the changes to the DIALOG file (#.84).

Table : Dialog (Instructions)

| Dialog | Instructions |
| --- | --- |
| Dialog Message (Description) | List the specific message affected or needed by the changes being designed. |
| Enhancement Category | Select the appropriate category: New, Modify, Delete, or No Change. |
| Dialog Message (Description) Condition | Describe the dialog message (description) functionality. An example of such a description would be the condition that would trigger the output of the message (dialog). This section refers to the condition generating the message (dialog). |
| Current Dialog Message (Description) | Define the current dialog message (description) that the design will modify. If this is a new dialog message (description) enter N/A. |
| Modified Dialog Message (Description)  (Changes are in bold) | Define the dialog message (description) that the design will implement. |

Table : Dialog

| Dialog | Instructions | | | |
| --- | --- | --- | --- | --- |
| Dialog Message (Description) |  | | | |
| Enhancement Category | New | Modify | Delete | No Change |
| Dialog Message (Description) Condition |  | | | |
| Current Dialog Message (Description) |  | | | |
| Modified Dialog Message (Description)  (Changes are in bold) |  | | | |

##### Help Frame

A short description of what change will be made to the Help Frame text and/or new text should be included in this section. Help frames may be associated with options or with data dictionary fields to provide on-line instruction.

Table : Help Frame (Instructions)

| Help Frame | Instructions |
| --- | --- |
| Help Frame Text | List the text affected or needed by the changes being designed. |
| Enhancement Category | Check the appropriate box: New, Modify, Delete, or No Change. |
| Help Frame Text Calling Mechanism | Provide a short description of the mechanism used to call the Help Frame text in this section. An example of a mechanism would be the name of the routine or an explanation of how the Help Frame is called. An example of a calling mechanism would be the Standard VA FileMan API and the keystroke(s) that would trigger the output of the text. |
| Current Help Frame Text | List the current Help Frame Text that the design will modify. If new text enter N/A. |
| Modified Help Frame Text (Changes are in bold) | List the Help Frame Text that the design will modify. |

Table : Help Frame

| Help Frame | Description | | | |
| --- | --- | --- | --- | --- |
| Help Frame Text |  | | | |
| Enhancement Category | New | Modify | Delete | No Change |
| Help Frame Text Calling Mechanism |  | | | |
| Current Help Frame Text | | | | |
|  | | | | |
| Modified Help Frame Text (Changes are in bold) | | | | |
|  | | | | |

##### HL7 Application Parameter

Table : HL7 Application Parameter (Instructions)

| HL7 Application Parameter | Instructions |
| --- | --- |
| HL7 Application Parameter Name | List the HL7 Application Parameter affected or needed by the changes being designed. |
| Enhancement Category | Check the appropriate box: New, Modify, Delete, or No Change. |
| Application Status | Check the appropriate box in the applicable column for Current and Modified |
| Facility Name | List the current and modified value in the appropriate column. |
| Country Code | List the current and modified value in the appropriate column. |
| HL7 Field Separator | List the current and modified value in the appropriate column. |
| HL7 Encoding Characters | List the current and modified value in the appropriate column. |
| Mail Group | List the current and modified value in the appropriate column. |

Table : HL7 Application Parameter

| HL7 Application Parameter Name | Description | | | | |
| --- | --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | | Delete | No Change |
| Application Status | Active | Inactive | | Active | Inactive |
| Enhancement Category | Current | | Modified | | |
| Facility Name |  | |  | | |
| Country Code |  | |  | | |
| HL7 Field Separator |  | |  | | |
| HL7 Encoding Characters |  | |  | | |
| Mail Group |  | |  | | |

##### HL7 Logical Link

Table : HL7 Logical Link (Instructions)

| HL7 Logical Link | Instructions |
| --- | --- |
| HL7 Logical Link Parameter (LLP) Name | List the specific HL7 Logical Link affected or needed by the changes being designed. |
| Enhancement Category | Check the appropriate box: New, Modify, Delete, or No Change. |
| Node | List the current and modified value in the appropriate column. |
| Institution | List the current and modified value in the appropriate column. |
| Domain | List the current and modified value in the appropriate column. |
| Autostart | List the current and modified value in the appropriate column. |
| Queue Size | List the current and modified value in the appropriate column. |
| LLP Type | List the current and modified value in the appropriate column. |

Table : HL7 Logical Link

| HL7 Logical Link | Description | | | | |
| --- | --- | --- | --- | --- | --- |
| HL7 Logical Link Parameter Name |  | | | | |
| Enhancement Category | New | Modify | | Delete | No Change |
| Enhancement Category | Current | | Modified | | |
| Node |  | |  | | |
| Institution |  | |  | | |
| Domain |  | |  | | |
| Autostart |  | |  | | |
| Queue Size |  | |  | | |
| LLP Type |  | |  | | |

##### COTS Interface

The specific communication method(s) and Application Interface(s) that will be created or modified for the COTS system being interfaced should be described in this section. A short description of the existing tools that will be used and any new tools that will be developed should also be included.

Table : COTS Interface (Instructions)

| COTS Interface | Instructions |
| --- | --- |
| Communication Method | List the specific communication method created or modified for the functionality being designed. |
| Application Interface | List the specific application interface created or modified for the functionality being designed. |

Table : COTS Interface

| COTS Interface | Description |
| --- | --- |
| Communication Method |  |
| Application Interface |  |

## Network Detailed Design

* Please refer to [Section 4.3](#_Network_Architecture) Network Design.

## Security and Privacy

### Security

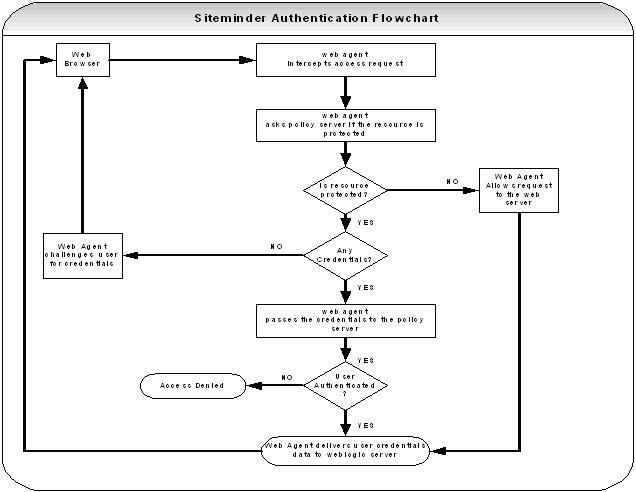
The Inbound eRx security controls have been implemented to comply with VA Handbook 6500. The following describes at a high-level the key security controls required for technical and operational implementation. For additional information related to full details of security, refer to Risk Vision ATO documentation.

| Security Control | Implementation |
| --- | --- |
| Access Control, Identification, Authentication, and Authorization | **Note**: Inbound eRx assumes that all users will be authenticated to the One VA network through VA SSOi service using their PIV card and Identity and Access Management (SAC) service using role based (RBAC) access functionality.  Note: Inbound eRx assumes that external connections (i.e. prescription clearinghouse, such as Change Healthcare) will follow the DAS security controls.  Privileged users will access Inbound eRx through Enterprise Operations (EO) generated accounts using 9957 process, which will be specific to their function and are issued and monitored from EO.  Messages will be encrypted using two-way SSL at the transport/session layer and via IP/Port identification at the network layer.  Inbound eRx shall follow VA Authentication Federation Infrastructure (VAAFI) standards when interacting with the Master Veteran Index (MVI) and other VA interfaces. |
| Auditing | All auditing to include infrastructure and application will be monitored by EO. |
| Media Protection, Transmission Confidentiality and Integrity | ePHI/PII data stored in the eRx Processing Hub is encrypted using oracle’s Transparent Data Encryption (TDE) and protected as per the VA VM and EO policies.  Messages are encrypted using two-way SSL at the transport/session layer and via IP/Port identification at the network layer. |
| Contingency planning /Disaster Recovery | EO is responsible for the Infrastructure and Inbound eRx is responsible for the Application. |
| Incident Response, Security Alerting | EO is responsible for Incident response and will notify Inbound eRx if there are any incidents. |
| Maintenance | EO is responsible for the maintenance of the Inbound eRx infrastructure. Inbound eRx is responsible for the application maintenance. |
| Physical and Environmental | EO is responsible for the Physical and environmental security controls. |
| Boundary Protection, network | EO is responsible for the boundary protection using Load balancers, firewalls, etc. |
| Cryptography | DAS processing tier is responsible for all external to VA communications. DAS has implemented two-way SSL which will be decrypted at the Layer 7 and IB. Certificates are provided by VA Public Key Infrastructure office and are compliant with VA Handbook 6500.  Internal to VA Cryptography (including database) is provided by EO. |
| Patch Management, Vulnerability Assessments | EO is responsible for patch management of the infrastructure and Inbound eRx is responsible for the applications. Both Inbound eRx and EO CM processes are followed. Vulnerability scanning is the responsibility of the National Security Operations Center, and the remediation of findings is the responsibility of both EO and Inbound eRx. |

#### Authentication and Authorization

Administrator UI in eRx Processing Hub uses the VASingle Sign-On (SSOi) service and VA Identity Access Management (IAM) solution for authentication (SiteMinder) (Figure 32). Authorization is role-based and is passed from SiteMinder via SAML2. The eRx Processing Hub maintains a mapping of roles to application functions in the eRx\_Config table. Spring Security is used to implement authorization.

