Joint Legacy Viewer (JLV) 2.5.2

System Design Document (SDD)

April 2017

Version 1.2

Department of Veterans Affairs

Revision History

Date	Version	Description	Author
04/07/2017	1.2	Document approved by reviewers	AbleVets
04/06/2017	1.2	Submitted for review three with client comment addressed	AbleVets
04/04/2017	1.1	Submitted for final review, with client comments addressed	AbleVets
03/13/2017	1.0	Draft submitted for review one	AbleVets
01/12/2017	0.1	Initial draft of the document	AbleVets

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.

Table of Contents

Table of Figures

Table of Tables

1. Introduction

JLV is a centrally hosted, java-based web application that is managed as a single code baseline, and deployed in separate Department of Defense (DoD) and Department of Veteran Affairs (VA) environments. JLV is a browser-based, graphical user interface (GUI) that provides an integrated, read-only view of Electronic Health Record (EHR) data from the DoD, VA, and Virtual Lifetime Electronic Record (VLER) Health Information Exchange (HIE) partners, within a single application.

The JLV GUI retrieves and displays clinical data from a number of native data sources and systems. The data is then presented to the user in a simple to use, web-based interface, through widgets. Each widget corresponds to a clinical data domain. JLV eliminates the need for VA and DoD clinicians to access separate, disparate viewers. Born from a joint DoD-VA venture called JANUS, JLV was directed by the Secretary of Defense and Secretary of Veterans Affairs in early 2013 to further support interoperability between the two departments.

JLV users can create and personalize tabs, drag and drop widgets onto tabs, sort data within a widget's columns, set date filters, and expand a widget for a detailed view of patient information. Within each widget, a circular, blue icon indicates the data retrieved is from a VA source; a square orange icon indicates that the data retrieved is from a DoD source; and a hexagonal, purple icon indicates data that the data retrieved is from VA VLER HIE partners.

1.1. Scope

For detailed information about the scope of the project, please see the *Business Requirements Document (BRD)*, available on the Technical Services Project Repository (TSPR)[1].

1.2. User Profiles

Table 1 describes the authorized JLV users and their responsibilities.

Table 1: JLV User Descriptions and Responsibilities

User	Description and Responsibilities
Veterans Health Administration (VHA) Clinician	VA clinicians access the patient EHRs.
DoD Clinician	DoD clinicians access the patient EHRs.
Veterans Benefits Administrator (VBA)	Access patient EHRs to assist in Veterans benefit adjudication.
VA Program Staff	Access patient EHRs for administrative purposes.

[1] **NOTE:** Access to TSPR is restricted, and must be requested.

2. Background

The following subsections provide an overview of the JLV initiative.

2.1. Overview of the System

JLV is a patient-centric, presentation system that pulls information from disparate health care systems, in real time, for viewing in a web browser. The web application provides the ability to view specific clinical data within patients' longitudinal health records that are stored in EHR systems available to the VA, DoD, and VLER partners.

2.2. Overview of the Business Process

JLV provides clinicians the ability to view specific data within patients' longitudinal health records, stored in EHR systems available to the VA, DoD, and VLER partners. JLV delivers standard-based, integrated VA, DoD, and private sector clinical data information faster to VA clinicians, resulting in more timely, higher quality examinations. Further, it provides valuable and timely clinical information to VA administrators evaluating compensation and pension benefits.

For an overview of the business processes that JLV supports, please see the *JLV 2.5.2 Requirements Specification Document (RSD)*. All referenced documentation for JLV 2.5.2, once approved, is available on the TSPR[2].

2.3. Overview of the Significant Requirements

In accordance with the requirements outlined in the RSD and the Information System Contingency Plan (ISCP), performance and disaster recovery guidelines have been developed to ensure that the uptime for JLV is consistent with its evolving role. JLV has deployed production load balanced processing environments at both the Austin Information Technology Center (AITC) and the Philadelphia Information Technology Center (PITC). In the event of an unplanned outage at one of the locations, the other location is fully capable of processing all user requests, in order to continue operations. Current system performance has been deemed acceptable during User Acceptance Testing (UAT). In anticipation of user growth, monitoring capabilities have been deployed within the JLV production environment to ensure consistent operational performance and capacity planning.

For more detailed information, please see the *JLV 2.5.2 RSD*, the *JLV 2.5.2 Requirements Traceability Matrix (RTM)*, and the *BRD*. All referenced documentation, once approved, is available on the TSPR.

3. Conceptual Design

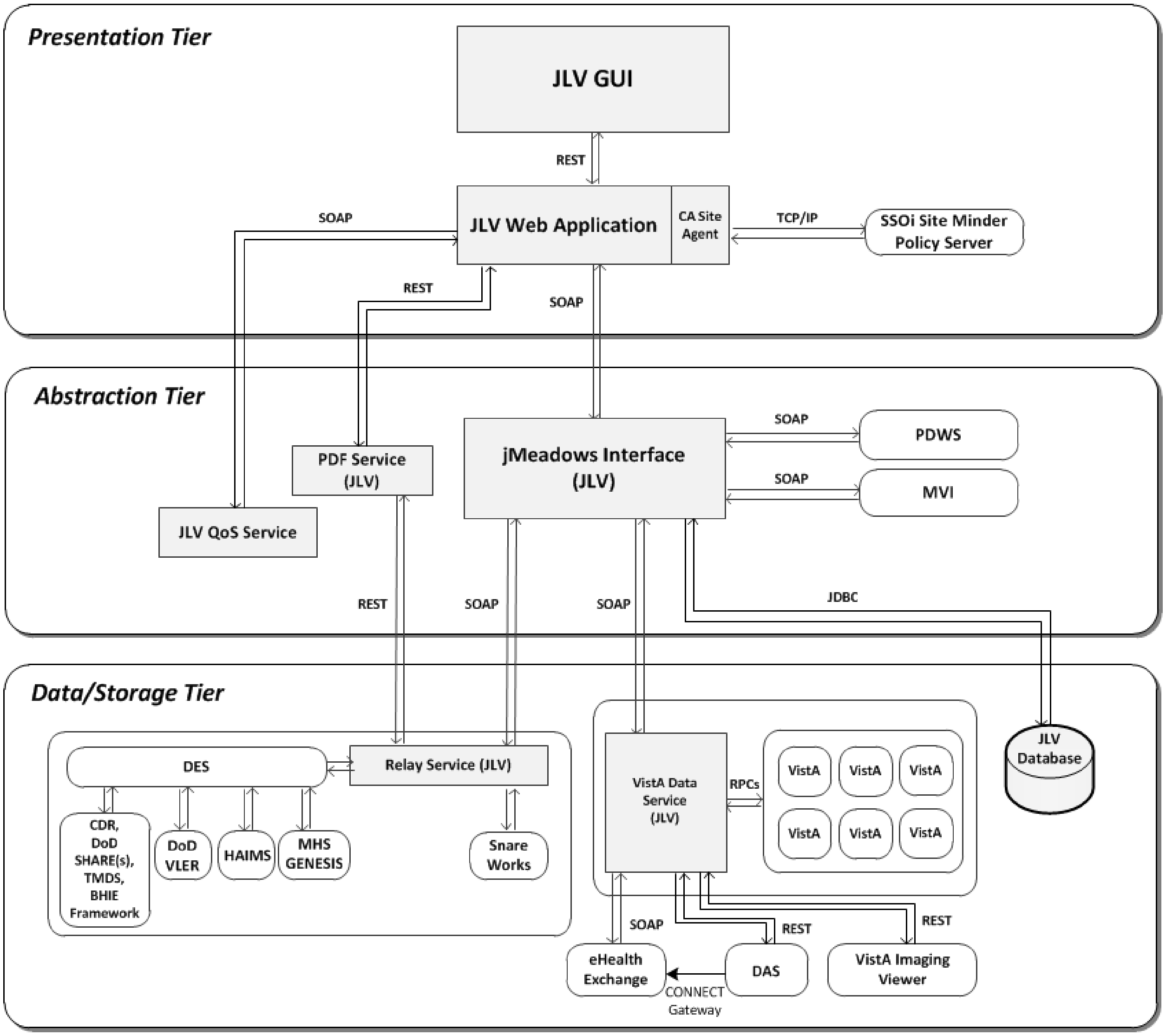
3.1. Conceptual Application Design

3.1.1. Application Context

Figure 1 provides a diagram that depicts the context of the JLV application.

[2] **NOTE:** Access to TSPR is restricted, and must be requested.

Figure 1: JLV Context Diagram



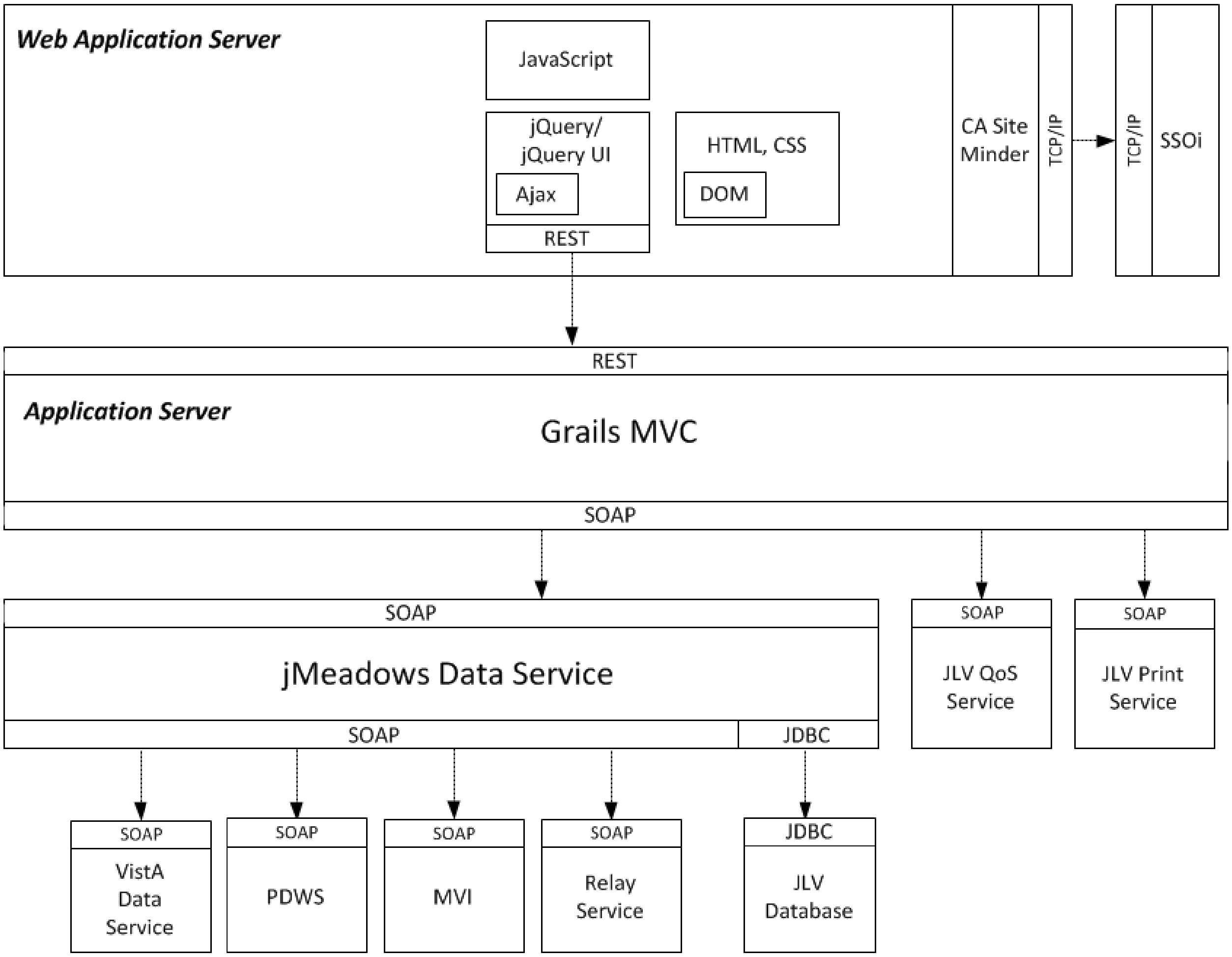
3.1.2. High-Level Application Design

Figure 2 illustrates the main components of JLV, and the messaging protocols that communicate within and between tiers in the system. JLV is a read-only GUI interface. See Section 4.2, Software Architecture, for more detailed information.

3.1.2.1. JLV GUI Framework

JLV differentiates between client side and server side technologies in its GUI framework, as seen in Figure 2.

Figure 2: JLV Client/Server Technologies in the Stack



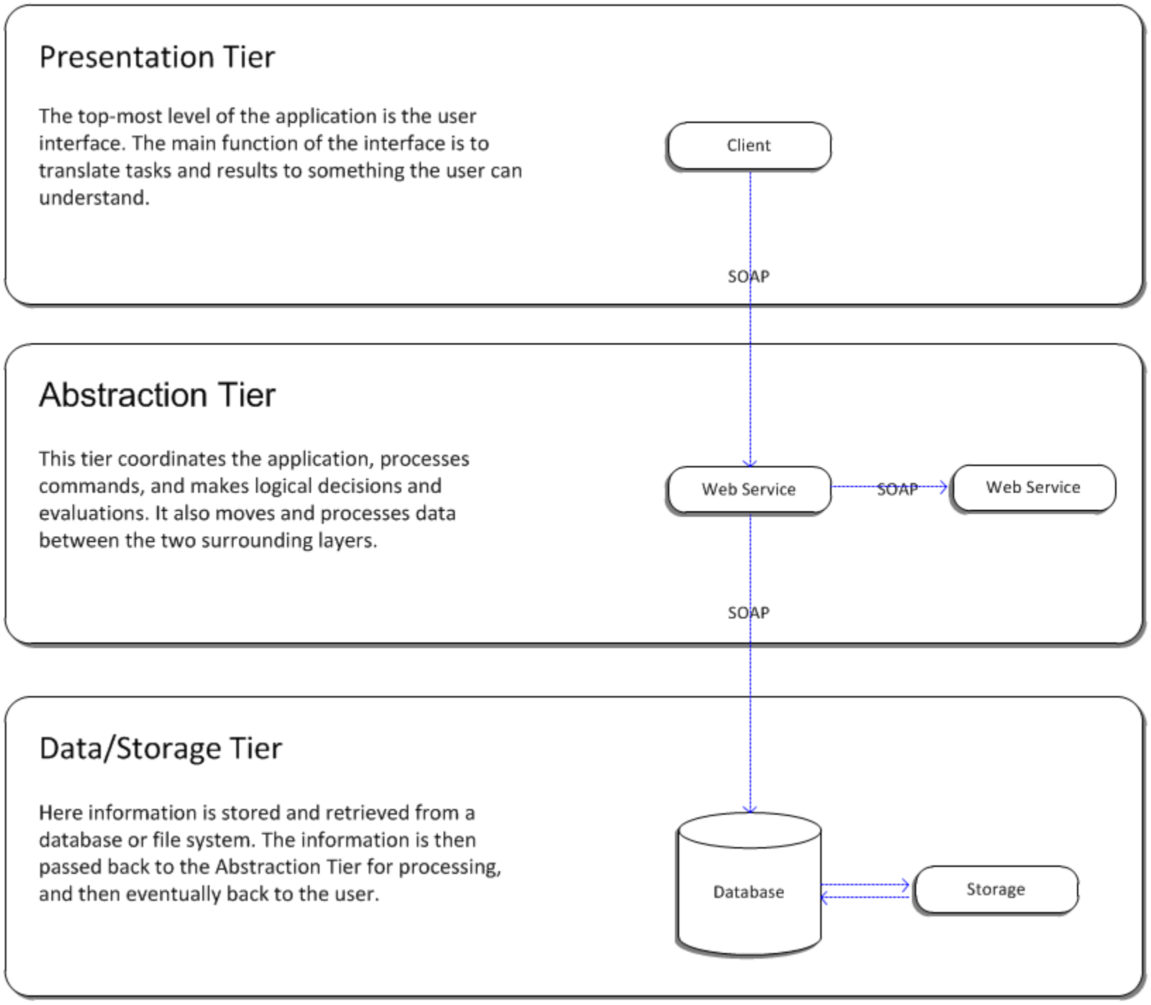
3.1.3. Application Locations

The JLV application is hosted in VA's AITC and PITC datacenters. Refer to Section 4.1, Hardware Architecture, for location and related server information.

3.1.4. System Framework Model

The JLV framework is an n-tier hierarchical model, comprised of the presentation, abstraction, and data/storage tiers, as shown in Figure 3.

Figure 3: Example of N-Tier Architecture Structure


Presentation Tier
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.
Client
SOAP
Abstraction Tier
This tier coordinates the application, processes commands, and makes logical decisions and evaluations. It also moves and processes data between the two surrounding layers.
Web Service
SOAP
Web Service
SOAP
Data/Storage Tier
Here information is stored and retrieved from a database or file system. The information is then passed back to the Abstraction Tier for processing, and then eventually back to the user.
Database
Storage

3.1.5. Architecture Tiers

Each element in the hierarchy, shown in Figure 3, has a specific set of functions and services that it offers, and a specific role to play in each tier of the design.

3.1.5.1. Presentation Tier

The presentation tier, or client tier, is the top-most level of the n-tier architecture, and is also considered the user interface. The main function of the interface is to translate tasks and results for the client. JLV provides the ability to view specific clinical data stored in any EHR systems available to the abstraction tier.

VA users must present their Personal Identification Verification (PIV) identification, Windows Active Directory (AD) Account credentials, or Kerberos credentials before gaining access to JLV. Based on the user’s credentials, jMeadows retrieves the user’s profile information from the JLV database. The user’s default host location, custom widget layout, and other user-specific data are returned.

Once users launch the presentation layer, the user is prompted to enter their credentials. JLV sends these credentials to jMeadows which then authenticates the users to their host EHR system, granting access to JLV. User authentication takes place before JLV interfaces with jMeadows.

3.1.5.2. Abstraction Tier

The abstraction, or application, tier is the tier that the presentation and the data/storage tiers use to communicate with each other. The abstraction tier moves and processes data between the presentation and the data/storage tiers. The abstraction tier coordinates the application, processes commands, and makes logical decisions and evaluations. The process of abstracting the data sources from the application takes place here.

3.1.5.3. Data/Storage Tier

The data/storage tier is where the source application's data is stored, and from where data is retrieved.

3.1.5.3.1. Data Source Interfaces

The following web services within the JLV system retrieve clinical data:

- jMeadows Data Service (jMeadows)
- VistA Data Service (VDS)
- Relay Service

3.1.5.3.2. jMeadows Data Service

The jMeadows Data Service (jMeadows) is a web service that aggregates patient and provider data for clinical domains. It is an essential component of the JLV GUI framework, which uses Java-based web services technology and request- and response-driven transactions for its web service system interfaces.

JLV utilizes Defense Manpower Data Center (DMDC), Patient Discovery Web Service (PDWS), and Master Veteran Index (MVI) for patient searches. Within the JLV system, jMeadows sends search requests, retrieves, and aggregates patient data. jMeadows uses the Simple Object Access Protocol (SOAP) version 2.0 messaging protocol to communicate with PDWS, which contains all federal employees, and provides their enterprise Federal Identification (ID) for patient lookup; the VA MVI Enterprise Central Web Service, which provides the VA enterprise unique patient identifier information; and data source interfaces, such as VDS, and the Relay Service, that are used to call each EHR system in which a patient is registered.

See the *jMeadows Data Service Interface Control Document*, submitted with this release, for complete information.

3.1.5.3.3. VistA Data Service

VDS is a web service that retrieves VA-specific clinical data from all VistA host systems at which a patient is registered, community health data from eHealth Exchange (eHX), and radiology images from VistA Imaging Exchange (VIX). To retrieve data from a VistA host system, the VistA Data Service uses remote procedure calls (RPCs) to pull clinical information. VDS also interfaces with jMeadows for VA user authentication from local VistA host systems.

To retrieve VA VLER patient data, VistA Data Service interfaces with eHX, formerly the Nationwide Health Information Network (NwHIN). VDS utilizes Data Access Service (DAS) to retrieve and parse VA VLER for integration of patient data in the Demographics, Allergies, Immunizations, Problems List, Procedures, and Vitals widgets. For VA VLER data displayed in the Community Health Summaries - VA widget, VDS does not utilize DAS, and instead queries the eHX directly.

See the *VistA Data Service Interface Control Document*, submitted with this release, for complete information.

3.1.5.3.4. Relay Service

Relay Service is a lightweight service that serves as a proxy to the Data Exchange Service (DES) and the SnareWorks Authentication Server (JLV DoD users). From DES, the Relay Service requests data from the DoD components that JLV interacts with: Armed Forces Health Longitudinal Technology Application (AHLTA)/Clinical Data Repository (CDR), Theater Medical Data Store (TMDS), Composite Health Care System (CHCS), Essentris, Healthcare Artifact and Image Management Solution (HAIMS), and DoD VLER.

See the *Relay Service Interface Control Document*, submitted with this release, for complete information.

3.1.5.3.5. Portable Document Format (PDF) Service

The PDF Service is a utility service that is utilized to print and convert documents and other clinical records that JLV receives in HTML and Rich Text Format (RTF) formats from a source data system. The service is a REST service, independent of the JLV web application, that allows users to print clinical records to PDF, or converts HTML and RTF-formatted documents into PDFs for display or printing during an active user session. The PDF Service receives a HTTP POST request for a document conversion from either the JLV web application or the Relay Service, performs the conversion, and returns the binary file (PDF). It is recommended that JLV users have the latest Adobe Reader installed on the system from which they access JLV.

3.2. Conceptual Data Design

3.2.1. Project Conceptual Data Model

Please see Section 5, Data Design, for a description of the database, including the database tables, columns, and stored procedures. Additional information about data mappings utilized for terminology normalization can be found in the *JLV Normalized Data Design Document*, submitted with this release.

3.2.2. Database Information

The JLV database is a SQL Server 2012 database and is used to store user profile information and audit records. JLV database also stores both local terminology and national standard terminology mappings for VA and DoD. The JLV database does not store patient or provider data from DoD and VA EHR systems, either long-term or temporarily.

The JLV database resides on a dedicated server within a deployed JLV environment, alongside the server hosting the JLV web application, and VDS. The JLV web application and components of the JLV system, including jMeadows, are the only components to connect and utilize the JLV database.

3.2.3. User Interface Data Mapping

JLV is made up of widgets that display clinical data. Table 2 identifies the source systems for the clinical data presented in the JLV web application. Sources for records appearing in the Documents widget are summarized in Table 3.

Table 2: JLV Data Sources

JLV Widget	Data Content & Structure	Data Sources									Bean
	Data Type LOINC | Application Program Interface (API) Version	AHLTA (CDR)	DoD or VA VLER	Essentris	MHS GENESIS	SHARE	TMDS	VA (RPC)	VistA Imaging	Federal Health Information Exchange Repository	
Admissions	Admissions 52536-0 | v4	●						●			Admission
Allergies	Allergies 48765-2 | v4	●	VA					●			Allergy
Appointments	Appointments 56446-8 | v4	●						●			Appointment
Clinical Reminders	N/A							●			Clinical Reminders
Community Health Summaries and Documents (VA)	See Table X for full list of LOINCs.		eHX								N/A
Consult Encounters	Notes – Clinical 11536-0 | v4	●					●[3]	●			Note
Demographics	Demographics 52536-0 | v4	●	DoD, VA					●			Patient Demographics
Documents See Table 3											
Immunizations	Immunizations 11369-6 | v4	●	VA					●			Immunization
Inpatient Meds Outpatient Meds	Medications 10160-0 | v4	●				●	●	●			Prescription
Inpatient Summaries	Notes - Clinical – Inpatient 28563-5 | v4					●		●			Note
Inpatient Summaries	Discharge Summaries 18842-5 | v4			●		●					Note

[3] TMDS data (outpatient notes) is synced to the CDR and passed through DES.

JLV Widget	Data Content & Structure	Data Sources									Bean
	Data Type LOINC | Application Program Interface (API) Version	AHLTA (CDR)	DoD or VA VLER	Essentris	MHS GENESIS	SHARE	TMDS	VA (RPC)	VistA Imaging	Federal Health Information Exchange Repository	
Inpatient Summaries	History & Physicals (H&P) 34117-2 | v4			●							Note
Insurance (Demographics)	Payers 48768-6 | v4	●	DoD					●			Payer
Lab Panel Results; Lab Results	Laboratories - Anatomic Pathology 26439-0 | v4	●				●	●	●			Lab Anatomic Pathology
Lab Panel Results; Lab Results	Laboratories – Chemistry 11502-2 | v4	●				●	●	●			Lab Chemistry
Lab Panel Results; Lab Results	Laboratories – Microbiology 18725-2 | v4	●				●	●	●			Lab Microbiology
MHS GENESIS	Documents 34794-8 | v4				●						N/A
Orders	Orders 46209-3 | v4	●						●			Order
Outpatient Encounters	Encounters 46240-8 | v4	●						●	●		Encounter
Problem List	Problem Lists 11450-4 | v4	●	VA					●			Problem
Procedures	Procedures 47519-4 | v4	●	VA					●	●		Procedure
Progress Notes	Notes – Clinical 11536-0 | v4	●					●	●	●		Note[4]
Progress Notes	Notes – Encounter	●						●			Note

[4] TMDS data (outpatient notes) is synced to the CDR and passed through DES.

JLV Widget	Data Content & Structure	Data Sources									Bean
	Data Type LOINC | Application Program Interface (API) Version	AHLTA (CDR)	DoD or VA VLER	Essentris	MHS GENESIS	SHARE	TMDS	VA (RPC)	VistA Imaging	Federal Health Information Exchange Repository	
	Report 34109-9 | v4										
Questionnaires and Deployment Assessments	Deployment Forms 51847-2 | v4									●	Note
Questionnaires and Deployment Assessments	Questionnaires 10187-3 | v4	●									Questionnaire
Radiology Reports	Radiology Reports 18726-0 | v4	●				●	●	●	●		RadExam
Social, Family, and Other Past Histories	Histories – Family 10157-6 | v4	●									History Family
Social, Family, and Other Past Histories	Histories - Other Past Medical 11348-0 | v4	●									History
Social, Family, and Other Past Histories	Histories – Social 29762-2 | v4	●									History Social
Vitals	Vital Signs 8716-3 | v4	●	VA							●	Vitals Panel

Table 3: Documents Widget Data Sources

Note Types	LOINC	Source		
		SHARE	Essentris	Other
Consults	11488-4	●	●	
Deployment Forms	51847-2			FHIE Repository
Encounters	46240-8			AHLTA (CDR) VA (RPC)
Evaluation and Managements -Post-Operative	34875-5	●	●	
Evaluation and Managements -Pre-Operative	34876-3	●	●	
General Inpatients	34112-3		●	
HAIMS	34794-8			HAIMS
Initial Evaluations	28636-9		●	
Notes - Clinical - Inpatient notes	28563-5	●		VA (RPC)
Notes - Clinical (Progress Note)	11536-0			AHLTA (CDR) TMDS VA (RPC) VistA Imaging
Outpatient Encounters	34108-1		●	
Procedures	28570-0		●	
Procedures	28570-1		●	
Questionnaires	10187-3			AHLTA (CDR)
Radiology Reports	18726-0	●		AHLTA (CDR) TMDS VA (RPC)
Studies	47045-0		●	
Surgical & Operative Reports	11504-8		●	
Transfer Summarizations	18761-7		●	

Table 4: LOINCs Utilized in Community Health Summaries (VA) Query to eHX

Concept Code	LOINC Component Name
Health Summary (C32 or C-CDA CCD)	
34133-9	Summarization of Episode Note
C62 Clinical Notes	
Discharge Summaries	
18842-5	Discharge Summarization Note
18761-7	Transfer Summarization Note
47046-8	Summary of Death

Concept Code	LOINC Component Name
Consults/Referrals	
11488-4	Consultation Note
34140-4	Transfer of Care Referral Note
History & Physicals/Progress Notes	
34117-2	History and physical note
47039-3	Admission History and Physical Note
11506-3	Progress Note
Results of Diagnostics Studies OR Procedure Notes	
26441-6	Cardiology Studies
26442-4	Obstetrical Studies
27895-2	Gastroenterology Endoscopy Studies
27896-0	Pulmonary Studies
28619-5	Ophthalmology/Optometry Studies
27897-8	Neuromuscular Electrophysiology Studies
28634-4	Miscellaneous Studies
47045-0	Study Report
28570-0	Procedure Note
34121-4	Interventional Procedure note
Radiology Studies	
18726-0	Radiology Studies
Pathology Reports	
27898-6	Pathology Studies
34122-2	Pathology Procedure Note
29752-3	Perioperative Records

3.2.3.1. Application Screen Interface

JLV is a read only application that does not utilize user input screens that write data to a database.

3.3. Conceptual Infrastructure Design

The conceptual infrastructure design of JLV includes the usage of two data centers, at AITC and PITC. The two geographically separate sites provide a hot site for disaster recovery.

3.3.1. System Criticality and High Availability

The key components of the JLV production infrastructure to meet system criticality and high availability include the utilization of the following:

- Distributed sites for disaster recovery between the two data centers to provide failover.
  - Manual intervention and service restart is needed.

- GTM for VDS provides fault tolerance and redundancy for VA JLV enterprise production infrastructure within the AITC and PITC data centers.
- GTM for jMeadows provides fault tolerance and redundancy for VA JLV enterprise production infrastructure within the AITC and PITC data centers.
- Cloud infrastructure that makes it easier to clone and scale the application.

3.3.2. Special Technology

Not applicable to JLV.

3.3.3. Technology Locations

The JLV application has environments at VA’s AITC and PITC datacenters. The primary operating environment is AITC, with PITC serving as the failover site.

Refer to Section 4.1, Hardware Architecture, for location and related server information.

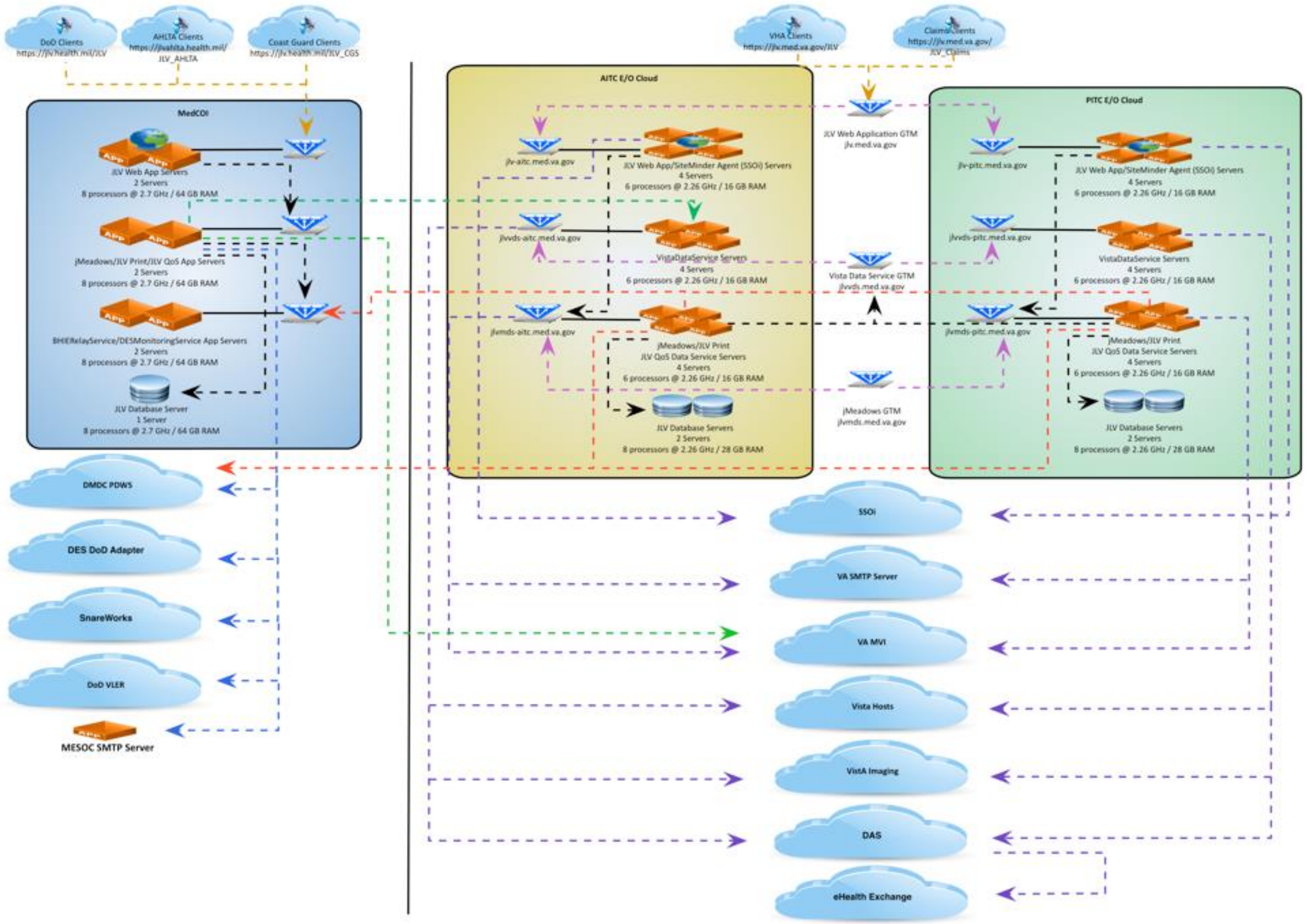
3.3.4. Conceptual Infrastructure Diagram

Refer to Figure 4.

3.3.4.1. Location of Environments and External Interfaces

Figure 4 provides an overview of the JLV production infrastructure hosted at DoD MESOC, and VA AITC and PITC.