Figure : SiteMinder Authentication Flowchart



### Privacy

The Inbound eRx complies with all VA privacy policies. The following describes at a high-level the key privacy controls required for technical and operational implementation.

| Privacy Control | Implementation |
| --- | --- |
| Governance and Privacy | A governance program has been established. Please refer to governance documentation for compliance. |
| Privacy Impact and Risk Assessment | As part of the ATO, a Privacy Impact Assessment and Risk Assessment are being performed for Inbound eRx. |
| Privacy Enhanced System Design | The system design for Inbound eRx employs technologies and system capabilities that allow PII/PHI data to flow through an ESB to eliminate additional copies of this type of data, automate collection, use, retention, and disclosure of PII. |
| Data Integrity | Data Integrity is handled through two-way SSL, service IP/Port identification, and, in through IAM. |
| Data Retention and Disposal | For stored data, encryption methods will be deployed according to VA Handbook 6500. Currently, there are no defined data archiving requirements for Inbound eRx. |
| PII Used in Testing | No PII data is using in testing scenarios. |
| Incident Response | Incidents for PII are reported to the VA Privacy Officer. |

## Service Oriented Architecture / ESS Detailed Design

Inbound eRx is designed as a service oriented architecture. It involves loose coupling between services, service reusability, service autonomy, service abstraction and utilizes services that are self-contained. Service-orientation is independent of any vendor, product or technology.

This section provides details of provided and consumed services as follows:

### Service Description for MVI

[MVI Service Description](http://DNS       .med.DNS   /warboard/ProjectDocs/MVI/MVI_Service_Description.pdf)

### Service Description for E&E

[Enrollment and Eligibility Service Description](http://DNS  DNS       .DNS   /sites/vrm/MSTI/Member%20Services/Member%20Services%20Program%20Artifacts/Archive/Data%20and%20Dependencies/Enrollment/ESR%20Enrollment%20Service%20Design%20Document.doc)

### Service Description for HDR

To be identified

### Service Design for PatientEligibility

Please see [PatientEligibility](#_PatientEligibility) for introduction, purpose and scope of service and service details.

#### Introduction

##### Purpose and Scope of Service

This service was described at a high level in the charter document. Please refer to it here via a link.

##### Links to Other Documents

This section will be addressed in the next increment.

Provide links to other documents created for this service so far in the SOA lifecycle. At a minimum, provide links to:

* Service Charter
* Service Roadmap
* Service Description

#### Service Details

##### Service Identification

This section will be addressed in the next increment.

| Service Attribute | Value |
| --- | --- |
| Name and Alias (if any) | Name of the service and other names for the service, which might be used by someone searching for this service. Please follow ESS naming standards. |
| Overview | Brief textual overview of the service. |
| Version | Version number of the service being described here |
| Latest Status | This field shows the latest status for the above referenced version of this service! The status of a service shows the progress of the service from initiation through development, deployment, and eventual retirement. The status also has a status date associated with the status - and we will be using the latest one here in this document. Valid values include: Inception, Design, Provisioning, Certification / Testing, Operation, Deprecated, Retired, Rejected - Owner has decided not to develop the service. |
| Service Type | Used to define applicable architecture patterns. Examples (from Open Group):  • Interaction  • Process  • Information  • Partner  • Business Application  • Access  • Service Connectivity |
| Architecture Layer | Referred to as class in VA Service template. Used to define applicable architecture patterns and relationships to governing bodies. Examples:  • Solution  • Process  • Information  • Utility  • Underlying |
| Business Domain | Business Vertical or Business Division where this service belongs. |
| Service Domain | The service or technical domain that the service belongs to. Can be used to establish the namespace. |
| Business Organization and Owner | Person who approves this service & any changes. Include email. |
| Technical Organization and Owner | Person responsible for provisioning (specifying, acquiring certifying) this service. Include email. |
| Development Organization and Owner | Person who is responsible for the development processes and activities for this service. Include email. |
| Support Organization and Owner | Person who is responsible for the support of this service while in production. Include email. |
| Target Consumer Organization(s) and Owner(s) | Organizations and/or developers roles that service is intended for. |

##### Service Versions

This section will be addressed in the next increment.

|  |  |  |
| --- | --- | --- |
| Version Numbers | Current Status of Version | A Brief Description of the change implemented in that version |
| This version | Being Designed |  |
| Example: version 2 | Example: In production. Will be retired with this release. | Example: This release added the ability to look up a person by address.  Provide a link to each version of the service. |
| Example: version 1 | Example: Retired. | Example: This release provided the base minimum functionality to look up a person by name.  Provide a link to each version of the service. |

##### Summary of Design and Platform Details

###### SOA Pattern(s) Implemented

The SOA patterns to be implemented for this service are Secured Message pattern and Request/Reaction pattern.

###### COTS Platform vendor names and versions for hosting platform

Oracle WebLogic 12c.

#### Dependencies

This service depends on MVI for patient check and E&E for patient enrollment and eligibility check.

#### Service Design Details

This section will be addressed in the next increment.

##### Interface Technical Specs

This section will be addressed in the next increment.

###### Service Invocation Type

REST.

###### Service Interface Type

URIs

###### Service Name

PatientEligibility

###### Interface

This section will be addressed in the next increment.

###### End Points

This section will be addressed in the next increment.

###### Operations or Methods

This section will be addressed in the next increment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Operation Name | Inputs | Outputs | Transactional Qualities if relevant (Updating?, Atomic?, Can participate in transaction?) | Pre and Post Conditions | Exception (s) |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

###### Message Schemas

This section will be addressed in the next increment.

##### Information Model

This section will be addressed in the next increment.

###### Class Diagram and Description of Entities Involved

This section will be addressed in the next increment.

###### Mappings from ELDM to Standards Based Schemas

This section will be addressed in the next increment.

##### Behavior Model (AKA Use Case Realization)

Please see [PatientEligibility](#_PatientEligibility)

###### Use Cases (Use Case Model)

Describe how this service fits into the larger use case model of the consumer. You may need multiple models for multiple consumers. Focus is **not** on the internal workings of the new service instead of the calls made from external consumers. Just a summary or the Use Case Diagram may be sufficient. List the alternative and exception flows. Reference the detailed design documents via a URL.

###### Interaction Diagrams

Please see [PatientEligibility](#_PatientEligibility)

#### Gap Analysis

Provide a Gap Analysis (Reference) to demonstrate compliance of this service with various standards, policies, guidelines and laws. The Gap Analysis may take the form of a matrix as shown in the sample below. This will help the governance boards expedite your request.

| Design Elements🡪  Policies / SLD elements etc.↓ | Design  Element A | Design  Element B | Design  Element C | Comment for non-conformance |
| --- | --- | --- | --- | --- |
| Policy X | Match |  |  |  |
| Policy Y |  | Partial |  |  |
| Policy Z |  |  |  | Commercial encryption server in prod will have to address this policy. |
| Policy A |  |  |  | Compliance with this policy not required until next year. |
| New / Additional Features |  |  | New element minimizes manual intervention |  |

##### Variances from Enterprise Target Architecture

This list of “variances” will become a submission to the ESS dispensation process.

##### Variances from SLDs

This list of “variances” will become a submission to the ESS dispensation process.

##### Variances from Standards and Policies

This list of “variances” will become a submission to the ESS dispensation process.

##### Justification for Exceptions and Mitigation

This section will list out any non-functional and functional requirements that are not being met. The non-conformance may be in violation of elements of SLDs, enterprise architecture (TRM Technology Reference Model), privacy policies or guidelines. For each exception provide:

1. Reasons for non-conformance (cost, time, technology, etc.)
2. Mitigating actions taken to reduce the impact of non-conformance
3. Plan (roadmap) to come back into conformance

This list can grow depending on what the Review bodies may ask for.

### Service Design for PrescriptionProviders

Please see [PrescriptionProviders](#_PrescriptionProviders) for introduction, purpose and scope of service and service details.

#### Introduction

##### Purpose and Scope of Service

This service was described at a high level in the charter document. Please refer to it here via a link.

##### Links to Other Documents

This section will be addressed in the next increment.