Figure 4: JLV Production Infrastructure in the VA and DoD Environments


DoD Clients
https://jlv.health.mil/JLV
AHLTA Clients
https://jlvahlta.health.mil/JLV_AHLTA
Coast Guard Clients
https://jlv.health.mil/JLV_CGS
MedCOI
JLV Web App Servers
2 Servers
8 processors @ 2.7 GHz / 64 GB RAM
jMeadows/JLV Print/JLV QoS App Servers
2 Servers
8 processors @ 2.7 GHz / 64 GB RAM
BHIERelayService/DESMonitoringService App Servers
2 Servers
8 processors @ 2.7 GHz / 64 GB RAM
JLV Database Server
1 Server
8 processors @ 2.7 GHz / 64 GB RAM
DMDC PDWS
DES DoD Adapter
SnareWorks
DoD VLER
MESOC SMTP Server
VHA Clients
https://jlv.med.va.gov/JLV
Claims Clients
https://jlv.med.va.gov/JLV_Claims
AITC E/O Cloud
jlv-aitc.med.va.gov
JLV Web App/SiteMinder Agent (SSOi) Servers
4 Servers
6 processors @ 2.26 GHz / 16 GB RAM
jlvvds-aitc.med.va.gov
VistaDataService Servers
4 Servers
6 processors @ 2.26 GHz / 16 GB RAM
jlvmds-aitc.med.va.gov
jMeadows/JLV Print
JLV QoS Data Service Servers
4 Servers
6 processors @ 2.26 GHz / 16 GB RAM
JLV Database Servers
2 Servers
8 processors @ 2.26 GHz / 28 GB RAM
JLV Web Application GTM
jlv.med.va.gov
Vista Data Service GTM
jlvvds.med.va.gov
jMeadows GTM
jlvmds.med.va.gov
PITC E/O Cloud
jlv-pitc.med.va.gov
JLV Web App/SiteMinder Agent (SSOi) Servers
4 Servers
6 processors @ 2.26 GHz / 16 GB RAM
jlvvds-pitc.med.va.gov
VistaDataService Servers
4 Servers
6 processors @ 2.26 GHz / 16 GB RAM
jlvmds-pitc.med.va.gov
jMeadows/JLV Print
JLV QoS Data Service Servers
4 Servers
6 processors @ 2.26 GHz / 16 GB RAM
JLV Database Servers
2 Servers
8 processors @ 2.26 GHz / 28 GB RAM
SSOi
VA SMTP Server
VA MVI
Vista Hosts
VistA Imaging
DAS
eHealth Exchange

The legend for the connections shown above in Figure 4 can be seen in Table 5.

Table 5: Legend for Figure 4

Connector	Description
– – – – – (Black)	JLV server to server connections
– – – – – (Blue)	JLV MESOC MEDCOI to external DoD connections
– – – – – (Red)	JLV EO Cloud to DoD connections
– – – – – (Green)	JLV MESOC MEDCOI to VA connections
– – – – – (Purple)	JLV EO Cloud to external VA connections
– – – – – (Gold)	Client to JLV web application connections
– – – – – (Fuchsia)	VA global load balancing traffic

3.3.4.2. Conceptual Production String Diagram

Not applicable to JLV.

4. System Architecture

4.1. Hardware Architecture

This section highlights the hardware architecture for JLV 2.5.2.

4.1.1. JLV Enterprise Servers

Table 6 lists the server types and specifications for the JLV Enterprise servers hosted at AITC.

Table 6: AITC JLV Enterprise VM Server Configuration

Server Type	Server Specifications
JLV Web Application Servers	Four (4) virtual servers each with six (6) processors and 16 GB RAM
VistADataService Servers	Four (4) virtual servers each with six (6) processors and 16 GB RAM
jMeadows Service Servers	Four (4) virtual servers each with six (6) processors and 16 GB RAM
Database Servers	Two (2) virtual servers each with eight (8) processors and 28 GB RAM

Table 7 describes the server configuration for JLV Enterprise production infrastructure hosted at PITC.

Table 7: PITC JLV Enterprise VM Server Configuration

Server Type	Server Specifications
JLV Web Application Servers	Four (4) virtual servers each with six (6) processors and 16 GB RAM
VistADataService Servers	Four (4) virtual servers each with six (6) processors and 16 GB RAM

Server Type	Server Specifications
jMeadows Service Servers	Four (4) virtual servers each with six (6) processors and 16 GB RAM
Database Servers	Two (2) virtual servers each with eight (8) processors and 28 GB RAM

4.2. Software Architecture

See Figure 2 for information.

4.2.1. JLV Development Technologies

4.2.1.1. Client Side Technologies

The following technologies were utilized for the development of client-side components within the JLV system:

- Adobe Portable Document Format (PDF) (ver. 1.4) is used within the JLV system when the JLV renders XHTML data to PDF data and returns PDF data to the client system during the print process.
- Apache PDFBox library (v2.0.4) is an open source Java tool for working with PDF documents, allowing for the creation and manipulation of PDFs, as well as extraction of content from PDFs.
- Backbone.js (v1.1.2) provides structure to web applications by providing models with key-value binding and custom events, collections with a rich API of enumerable functions, views with declarative event handling, and connects to existing API over a REpresentational State Transfer (REST)ful JavaScript Object Notation (JSON) interface. The Architecture and Engineering Review Board (AERB) has granted a conditional operating waiver for use of this library.
- Cascading Style Sheets (CSS) is the language for describing the presentation (i.e., the formatting and layout) of a HyperText Markup Language (HTML) document. CSS is designed to enable the separation of document control from the details of how it should be presented, including the typography, positioning, colors, and margins. This separation improves content accessibility and provides more flexibility in controlling presentation characteristics. The application is compliant with CSS Level 3.
- eXtensible Markup Language (XML) is a set of rules for marking up documents. It is widely used to transmit arbitrarily structured data in mixed client/server environments. XML and HTML are compatible members of a family of markup languages called Standard Generalized Markup Language (SGML). HTML is an SGML language with a specific Document Object Model (DOM) focused on describing hypertext documents. The application is compliant with XML version 1.0, 5th Edition.
- Extensible Stylesheet Language (XSL) Transformations (XSLT) is a language used with XML documents to transform XML documents into other formats or objects. The application is compliant with XSLT version 3.
- Flot (v0.8.3) is a pure JavaScript plotting library for jQuery, utilized in JLV for on-screen graph displays. AERB has granted a conditional operating waiver for use of this library.

- HyperText Markup Language (HTML) is a markup language for web pages that provides a means to create structured documents using semantic tags. Images and other media objects can be embedded and can be used to create interactive forms. The application is compliant with HTML 5.
- JavaScript is an object-oriented scripting language. Although JavaScript has other uses, it is client-side JavaScript—the version that runs inside a user's browser and manipulates HTML page elements—that is being used. Client-side JavaScript is used to turn static HTML documents into interactive web applications. The application is compliant with JavaScript 5.1.
- JavaScript Object Notation (JSON) is a language-independent system for representing data objects, although it is based on JavaScript. It is simpler than XML and is often used as an alternative to XML in Ajax applications to transfer data objects between a server and a script running in a user's browser. The application is compliant with JSON 1.0.
- jQuery (v1.11.1) is a feature-rich JavaScript library and easy-to-use API that that works across a multitude of browsers and simplifies development with HTML document traversal and manipulation, event handling, animation, and Ajax.
- jQuery User Interfaces (UI) (v1.12) is a set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library.
- Lodash (v3.9.0) (a fork of the Underscore v1.8.3 library) is a JavaScript utility library that is utilized to simplify and improve JavaScript usage through a toolkit of JavaScript functions. AERB has granted a conditional operating waiver for use of this library.
- wkhtmltopdf (v0.12.4) is an open source command line tool that enables rendering of HTML into PDF and various image formats using the Qt WebKit rendering engine.

4.2.1.2. Server Side Technologies

The following technologies were utilized for the development of server-side capabilities within the JLV system:

- **Grails Model-View-Controller** (MVC) (v2.4.4) is an open source web application framework that has been designed according to the MVC paradigm. MVC is a software architectural pattern that isolates domain logic (i.e., the application logic for the user) from the user interface (i.e., input and presentation).
- **SOAP** (v2.0) is utilized as the messaging protocol used to communicate between the web services within JLV and the data source. When SOAP requests are initiated from the Grails MVC framework running on the JLV Server, the system waits for a response, as the request is synchronous. If the response is not given with a finite period of time (default is 100 seconds), the connection terminates, the user receives a connection error message, and the system will not initiate any new requests until action is taken by the user. All SOAP messages are digitally signed.
- **TextControl.NET Server** (v23 SP1) is a HTML5-based Web editor and reporting template designer for cross-platform report templates for use with the JLV PDF Service.
- **Java** is the language used for development. Java v7u99 is required for Oracle WebLogic 12c on all application servers.

4.2.2. Additional Design Considerations

The JLV system is designed to run within both IPv4- and IPv6-based environments. WebSphere Application Server and WebLogic Server are IPv4 and IPv6 compliant. JLV does not use Internet Protocol (IP) addresses in its configurations.

JLV implements session management and keeps track of a user's activity across sessions of the JLV system. Session management allows the state of application(s) that are running to be saved and remembered.

JLV implements session state management on the server side. The process of knowing the values of controls and variables is referred to as state management. Session state is server side. In session state, a special session ID is stored on the server. This session ID identifies a specific application. The session ID is assigned to the calling browser.

4.3. Network Architecture

JLV utilizes the network infrastructure provided by the AITC and PITC data centers, as displayed in Figure 4.

4.4. Service Oriented Architecture/ESS

The JLV system does not participate in a VA Enterprise Service Bus at this time.

4.5. Enterprise Architecture

See Figure 4 for a graphic representation of the Enterprise Architecture.

4.5.1. Enterprise Services

4.5.1.1. VA MVI Enterprise Central Web Service

MVI is a database that holds more than 17 million unique patient identity entries, populated from all VA facilities nationwide. MVI matches/links system records together across the VA systems, and establishes a unique enterprise identifier for each of the VA unique person records. The identifier is called an Integration Control Number (ICN).

Both MVI and the VA Authentication Federation Infrastructure (VAAFI) are managed by the Identity and Access Management (IAM) division of the VA, and provide identity and security services. MVI partners with VAAFI to provide a web service proxying and authentication capability. The web service capability is made possible by the IAM Service Oriented Architecture (SOA).

4.5.1.2. DMDC PDWS

The DMDC PDWS is a patient lookup service that acts as a retrieval mechanism for patients. This service is used by JLV to obtain a set of patients that match specific search criteria. The service returns lists of patients with a corresponding DoD Electronic Data Interchange Personal Identifier (EDIPI), used to cross-reference patients for a more complete view of the clinical information for providers. PDWS provides an external interface for a patient search based on probabilistic search/analysis capabilities.

PDWS conforms to the interface requirements for Person Search Inquiry in the HL7-based IHE Patient Demographics Query (PDQ) service, referenced by the eHX Patient Discovery Web Service specification. The PDWS implementation helps find, prevent, and resolve duplicate identities. It also helps to provide sophisticated person search capabilities to DMDC and partner applications.

4.5.1.3. DES

DES is an enterprise service that JLV uses to retrieve DoD data, DoD community partner data, and other external sources. DES retrieves data from AHLTA/CDR, TMDS, CHCS, Essentris, HAIMS, DoD VLER, MHS GENESIS, and FHIE repositories.

4.5.1.4. DAS

DAS is an enterprise service that utilizes the CONNECT Gateway, together with VA eHX, to enable the VA to share patient health data with other federal partners and private providers.

DAS is responsible for the transport and temporary storage (caching) of both structured and non-structured data between internal VA and external producers and consumers. JLV interfaces with DAS to retrieve Clinical Document Architecture (CDA) documents from VA eHX partners.

4.5.1.5. SnareWorks

The SnareWorks application is a security framework that provides data protection facilities, scalable access control, remote security management, and secure single sign-on for enterprise applications. SnareWorks utilizes the web architecture of the enterprise to ensure a secure communications medium between client and server systems. DoD JLV users will be authenticated against a SnareWorks server configured within environments where JLV is deployed.

4.5.1.6. Single Sign On Internal (SSOi)

Single Sign On Internal (SSOi) is an authentication solution for internal-facing VA applications utilized by VA employees, contractors, and volunteers. JLV receives VA user PIV authentication and user credentials from SSOi during the JLV login process to enable user access to the JLV GUI.

4.5.1.7. eHX

The VA eHX is a health information exchange network used for securely sharing clinical information over the Internet, nationwide, with VA partners. The eHX program is primarily focused on exchanging standards-based healthcare information between different health information exchanges (HIEs). eHX interfaces with the eHX Gateway, an implementation of the CONNECT Gateway.

5. Data Design

The JLV database is a relational database used to store user profile information and audit data for users of the JLV web application.

The JLV database also stores VA and DoD terminology mappings (both local terminology and national standards). The JLV database does not store, either long term or temporarily, patient or provider electronic healthcare records from DoD, VA, and VLER EHR systems. The JLV database resides in the data/storage tier (shown in Figure 1), where the source application's data is stored, and from where data is retrieved.

The following objects are data stores defined within the database schema, or contained within the JLV database:

- Tables, columns, and column variable types (detailed in Section 5.1, DBMS Files).
- Mappings for external terminology, classification, and medical data coding standards (detailed in the *Terminology Normalization Design Document*, submitted with this release).

5.1. DBMS Files

The following sections describe the group of base tables that are used throughout the JLV database. Where available, the primary key (PK) for the table is identified with PK, appearing next to the column name. Some tables do not have a primary key.

5.1.1. AUDIT Table

Table 8 describes the AUDIT table. This table is used to hold auditing information for JLV user and data access auditing capabilities.

Table 8: AUDIT Table

Column Name	Type	Description
auditID (PK)	int	Unique ID of each entry
entryDate	datetime	Date at which the audit was entered
startDate	smalldatetime	Timestamp of when the audited action began
endDate	smalldatetime	Timestamp of when the audited action ended
systemID	varchar(50)	User's login site identifier
userID	varchar(25)	User's identifier
userNPI	varchar(25)	User's identifier
userName	varchar(50)	Name of user to be recorded in audit
patID	varchar(25)	Patient's identifier
category	varchar(200)	Query action (login, select patient, patient lookup, get patient allergies)
queryType	varchar(50)	Application name
cardID	varchar(25)	User's unique PIV ID.
ipAddress	varchar(15)	IP address of the machine from which the user is logging in

5.1.2. AUTH_USER Table

Table 9 describes the AUTH_USER table. This table is the JLV whitelist; the official, master list of users authorized to access the JLV web application.

Table 9: AUTH_USER Table

Column Name	Type	Description
id (PK)	int	Sequence number to identify the user
cardId	varchar(20)	User’s smartcard ID
organization	varchar(10)	Organization of user
facility	varchar(10)	Facility
name	varchar(100)	Name of user
email	varchar(50)	Email address of user
phone	varchar(20)	Phone number of user
subjectDN	varchar(500)	Subject DN string from the user’s smartcard certificate

5.1.3. CPT_CVX Table

Table 10 describes the Current Procedural Terminology (CPT)_CVX table. This table is used to cross-reference CPT national standards to CVX national standards for immunization-related data.

Table 10: CPT_CVX Table

Column Name	Type	Description
id (PK)	int	Sequence number to identify the code
cptCode	varchar(20)	CPT code
cptDescription	varchar(300)	CPT description
cptCodeStatus	varchar(20)	CPT code status
cvxCode	varchar(20)	CVX code
cvxShortDescription	varchar(200)	CVX description (short)
cvxFullDescription	varchar(300)	CVX description (full)
vaccineStatus	varchar(20)	Vaccine status

5.1.4. CVX Table

Table 11 describes the CVX table. This table is used to hold terminology mappings and national standards for immunization data.

Table 11: CVX Table

Column Name	Type	Description
CVX_Code (PK)	varchar(50)	CVX code
CVX_Short_Description	varchar(200)	CVX description (short)
Full_Vaccine_Name	varchar(500)	Full name of vaccine

5.1.5. DoD_ALLERGIES Table

Table 12 describes the DoD_ALLERGIES table. This table is used to hold terminology mappings and national standards for allergy data.

Table 12: DoD_ALLERGIES Table

Column Name	Type	Description
id (PK)	int	Sequence number to identify allergy
chcsAllergyIEN	varchar(20)	CHCS internal identifier
chcsName	varchar(200)	Name of allergy as displayed in CHCS
dodNcid	varchar(20)	DoD NCID
mmmName	varchar(300)	Not used
dodName	varchar(300)	Not used
rxnorm	varchar(50)	RxNorm value
umlsCui	varchar(50)	UMLSCUI value

5.1.6. DoD_LABS Table

Table 13 describes the DoD_LABS table. This table is used to hold terminology mappings and national standards for laboratory data.