Provide links to other documents created for this service so far in the SOA lifecycle. At a minimum, provide links to:

* Service Charter
* Service Roadmap
* Service Description

#### Service Details

##### Service Identification

This section will be addressed in the next increment.

| Service Attribute | Value |
| --- | --- |
| Name and Alias (if any) | Name of the service and other names for the service, which might be used by someone searching for this service. Please follow ESS naming standards. |
| Overview | Brief textual overview of the service. |
| Version | Version number of the service being described here |
| Latest Status | This field shows the latest status for the above referenced version of this service! The status of a service shows the progress of the service from initiation through development, deployment, and eventual retirement. The status also has a status date associated with the status - and we will be using the latest one here in this document. Valid values include: Inception, Design, Provisioning, Certification / Testing, Operation, Deprecated, Retired, Rejected - Owner has decided not to develop the service. |
| Service Type | Used to define applicable architecture patterns. Examples (from Open Group):  • Interaction  • Process  • Information  • Partner  • Business Application  • Access  • Service Connectivity |
| Architecture Layer | Referred to as class in VA Service template. Used to define applicable architecture patterns and relationships to governing bodies. Examples:  • Solution  • Process  • Information  • Utility  • Underlying |
| Business Domain | Business Vertical or Business Division where this service belongs. |
| Service Domain | The service or technical domain that the service belongs to. Can be used to establish the namespace. |
| Business Organization and Owner | Person who approves this service & any changes. Include email. |
| Technical Organization and Owner | Person responsible for provisioning (specifying, acquiring certifying) this service. Include email. |
| Development Organization and Owner | Person who is responsible for the development processes and activities for this service. Include email. |
| Support Organization and Owner | Person who is responsible for the support of this service while in production. Include email. |
| Target Consumer Organization(s) and Owner(s) | Organizations and/or developers roles that service is intended for. |

##### Service Versions

This section will be addressed in the next increment.

|  |  |  |
| --- | --- | --- |
| Version Numbers | Current Status of Version | A Brief Description of the change implemented in that version |
| This version | Being Designed |  |
| Example: version 2 | Example: In production. Will be retired with this release. | Example: This release added the ability to look up a person by address.  Provide a link to each version of the service. |
| Example: version 1 | Example: Retired. | Example: This release provided the base minimum functionality to look up a person by name.  Provide a link to each version of the service. |

##### Summary of Design and Platform Details

###### SOA Pattern(s) Implemented

The SOA pattern to be implemented for this service is Request/Reaction pattern.

###### COTS Platform vendor names and versions for hosting platform

Oracle WebLogic 12c.

#### Dependencies

This service depends on VistA OP RPCs to function properly.

#### Service Design Details

This section will be addressed in the next increment.

##### Interface Technical Specs

This section will be addressed in the next increment.

###### Service Invocation Type

REST

###### Service Interface Type

URIs

###### Service Name

PrescriptionProviders

###### Interface

This section will be addressed in the next increment.

###### End Points

This section will be addressed in the next increment.

###### Operations or Methods

This section will be addressed in the next increment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Operation Name | Inputs | Outputs | Transactional Qualities if relevant (Updating?, Atomic?, Can participate in transaction?) | Pre and Post Conditions | Exception (s) |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

###### Message Schemas

This section will be addressed in the next increment.

##### Information Model

This section will be addressed in the next increment.

###### Class Diagram and Description of Entities Involved

This section will be addressed in the next increment.

###### Mappings from ELDM to Standards Based Schemas

This section will be addressed in the next increment.

##### Behavior Model (AKA Use Case Realization)

Please see [PrescriptionProviders](#_PrescriptionProviders)

###### Use Cases (Use Case Model)

Describe how this service fits into the larger use case model of the consumer. You may need multiple models for multiple consumers. Focus is **not** on the internal workings of the new service instead of the calls made from external consumers. Just a summary or the Use Case Diagram may be sufficient. List the alternative and exception flows. Reference the detailed design documents via a URL.

###### Interaction Diagrams

Please see [PrescriptionProviders](#_PrescriptionProviders)

#### Gap Analysis

Provide a Gap Analysis (Reference) to demonstrate compliance of this service with various standards, policies, guidelines and laws. The Gap Analysis may take the form of a matrix as shown in the sample below. This will help the governance boards expedite your request.

| Design Elements🡪  Policies / SLD elements etc.↓ | Design  Element A | Design  Element B | Design  Element C | Comment for non-conformance |
| --- | --- | --- | --- | --- |
| Policy X | Match |  |  |  |
| Policy Y |  | Partial |  |  |
| Policy Z |  |  |  | Commercial encryption server in prod will have to address this policy. |
| Policy A |  |  |  | Compliance with this policy not required until next year. |
| New / Additional Features |  |  | New element minimizes manual intervention |  |

##### Variances from Enterprise Target Architecture

None at this time

##### Variances from SLDs

None at this time

##### Variances from Standards and Policies

None at this time

##### Justification for Exceptions and Mitigation

This section will list out any non-functional and functional requirements that are not being met. The non-conformance may be in violation of elements of SLDs, enterprise architecture (TRM Technology Reference Model), privacy policies or guidelines. For each exception provide:

1. Reasons for non-conformance (cost, time, technology, etc.)
2. Mitigating actions taken to reduce the impact of non-conformance
3. Plan (roadmap) to come back into conformance

This list can grow depending on what the Review bodies may ask for.

### Service Design for PrescriptionDrugs

Please see [PrescriptionDrugs](#_PrescriptionDrugs) for introduction, purpose and scope of service and service details.

#### Introduction

##### Purpose and Scope of Service

This service was described at a high level in the charter document. Please refer to it here via a link.

##### Links to Other Documents

This section will be addressed in the next increment.

Provide links to other documents created for this service so far in the SOA lifecycle. At a minimum, provide links to:

* Service Charter
* Service Roadmap
* Service Description

#### Service Details

##### Service Identification

This section will be addressed in the next increment.

| Service Attribute | Value |
| --- | --- |
| Name and Alias (if any) | Name of the service and other names for the service, which might be used by someone searching for this service. Please follow ESS naming standards. |
| Overview | Brief textual overview of the service. |
| Version | Version number of the service being described here |
| Latest Status | This field shows the latest status for the above referenced version of this service! The status of a service shows the progress of the service from initiation through development, deployment, and eventual retirement. The status also has a status date associated with the status - and we will be using the latest one here in this document. Valid values include: Inception, Design, Provisioning, Certification / Testing, Operation, Deprecated, Retired, Rejected - Owner has decided not to develop the service. |
| Service Type | Used to define applicable architecture patterns. Examples (from Open Group):  • Interaction  • Process  • Information  • Partner  • Business Application  • Access  • Service Connectivity |
| Architecture Layer | Referred to as class in VA Service template. Used to define applicable architecture patterns and relationships to governing bodies. Examples:  • Solution  • Process  • Information  • Utility  • Underlying |
| Business Domain | Business Vertical or Business Division where this service belongs. |
| Service Domain | The service or technical domain that the service belongs to. Can be used to establish the namespace. |
| Business Organization and Owner | Person who approves this service & any changes. Include email. |
| Technical Organization and Owner | Person responsible for provisioning (specifying, acquiring certifying) this service. Include email. |
| Development Organization and Owner | Person who is responsible for the development processes and activities for this service. Include email. |
| Support Organization and Owner | Person who is responsible for the support of this service while in production. Include email. |
| Target Consumer Organization(s) and Owner(s) | Organizations and/or developers roles that service is intended for. |

##### Service Versions

This section will be addressed in the next increment.

|  |  |  |
| --- | --- | --- |
| Version Numbers | Current Status of Version | A Brief Description of the change implemented in that version |
| This version | Being Designed |  |
| Example: version 2 | Example: In production. Will be retired with this release. | Example: This release added the ability to look up a person by address.  Provide a link to each version of the service. |
| Example: version 1 | Example: Retired. | Example: This release provided the base minimum functionality to look up a person by name.  Provide a link to each version of the service. |

##### Summary of Design and Platform Details

###### SOA Pattern(s) Implemented

The SOA pattern to be implemented for this service is Request/Reaction pattern.

###### COTS Platform vendor names and versions for hosting platform

Oracle WebLogic 12c.

#### Dependencies

This service depends on local VistA OP for drug information lookup.

#### Service Design Details

This section will be addressed in the next increment.

##### Interface Technical Specs

This section will be addressed in the next increment.

###### Service Invocation Type

REST

###### Service Interface Type

URIs

###### Service Name

PrescriptionDrugs

###### Interface

This section will be addressed in the next increment.

###### End Points

This section will be addressed in the next increment.

###### Operations or Methods

This section will be addressed in the next increment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Operation Name | Inputs | Outputs | Transactional Qualities if relevant (Updating?, Atomic?, Can participate in transaction?) | Pre and Post Conditions | Exception (s) |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

###### Message Schemas

This section will be addressed in the next increment.

##### Information Model

This section will be addressed in the next increment.

###### Class Diagram and Description of Entities Involved

This section will be addressed in the next increment.

###### Mappings from ELDM to Standards Based Schemas

This section will be addressed in the next increment.

##### Behavior Model (AKA Use Case Realization)

Please see [PrescriptionDrugs](#_PrescriptionDrugs)

###### Use Cases (Use Case Model)

Describe how this service fits into the larger use case model of the consumer. You may need multiple models for multiple consumers. Focus is **not** on the internal workings of the new service instead of the calls made from external consumers. Just a summary or the Use Case Diagram may be sufficient. List the alternative and exception flows. Reference the detailed design documents via a URL.