Table 13: DoD_LABS Table

Column Name	Type	Description
id (PK)	int	Sequence number to identify lab
dodNcid	varchar(20)	DoD NCID
dodName	varchar(500)	Not used
mmmName	varchar(500)	Not used
loinc	varchar(20)	LOINC code
loincName	varchar(300)	LOINC name

5.1.7. DoD_MEDICATIONS Table

Table 14 describes the DOD_MEDICATIONS table. This table is used to hold standardized medications terms for DoD clinical data normalization.

Table 14: DoD_MEDICATIONS Table

Column Name	Type	Description
id (PK)	int	Sequence identifier to medication
dodNcid	varchar(20)	DoD NCID
dodName	varchar(500)	Medication name
mmmName	varchar(500)	Not used
rxnorm	varchar(20)	RxNORM code

5.1.8. DoD_NOTES Table

Table 15 describes the DOD_NOTES table. This table is used to hold standardized notes terms for DoD clinical data normalization.

Table 15: DoD_NOTES Table

Column Name	Type	Description
id (PK)	int	Internal database ID
dodNcid	varchar(50)	DOD NCID
mmmName	varchar(500)	Not used
dodName	varchar(500)	Not used
loinc	varchar(50)	LOINC code
loincName	varchar(300)	LOINC name

5.1.9. DoD_PAYERS Table

Table 16 describes the DoD_PAYERS table. This table is used to hold DoD insurance data.

Table 16: DoD_PAYERS Table

Column Name	Type	Description
Domain	varchar(50)	Not used
DoD_Local_Code	varchar(50)	DoD identifier
DoD_Local_Description	varchar(50)	Not used
Standard (PK)	varchar(50)	Standard code
Standard_Code_Description	varchar(50)	Standard code description

5.1.10. DoD_RACE Table

Table 17 describes the DoD_RACE table. This table is used to hold DoD race data.

Table 17: DoD_RACE Table

Column Name	Type	Description
Domain	varchar(50)	Not used
DoD_Local_Code	varchar(50)	DoD identifier
DoD_Local_Description	varchar(50)	Not used
Standard_Code (PK)	varchar(50)	Unique identifier for Race classification
Standard_Code_Description	varchar(50)	Standard code description

5.1.11. DoD_RADIOLOGY Table

Table 18 describes the DoD_RADIOLOGY table. This table is used to hold DoD radiology data.

Table 18: DoD_RADIOLOGY Table

Column Name	Type	Description
dodNcid	varchar(20)	DoD identifier
dodName	varchar(500)	DoD description
mmmName	varchar(500)	Not used
Loinc	varchar(20)	Loinc code
loincName	varchar(300)	Loinc description
id	int	database id

5.1.12. DoD_REACTANTS Table

Table 19 describes the DoD_REACTANTS table. This table is used to hold DoD reactant data.

Table 19: DoD_REACTANTS Table

Column Name	Type	Description
Local_Code	varchar(20)	DoD identifier
Local_Description	varchar(500)	DoD description
Target_Code	varchar(20)	RxNorm code
Target_Code_Description	varchar(500)	RxNorm description

5.1.13. DRUGS Table

Table 20 describes the DRUGS table. This table is used to hold RxNorm terminology standards for prescriptions and medications.

Table 20: DRUGS Table

Column Name	Type	Description
rxNormCode	varchar(20)	RxNorm code
VUID	varchar(20)	Local VistA identifier
VistAText	varchar(300)	Local VistA description
rxNormText	varchar(300)	RxNorm description
id (PK)	int	Internal identifier

5.1.14. ENDPOINTS Table

Table 21 describes the ENDPOINTS table. This table is used to hold the information necessary to perform status checks on local SHARE endpoints.

Table 21: ENDPOINTS Table

Column Name	Type	Description
id (PK)	int	Internal identifier
siteid	int	Code to identify site

Column Name	Type	Description
protocol	varchar(100)	Type of protocol utilized to the endpoint
host	varchar(2000)	IP address or FQDN of the endpoint
port	int	Network port of the listening endpoint
status	varchar(30)	Description of the status obtained in the status check
uname	varchar(100)	Username
pword	varchar(100)	Password
sitecode	varchar(10)	Code to identify site
modality	varchar(20)	RPC modality configuration
env	varchar(30)	Environment where JLV is deployed
accessionprefix	varchar(100)	Accession prefix for PACS/MedWeb configuration
aetitle	varchar(100)	Aetitle for PACS/MedWeb configuration
timezone	varchar(5)	Time zone where environment is hosted

5.1.15. ICD9_SNOMED Table

Table 22 describes the ICD9_ Systematized Nomenclature of Medicine (SNOMED) table. This table is used to hold ICD9 to SNOMED, one to one, terminology mappings.

Table 22: ICD9_SNOMED Table

Column Name	Type	Description
id (PK)	int	Internal identifier
ICD_CODE	varchar(20)	ICD code
ICD_NAME	varchar(300)	ICD name
IS_CURRENT	smallint	Not used
SNOMED_CID	varchar(20)	Not used
SNOMED_FSN	varchar(300)	Not used
IN_CORE	smallint	Not used

5.1.16. INSURANCE_TYPE Table

Table 23 describes the INSURANCE_TYPE table. This table is used to hold insurance information for payers data normalization.

Table 23: INSURANCE_TYPE Table

Column Name	Type	Description
[CONCEPT CODE] (PK)	varchar(10)	Concept code
[CONCEPT NAME]	varchar(200)	Concept name

5.1.17. LOINC Table

Table 24 describes the Logical Observation Identifiers Names and Codes (LOINC) table. This table is used to hold standardized LOINC terms for clinical data normalization.

Table 24: LOINC Table

Column Name	Type	Description
LOINC_NUM (PK)	nvarchar(10)	Standard LOINC code
COMPONENT	nvarchar(255)	Not used
SHORTNAME	nvarchar(40)	Not used
LONG_COMMON_NAME	nvarchar(255)	Standard LOINC description

5.1.18. MEDCIN_SNOMED Table

Table 25 describes the MEDCIN_SNOMED table. This table is used to hold standardized SNOMED terms for clinical data normalization.

Table 25: MEDCIN_SNOMED Table

Column Name	Type	Description
id (PK)	int	Internal identifier
medcinId	varchar(20)	Medcin ID
medcinDescription	varchar(300)	Medcin description
snomedCode	varchar(20)	SNOMED code

5.1.19. PAYERS Table

Table 26 describes the PAYERS table. This table is used to hold terminology mappings and national standards for insurance (payers) data.

Table 26: PAYERS Table

Column Name	Type	Description
ien (PK)	int	Internal identifier
name	varchar(100)	Insurance name
abbreviation	varchar(50)	Insurance name abbreviation
major_category	varchar(50)	Insurance category
standard_code	varchar(20)	Standard code

5.1.20. PERMISSIONS Table

Table 27 describes the PERMISSIONS table. This table is used to hold a user’s system permissions.

Table 27: PERMISSIONS Table

Column Name	Type	Description
id (PK)	int	User ID
description	varchar(50)	Permission type

5.1.21. Quality of Service (QOS)_LOGS Table

Table 28 describes the QOS_LOGS table. This table is used in the system status check process to hold the results of each tested endpoint of services utilized by JLV.

Table 28: QoS_LOGS Table

Column Name	Type	Description
date	datetime	Date and time of log entry
service	varchar(50)	Service type
status	varchar(10)	Service status description
message	varchar(MAX)	Message
id (PK)	bigint	Internal identifier

5.1.22. RACE Table

Table 29 describes the RACE table. This table is used to hold national standards and terminology for race.

Table 29: RACE Table

Column Name	Type	Description
id (PK)	int	ID
concept_code	varchar(10)	Concept code
concept_name	varchar(200)	Concept name

5.1.23. RECENTLY_VIEWED_PATIENTS Table

Table 30 describes the RECENTLY_VIEWED_PATIENTS table. This table is used to hold information regarding recently viewed patients.

Table 30: RECENTLY_VIEWED_PATIENTS Table

Column Name	Type	Description
id	int	Internal identifier
dateTime	datetime	Timestamp of record entry
userId	Int	User ID
name	nvarchar(255)	Patient’s name
edipi	varchar(16)	Patient’s EDIPI
ssn	varchar(16)	Patient’s SSN
sponsorSsn	varchar(16)	Sponsor’s SSN
gender	varchar(8)	Patient’s gender
dob	varchar(16)	Patient’s date of birth
sensitive	varchar(8)	Sensitive flag

Column Name	Type	Description
Icn	varchar(50)	Patient’s ICN

5.1.24. REGIONS Table

Table 31 describes the REGIONS table. This table is used to hold user geographic data.

Table 31: REGIONS Table

Column Name	Type	Description
id (PK)	varchar(10)	Internal identifier
name	varchar(100)	Region name

5.1.25. SITES Table

Table 32 describes the SITES table. This table is used to hold patient geographic data.

Table 32: SITES Table

Column Name	Type	Description
id (PK)	int	Internal identifier
moniker	varchar(20)	Short name of site
name	varchar(100)	Name of site
agency	varchar(10)	Agency of site
sitecode	varchar(10)	Internal site identifier
regionid	varchar(10)	Internal region identifier
DMISID	varchar(20)	DOD DMIS ID
MTFCODE	varchar(20)	DOD MTF Code
status	varchar(10)	Active flag (‘active’ = on)

5.1.26. SITES_PERMISSIONS Table

Table 33 describes the SITES_PERMISSIONS table. This table is used to hold site-based permissions.

Table 33: SITES_PERMISSIONS Table

Column Name	Type	Description
siteid (PK)	int	Internal identifier
permid	int	Permission ID

5.1.27. SNOMEDCT Table

Table 34 describes the SNOMEDCT table. This table is used to hold SNOMED-CT national standard terminology.

Table 34: SNOMEDCT Table

Column Name	Type	Description
id (PK)	int	Internal identifier
conceptid	varchar(100)	SNOMED concept ID
name	varchar(300)	SNOMED description
snomedid	varchar(50)	SNOMED ID

5.1.28. USERS Table

Table 35 describes the USERS table. This table is used to hold end user information and profile configuration(s).

Table 35: USERS Table

Column Name	Type	Description
id (PK)	int	Internal database ID
agency	varchar(20)	User’s agency (VA or DoD)
cardid	varchar(20)	User’s smartcard ID
lastname	varchar(30)	Not used
firstname	varchar(30)	Not used
middlename	varchar(30)	Not used
loginSite	varchar(20)	User’s login site
cfg	varchar(MAX)	User’s configuration
cfg_bak	varchar(MAX)	Not used
cfg_bak_date	smalldatetime	Not used
flags	varchar(4000)	User’s permissions
last_login	smalldatetime	Timestamp of last successful login

5.1.29. VA_DOCUMENTS Table

Table 36 describes the VA_DOCUMENTS table. This table is used to hold the terminology values used to map VA notes.

Table 36: VA_DOCUMENTS Table

Column Name	Type	Description
SourceCode	varchar(20)	VistA identifier
SourceCodeText	varchar(200)	VistA description
TargetCode	varchar(20)	Loinc code
TargetCodeText	varchar(300)	Loinc Description

5.1.30. VA_LABS Table

Table 37 describes the VA_LABS table. This table holds the terminology values used to map VA labs.

Table 37: VA_LABS Table

Column Name	Type	Description
id (PK)	int	Sequence number to identify the record
station	varchar(10)	Not used
labChemTestIEN	varchar(20)	VistA identifier
labChemTestName	varchar(300)	VistA name
loinc	varchar(20)	LOINC code
loincName	varchar(300)	Not used

5.1.31. VA_MEDICATIONS Table

Table 38 describes the VA_MEDICATIONS table. This table holds the terminology values used to map VA medications.

Table 38: VA_MEDICATIONS Table

Column Name	Type	Description
SourceCode	varchar(20)	VistA identifier
SourceCodeText	varchar(200)	VistA description
TargetCode	varchar(20)	RxNorm code
TargetCodeText_Short	varchar(500)	RxNorm short description
Target_Terminology	varchar(20)	Not used

5.1.32. VA_REACTANTS Table

Table 39 describes the VA_REACTANTS table. This table is used to hold the terminology values used to map VA reactants.

Table 39: VA_REACTANTS Table

Column Name	Type	Description
UMLSCode	nvarchar(255)	UMLS code
VUID	nvarchar(255)	VUID
VistAText	nvarchar(255)	VistA text
UMLSText	nvarchar(255)	UMLS text
id (PK)	int	Identifier

5.1.33. VATermMappingMaster_v002 Table

Table 40 describes the VATermMappingMaster_v002 table. This table is used to hold various VA-provided terminology maps.

Table 40: VATermMappingMaster_v002 Table

Column Name	Type	Description
RowID (PK)	int	Row number within the mapping spreadsheet
MapPathway_ID	smallint	Sequence number to identify the map pathway which is a Foreign Key to VATermMappingPathway_v002
SourceCode	varchar(50)	VistA unique identifier
SourceCodeText	varchar(300)	VistA description
TargetCode	varchar(50)	Target code
TargetCodeText	varchar(300)	Target description
OpCode	char(1)	Not used
CreateDate	datetime	Not used
EditDate	datetime	Not used

5.1.34. VATermMappingPathway_v002 Table

Table 41 describes the VATermMappingPathway_v002 table. This table is used to hold the domain and mapping descriptions of the MapPathway_ID from the VATermMappingMaster_v002 table.

Table 41: VATermMappingPathway_v002 Table

Column Name	Type	Description
MapPathway_ID (PK)	int	Internal identifier used to distinguish mapping domain
MapPathway_Desc	varchar(50)	Domain and map description
Domain	varchar(50)	Domain
SourceCodeType	varchar(50)	Source code type
TargetCodeType	varchar(50)	Target code type
OpCode	char(1)	Not used
CreateDate	datetime	Not used
EditDate	datetime	Not used

5.1.35. VITALS Table

Table 42 describes the VITALS table. This table is used to hold vitals terminology values.

Table 42: VITALS Table

Column Name	Type	Description
id (PK)	int	Sequence number of the record
dodNcid	varchar(20)	DoD identifier
vuid	varchar(20)	VistA identifier
loinc	varchar(20)	LOINC code
loincName	varchar(200)	LOINC name

5.1.36. VLER_FACILITIES Table

Table 43 describes the VLER_FACILITIES table. This table is used to hold the VLER community site locations.

Table 43: VLER_FACILITIES Table

Column Name	Type	Description
FACILITY_ID (PK)	int	Identification of the site
FACILITY_NUMBER	varchar(20)	Number of the site
HOME_COMMUNITY_ID	varchar(50)	Unique identifier of the site (OID)
FACILITY_NAME	varchar(100)	Name of the facility
FULL_HOME_COMMUNITY_ID	varchar(100)	Unique identifier of the site (formatted OID)

5.2. Non-DBMS Files

Not applicable to JLV.

5.3. Data View

The design of the JLV system includes the development and use of stored procedures, sets of operations, or queries sent to a database. Stored procedures for the JLV database are executed through jMeadows. Table 44 lists the stored procedures utilized within the JLV system.

Table 44: JLV System Stored Procedures

Stored Procedure Name	Description
addQoSReport	Adds records to the QoS Logs table
backupUserCfg	Copies the user’s configuration to the users.cfg_bak column
createCalisaUsers	Generates users for testing purposes
deletePermission	Removes permissions from a site
getAuthUser	Retrieves a record from the AUTH_USER table
getEndpoints	Gets the configured endpoints for a site
getIehrUserProfile	Retrieves a user’s profile
getLoginInfo	Retrieves user’s previous login attempts

Stored Procedure Name	Description
getPermissions	Queries the PERMISSIONS table for user permissions.
getProfile	Retrieves user’s profile
getRecentQoSReport	Retrieves the most recent QoS log entries
getRecentQoSServiceErrors	Retrieves the most recent QoS log errors
getRegions	Retrieves the available Regions
getSiteEndpoints	Retrieves the active site endpoints
getSitePermissions	Retrieves the permissions pertaining to a site
getSites	Retrieves all active sites
getUserCfg	Queries the USERS table for the user’s profile
getUserFlags	Retrieves the user’s permission configuration
getVLERSites	*Reserved for future use*
mapCodeList	Queries the table definitions for VA terminology mappings and returns national standard terminology to display in the CCP GUI (widgets)
mapCodeListAll	Queries the table definitions for VA terminology mappings and returns national standard terminology to display in the CCP GUI (widgets). Used in jMeadows “cached terminology” mode.
restoreUserCfg	Copies the value in users.cfg_bak into the users.cfg column
setAudit	Adds a record to the AUDIT table
setIehrUserProfile	Updates the user’s profile configuration
setLoginAudit	Adds a login record to the AUDIT table
setPermission	Sets a user’s permission
setProfile	Creates/updates a user’s profile
setUserCfg	Updates a user’s configuration
updateSubjectDN	Updates a user’s SubjectDN field

5.4. System Audit and Log Capabilities

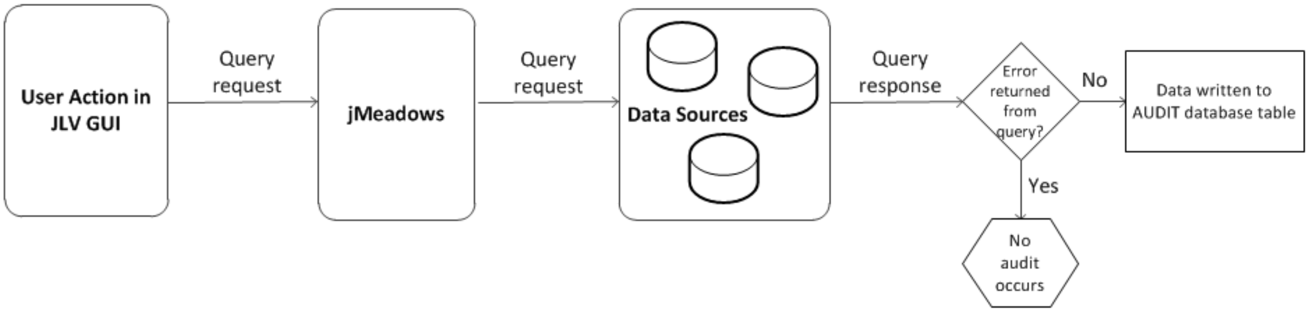
JLV writes a record of system use and user views of patient data to the JLV database. This allows the retention of application usage, and access to data for traceability for auditing purposes; specifically, to see what actions or calls are being made, including from where, and by whom, they originated.

5.4.1. Application Auditing

The JLV system audits actions that a user executes within the JLV web application, including, but not limited to, logging in and out, and the successful retrieval of patient data.

Figure 5 provides a sample auditing sequence, beginning with a user action. When an event is successful, or data is retrieved successfully from a data source, event data is recorded to the AUDIT table within the JLV database. In the AUDIT table, the query is recorded in the Category column. If the query response received from the data source has an error, no audit occurs.

Figure 5: JLV Audit Sequence Overview



Detailed information about JLV database tables can be found in Section 5.1, DBMS Files.

Note: Closing a widget utilizes the 'setUserConfig' query. This action represents a change to the user profile, and is not considered access to patient data.

5.4.2. Enhanced Error Handling for Performance Monitoring

Within the JLV system, the jMeadows Data Service, VistA Data Service, and Relay Service projects utilize a generic method, logError(), that is inserted in either the web service handler class or the web service class itself. The logError() method is used each time an error is caught during a query attempt, allowing for enhanced performance monitoring, and better reporting of errors that may occur within JLV.

The logError() method exposes errors that may be occurring in JLV and its subcomponents to the third-party Introscope tool, installed and managed by the VA for purposes of JLV monitoring, in order for Introscope to read the contents of the errors. Through its inspection processes, Introscope will be able to see if and when the logError() method is utilized during performance monitoring.

Table 45 is a snippet of logError() defined in VDS (VistaData.java):

Table 45: logError() Example 1

```
private void logError(Exception e) {
       LOGGER.error("VistaDataService ERROR: " + e.getMessage(), e);
}
```

Table 46 is a snippet of logError() called by the "getPatientAllergies" method in VDS (VistaData.java) if an error occurs:

Table 46: logError() Example 2

```
@WebMethod(operationName = "getPatientAllergies")
    public List<Allergy> getPatientAllergies(@WebParam(name = "queryBean")
                                             QueryBean queryBean) throws
VDSException {
        long begin = System.currentTimeMillis();
        try {
            AllergyDataService allergyService = new AllergyDataService();
            List<Allergy> rtc = allergyService.getPatientAllergies(queryBean);
            logResponse(queryBean, begin, "", rtc);
            return rtc;
        } catch (VDSException e) {
            logResponse(queryBean, begin, e.getMessage(), 0);
            logError(e);
            throw e;
        }
    }
```

5.4.3. Retrieval of Audit Information

The audit log information is stored in the AUDIT table within the JLV database. Audit data is maintained for the life of the application, and is not purged. Audit data can be retrieved by authorized Network Administrators or members of the JLV Support team, on an as-needed basis, using standard SQL queries.

Table 47 presents JLV database column items, and possible data types, that are displayed in a JLV audit log.

Table 47: JLV Database Audit Column Items and Data Types

Column Item	Data Type
ID or auditID	The unique ID of each entry
entryDate	The date and time at which the audit was entered
startDate	Works with endDate to set the date range for the data request
endDate	Works with startDate to set the date range for the data request
systemID	The user's login site identifier. On the DoD side, systemID specifies the host system to which the userID is associated. On the VA side, systemID specifies the VistA host system to which the userID is associated.
userNPI	The user's NPI (National Provider Identifier)
userID	The user's identifier. On the DoD side, userID is the IEN (Internal Entry Number) of the host system that is associated with the user's Access/Verify codes. On the VA side, userID is the VistA IEN of the VistA host system that is associated with the user's Access/Verify codes.
userName	The name of the user
patID	The patient's EDIPI
category	The query action (e.g., login, select patient, patient lookup, clinical domain data
queryType	The application name (JLV or unit_test)
cardID	The CAC or PIV identifier

Column Item	Data Type
ipAddress	The IP address of the user’s computer

5.4.4. Break the Glass Audit

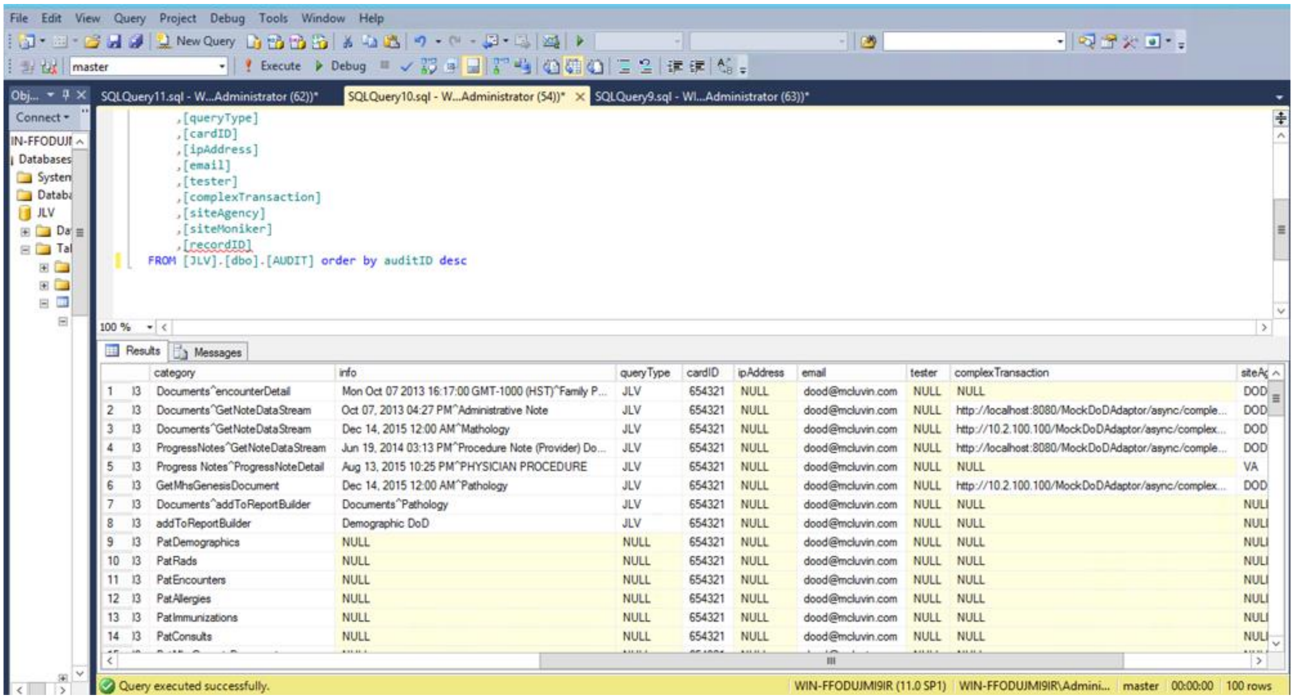
The phrase break the glass, as it relates to JLV, refers to breaking the barrier between the user and restricted access patient information. The action of breaking the glass involves the user acknowledging that they are about to access restricted patient information, and agreeing that any action they take within JLV on the restricted access patient information will be tracked and audited (Figure 6). The Break the Glass feature provides auditing for two functional areas:

- VA providers and VHA users are permitted to view the records of DoD-only patients (i.e., there are no VA identifiers, or the patient is **not** registered in MVI), but the VA requires that these actions be audited. If an attempt is made to access DoD-only patient records, the user is asked to specify the purpose for access.
  - Veterans Benefits Administration users are not permitted access to DoD-only patients (patients **not** registered in MVI).
- DoD and VA users will be audited each time a sensitive DoD record (domains: sensitive notes, outpatient encounters, and labs) is viewed, regardless of how many times the user has previously viewed it, including viewing the data multiple times in the same JLV session. When a user accesses and closes the sensitive record, then re-opens the same record and views it a second time, the user will again be asked to agree to be audited.

The following information will be captured for each attempt to access sensitive DoD data:

- Organization (e.g., VHA, VBA, DoD)
- Username
- User PIV
- User Location
- Patient (Patient Last, First Name, Middle Initial (MI); MVI; DOB)
- Sensitive data accessed (e.g., unique note identifier)
- Date/Time accessed
- Reason for access (e.g., Emergent Care, Clinical Care, and Authorized Administrative Use)

Figure 6: Sample Audit Log



```
      ,[queryType]
      ,[cardID]
      ,[ipAddress]
      ,[email]
      ,[tester]
      ,[complexTransaction]
      ,[siteAgency]
      ,[siteMoniker]
      ,[recordID]
  FROM [JLV].[dbo].[AUDIT] order by auditID desc
```

		category	info	queryType	cardID	ipAddress	email	tester	complexTransaction	siteAg
1	13	Documents^encounterDetail	Mon Oct 07 2013 16:17:00 GMT-1000 (HST)^Family P...	JLV	654321	NULL	dood@mcluvin.com	NULL	NULL	DOD
2	13	Documents^GetNoteDataStream	Oct 07, 2013 04:27 PM^Administrative Note	JLV	654321	NULL	dood@mcluvin.com	NULL	http://localhost:8080/MockDoDAdaptor/async/comple...	DOD
3	13	Documents^GetNoteDataStream	Dec 14, 2015 12:00 AM^Mathology	JLV	654321	NULL	dood@mcluvin.com	NULL	http://10.2.100.100/MockDoDAdaptor/async/complex...	DOD
4	13	ProgressNotes^GetNoteDataStream	Jun 19, 2014 03:13 PM^Procedure Note (Provider) Do...	JLV	654321	NULL	dood@mcluvin.com	NULL	http://localhost:8080/MockDoDAdaptor/async/comple...	DOD
5	13	Progress Notes^ProgressNoteDetail	Aug 13, 2015 10:25 PM^PHYSICIAN PROCEDURE	JLV	654321	NULL	dood@mcluvin.com	NULL	NULL	VA
6	13	GetMhsGenesisDocument	Dec 14, 2015 12:00 AM^Pathology	JLV	654321	NULL	dood@mcluvin.com	NULL	http://10.2.100.100/MockDoDAdaptor/async/complex...	DOD
7	13	Documents^addToReportBuilder	Documents^Pathology	JLV	654321	NULL	dood@mcluvin.com	NULL	NULL	NUL
8	13	addToReportBuilder	Demographic DoD	JLV	654321	NULL	dood@mcluvin.com	NULL	NULL	NUL
9	13	PatDemographics	NULL	NULL	654321	NULL	dood@mcluvin.com	NULL	NULL	NUL
10	13	PatRads	NULL	NULL	654321	NULL	dood@mcluvin.com	NULL	NULL	NUL
11	13	PatEncounters	NULL	NULL	654321	NULL	dood@mcluvin.com	NULL	NULL	NUL
12	13	PatAllergies	NULL	NULL	654321	NULL	dood@mcluvin.com	NULL	NULL	NUL
13	13	PatImmunizations	NULL	NULL	654321	NULL	dood@mcluvin.com	NULL	NULL	NUL
14	13	PatConsults	NULL	NULL	654321	NULL	dood@mcluvin.com	NULL	NULL	NUL

Query executed successfully.   WIN-FFODUJMI9IR (11.0 SP1)   WIN-FFODUJMI9IR\Admini...   master   00:00:00   100 rows

5.4.5. Data Service Response Time Logs

Query times for each web service call into the Relay Service, jMeadows, and VDS will be recorded to a file in the D:\Log directory on the server where the services are installed. Sample log file output for the jMeadows Data Service is provided in Figure 7.

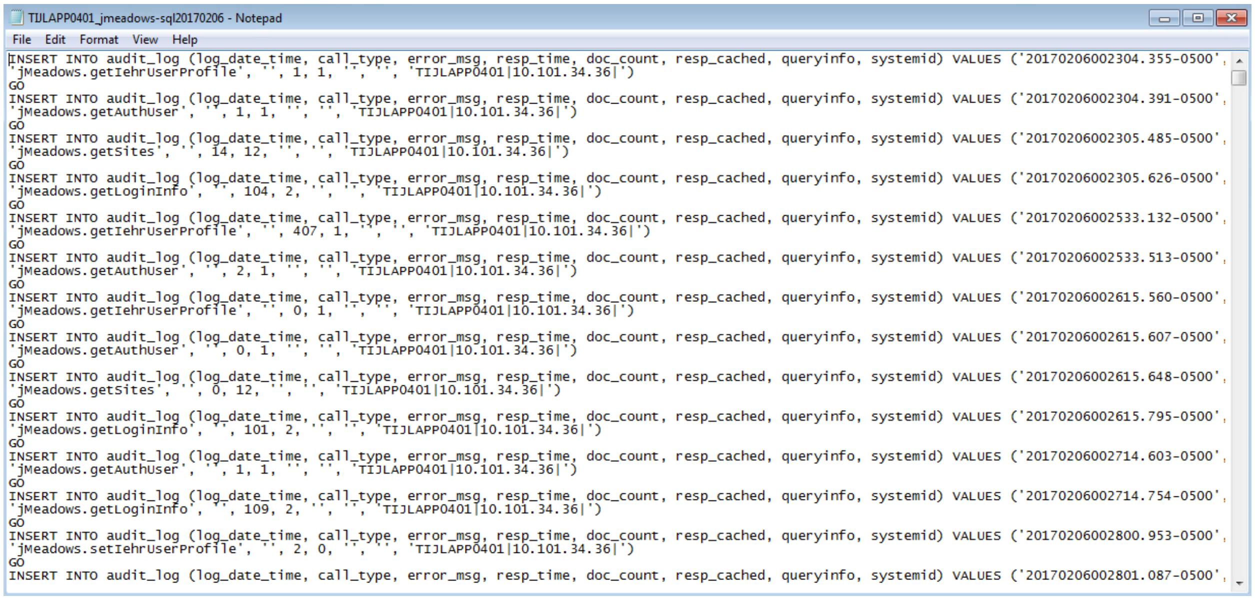
Table 48: Response Time Log Location

Data Service	Log File Name
jMeadows Data Service	jmeadows-sql.txt
Relay Service	bhie-sql.txt*
VistA Data Service	vds-sql.txt
** Bidirectional Health Information Exchange (BHIE)*	

5.4.5.1. Sample Response Time Logs

A sample query time log file output for the jMeadows Data Service is provided in Figure 7.