###### Interaction Diagrams

Please see [PrescriptionDrugs](#_PrescriptionDrugs)

#### Gap Analysis

Provide a Gap Analysis (Reference) to demonstrate compliance of this service with various standards, policies, guidelines and laws. The Gap Analysis may take the form of a matrix as shown in the sample below. This will help the governance boards expedite your request.

| Design Elements🡪  Policies / SLD elements etc.↓ | Design  Element A | Design  Element B | Design  Element C | Comment for non-conformance |
| --- | --- | --- | --- | --- |
| Policy X | Match |  |  |  |
| Policy Y |  | Partial |  |  |
| Policy Z |  |  |  | Commercial encryption server in prod will have to address this policy. |
| Policy A |  |  |  | Compliance with this policy not required until next year. |
| New / Additional Features |  |  | New element minimizes manual intervention |  |

##### Variances from Enterprise Target Architecture

None at this time

##### Variances from SLDs

None at this time

##### Variances from Standards and Policies

None at this time

##### Justification for Exceptions and Mitigation

This section will list out any non-functional and functional requirements that are not being met. The non-conformance may be in violation of elements of SLDs, enterprise architecture (TRM Technology Reference Model), privacy policies or guidelines. For each exception provide:

1. Reasons for non-conformance (cost, time, technology, etc.)
2. Mitigating actions taken to reduce the impact of non-conformance
3. Plan (roadmap) to come back into conformance

This list can grow depending on what the Review bodies may ask for.

### Service Design for PharmacyTransfers

Pleasesee [PharmacyTransfers](#_PharmacyTransfers) for introduction, purpose and scope of service and service details.

#### Introduction

##### Purpose and Scope of Service

This service was described at a high level in the charter document. Please refer to it here via a link.

##### Links to Other Documents

This section will be addressed in the next increment.

Provide links to other documents created for this service so far in the SOA lifecycle. At a minimum, provide links to:

* Service Charter
* Service Roadmap
* Service Description

#### Service Details

##### Service Identification

This section will be addressed in the next increment.

| Service Attribute | Value |
| --- | --- |
| Name and Alias (if any) | Name of the service and other names for the service, which might be used by someone searching for this service. Please follow ESS naming standards. |
| Overview | Brief textual overview of the service. |
| Version | Version number of the service being described here |
| Latest Status | This field shows the latest status for the above referenced version of this service! The status of a service shows the progress of the service from initiation through development, deployment, and eventual retirement. The status also has a status date associated with the status - and we will be using the latest one here in this document. Valid values include: Inception, Design, Provisioning, Certification / Testing, Operation, Deprecated, Retired, Rejected - Owner has decided not to develop the service. |
| Service Type | Used to define applicable architecture patterns. Examples (from Open Group):  • Interaction  • Process  • Information  • Partner  • Business Application  • Access  • Service Connectivity |
| Architecture Layer | Referred to as class in VA Service template. Used to define applicable architecture patterns and relationships to governing bodies. Examples:  • Solution  • Process  • Information  • Utility  • Underlying |
| Business Domain | Business Vertical or Business Division where this service belongs. |
| Service Domain | The service or technical domain that the service belongs to. Can be used to establish the namespace. |
| Business Organization and Owner | Person who approves this service & any changes. Include email. |
| Technical Organization and Owner | Person responsible for provisioning (specifying, acquiring certifying) this service. Include email. |
| Development Organization and Owner | Person who is responsible for the development processes and activities for this service. Include email. |
| Support Organization and Owner | Person who is responsible for the support of this service while in production. Include email. |
| Target Consumer Organization(s) and Owner(s) | Organizations and/or developers roles that service is intended for. |

##### Service Versions

This section will be addressed in the next increment.

|  |  |  |
| --- | --- | --- |
| Version Numbers | Current Status of Version | A Brief Description of the change implemented in that version |
| This version | Being Designed |  |
| Example: version 2 | Example: In production. Will be retired with this release. | Example: This release added the ability to look up a person by address.  Provide a link to each version of the service. |
| Example: version 1 | Example: Retired. | Example: This release provided the base minimum functionality to look up a person by name.  Provide a link to each version of the service. |

##### Summary of Design and Platform Details

###### SOA Pattern(s) Implemented

The SOA patterns to be implemented for this service are Secured Message pattern and Transactional Service pattern.

###### COTS Platform vendor names and versions for hosting platform

Oracle WebLogic 12c.

#### Dependencies

This service depends on DAS JMS Queue to function properly.

#### Service Design Details

This section will be addressed in the next increment.

##### Interface Technical Specs

This section will be addressed in the next increment.

###### Service Invocation Type

REST

###### Service Interface Type

URIs

###### Service Name

PharmacyTransfers

###### Interface

This section will be addressed in the next increment.

###### End Points

This section will be addressed in the next increment.

###### Operations or Methods

This section will be addressed in the next increment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Operation Name | Inputs | Outputs | Transactional Qualities if relevant (Updating?, Atomic?, Can participate in transaction?) | Pre and Post Conditions | Exception (s) |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

###### Message Schemas

This section will be addressed in the next increment.

##### Information Model

This section will be addressed in the next increment.

###### Class Diagram and Description of Entities Involved

This section will be addressed in the next increment.

###### Mappings from ELDM to Standards Based Schemas

This section will be addressed in the next increment.

##### Behavior Model (AKA Use Case Realization)

Please see [PharmacyTransfers](#_PharmacyTransfers)

###### Use Cases (Use Case Model)

Describe how this service fits into the larger use case model of the consumer. You may need multiple models for multiple consumers. Focus is **not** on the internal workings of the new service instead of the calls made from external consumers. Just a summary or the Use Case Diagram may be sufficient. List the alternative and exception flows. Reference the detailed design documents via a URL.

###### Interaction Diagrams

Please see [PharmacyTransfers](#_PharmacyTransfers)

#### Gap Analysis

Provide a Gap Analysis (Reference) to demonstrate compliance of this service with various standards, policies, guidelines and laws. The Gap Analysis may take the form of a matrix as shown in the sample below. This will help the governance boards expedite your request.

| Design Elements🡪  Policies / SLD elements etc.↓ | Design  Element A | Design  Element B | Design  Element C | Comment for non-conformance |
| --- | --- | --- | --- | --- |
| Policy X | Match |  |  |  |
| Policy Y |  | Partial |  |  |
| Policy Z |  |  |  | Commercial encryption server in prod will have to address this policy. |
| Policy A |  |  |  | Compliance with this policy not required until next year. |
| New / Additional Features |  |  | New element minimizes manual intervention |  |

##### Variances from Enterprise Target Architecture

None at this time

##### Variances from SLDs

None at this time

##### Variances from Standards and Policies

None at this time

##### Justification for Exceptions and Mitigation

This section will list out any non-functional and functional requirements that are not being met. The non-conformance may be in violation of elements of SLDs, enterprise architecture (TRM Technology Reference Model), privacy policies or guidelines. For each exception provide:

1. Reasons for non-conformance (cost, time, technology, etc.)
2. Mitigating actions taken to reduce the impact of non-conformance
3. Plan (roadmap) to come back into conformance

This list can grow depending on what the Review bodies may ask for.

### Service Design for VistAErxTransmissions

Please see [VistAErxTransmissions](#_VistAErxTransmissions) for introduction, purpose and scope of service and service details.

#### Introduction

##### Purpose and Scope of Service

This service was described at a high level in the charter document. Please refer to it here via a link.

##### Links to Other Documents

This section will be addressed in the next increment.

Provide links to other documents created for this service so far in the SOA lifecycle. At a minimum, provide links to:

* Service Charter
* Service Roadmap
* Service Description

#### Service Details

##### Service Identification

This section will be addressed in the next increment.

| Service Attribute | Value |
| --- | --- |
| Name and Alias (if any) | Name of the service and other names for the service, which might be used by someone searching for this service. Please follow ESS naming standards. |
| Overview | Brief textual overview of the service. |
| Version | Version number of the service being described here |
| Latest Status | This field shows the latest status for the above referenced version of this service! The status of a service shows the progress of the service from initiation through development, deployment, and eventual retirement. The status also has a status date associated with the status - and we will be using the latest one here in this document. Valid values include: Inception, Design, Provisioning, Certification / Testing, Operation, Deprecated, Retired, Rejected - Owner has decided not to develop the service. |
| Service Type | Used to define applicable architecture patterns. Examples (from Open Group):  • Interaction  • Process  • Information  • Partner  • Business Application  • Access  • Service Connectivity |
| Architecture Layer | Referred to as class in VA Service template. Used to define applicable architecture patterns and relationships to governing bodies. Examples:  • Solution  • Process  • Information  • Utility  • Underlying |
| Business Domain | Business Vertical or Business Division where this service belongs. |
| Service Domain | The service or technical domain that the service belongs to. Can be used to establish the namespace. |
| Business Organization and Owner | Person who approves this service & any changes. Include email. |
| Technical Organization and Owner | Person responsible for provisioning (specifying, acquiring certifying) this service. Include email. |
| Development Organization and Owner | Person who is responsible for the development processes and activities for this service. Include email. |
| Support Organization and Owner | Person who is responsible for the support of this service while in production. Include email. |
| Target Consumer Organization(s) and Owner(s) | Organizations and/or developers roles that service is intended for. |

##### Service Versions

This section will be addressed in the next increment.

|  |  |  |
| --- | --- | --- |
| Version Numbers | Current Status of Version | A Brief Description of the change implemented in that version |
| This version | Being Designed |  |
| Example: version 2 | Example: In production. Will be retired with this release. | Example: This release added the ability to look up a person by address.  Provide a link to each version of the service. |
| Example: version 1 | Example: Retired. | Example: This release provided the base minimum functionality to look up a person by name.  Provide a link to each version of the service. |

##### Summary of Design and Platform Details

###### SOA Pattern(s) Implemented

The SOA patterns to be implemented for this service e are Secured Message pattern and Request/Reaction pattern.

###### COTS Platform vendor names and versions for hosting platform

Oracle WebLogic 12c.

#### Dependencies

This service depends on VistA OP RPC for its operation.

#### Service Design Details

This section will be addressed in the next increment.

##### Interface Technical Specs

This section will be addressed in the next increment.

###### Service Invocation Type

REST

###### Service Interface Type

URIs

###### Service Name

VistAErxTransmissions

###### Interface

This section will be addressed in the next increment.

###### End Points

This section will be addressed in the next increment.

###### Operations or Methods

This section will be addressed in the next increment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Operation Name | Inputs | Outputs | Transactional Qualities if relevant (Updating?, Atomic?, Can participate in transaction?) | Pre and Post Conditions | Exception (s) |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

###### Message Schemas

This section will be addressed in the next increment.

##### Information Model

This section will be addressed in the next increment.

###### Class Diagram and Description of Entities Involved

This section will be addressed in the next increment.

###### Mappings from ELDM to Standards Based Schemas

This section will be addressed in the next increment.

##### Behavior Model (AKA Use Case Realization)

Please see [VistAErxTransmissions](#_VistAErxTransmissions)

###### Use Cases (Use Case Model)

Describe how this service fits into the larger use case model of the consumer. You may need multiple models for multiple consumers. Focus is **not** on the internal workings of the new service instead of the calls made from external consumers. Just a summary or the Use Case Diagram may be sufficient. List the alternative and exception flows. Reference the detailed design documents via a URL.

###### Interaction Diagrams

Please see [VistAErxTransmissions](#_VistAErxTransmissions)

#### Gap Analysis

Provide a Gap Analysis (Reference) to demonstrate compliance of this service with various standards, policies, guidelines and laws. The Gap Analysis may take the form of a matrix as shown in the sample below. This will help the governance boards expedite your request.

| Design Elements🡪  Policies / SLD elements etc.↓ | Design  Element A | Design  Element B | Design  Element C | Comment for non-conformance |
| --- | --- | --- | --- | --- |
| Policy X | Match |  |  |  |
| Policy Y |  | Partial |  |  |
| Policy Z |  |  |  | Commercial encryption server in prod will have to address this policy. |
| Policy A |  |  |  | Compliance with this policy not required until next year. |
| New / Additional Features |  |  | New element minimizes manual intervention |  |

##### Variances from Enterprise Target Architecture

None at this time

##### Variances from SLDs

None at this time

##### Variances from Standards and Policies

None at this time

##### Justification for Exceptions and Mitigation

This section will list out any non-functional and functional requirements that are not being met. The non-conformance may be in violation of elements of SLDs, enterprise architecture (TRM Technology Reference Model), privacy policies or guidelines. For each exception provide:

1. Reasons for non-conformance (cost, time, technology, etc.)
2. Mitigating actions taken to reduce the impact of non-conformance
3. Plan (roadmap) to come back into conformance

This list can grow depending on what the Review bodies may ask for.

### Service Design for VistAErxStatuses

Please see [VistAErxStatuses](#_VistAErxStatuses) for introduction, purpose and scope of service and service details.

#### Introduction

##### Purpose and Scope of Service

This service was described at a high level in the charter document. Please refer to it here via a link.

##### Links to Other Documents

This section will be addressed in the next increment.

Provide links to other documents created for this service so far in the SOA lifecycle. At a minimum, provide links to:

* Service Charter
* Service Roadmap
* Service Description

#### Service Details

##### Service Identification

This section will be addressed in the next increment.

| Service Attribute | Value |
| --- | --- |
| Name and Alias (if any) | Name of the service and other names for the service, which might be used by someone searching for this service. Please follow ESS naming standards. |
| Overview | Brief textual overview of the service. |
| Version | Version number of the service being described here |
| Latest Status | This field shows the latest status for the above referenced version of this service! The status of a service shows the progress of the service from initiation through development, deployment, and eventual retirement. The status also has a status date associated with the status - and we will be using the latest one here in this document. Valid values include: Inception, Design, Provisioning, Certification / Testing, Operation, Deprecated, Retired, Rejected - Owner has decided not to develop the service. |
| Service Type | Used to define applicable architecture patterns. Examples (from Open Group):  • Interaction  • Process  • Information  • Partner  • Business Application  • Access  • Service Connectivity |
| Architecture Layer | Referred to as class in VA Service template. Used to define applicable architecture patterns and relationships to governing bodies. Examples:  • Solution  • Process  • Information  • Utility  • Underlying |
| Business Domain | Business Vertical or Business Division where this service belongs. |
| Service Domain | The service or technical domain that the service belongs to. Can be used to establish the namespace. |
| Business Organization and Owner | Person who approves this service & any changes. Include email. |
| Technical Organization and Owner | Person responsible for provisioning (specifying, acquiring certifying) this service. Include email. |
| Development Organization and Owner | Person who is responsible for the development processes and activities for this service. Include email. |
| Support Organization and Owner | Person who is responsible for the support of this service while in production. Include email. |
| Target Consumer Organization(s) and Owner(s) | Organizations and/or developers roles that service is intended for. |

##### Service Versions

This section will be addressed in the next increment.

|  |  |  |
| --- | --- | --- |
| Version Numbers | Current Status of Version | A Brief Description of the change implemented in that version |
| This version | Being Designed |  |
| Example: version 2 | Example: In production. Will be retired with this release. | Example: This release added the ability to look up a person by address.  Provide a link to each version of the service. |
| Example: version 1 | Example: Retired. | Example: This release provided the base minimum functionality to look up a person by name.  Provide a link to each version of the service. |

##### Summary of Design and Platform Details

###### SOA Pattern(s) Implemented

The SOA pattern to be implemented for this service is Transactional Service pattern.

###### COTS Platform vendor names and versions for hosting platform

Oracle WebLogic 12c.

#### Dependencies

This service depends on DAS OutboundErxMessages proxy service and eRx database to function properly.

#### Service Design Details

This section will be addressed in the next increment.

##### Interface Technical Specs

This section will be addressed in the next increment.

###### Service Invocation Type

REST

###### Service Interface Type

URIs

###### Service Name

VistAErxStatuses

###### Interface

This section will be addressed in the next increment.

###### End Points

This section will be addressed in the next increment.

###### Operations or Methods

This section will be addressed in the next increment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Operation Name | Inputs | Outputs | Transactional Qualities if relevant (Updating?, Atomic?, Can participate in transaction?) | Pre and Post Conditions | Exception (s) |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

###### Message Schemas

This section will be addressed in the next increment.

##### Information Model

This section will be addressed in the next increment.

###### Class Diagram and Description of Entities Involved

This section will be addressed in the next increment.

###### Mappings from ELDM to Standards Based Schemas

This section will be addressed in the next increment.

##### Behavior Model (AKA Use Case Realization)

Please see [VistAErxStatuses](#_VistAErxStatuses)

###### Use Cases (Use Case Model)

Describe how this service fits into the larger use case model of the consumer. You may need multiple models for multiple consumers. Focus is **not** on the internal workings of the new service instead of the calls made from external consumers. Just a summary or the Use Case Diagram may be sufficient. List the alternative and exception flows. Reference the detailed design documents via a URL.

###### Interaction Diagrams

Please see [VistAErxStatuses](#_VistAErxStatuses)

#### Gap Analysis

Provide a Gap Analysis (Reference) to demonstrate compliance of this service with various standards, policies, guidelines and laws. The Gap Analysis may take the form of a matrix as shown in the sample below. This will help the governance boards expedite your request.

| Design Elements🡪  Policies / SLD elements etc.↓ | Design  Element A | Design  Element B | Design  Element C | Comment for non-conformance |
| --- | --- | --- | --- | --- |
| Policy X | Match |  |  |  |
| Policy Y |  | Partial |  |  |
| Policy Z |  |  |  | Commercial encryption server in prod will have to address this policy. |
| Policy A |  |  |  | Compliance with this policy not required until next year. |
| New / Additional Features |  |  | New element minimizes manual intervention |  |

##### Variances from Enterprise Target Architecture

None at this time

##### Variances from SLDs

None at this time

##### Variances from Standards and Policies

None at this time

##### Justification for Exceptions and Mitigation

This section will list out any non-functional and functional requirements that are not being met. The non-conformance may be in violation of elements of SLDs, enterprise architecture (TRM Technology Reference Model), privacy policies or guidelines. For each exception provide:

1. Reasons for non-conformance (cost, time, technology, etc.)
2. Mitigating actions taken to reduce the impact of non-conformance
3. Plan (roadmap) to come back into conformance

This list can grow depending on what the Review bodies may ask for.

### Service Design for ClinicalExchangePharmacyInformation

Please see [ClinicalExchangePharmacyInformation](#_ClinicalExchangePharmacyInformation) for introduction, purpose and scope of service and service details.