Figure 7: Sample jMeadows Log File Output



TIJLAPP0401_jmeadows-sql20170206 - Notepad
File Edit Format View Help

```
INSERT INTO audit_log (log_date_time, call_type, error_msg, resp_time, doc_count, resp_cached, queryinfo, systemid) VALUES ('20170206002304.355-0500',
'jMeadows.getIehrUserProfile', '', 1, 1, '', '', 'TIJLAPP0401|10.101.34.36|')
GO
INSERT INTO audit_log (log_date_time, call_type, error_msg, resp_time, doc_count, resp_cached, queryinfo, systemid) VALUES ('20170206002304.391-0500',
'jMeadows.getAuthUser', '', 1, 1, '', '', 'TIJLAPP0401|10.101.34.36|')
GO
INSERT INTO audit_log (log_date_time, call_type, error_msg, resp_time, doc_count, resp_cached, queryinfo, systemid) VALUES ('20170206002305.485-0500',
'jMeadows.getSites', '', 14, 12, '', '', 'TIJLAPP0401|10.101.34.36|')
GO
INSERT INTO audit_log (log_date_time, call_type, error_msg, resp_time, doc_count, resp_cached, queryinfo, systemid) VALUES ('20170206002305.626-0500',
'jMeadows.getLoginInfo', '', 104, 2, '', '', 'TIJLAPP0401|10.101.34.36|')
GO
INSERT INTO audit_log (log_date_time, call_type, error_msg, resp_time, doc_count, resp_cached, queryinfo, systemid) VALUES ('20170206002533.132-0500',
'jMeadows.getIehrUserProfile', '', 407, 1, '', '', 'TIJLAPP0401|10.101.34.36|')
GO
INSERT INTO audit_log (log_date_time, call_type, error_msg, resp_time, doc_count, resp_cached, queryinfo, systemid) VALUES ('20170206002533.513-0500',
'jMeadows.getAuthUser', '', 2, 1, '', '', 'TIJLAPP0401|10.101.34.36|')
GO
INSERT INTO audit_log (log_date_time, call_type, error_msg, resp_time, doc_count, resp_cached, queryinfo, systemid) VALUES ('20170206002615.560-0500',
'jMeadows.getIehrUserProfile', '', 0, 1, '', '', 'TIJLAPP0401|10.101.34.36|')
GO
INSERT INTO audit_log (log_date_time, call_type, error_msg, resp_time, doc_count, resp_cached, queryinfo, systemid) VALUES ('20170206002615.607-0500',
'jMeadows.getAuthUser', '', 0, 1, '', '', 'TIJLAPP0401|10.101.34.36|')
GO
INSERT INTO audit_log (log_date_time, call_type, error_msg, resp_time, doc_count, resp_cached, queryinfo, systemid) VALUES ('20170206002615.648-0500',
'jMeadows.getSites', '', 0, 12, '', '', 'TIJLAPP0401|10.101.34.36|')
GO
INSERT INTO audit_log (log_date_time, call_type, error_msg, resp_time, doc_count, resp_cached, queryinfo, systemid) VALUES ('20170206002615.795-0500',
'jMeadows.getLoginInfo', '', 101, 2, '', '', 'TIJLAPP0401|10.101.34.36|')
GO
INSERT INTO audit_log (log_date_time, call_type, error_msg, resp_time, doc_count, resp_cached, queryinfo, systemid) VALUES ('20170206002714.603-0500',
'jMeadows.getAuthUser', '', 1, 1, '', '', 'TIJLAPP0401|10.101.34.36|')
GO
INSERT INTO audit_log (log_date_time, call_type, error_msg, resp_time, doc_count, resp_cached, queryinfo, systemid) VALUES ('20170206002714.754-0500',
'jMeadows.getLoginInfo', '', 109, 2, '', '', 'TIJLAPP0401|10.101.34.36|')
GO
INSERT INTO audit_log (log_date_time, call_type, error_msg, resp_time, doc_count, resp_cached, queryinfo, systemid) VALUES ('20170206002800.953-0500',
'jMeadows.setIehrUserProfile', '', 2, 0, '', '', 'TIJLAPP0401|10.101.34.36|')
GO
INSERT INTO audit_log (log_date_time, call_type, error_msg, resp_time, doc_count, resp_cached, queryinfo, systemid) VALUES ('20170206002801.087-0500',
```

6. Detailed Design

6.1. Hardware Detailed Design

Refer to the tables in Section 4.1, Hardware Architecture, for a description of the server configuration for JLV Enterprise production infrastructure.

6.2. Software Detailed Design

This section provides conceptual and detailed information associated with the design of the software being delivered. For more information, see Section 4.2.1, JLV Development Technologies.

6.2.1. Access and Authorization Design

The JLV system restricts access to the JLV GUI to authorized users within the VA and DoD enterprise. User access to JLV, for all VA users and most DoD users, is via a Uniform Resource Locator (URL), provided to them by System Administrators.

VA users must present their PIV identification and Personal Identification Number (PIN), Windows Active Directory (AD) Account credentials, or Kerberos credentials before gaining access to JLV. When presenting a PIV/PIN for log in, the following credentials must also be provided:

- VistA/Computerized Patient Record System (CPRS) access and verify codes (VHA/clinical users)
- VistA/Compensation and Pension Record Interchange (CAPRI) access and verify codes (for VBA/benefits users)

User access control and authentication takes place before JLV interfaces with jMeadows. The user is authenticated to his/her host EHR system, granting the user access to the presentation layer. Based on their credentials, jMeadows retrieves the user's profile information from the JLV database. The user's default host location, custom widget layout, and other user-specific data are returned.

Note: The JLV system does not directly manage user roles. There is no administrative user access into the JLV web application.

If the user's PIV card is used for log in, the must insert the card into the computer before entering the URL of the JLV application into a browser window. The JLV log in pages guide the user through the log in process, including, where necessary, fields to enter user credentials such as PIN, agency, and site. A detailed overview of this process from the user's perspective is included in the *JLV 2.5.2 User Guide*, submitted with the JLV release package.

6.2.2. Conceptual Design

JLV's GUI framework is built on a simple architecture consisting of portals, tokens, widgets, and sessions. These elements, including the definition and/or purpose of each and how they are used in the GUI, are summarized in Table 49.

Table 49: Framework Elements and Implementation

Element	Implementation
Portal: A gateway for a web site or web application that is, or proposes to be, a major starting site for users when they get connected to the web or that users tend to visit as an anchor	The interface has two portals: a *provider* portal (where CCP information and access is managed), and a *patient* portal (where CCPs view patient data). Each portal does the following: • Pertains to a particular subject or topic. • Includes a library of widgets. • Provides a column-based widget layout and layout customization. Provides a tabular layout design and the ability to view any number of widget layouts.
Token: An object that represents something else, such as another object (either physical or virtual), or an abstract concept	The GUI uses these types of tokens: a *patient* token and a *record* token. A *patient* token: • Consists of patient ID, patient site code, and date/timestamp. • Is tied to an active session that is initiated by the provider upon log in to the system. • Is generated in Grails and encrypted. Data encryption is provided by the Advanced Encryption Standard (AES). A *record* token is used to retrieve specific details.
Widget: An element of a GUI that displays information or provides a specific way for a user to interact with the operating system and the application. Widgets include icons, pull-down menus, buttons, selection boxes, progress indicators, on-off checkmarks, scroll bars, windows, window edges (that allow the resizing of a window), toggle buttons, forms, and many other devices for displaying information	Each widget: • Is a mini-application, running within a larger application. • A generic container to which provider data, or clinical data, can be ported. • Contains data coming from one source; in this case, from the Representational State Transfer (REST) layer. Requires a patient token to retrieve data.

Element	Implementation
and for inviting, accepting, and responding to user actions.	
Session: A session is initiated when an authorized user logs into the JLV application.	During an active session, a user has access to the following capabilities: • View/Edit user profiles. • Change on-screen user interface themes. • Search for patient records. By default, a session terminates automatically after a period of inactivity.

6.2.2.1. Product Perspective

Refer to Section 3, Conceptual Design.

6.2.2.1.1. User Interfaces

6.2.2.1.1.1. Status Indicator Displays

JLV provides the following on-screen status indicators within the web application GUI:

- **System Status:** The JLV system includes a health monitoring service that communicates the status of external systems and services on the Login and Portal pages. Monitored services include DMDC, PDWS, MVI, VA VDS, Relay Service, jMeadows Data Service, and SnareWorks. See the *JLV Health Monitor Design Document*, submitted with this release, for a detailed overview of the JLV system status implementation, and GUI status messages.
- **Interface Status:** JLV provides interface status buttons in the toolbar of multiple Patient Portal widgets that display the status of the data source for that clinical domain. The information icon ℹ indicates that all sources are available. The warning icon ⚠ indicates that one or more data sources are unavailable. Both icons provide status for DoD, VA, and community partner data sources. Clicking either status icon will open the interface status details in a separate window.
- **Widget Banner:** A yellow banner will be displayed over a widget when one or more sources are unavailable, indicating sources could not be connected, and some records may not appear. Interface status notifications accessed from a widget show the connection status at the domain level.

6.2.2.1.2. Hardware Interfaces

Not applicable to JLV.

6.2.2.1.3. Software Interfaces

Not applicable to JLV, as it does not interface with Commercial Off-the-Shelf (COTS) products or systems.

6.2.2.1.4. Communications Interfaces

This section details the key JLV sequences utilizing the data services, including data retrieval from DoD, VA, and VLER sources. Section 6.2.2.1.4.1, Data Request/Response Sequence, provides an overview of the process that occurs when a user searches for a patient. Section

6.2.2.1.4.2, VA VLER Data Request/Response Sequence, provides an overview of the process that occurs when a user searches for VA VLER data for a patient.

JLV is comprised of the following data services that retrieve clinical data:

- VDS retrieves clinical data from all VA VistA EHR systems, VA VLER, and VistA Imaging; and,
- The Relay Service retrieves DoD clinical data from DES, which interfaces to AHLTA/CDR, TMDS, CHCS, Essentris, FHIE repositories, HAIMS, MHS GENESIS, and DoD VLER.

6.2.2.1.4.1. Data Request/Response Sequence

The process of JLV requesting patient data from sources, and the resulting response to JLV, is detailed in the steps listed below. This sequence occurs after a patient is selected. Figure 8 provides a diagram of the request/response relationship.

1. A widget requests data for a clinical domain from the REST service.
2. The REST service calls a corresponding SOAP service.
3. The jMeadows SOAP service layer makes the corresponding clinical domain request from jMeadows.
4. jMeadows returns a SOAP response that contains a VistA bean.
5. The VistA bean is mapped to a GUI bean.
6. The REST service returns the GUI bean to the widget.
7. The GUI bean is communicated back to the GUI, which then returns the response to the widget.

6.2.2.1.4.2. VA VLER Data Request/Response Sequence

The sequence for VA VLER data retrieval, (steps detailed below), occurs once a patient is selected in JLV.

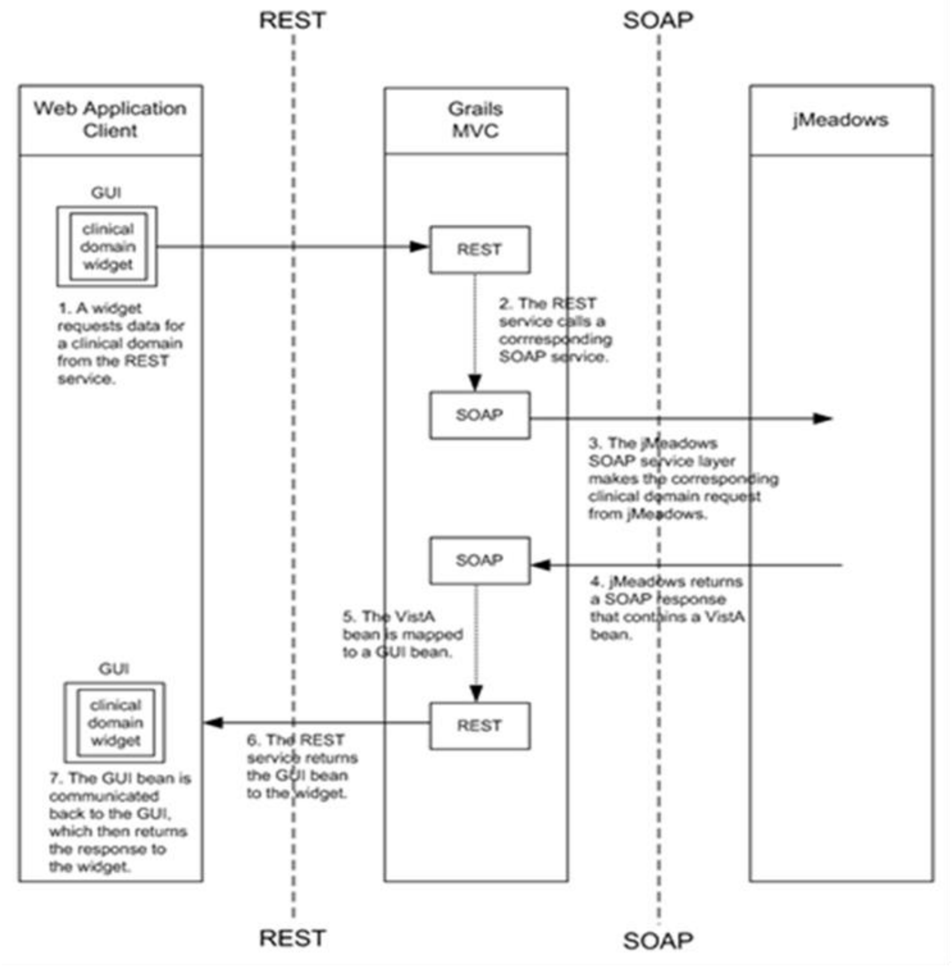
1. jMeadows sends a patient request, with ICN, to DAS through VDS
2. A series of document requests and retrieval transactions occur between DAS, CONNECT Gateway, and eHX
3. Requests are also sent from DAS to the central database for documents, and documents are returned from the database to DAS
4. DAS returns patient data to VDS
5. VDS returns VLER patient data to jMeadows
6. jMeadows aggregates the data and returns the data to JLV

6.2.2.1.4.3. VistA Imaging Viewer Request/Response Sequence

JLV communicates with VIX to retrieve a viewer URL that will launch a VistA Imaging Viewer instance for VA image records. This retrieval occurs after a user clicks one of a two areas in the JLV GUI: the camera icon in a supported widget's Image column and the image thumbnail, or the link displayed in the Details view of a VA record.

1. jMeadows calls VistA Data Service to retrieve the viewer URL for that record from VIX
    a. jMeadows constructs contextID as needed by the VIX API
2. VistA Data Service calls VIX to retrieve viewer URL
3. VIX API returns object
4. VistA Data Service parses the object from VIX to generate a string for the context URL
5. Vista Data Service passes the string (URL) back to the JLV web application
6. jMeadows returns viewer URL to JLV web application
7. Using this string, JLV will open an instance of the VistA Imaging Viewer in a separate browser tab or window

Figure 8: Sequence Diagram of the Request/Response Relationship


REST
SOAP
Web Application Client
Grails MVC
jMeadows
GUI
clinical domain widget
REST
1. A widget requests data for a clinical domain from the REST service.
2. The REST service calls a corresponding SOAP service.
SOAP
3. The jMeadows SOAP service layer makes the corresponding clinical domain request from jMeadows.
SOAP
4. jMeadows returns a SOAP response that contains a VistA bean.
5. The VistA bean is mapped to a GUI bean.
GUI
clinical domain widget
REST
6. The REST service returns the GUI bean to the widget.
7. The GUI bean is communicated back to the GUI, which then returns the response to the widget.
REST
SOAP

6.2.2.1.5. Memory Constraints

There are no known memory constraints.

6.2.2.1.6. Special Operations

Special operations, such as disaster recovery, are provided by the AITC and PITC data centers.

6.2.2.2. Product Features

For a detailed overview of product features from the user's perspective, please see the JLV 2.5.2 User Guide, submitted with the JLV release package.

6.2.2.3. User Characteristics

For a list of user roles and responsibilities, please see Section 1.2, User Profiles.

6.2.2.4. Dependencies and Constraints

6.2.2.4.1. External Data Sources

JLV is dependent on data provided from external data sources, VA MVI, VistA, VA VLER, CAPRI, DMDC - Defense Enrollment Eligibility Reporting System (DEERS), PDWS, and DES.

6.2.3. Specific Requirements

6.2.3.1. Database Repository

Please see Section 5, Data Design.

6.2.3.2. System Features

Requirements for JLV 2.5.2 can be found in the *JLV 2.5.2 Requirements Specifications Document*.

6.2.3.3. Design Element Tables

Not applicable to JLV.

6.2.3.3.1. Routines (Entry Points)

Not applicable to JLV.

6.2.3.3.2. Templates

Not applicable to JLV.

6.2.3.3.3. Bulletins

Not applicable to JLV.

6.2.3.3.4. Data Entries Affected by the Design

Not applicable to JLV.

6.2.3.3.5. Unique Record(s)

Not applicable to JLV.

6.2.3.3.6. File or Global Size Changes

Not applicable to JLV.

6.2.3.3.7. Mail Groups

Not applicable to JLV.

6.2.3.3.8. Security Keys

Not applicable to JLV.

6.2.3.3.9. Options

Not applicable to JLV.

6.2.3.3.10. Protocols

Not applicable to JLV.

6.2.3.3.11. Remote Procedure Call (RPC)

JLV leverages existing RPCs; therefore, no new RPCs are required. For a detailed list of RPCs utilized by JLV, please see the *VistA Data Service Interface Control Document*, submitted with this release. The two VistA Imaging Viewer calls are detailed in the following subsection.

6.2.3.3.11.1. VistA Imaging Viewer

JLV retrieves VA records with images from VA RPCs, and parses the record data into the JLV widget. Retrieving the viewer URL for the VistA Imaging Viewer window is on-demand. When a user clicks a camera icon, a process to retrieve the viewer URL for the VistA Imaging Viewer window is initiated.

For DoD radiology exam images, JLV queries VIX through jMeadows and VistA Data Service using the *getPatientRads* call, then maps image(s) to radiology reports. The viewer URL for VistA Imaging Viewer window is retrieved during this request and response sequence. No further queries are made to VIX after DoD image data is displayed in the JLV widget.

6.2.3.3.12. Constants Defined in Interface

Not applicable to JLV.

6.2.3.3.13. Variables Defined in Interface

Not applicable to JLV.

6.2.3.3.14. Types Defined in Interface

Not applicable to JLV.

6.2.3.3.15. GUI

Not applicable to JLV.