#### Introduction

##### Purpose and Scope of Service

This service was described at a high level in the charter document. Please refer to it here via a link.

##### Links to Other Documents

This section will be addressed in the next increment.

Provide links to other documents created for this service so far in the SOA lifecycle. At a minimum, provide links to:

* Service Charter
* Service Roadmap
* Service Description

#### Service Details

##### Service Identification

This section will be addressed in the next increment.

| Service Attribute | Value |
| --- | --- |
| Name and Alias (if any) | Name of the service and other names for the service, which might be used by someone searching for this service. Please follow ESS naming standards. |
| Overview | Brief textual overview of the service. |
| Version | Version number of the service being described here |
| Latest Status | This field shows the latest status for the above referenced version of this service! The status of a service shows the progress of the service from initiation through development, deployment, and eventual retirement. The status also has a status date associated with the status - and we will be using the latest one here in this document. Valid values include: Inception, Design, Provisioning, Certification / Testing, Operation, Deprecated, Retired, Rejected - Owner has decided not to develop the service. |
| Service Type | Used to define applicable architecture patterns. Examples (from Open Group):  • Interaction  • Process  • Information  • Partner  • Business Application  • Access  • Service Connectivity |
| Architecture Layer | Referred to as class in VA Service template. Used to define applicable architecture patterns and relationships to governing bodies. Examples:  • Solution  • Process  • Information  • Utility  • Underlying |
| Business Domain | Business Vertical or Business Division where this service belongs. |
| Service Domain | The service or technical domain that the service belongs to. Can be used to establish the namespace. |
| Business Organization and Owner | Person who approves this service & any changes. Include email. |
| Technical Organization and Owner | Person responsible for provisioning (specifying, acquiring certifying) this service. Include email. |
| Development Organization and Owner | Person who is responsible for the development processes and activities for this service. Include email. |
| Support Organization and Owner | Person who is responsible for the support of this service while in production. Include email. |
| Target Consumer Organization(s) and Owner(s) | Organizations and/or developers roles that service is intended for. |

##### Service Versions

This section will be addressed in the next increment.

|  |  |  |
| --- | --- | --- |
| Version Numbers | Current Status of Version | A Brief Description of the change implemented in that version |
| This version | Being Designed |  |
| Example: version 2 | Example: In production. Will be retired with this release. | Example: This release added the ability to look up a person by address.  Provide a link to each version of the service. |
| Example: version 1 | Example: Retired. | Example: This release provided the base minimum functionality to look up a person by name.  Provide a link to each version of the service. |

##### Summary of Design and Platform Details

###### SOA Pattern(s) Implemented

The SOA pattern to be implemented for this service is Request/Reaction pattern.

###### COTS Platform vendor names and versions for hosting platform

Oracle WebLogic 12c.

#### Dependencies

This service depends on eRx database to function properly.

#### Service Design Details

This section will be addressed in the next increment.

##### Interface Technical Specs

This section will be addressed in the next increment.

###### Service Invocation Type

REST

###### Service Interface Type

URIs

###### Service Name

ClinicalExchangePharmacyInformation

###### Interface

This section will be addressed in the next increment.

###### End Points

This section will be addressed in the next increment.

###### Operations or Methods

This section will be addressed in the next increment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Operation Name | Inputs | Outputs | Transactional Qualities if relevant (Updating?, Atomic?, Can participate in transaction?) | Pre and Post Conditions | Exception (s) |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

###### Message Schemas

This section will be addressed in the next increment.

##### Information Model

This section will be addressed in the next increment.

###### Class Diagram and Description of Entities Involved

This section will be addressed in the next increment.

###### Mappings from ELDM to Standards Based Schemas

This section will be addressed in the next increment.

##### Behavior Model (AKA Use Case Realization)

Please see [ClinicalExchangePharmacyInformation](#_ClinicalExchangePharmacyInformation)

###### Use Cases (Use Case Model)

Describe how this service fits into the larger use case model of the consumer. You may need multiple models for multiple consumers. Focus is **not** on the internal workings of the new service instead of the calls made from external consumers. Just a summary or the Use Case Diagram may be sufficient. List the alternative and exception flows. Reference the detailed design documents via a URL.

###### Interaction Diagrams

Please see [ClinicalExchangePharmacyInformation](#_ClinicalExchangePharmacyInformation)

#### Gap Analysis

Provide a Gap Analysis (Reference) to demonstrate compliance of this service with various standards, policies, guidelines and laws. The Gap Analysis may take the form of a matrix as shown in the sample below. This will help the governance boards expedite your request.

| Design Elements🡪  Policies / SLD elements etc.↓ | Design  Element A | Design  Element B | Design  Element C | Comment for non-conformance |
| --- | --- | --- | --- | --- |
| Policy X | Match |  |  |  |
| Policy Y |  | Partial |  |  |
| Policy Z |  |  |  | Commercial encryption server in prod will have to address this policy. |
| Policy A |  |  |  | Compliance with this policy not required until next year. |
| New / Additional Features |  |  | New element minimizes manual intervention |  |

##### Variances from Enterprise Target Architecture

None at this time

##### Variances from SLDs

None at this time

##### Variances from Standards and Policies

None at this time

##### Justification for Exceptions and Mitigation

This section will list out any non-functional and functional requirements that are not being met. The non-conformance may be in violation of elements of SLDs, enterprise architecture (TRM Technology Reference Model), privacy policies or guidelines. For each exception provide:

1. Reasons for non-conformance (cost, time, technology, etc.)
2. Mitigating actions taken to reduce the impact of non-conformance
3. Plan (roadmap) to come back into conformance

This list can grow depending on what the Review bodies may ask for.

# External System Interface Design

## Interface Architecture

The Inbound eRx Processing Hub will interface with a “clearinghouse” system hosted outside of the VA network in order to facilitate eRx message retrieval and delivery between non-VA prescribers and the VA Pharmacies. Change Healthcare (currently in the process of a name change to Change Healthcare) is the clearinghouse provider used for this purpose in the current design. Change Healthcare provides commercial ePrescribing solutions, and for the purposes of the Inbound eRx implementation serves as a gateway to all ePrescribing providers nationwide. The Change Healthcare system will interface to the Inbound eRx Processing Hub through DAS as shown in Section 4.3 Network Architecture. DAS and Change Healthcare will communicate using https requests over a secured network.

## Interface Detailed Design

Below is the description of the Change Healthcare connection, authentication, authorization and interface specifications. Recognizing the PHI and PII nature of the messages on its network, Change Healthcare will support and adhere to standard VA guidelines on message encryption and service authentication/authorization.

### Connectivity Methods

The recommended network connection method for eRx Network is HTTPS (an SSL Certificate is required). HTTPS is reliable and widely used for the exchange of information over TCP/IP.

### SSL Certificate

An SSL certificate/Fully qualified Domain Name is required if the connection will be made via a public HTTPS connection. The SSL certificate must meet the following guidelines:

1. The SSL certificate must have at least 1024-but key and must be issued from a trusted certificate authority (GeoTrust, GlobalSign etc.).
2. The SSL certificate cannot be self-signed.

### Timeout Limits

Timeout limits (maximum wait times) for:

* Submitted transactions – 30 seconds before timeout.
* Received transactions – 27 seconds before timeout.

### Two-way Mutual Certification Authentication

Change Healthcare will support mutual authentication between DAS and the Change Healthcare servers.

### Network Communications Header

Transactions sent across eRx Network must use the eRx Network Communications Header format to allow Change Healthcare to properly identify and route ePrescribing transactions.

This protocol can be either synchronous or asynchronous. It is possible to send each transaction via a separate socket connection if desired.

### eRx XML Wrapper

Transactions sent across eRx Network require a standard eRx XML wrapper. For XML transactions, the wrapper information is included in the XML header.

Snippet of XML Wrapper Code :

<?xml version="1.0" encoding="UTF-8"?>

<Message xmlns="https://erxpad.erxnetwork.com">

<Header>

<To/>

<From/>

<MessageID/>

<RelatesToMessageID/>

<SentTime/>

<Security/>

<Mailbox/>

<TestMessage/>

<DigitalSignature>

<DigestValue/>

<SignatureValue/>

<X509Data/>

</DigitalSignature>

</Header>

<erx>

<Application/>

<SendingEntity/>

<SendingEntityType/>

<ERXMessageID/>

<BodyType/>

<BodyVersion/>

<MessageType/>

<ControlledDrug/>

<CustomerID/>

</erx>

<Body/>

</Message>

The eRx XML wrapper consists of these elements:

Table : Header Segment

| Element | Description |
| --- | --- |
| <To> | Identifies the intended receiver of the message. Usually a pharmacy or prescriber.  **SCRIPT Reference: UIB-070-01 for EDIFACT**  **XML Reference: Header>To** |
| <From> | Identifies the sender of the message. Usually a pharmacy or prescriber. **SCRIPT Reference: UIB-060-01 for EDIFACT**  **XML Reference: Header>From** |
| <MessageID> | Unique message identifier used to track messages and identify duplicate messages.  **SCRIPT Reference: UIB-030-01 for EDIFACT**  **XML Reference: Header>MessageID** |
| <RelatesToMessageID> | Identifies previously sent messages related to the message being sent.  **SCRIPT Reference: UIB-030-02 for EDIFACT**  **XML Reference: Header>RelatesToMessageID** |
| <SentTime> | Date and time message was sent.  Expressed as an XML UTC time string.  Example: 2007-01-01T11:12:13Z  **SCRIPT Reference: UIB-080-01 and UIB-080-02 for EDIFACT**  **XML Reference: Header>SentTime** |
| <Security> | Not used. |
| <Mailbox> | Not used. |
| <TestMessage> | Indicates whether or not the message is a test.  0 = False  1 = True  **SCRIPT Reference: UIB-100 for EDIFACT**  **XML Reference: Header>TestMessage** |
| <DigitalSignature> | EPCS only – Digital code used to authenticate the identity of the sender.  The DigitalSignature is made up of these elements:  DigestValue  SignatureValue  X509Data  The data contained in each of these elements must be base64 encoded.  For additional EPCS and Digital Signature information and suggestions, see Steps for Creating a Digital Signature. |
| <DigestValue> | EPCS only – All required message fields encoded. |
| <SignatureValue> | EPCS only – DigestValue that has been signed with a Private Key, base64 encoded |
| <X509Data> | EPCS only – Base64-encoded raw bytes of the X509 certificate (which contains the Public Key related to the Private Key in the SignatureValue element). |

Table : eRx Segment

|  |  |
| --- | --- |
| Element | Description |
| <Application> | Set to “ESCRIPTS” |
| <SendingEntity> | Agreed upon value to be obtained by eRx. |
| <SendingEntityType> | Set to type of entity sending the message:  Pharmacy = “PHARMACY”  Prescriber = “CLINIC” |
| <ERXMessageID> | eRx Internal Message ID – eRx Network supplies this on responses or new messages it sends. |
| <BodyType> | Defines the format of the message inside the <Body>.  Set to the appropriate message format:  EDIFACT = “SCRIPT”  XML = “XML” |
| <BodyVersion> | Defines the version of the message being sent.  Example: “10.6” or “8.1” |
| <MessageType> | Message Function  **Script Reference: UIH-010-040**  **XML reference: Body>MessageType** |
| <ControlledDrug> | Not supported |
| <CustomerID> | ID provided by the customer.  eRx returns the provided CustomerID value in responses |

Table : Body Segment

|  |  |  |
| --- | --- | --- |
| Element | Description | |
| <Body> | Contains the entire EDIFACT or XML message.  Includes the UNA, UIB, and UIH segments |

eRx Network Certification can include the following transaction types:

* NEWRX/NewRx
* REFREQ/RefillRequest
* REFRES/RefillResponse
* CANRX/CancelRx
* CANRES/CancelRxResponse
* RXCHG/RxChange
* CHGRES/RxChangeResponse
* RXFILL/RxFill
* VERIFY/Verify
* ERROR/Error (synchronous and asynchronous)
* STATUS/Status
* GETMSG/GetMessage
* PASCHG/PasswordChange
* RESUPP/Resupply
* CENSUS/Census

**Sender and Receiver Fields**

The Sender and Receiver fields appear in all supported transaction types. These fields must contain specific values to ensure transactions are routed properly. The sender and receiver fields are determined and set by Change Healthcare during the implementation process.

# Human-Machine Interface

The Inbound eRx application is hosted on a WebLogic application server and provides a web-based front end for user control. This section details the design of this web-based user interface. The VistA OP enhancements will include new “roll and scroll” screens using List Manager as well as modifications to existing screens.

## Interface Design Rules

The layout of the Inbound eRx screens follows that of other VA web-based interfaces. All web pages in the eRx application must be 508 compliant. 508 compliance testing will be performed during development testing to ensure a readily 508 certified system. The System is designed to work with a screen resolution of 1280 X 1024 and to work with the Internet Explorer version 11 or greater or other TRM compliant web browsers. All the user interface mockups in this section are notional and may change in future increments.

## Inputs

All data input by the VA Pharmacists is collected from the web pages served by the eRx application. All the functionality defined for eRx is controlled via the eRx processing hub, eRx GUI screens and VistA OP. Only VA Pharmacists and VA Pharmacy Technicians authenticated with the VA’s SSOi service and authorized by the application are allowed access. Authorization is role-based and is supported by SiteMinder via SAML2. The eRx Processing Hub maintains a mapping of roles to application functions in the eRx\_Config table. Spring Security is used to implement authentication. Additionally VA Identity and Access Management (SAC) service using role based (RBAC) can be used for role based authentication.

## Outputs

All data output to the VA Pharmacists and VA Pharmacy Technicians is presented via web pages served by the eRx application and VistA OP.

## Navigation Hierarchy

The following list describes the navigation hierarchy for the eRx user interface:

* Login
* Home
* View eRx
* Start/Stop
* Track/Audit
* Reports
* Help

Note: Please refer to [Section 3.2.3](#_User_Interface_Data) for the above screens.

### Vista OP Screen

Vista OP user interface will include new “roll and scroll” screens and modification to existing screens and will be addressed in a future increment.

# Attachment A – Approval Signatures

This section is used to document the approval of the System Design Document. The review should be conducted face to face where signatures can be obtained ‘live’ during the review. If unable to conduct a face-to-face meeting then it should be held via LiveMeeting and concurrence captured during the meeting. The Scribe should add /es/name by each position cited. Example provided below.

The Business Sponsor and Project Manager are required to sign.

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

Signed: Date:

Lynn Sanders

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

Signed: Date:

EJ Marine

1. Additional Information
   1. Relationship to Other Documents and Plans

This design document was developed according to the specification and functionality as found in the following documents:

* Inbound prescribing Requirements Specification Document, Version 2.1, November 2015
* Inbound ePrescribing Business Requirements Document, Version 1.0, October 2014

All documents are available on the VA SharePoint site, and are updated as needed pending further discussions and elaboration with VA personnel.

* 1. Acronyms and Abbreviations

Relevant definitions, acronyms and abbreviations for the Inbound ePrescribing application are outlined below.

| **Term** | **Definition** |
| --- | --- |
| ANR | Automated Notification Reporting |
| API | Application Program Interface |
| ARRA | American Recovery and Reinvestment Act |
| BN | Business Need |
| BRD | Business Requirements Document |
| CBO | Central Business Office |
| CCD | Continuity of Care Document |
| CHAMPVA | Civilian Health and Medical Program of the VA |
| COTS | Commercial-Off-the-Shelf |
| CPRS | Computerized Patient Record System |
| DAS | Data Access Services |
| DME | Durable Medical Equipment |
| DoD | Department of Defense |
| E&E | Eligibility And Enrollment |
| EDES | Emergency Department Encounter Summary |
| EES | Employee Education System |
| EHR | Electronic Health Record |
| EAR | Enterprise Archive, A file format used by Java EE for packaging one or more modules. |
| EMR | Electronic Medical Record |
| ENTR | Enterprise Requirement |
| ePrescribing | Electronic Prescribing |
| eRx | Electronic Prescribing or Electronic Prescription |
| FB | Fee Basis |
| FIPS | Federal Information Processing Standard |
| FTE | Full Time Equivalent |
| FY | Fiscal Year |
| GUI | Graphical User Interface |
| HARB | Health Architecture Review Board |
| HDR | Health Data Repository |
| HIPAA | Health Insurance Portability and Accountability Act |
| HITSP | Health Information Technology Standards Panel |
| HL7 | Health Level Seven |
| IEC | International Electrochemical Commission |
| IHE | Integrating the Healthcare Enterprise |
| ISO | International Organization for Standardization |
| IT | Information Technology |
| JEC | Joint Executive Council |
| JEE | Java Enterprise Edition |
| JSP | Joint Strategic Plan |
| LOINC | Logical Observation Identifiers, Names, and Codes |
| MbM | Medications by Mail |
| MOCHA | Medication Order Check Healthcare Application |
| MTF | Military Treatment Facility |
| MU | Meaningful Use |
| MVI | Master Veteran Index |
| NCHS | National Center for Health Statistics |
| NCPDP | National Council for the Prescription Drug Programs |
| NDF | National Drug File |
| NIST | National Institute of Standards and Technology |
| NISTIR | National Institute of Standards and Technology Interagency Report |
| nonf | Non-Functional Requirement |
| NSR | New Service Request |
| NTRT | New Term Rapid Turnaround |
| OIA | Office of Informatics and Analytics |
| OI&T | Office of Information and Technology |
| ONCHIT | Office of the National Coordinator for Health Information Technology |
| OP | Outpatient Pharmacy |
| OWNR | Owner Requirement |
| PBM | Pharmacy Benefits Management |
| PCS | Patient Care Services |
| PD | Product Development |
| PHIS | Pharmacy Hospital Information System |
| PHR | Personal Health Record |
| PRE | Pharmacy Re-engineering |
| RDM | Requirements Development and Management |
| RED | Requirements Elaboration Document |
| ReqPro | Rational© RequisitePro |
| RSD | Requirements Specification Document |
| Rx | Prescription |
| SDS | Standard Data Services |
| SIM | Strategic Investment Management |
| SLA | Service Level Agreement |
| SME | Subject Matter Expert |
| SNOMED CT | Systematized Nomenclature of Medicine Clinical Terms |
| UCD | User Centered Design |
| UI | User Interface |
| VA | Department of Veterans Affairs |
| VAMC | VA Medical Center |
| VE | VistA Evolution |
| VETS | VA Enterprise Terminology Services |
| VHA | Veterans Health Administration |
| VistA | Veterans Health Information Systems and Technology Architecture |