6.2.3.3.16. GUI Classes

Not applicable to JLV.

6.2.3.3.17. Current Form

Not applicable to JLV.

6.2.3.3.18. Modified Form

Not applicable to JLV.

6.2.3.3.19. Components on Form

Not applicable to JLV.

6.2.3.3.20. Events

Not applicable to JLV.

6.2.3.3.21. Methods

Not applicable to JLV.

6.2.3.3.22. Special References

Not applicable to JLV.

6.2.3.3.23. Class Events

Not applicable to JLV.

6.2.3.3.24. Class Methods

Not applicable to JLV.

6.2.3.3.25. Class Properties

Not applicable to JLV.

6.2.3.3.26. Uses Clause

Not applicable to JLV.

6.2.3.3.27. Forms

Not applicable to JLV.

6.2.3.3.28. Functions

Not applicable to JLV.

6.2.3.3.29. Dialog

Not applicable to JLV.

6.2.3.3.30. Help Frame

Not applicable to JLV.

6.2.3.3.31. HL7 Application Parameter

Not applicable to JLV.

6.2.3.3.32. HL7 Logical Link

Not applicable to JLV.

6.2.3.3.33. COTS Interface

Not applicable to JLV.

6.3. Network Detailed Design

Refer to Section 4.3, Network Architecture, for more information.

6.4. Security and Privacy

Security and privacy mechanisms are described in the following sections.

6.4.1. Security

6.4.1.1. Security Design Principles

The following security design principles are applied to the JLV system to ensure a system that follows security protocol standards for secured systems:

- **Session security:** By the use of secured, unique session tokens generated using a 128-bit hash from a secure random number generator for each authenticated user, the system ensures prevention of communication session hijacking. Once the user logs out of the system, the session is immediately destroyed, and the session hash can no longer be used. Also, if in some instance the session-id were to be obtained, the user cannot paste it as part of a URL string to gain access.
- **Data Encryption:** Using Secure Sockets Layer (SSL) with Transport Layer Security (TLS) 1.0 ensures that all server communication is encrypted, which limits the ability to perform Man in the Middle (MITM) attacks.
- **Database Encryption at Rest:** Using Microsoft SQL Server Transparent Data Encryption (TDE) Encryption level Advanced Encryption Standard (AES) 256-bit to encrypt Personal Identification Information (PII)/PHI data at rest.
- **Schema Validation:** Web Services used in JLV employ Schema Validation. This helps prevent Denial of Service (DoS) attacks by preventing the invocation of XML bombs.

6.4.1.2. Interface Transactions

JLV implements proper transport security, in accordance with Information Assurance (IA) guidelines. The transport security mechanism protects the application during transport, by using SSL for authentication and confidentiality. Transport layer security is provided by the transport mechanisms used to transmit information over the wire between clients and providers, thus transport layer security relies on secure HTTP transport (HTTPS) using SSL. Transport security is a point-to-point security mechanism that can be used for authentication, message integrity, and confidentiality. When running over an SSL-protected session, the server and client can authenticate one another, and negotiate an encryption algorithm and cryptographic keys before the application protocol transmits or receives its first byte of data. Security is live from the time it leaves the JLV user until it arrives at a source, as well as from the time it leaves a source and returns to the JLV user.

Digital certificates are necessary when running HTTPS using SSL. Digital VA Public Key Infrastructure (PKI) certificates are in use with the transmission of data in JLV.

6.4.1.3. Data Service Communication

jMeadows interfaces with two main data services to retrieve and aggregate clinical data:

1. VDS retrieves clinical data from all VA VistA EHR systems, community partner data from eHX, and data from VIX servers for VistA Imaging Viewer integration, and,
2. The Relay Service retrieves clinical data from AHLTA/CDR, TMDS, CHCS (SHARE), Essentris, HAIMS, MHS GENESIS, and VLER community partners

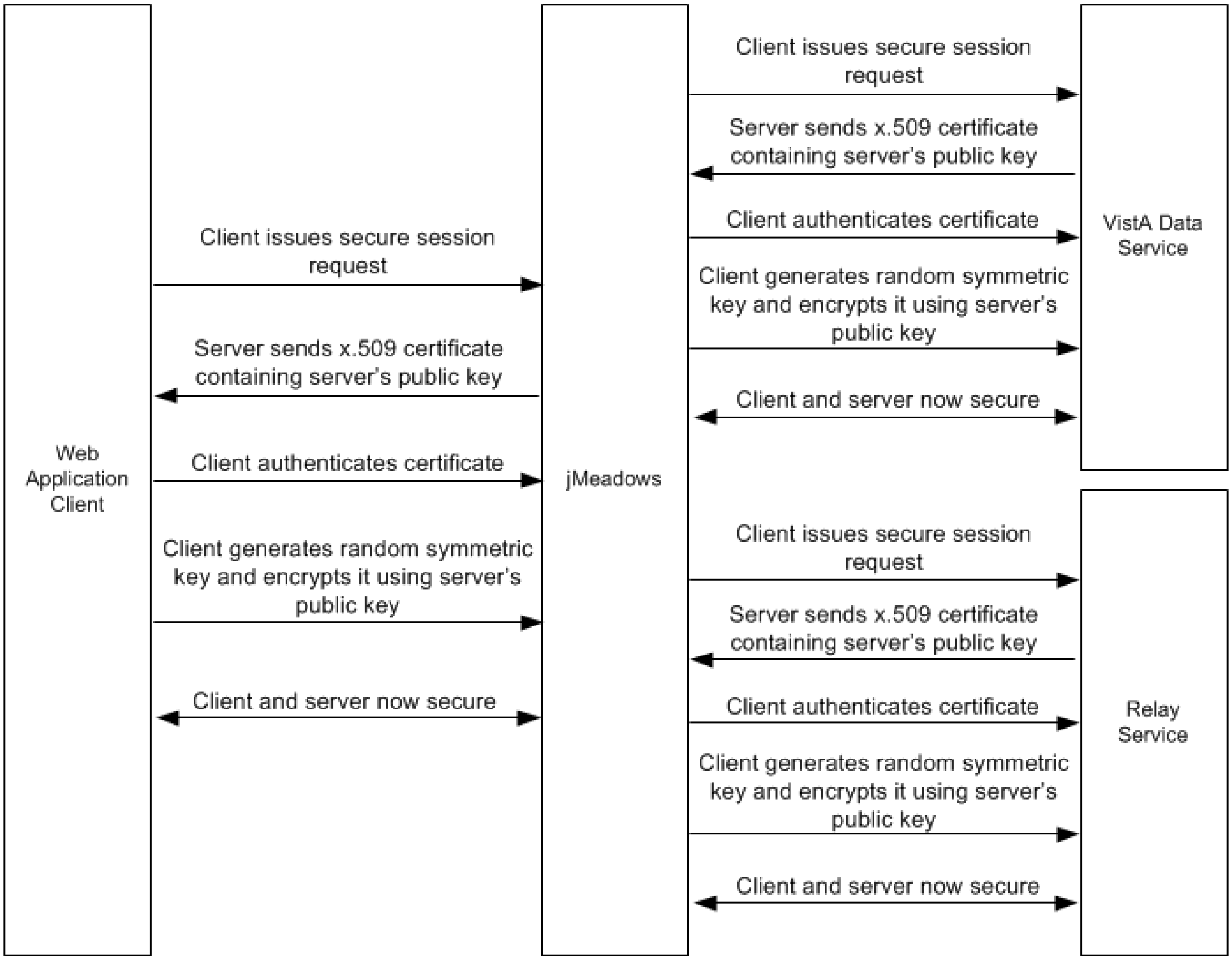
All communication to VDS and the Relay Service from jMeadows, the main aggregate service, is through HTTPS SSL/TLS basic authentication. Before any connection to the service is made, it is required that the exchange of valid server certificates and valid service/user name and password are provided for each service.

For example, when jMeadows requests VA data from VDS, the jMeadows server must first present the server certificate to the VDS server, along with the server/user name and password. If the provided server certificate and server/user name are valid, the request for data is executed and the data is returned to jMeadows. This process occurs for each data request: jMeadows to VDS, and jMeadows to the Relay Service.

JLV does not expose any services for consumption. JLV only requests data from supporting services. Therefore, JLV does not require a mechanism for detecting resubmitted SOAP messages, as it does not receive any incoming SOAP messages.

Figure 9 diagrams the secure service communication sequence.

Figure 9: Sequence Diagram for Secure Service Communication


Web Application Client
jMeadows
VistA Data Service
Relay Service
Client issues secure session request
Server sends x.509 certificate containing server's public key
Client authenticates certificate
Client generates random symmetric key and encrypts it using server's public key
Client and server now secure
Client issues secure session request
Server sends x.509 certificate containing server's public key
Client authenticates certificate
Client generates random symmetric key and encrypts it using server's public key
Client and server now secure
Client issues secure session request
Server sends x.509 certificate containing server's public key
Client authenticates certificate
Client generates random symmetric key and encrypts it using server's public key
Client and server now secure

6.4.1.4. Session Management Authentication, PKI Authentication

Session management authentication controls are inherited from VistA. The user name and password generation policies are those of the legacy systems, as JLV does not provision users.

Server password policies are inherited from the site. JLV also inherits policies for unsuccessful login attempts from VistA, and JLV provides notification of the unsuccessful attempts.

Session management security in JLV is outlined in the following procedure:

1. The provider launches the JLV GUI via a web browser. The JLV GUI is accessible from all potential VA facilities.
2. The provider logs in to the system by using his Access and Verify codes.
3. JLV sends the Access/Verify codes and user host location to jMeadows. This authenticates the provider to his host EHR system (VistA).
4. The provider performs a patient search. All patient searches are initiated against the PDWS service.
5. The patient search returns a list of GUI patient objects.
6. Each GUI patient object contains a patient token, which is generated in Grails MVC and is encrypted. Symmetric key encryption is used. A unique security/encrypted key is generated for each provider session. The encryption key is stored in the provider's session object on the server.
7. Each patient token is comprised of a patient ID, patient site code, and timestamp.
8. The patient token is included in every patient-centric request that is available from a clinical domain.

PKI authentication is a system that provides for trusted, third-party user identity inspection and assurance. This is done by a certificate authority, and uses cryptography. JLV supports PKI challenge.

6.4.1.5. Transport Security and Message Authentication

The transport security mechanism protects the application during transport using SSL for authentication and confidentiality. Transport-layer security is provided by the transport mechanisms used to transmit information over the wire between clients and providers, thus transport-layer security relies on HTTPS using SSL. Transport security is a point-to-point security mechanism that can be used for authentication, message integrity, and confidentiality. When running over an SSL-protected session, the server and client can authenticate one another and negotiate an encryption algorithm and cryptographic keys before the application protocol transmits or receives its first byte of data. Security is live from the time it leaves the consumer until it arrives at the provider or vice versa. The problem is that it is not protected once it gets to its destination. For protection of data after it reaches its destination, one of the security mechanisms that uses SSL and that also secures data at the message level will be utilized.

Digital certificates are necessary when running HTTPS using SSL. The HTTPS service of most web servers will not run unless a digital certificate has been installed. Digital certificates have been created for the web services server, and the default certificates are sufficient for running this mechanism, and are required when using atomic transactions. However, the message security mechanisms require a newer version of certificates than is available with the web services server.

The message authentication over SSL mechanism attaches a cryptographically secured identity or authentication token with the message and uses SSL for confidentiality protection.

Two-way SSL authentication is a security requirement for PDWS. When JLV attempts to establish a connection with the PDWS server, the PDWS server will transmit its certificate to the JLV server. If the JLV server recognizes the certificate as a trusted certificate, it will then proceed to transmit its certificate to the PDWS server. Only when the PDWS server recognizes the JLV server's certificate as authentic and trusted will a connection be established. The certificate root must be approved and recognized by DMDC for use with two-way SSL authentication.

Required certificates are for fully qualified domain names (FQDNs), as in the following examples:

- janusap-aitc.va.gov (application)
- janusds-aitc.va.gov (data service)
- janusdb-mesa.health.mil (database)

6.4.1.6. Input Validation

JLV employs input validation controls in components containing user input fields in order to prevent insertion of executable scripts, SQL injection, and invalid data types/sizes (see Table 50). The following JLV functions apply input validation:

- User Login: JLV inherits the authentication process of VistA. VistA has particular rules around user input with which the system does not interfere. Any notifications of input validation failure are provided to the user via the JLV login page. The login field input is limited to 100 characters, which is a known length that will not trigger a buffer overflow. All users have the same privileges upon usage of the application; therefore, invalid input could never provide an elevated privilege.
- Patient Search: JLV utilizes interfaces from PDWS to perform patient search. If the PDWS interface rule sets are not followed, PDWS will throw an error. To avoid errors, JLV will validate search field entries in the Patient Search dialog box to ensure JLV users follow PDWS interface rule sets.

Table 50: Patient Search Dialog Box Input Validation

Patient Identifier	Input Validation
DoD ID	Input for the DoD ID field (EDIPI) field is limited to 10 characters, numbers only. If these conditions are not met, the user will see the Input Error message, *DoD ID must be only 10 digits.*
SSN (Patient)	Searching for a patient using a SSN also requires the input of the patient's last name OR the patient's DOB in the fields provided in the Patient Search dialog box. 1) If a user enters only the Patient SSN, the user will see the Input Error message, *Patient Last Name or DOB is required.* 2) If a user enters only the Last Name or DOB (without the SSN number), the user will see the Input Error message, *Patient SSN or Sponsor SSN is required with Last Name or DOB.*
Sponsor SSN	Searching for a patient using a family member's social security number (Sponsor SSN) also requires the input of the patient's last name OR the patient's DOB in the fields provided in the Patient Search dialog box.

Patient Identifier	Input Validation
	1) If a user enters only the Sponsor SSN, the user will see the Input Error message, *Patient Last Name is required.* 2) If a user enters only the Last Name or DOB (without the Sponsor SSN number), the user will see the Input Error message, *Patient SSN or Sponsor SSN is required with Last Name or DOB.*
Last Name, First Name, DOB, and Gender	When searching for a patient without a SSN, all four identifiers (Last Name, First Name, DOB, and Gender) must be entered. If a user does not meet this requirement, the user will see the Input Error message, *Last Name, First Name, DOB, and Gender are required.*

Input validation failure will block access to patient data. In addition, the system also checks for null values, and the field sizes are limited to 100 characters, which is a known length that will not trigger a buffer overflow.

6.4.2. Privacy

The following sections detail how JLV implements various VA policies related to restricted access to patient data.

6.4.2.1. Break the Glass Restricted Access for VA Users

The Memorandum of Agreement (MOA) between DoD, DHA and VA, and VBA for sharing data through JLV stipulates at the request of the DoD Office of General Counsel, that VBA is permitted to access PHI through JLV only for individuals listed on the MVI, and that JLV will block VBA user access in JLV for information on any individual not listed on the MVI.

Break the glass provides the capability to properly track the purpose and organization of a VA provider when accessing a DoD-only patient (i.e., there are no VA identifiers for this patient) to ensure access to information is properly audited. JLV displays a warning message when a VA user attempts to look up a patient who is not registered with the VA. Also, the user is informed that his acknowledgement of the message will be audited. The audit log information is stored within the JLV database server and is retrievable by the JLV developers on an as-needed basis.

JLV handles patient selection differently based on whether the user is a VHA or VBA users. For VHA Users, after performing a patient search and selecting a patient from the list presented, the VHA user is asked to specify the purpose of accessing the record. Options presented to the user are: Emergent Care, Clinical Care, or Authorized Administrative Use.

For VBA Users, there are two different dialog boxes that appear after performing a patient search, depending on whether the patient is registered in MVI:

- Patient registered in MVI. After performing a patient search and selecting a patient from the list presented, a VBA user will see a dialog box when he selects a patient whom is registered in MVI. After agreeing to the audit, the VBA user can access the patient’s record.
- Patient not registered in MVI. A VBA user cannot access the record of a patient not registered in MVI. After performing a patient search and selecting a patient from the list presented, a VBA user will see a dialog box when he selects a patient whom is not registered in MVI.

6.4.2.2. Restricted Access to DoD Sensitive Data

Break the glass functionality is applied to VA and DoD users accessing a sensitive DoD record (domains: sensitive notes, outpatient encounters, and labs). When a user attempts to access data masked as sensitive in the JLV GUI, the user is prompted to agree to be audited. The user has the option to agree to the audit and view the record, or cancel the request to view the record. Audit information is stored in the JLV database and is retrievable by the JLV developers on an as-needed basis.

A user will be prompted each time a sensitive record is viewed, regardless of how many times the user has previously viewed it, including viewing multiple times in the same user session. When a user accesses and closes the sensitive record and then opens the same record/views the record a second time, the user will be asked to agree to be audited again.

6.4.2.3. Additional Restricted Access Scenarios for VA Users

JLV will check the VA user’s access credentials after a patient is selected from the search results presented in Patient Search dialog and enforce additional patient data access restrictions for the following scenarios:

- JLV will deny a VA user the ability to view patient records when the user's SSN is not registered in the user’s VistA profile.
- JLV will deny a VA user the ability to view the user's own patient records.