* 1. Identification of Technology and Standards
* 2013 DoD/VA Target Health Standards Profile

<http://www.gao.gov/assets/320/315525.html>

* American Recovery and Reinvestment Act of 2009  
  <http://www.gpo.gov/fdsys/pkg/BILLS-111hr1enr/pdf/BILLS-111hr1enr.pdf>
* [Centers for Medicare and Medicaid Services  
  http://www.cms.gov/Medicare/E-Health/Eprescribing/index.html](http://www.cms.gov/Medicare/E-Health/Eprescribing/index.html)
* Department of Veterans Affairs FY2014-2020 Strategic Plan  
  [http://www.DNS /op3/docs/StrategicPlanning/VA2014-2020strategicPlan.PDF](http://www.DNS   /op3/docs/StrategicPlanning/VA2014-2020strategicPlan.PDF)
* National Center for Health Statistics (NCHS) Data Brief No. 143, January 2014  
  <http://www.cdc.gov/nchs/data/databriefs/db143.htm>
* NCPDP Prescription Transfer Standard 3.2  
  <http://www.ncpdp.org/Standards/Standards-Info>
* NCPDP SCRIPT Standard 10.6  
  [http://DNS infoshare.DNS /sites/vapharmacyinformatics/DEA\_EPCS/Shared%20Documents/SCRIPT%2010.6/Script\_imp\_guide\_v10.6.pdf](http://DNS  infoshare.DNS   /sites/vapharmacyinformatics/DEA_EPCS/Shared%20Documents/SCRIPT%2010.6/Script_imp_guide_v10.6.pdf)
* NSR #20130905 VistA Evolution  
  [http://vista.med.DNS /nsrd/ViewITRequestNSR.asp?RequestID=20130905](http://vista.med.DNS   /nsrd/ViewITRequestNSR.asp?RequestID=20130905)
* One-VA Enterprise Architecture Enterprise Technical Architecture  
  [http://DNS ea.oit.DNS /wp-content/uploads/2014/04/OneVA\_EA-\_ETA\_Compliance\_v3\_03312014.pdf](http://DNS  ea.oit.DNS   /wp-content/uploads/2014/04/OneVA_EA-_ETA_Compliance_v3_03312014.pdf)
* One-VA Technical Reference Model  
  http://DNS .DNS /
* VA Handbook 6500 – Information Security Program  
  [http://vaww1.DNS /vapubs/viewPublication.asp?Pub\_ID=638&FType=2](http://vaww1.DNS   /vapubs/viewPublication.asp?Pub_ID=638&FType=2)
* VA Medications by Mail Brochure  
  [http://www.DNS /hac/forbeneficiaries/meds/brochure/MbMBrochure.pdf](http://www.DNS   /hac/forbeneficiaries/meds/brochure/MbMBrochure.pdf)
* VHA’s Defining Excellence in the 21st Century  
  [https://DNS portal.DNS /sites/VHACOMMUNICATION/VHA%20Reorganization/Pages/default.aspx](https://DNS  portal.DNS   /sites/VHACOMMUNICATION/VHA%20Reorganization/Pages/default.aspx)
* VHA Handbook 1108.05 Outpatient Pharmacy Services  
  [http://DNS infoshare.DNS /sites/vapharmacyinformatics/Prro/PBMpolicy/VHA%20Handbooks/VHA%20Handbook%201108.05,%20Outpatient%20Pharmacy%20Services.pdf](http://DNS  infoshare.DNS   /sites/vapharmacyinformatics/Prro/PBMpolicy/VHA%20Handbooks/VHA%20Handbook%201108.05,%20Outpatient%20Pharmacy%20Services.pdf)

The tools to be utilized for the implementation of the Inbound eRx solution are all verified as being on the current TRM approved list.

| **Technology/Software/Tool** | **TRM** |
| --- | --- |
| Angular (version 1.3.x) | http://www.DNS /TRM/ToolPage.asp?tid=7842# |
| Apache CXF (version 3.0.4) | http://www.DNS /TRM/ToolPage.asp?tid=6933# |
| Apache HTTP Server (version 2.2) | http://www.DNS /TRM/ToolPage.asp?tid=5009# |
| Apache log4j (version v2.3) | http://www.DNS /TRM/ToolPage.asp?tid=15# |
| Apache Maven (version 3.3) | http://www.DNS /TRM/ToolPage.asp?tid=1101# |
| Apache Tiles (version 3.0) | http://www.DNS /TRM/ToolPage.asp?tid=6240# |
| Bootstrap (version 3.2) | http://www.DNS /TRM/ToolPage.asp?tid=7795# |
| Dojo Toolkit (version 1.10) | http://www.DNS /TRM/ToolPage.asp?tid=6250# |
| Eclipse Luna (version 4.4) | http://www.DNS /TRM/ToolPage.asp?tid=6316# |
| Hibernate ORM (version 4.3) | http://www.DNS /TRM/ToolPage.asp?tid=13# |
| Hibernate Validator (version 5.1) | http://www.DNS /TRM/ToolPage.asp?tid=6260# |
| HP Fortify (version 4.3) | http://www.DNS /TRM/ToolPage.asp?tid=6429# |
| Java 7 and Java 8 | http://www.DNS /TRM/StandardPage.asp?tid=22# |
| Oracle Database 11g | http://www.DNS /TRM/ToolPage.asp?tid=9# |
| Oracle SQL Developer (version 4.x) | http://www.DNS /TRM/ToolPage.asp?tid=5850# |
| Oracle WebLogic 12c (version 12.1) | http://www.DNS /TRM/ToolPage.asp?tid=7# |
| Pentaho Data Integration (version 6.0) | http://www.DNS /TRM/ToolPage.asp?tid=6653# |
| Quest TOAD Development Suite for Oracle (version 12.6) | http://www.DNS /TRM/ToolPage.asp?tid=6382# |
| Slf4j (version 1.7) | http://www.DNS /TRM/ToolPage.asp?tid=7735# |
| SoapUI (version 5.1) | http://www.DNS /TRM/ToolPage.asp?tid=6252# |
| Spring Framework version (version 4.1) | http://www.DNS /TRM/ToolPage.asp?tid=12# |
| Spring Security 4.0.x | http://www.DNS /TRM/ToolPage.asp?tid=1307# |

**The following tools and software may be used pending future TRM approval:**

* Pentaho Data Integration (version 6.0)
* Eclipse Mars (version 4.5)
* Oracle WebLogic 12c (version 12.3)
* Oracle Database 12c (version 12.1)
  1. Constraining Policies, Directives and Procedures

This design document is based on the system design document template standardized by the VA.

From the corresponding Business Requirements Document (BRD) for this effort, development of the Inbound ePrescribing system has the following constraints:

Joint health standards profile defined by the DoD/VA Health Architecture Review Board (HARB) in support of the DoD/VA Joint Executive Council (JEC) Joint Strategic Plan (JSP)

NCPDP SCRIPT Standard 10.6 or subsequent current version

NCPDP Prescription Transfer Standard 3.2 or subsequent current version

Utilizing DAS as a gateway for incoming prescription data and outgoing statuses to Change Healthcare

* 1. Requirements Traceability Matrix

As the RTM is a living document, an updated copy will be posted to VA SharePoint. The latest copy is here: [http://DNS DNS .DNS /projects/pre/PRE\_Inb\_eRx/Shared%20Documents/Requirement/PRE%20Inbound%20ePrescribing%20Requirements%20Traceability%20Matrix%20V1.1.xlsx](http://DNS  DNS       .DNS   /projects/pre/PRE_Inb_eRx/Shared%20Documents/Requirement/PRE%20Inbound%20ePrescribing%20Requirements%20Traceability%20Matrix%20V1.1.xlsx)

A Hard copy will be distributed upon request.

* 1. Packaging and Installation

Application code targeted for the WebLogic Server on the Inbound eRx Processing Hub will conform to the standard JEE specification for resources run inside a packaged EAR file. The EAR file will package one or more JEE modules into a single module to align class loading and deployment into a single server. VistA OP packaging and installation will conform to standardized VA practices details of which will be provided in a future increment.

* 1. Design Metrics

All software development by Inbound ePrescribing Developers shall conform to technology standards as defined in the VA Technical Reference Model (TRM).

Inbound ePrescribing will be developed and maintained using industry standard technologies. This project will add functionality that complements the existing VistA OP package. All necessary security protocols and requirements necessary for interacting with this package as well as others are outlined in the SDD and RSD.

1. [Meds by Mail Brochure](http://www.DNS   /hac/forbeneficiaries/meds/brochure/MbMBrochure.pdf) [↑](#footnote-ref-2)
2. A thorough history of all regular medication use (prescribed and non-prescribed) by an individual [↑](#footnote-ref-3)