6.4.2.4. VHA User Restricted Patient List

When access to patients is restricted for a VHA user, JLV will display a pre-determined list of patients that the VHA user is allowed to access after the user logs in. The user will only be able to select a patient from the displayed Patient List. The list represents the patient(s) the VHA user is authorized to view as configured in the user’s local VistA host.

In the JLV GUI, when access to patients is restricted for a VHA user, the patient search function is removed and is replaced with a ☰ Patient List link in the top-left corner of the portal pages. During an active user session, clicking the ☰ Patient List link will open the Patient List and will display the patients that user is allowed to view.

For the JLV patient lookup sequence for the VHA user with restricted patient selection, jMeadows queries VDS to access the user’s VistA settings and retrieves the patient information from the user’s local VistA host.

The modified login and patient lookup sequence for VHA users is as follows:

1. VHA user logs into JLV using credentials for his/her local VistA site.
2. JLV authenticates the user to his/her host system, granting that user access to the presentation layer.
3. jMeadows issues request to VistA Data Service.
4. VistA Data Service uses RPC to query the user’s local VistA host for the user’s VistA configuration.
5. If the RESTRICT PATIENT SELECTION: setting is set to YES, JLV will perform the following (utilizing VistA Data Service and jMeadows):

b. Pull the restricted patients list from the user’s VistA configuration.

c. Populate a Patient List in the JLV GUI with the patients from the user’s VistA configuration and display over the JLV portal.

d. Remove the Patient Search function from the portal pages and insert a Patient List link.

6.4.2.5. Patient Blacklist

The JLV system includes a mechanism to blacklist patients so that all JLV users are blocked from obtaining any information about a patient through the JLV GUI. When a user tries to search for one of the blacklisted patients by entering the blacklisted patient’s EDIPI or other patient identifiers in the Patient Search dialog box, the patient’s name will be suppressed and not seen in JLV's patient select list. The patient select list will display the text No Results, and there will be no further message displayed to the JLV user to indicate that the patient identifier entered in the Patient Search dialog box is on the blacklist.

Blacklisted EDIPIs are stored in the pdwsblacklist.properties file on the jMeadows server where the jMeadows WAR file resides. No other patient identifiers are recorded in the file. JLV checks this file during the patient search sequence when an EDIPI patient identifier is entered by a JLV user.

To add or remove blacklisted patients, the JLV system administrator would access the file on the jMeadows server and make the desired change(s). Modifications to the file require a restart of the jMeadows service.

6.5. Service Oriented Architecture/ESS Detailed Design

Not applicable to JLV.

6.5.1. Service Description for <Consumed Service Name>

Not applicable to JLV.

6.5.2. Service Design for <Provided Service Name>

Not applicable to JLV.

6.5.2.1. Introduction

6.5.2.1.1. Purpose and Scope of Service

Not applicable to JLV.

6.5.2.1.2. Links to Other Documents

Not applicable.

6.5.2.2. Service Details

6.5.2.2.1. Service Identification

Not applicable to JLV.

6.5.2.2.2. Service Versions

Not applicable to JLV.

6.5.2.2.3. Summary of Design and Platform Details

6.5.2.2.3.1. SOA Pattern(s) Implemented

Not applicable to JLV.

6.5.2.2.3.2. COTS Platform Vendor Names and Versions for Hosting Platform

Not applicable to JLV.

6.5.2.3. Dependencies

Not applicable to JLV.

6.5.2.4. Service Design Details

Not applicable to JLV.

6.5.2.4.1. Interface Technical Specs

Not applicable to JLV.

6.5.2.4.1.1. Service Invocation Type

Not applicable to JLV.

6.5.2.4.1.2. Service Interface Type

Not applicable to JLV.

6.5.2.4.1.3. Service Name

Not applicable to JLV.

6.5.2.4.1.4. Interface

Not applicable to JLV.

6.5.2.4.1.5. End Points

Not applicable to JLV.

6.5.2.4.1.6. Operations or Methods

Not applicable to JLV.

6.5.2.4.1.7. Message Schemas

Not applicable to JLV.

6.5.2.4.2. Information Model

Not applicable to JLV.

6.5.2.4.2.1. Class Diagram and Description of Entities Involved

Not applicable to JLV.

6.5.2.4.2.2. Mappings from ELDM to Standards Based Schemas

Not applicable to JLV.

6.5.2.4.3. Behavior Model (AKA Use Case Realization)

Not applicable to JLV.

6.5.2.4.3.1. Use Cases (Use Case Model)

Not applicable to JLV.

6.5.2.4.3.2. Interaction Diagrams

Not applicable to JLV.

6.5.2.5. Gap Analysis

Not applicable to JLV.

6.5.2.5.1. Variances from Enterprise Target Architecture

Not applicable to JLV.

6.5.2.5.2. Variances from SLDs

Not applicable to JLV.

6.5.2.5.3. Variances from Standards and Policies

Not applicable to JLV.

6.5.2.5.4. Justification for Exceptions and Mitigation

The exceptions for variances were not available at the time of this writing. All information will be supplemented when this information is available.

7. External System Interface Design

JLV utilizes jMeadows, VDS, and Relay Service to interface with external interfaces. Definitions and details of these external interfaces can be found in JLV Interface Control Documents (ICDs), submitted with this release:

- jMeadows ICD
- Relay Service ICD
- VistA Data Service ICD

7.1. Interface Architecture

Please see Figure 1 for detailed information.

7.2. Interface Detailed Design

Definitions and details of these external interfaces can be found in the JLV Interface Control Documents, submitted with this release:

- jMeadows Data Service
- Relay Service
- VistA Data Service

Please see Section 3.1, Conceptual Application Design, for more details.

8. Human-Machine Interface

For detailed descriptions, and examples of the JLV user interface, please refer to the *JLV 2.5.2 User Guide*, submitted with the JLV release package.

8.1. Interface Design Rules

Refer to Section 6.2.2, Conceptual Design, for framework elements and interface implementation.

8.2. Inputs

Not applicable to JLV. The JLV web application is read only, and does not have input forms.

8.3. Outputs

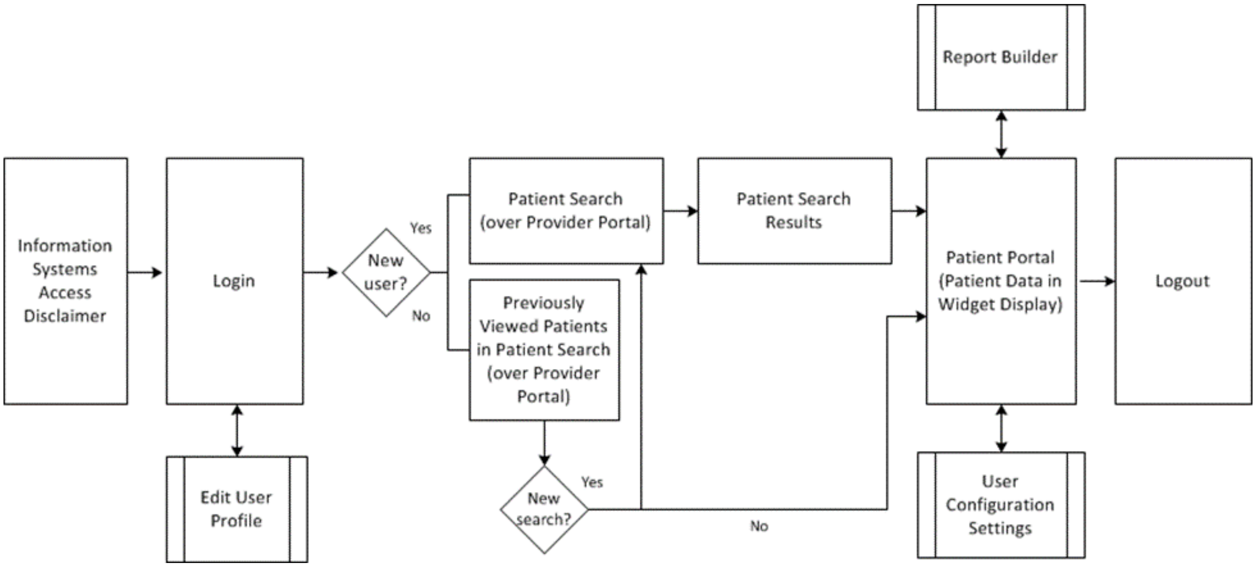
The JLV web application includes the Report Builder, a feature that enables a user to compile multiple patient records into a PDF document.

A small number of widgets within the Patient Portal include a Copy to Clipboard button that allows a user to copy the contents of the widget. Refer to the *JLV 2.5.2 User Guide*, submitted with the JLV release package, for additional information.

8.4. Navigation Hierarchy

Figure 10 provides an overview of JLV navigation within the web application.

Figure 10: JLV Navigation Hierarchy


Report Builder
Information Systems Access Disclaimer
Login
New user?
Yes
No
Patient Search (over Provider Portal)
Patient Search Results
Patient Portal (Patient Data in Widget Display)
Logout
Previously Viewed Patients in Patient Search (over Provider Portal)
Edit User Profile
New search?
Yes
No
User Configuration Settings

8.4.1. User Interface (UI) Screen

For examples, please refer to the *CLIN 0003AM JLV 2.5.2 User Guide*. Once approved all documentation for JLV 2.5.2 is available on the TSPR.

9. Attachment A – Approval Signatures

The Business Sponsor and Project Manager are required to sign.

Signed: Amanda M. Cournoyer 162029

Digitally signed by Amanda M. Cournoyer 162029
DN: dc=gov, dc=va, o=internal, ou=people,
0.9.2342.19200300.100.1.1=amanda.cournoyer@v
a.gov, cn=Amanda M. Cournoyer 162029
Date: 2017.04.07 12:21:42 -04'00'

Amanda Cournoyer, Business Sponsor Date

Signed: Latricia R. Facundus 703001

Digitally signed by Latricia R. Facundus 703001
DN: dc=gov, dc=va, o=internal, ou=people,
0.9.2342.19200300.100.1.1=latricia.facundus@va.g
ov, cn=Latricia R. Facundus 703001
Date: 2017.04.10 13:27:37 -04'00'

Latricia Renae Facundus, Project Manager Date

A. Additional Information

A.1. Identification of Technology and Standards

The following items relate to the technology and standards utilized in JLV.

- Section 508 Accessibility Compliance
- eHX Gateway
  - NwHIN is a set of standards, services, and policies that enable secure health information exchange across the Internet via the open-source CONNECT gateway. NwHIN provides the foundation for the exchange of health information across public and private healthcare entities.
  - JLV retrieves VA community health summary data retrieved from the eHX Gateway. VistA Data Service initiates a document query and passes the VA Integration Control Number (ICN) to eHX. The eHX interface uses the ICN to generate a list of documents to return to JLV.
- SOAP Messaging Protocol
  - SOAP version 2.0 is the messaging protocol used to communicate between the web services within JLV and the data sources.
- REST architecture is used between the GUI/browser and the JLV web application, as well as between components within the data/storage tiers of the JLV system.
- Java Database Connectivity (JDBC)
  - JDBC is a database connectivity technology used to connect the JLV Database and jMeadows. JDBC enables the transfer of data between the two components.
- SSL
  - The Transport Security mechanism protects the application during transport using SSL for authentication and confidentiality. Transport-layer security is provided by the transport mechanisms used to transmit information over the wire between clients and providers, thus transport-layer security relies on HTTPS using SSL.
- FHIR
  - HL7 Fast Healthcare Interoperability Resources (FHIR) is an API used to provide the architecture and the interface controls for the interface between VDS and DAS. FHIR enables the transfer of documents and data between the two services. The FHIR standard is used realizing RESTful web services.

A.2. Constraining Policies, Directives and Procedures

- VA/DOD Data Sharing Policy
- National Defense Authorization Act (NDAA), providing directives for DoD/VA data interoperability
- Health Insurance Portability and Accountability Act (HIPAA)

A.3. Requirements Traceability Matrix

Please see the *JLV 2.5.2 Requirements Traceability Matrix*. Once approved, the document will be available on the TSPR.

A.4. Packaging and Installation

Please see the *JLV 2.5.2 Deployment, Installation, Backout, and Rollback Guide*. Once approved, the document will be available on the TSPR.

A.5. Design Metrics

Design details are referenced throughout this System Design Document.

10. Acronyms and Abbreviations

Table 51 lists all acronyms and abbreviations used throughout this document, and their descriptions.

Table 51: Acronyms and Abbreviations

Acronym	Definition
AERB	Architecture and Engineering Review Board
AES	Advanced Encryption Standard
AHLTA	Armed Forces Health Longitudinal Technology Application
AITC	Austin Information Technology Center
Ajax	Asynchronous JavaScript and XML
API	Application Program Interface
AVHE	Application Virtualization Hosting Environment
BHIE	Bidirectional Health Information Exchange
BTG	Break the Glass
CA	Computer Associates
CAC	Common Access Card
CAPRI	Compensation and Pension Record Interchange
CCOW	Clinical Context Object Workgroup
CDC	Centers for Disease Control
CDR	Clinical Data Repository
CHCS	Composite Health Care System
CLIN	Contract Line Item Number
COTS	Commercial Off-the-Shelf
CPRS	Computerized Patient Record System
CPT	Current Procedural Terminology
CSS	Cascading Style Sheets
DEERS	Defense Enrollment Eligibility Reporting System
DES	Data Exchange Service
DFN	Data File Number
DHA	Defense Health Agency
DMDC	Defense Manpower Data Center
DMIX	Defense Medical Information Exchange
DOB	Date of Birth
DoD	Department of Defense
DOM	Document Object Model

Acronym	Definition
DoS	Denial of Service
EDIPI	Electronic Data Interchange Personal Identifier
EHDV	External Health Data Viewer
eHMP	enterprise Health Management Platform
EHR	Electronic Health Records
eHX	eHealth Exchange
EO	Enterprise Operations
ESS	Enterprise Shared Services
FHIE	Federal Health Information Exchange
FMP	Family Member Prefix
FOUO	For Official Use Only
FQDNs	Fully Qualified Domain Names
GB	Gigabyte
GTM	Global Traffic Manager
GUI	Graphical User Interface
HAIMS	Healthcare Artifact and Image Management Solution
HCP	Health Care Provider
HIE	Health Information Exchange
HIPAA	Health Insurance Portability and Accountability Act
HRG	Hawaii Resources Group
HTML	HyperText Markup Language
HTTP	Hypertext Transfer (or Transport) Protocol
HTTPS	Hypertext Transfer Protocol Secure
IA	Information Assurance
ICD	Interface Control Document
ICD9	International Classification of Diseases
ICN	Integration Control Number
ID	Identification
IEN	Employer Identification Number
IP	Internet Protocol
IPO	Interagency Program Office
IPT	Integrated Project Teams
ISCP	Information System Contingency Plan
IT	Information Technology

Acronym	Definition
JDBC	Java Database Connectivity
JLV	Joint Legacy Viewer
JSON	JavaScript Object Notation
KPP	Key Performance Parameters
LOINC	Logical Observation Identifiers Names and Codes
MESOC	Military Health System Enterprise Services Operations Center
MHS	Military Health System
MI	Middle Initial
MITM	Man-in-the-Middle
MOA	Memorandum of Agreement
MVC	Model-View-Controller
MVI	Master Veteran Index
NDAA	National Defense Authorization Act
NPI	National Provider Identifier
NwHIN	Nationwide Health Information Network
OI&T	Office of Information and Technology
OS	Operating System
PCM	Primary Care Management
PDF	Portable Document Format
PDWS	Patient Discovery Web Service
PHI	Personal Health Identifiers
PII	Personally Identifiable Information
PIN	Personal Identification Number
PITC	Philadelphia Information Technology Center
PIV	Personal Identification Verification
PKI	Public Key Infrastructure
QoS	Quality of Service
RAM	Random Access Memory
REST	REpresentational State Transfer
RPC	Remote Procedure Call
RSD	Requirements Specification Document
RTM	Requirements Traceability Matrix
SDD	System Design Document
SGML	Standard Generalized Markup Language

Acronym	Definition
SNOMED	Systematized Nomenclature of Medicine
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SSL	Secure Socket Layer
SSMS	SQL Server Management Studio
SSOi	Single Sign On Internal
SSN	Social Security Number
TDE	Transparent Data Encryption
TIU	Text Integration Utilities
TLS	Transport Layer Security
TMDS	Theater Medical Data Store
UAT	User Acceptance Testing
UCP	Utility Control Point
UI	User Interfaces
URL	Uniform Resource Locator
VA	Department of Veterans Affairs
VBA	Veterans Benefits Administration
VDS	VistA Data Service
VHA	Veterans Health Administration
VistA	Veterans Health Information Systems and Technology Architecture
VIX	VistA Imaging Exchange
VLER	Virtual Lifetime Electronic Record
VSA	VistA Services Assembler
WAR	Web Application Archive
WSDL	Web Services Description Language
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language (XSL) Transformations