Department of Veterans Affairs

**Digital Veterans Platform (DVP)**

**Integration with**

**Identity and Access Management Services**

**Master Veteran Index (MVI)**

Integration System Design Document



**November 2020**

Version 0.4

Revision History

Note: The revision history cycle begins once changes or enhancements are requested after the integration System Design Document has been baselined.

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 11/23/20 | 0.4 | Updated to include SaaS use-case | DVP Team |
| 10/13/17 | 0.3 | Updated Draft | Linda Reimonenq (MVI development) |
| 10/13/17 | 0.2 | Add DVP specific information. | DVP Team. |
| 9/8/2017 | 0.1 | Initial Draft | Linda Reimonenq, Brian Toval (MVI development) |

Table of Contents

[1 Introduction 1](#_Toc496710798)

[1.1 Purpose 1](#_Toc496710799)

[1.2 Identification 1](#_Toc496710800)

[1.3 Scope 1](#_Toc496710801)

[1.4 Relationship to Other Plans 2](#_Toc496710802)

[1.5 Methodology, Tools, and Techniques 2](#_Toc496710803)

[1.6 Policies, Directives, and Procedures 2](#_Toc496710804)

[1.7 Constraints 2](#_Toc496710805)

[1.8 Design Trade-offs 3](#_Toc496710806)

[1.9 User Characteristics 3](#_Toc496710807)

[1.9.1 User Problem Statement 3](#_Toc496710808)

[1.9.2 User Objectives 3](#_Toc496710809)

[2 Background 4](#_Toc496710810)

[2.1 Overview of the System 4](#_Toc496710811)

[2.2 Overview of the Business Process 4](#_Toc496710812)

[2.3 Business Benefits 5](#_Toc496710813)

[2.4 Assumptions and Constraints 5](#_Toc496710814)

[2.4.1 Design Assumptions 5](#_Toc496710815)

[2.4.2 Design Constraints 6](#_Toc496710816)

[2.5 Overview of the Significant Requirements 7](#_Toc496710817)

[2.5.1 Overview of the Significant Functional Requirements 7](#_Toc496710818)

[2.5.2 Functional Workload and Functional Performance Requirements 7](#_Toc496710819)

[2.5.3 Operational Requirements 8](#_Toc496710820)

[2.5.4 Overview of the Technical Requirements 8](#_Toc496710821)

[2.5.5 Overview of the Security or Privacy Requirements 8](#_Toc496710822)

[2.5.6 System Criticality and High Availability Requirements 10](#_Toc496710823)

[2.5.7 Special Device Requirements 10](#_Toc496710824)

[2.6 Legacy System Retirement 11](#_Toc496710825)

[3 Conceptual Design 11](#_Toc496710826)

[3.1 Conceptual Application Design 11](#_Toc496710827)

[3.1.1 Application Context 11](#_Toc496710828)

[3.1.2 High-Level Application Design 11](#_Toc496710829)

[3.1.3 Application Locations 13](#_Toc496710830)

[3.2 Conceptual Data Design 14](#_Toc496710831)

[3.2.1 Project Conceptual Data Model 14](#_Toc496710832)

[3.2.2 Database Information 14](#_Toc496710833)

[3.3 Conceptual Infrastructure Design 14](#_Toc496710834)

[3.3.1 System Criticality and High Availability 14](#_Toc496710835)

[3.3.2 Special Technology 14](#_Toc496710836)

[3.3.3 Technology Locations 14](#_Toc496710837)

[3.3.4 Conceptual Infrastructure Diagram 16](#_Toc496710838)

[3.4 Hardware Architecture 17](#_Toc496710839)

[3.5 Software Architecture 17](#_Toc496710840)

[3.6 Communications Architecture 17](#_Toc496710841)

[4 Data Design 18](#_Toc496710842)

[4.1 Database Management System Files 18](#_Toc496710843)

[4.2 Non-Database Management System Files 18](#_Toc496710844)

[5 Detailed Design 18](#_Toc496710845)

[5.1 Hardware Detailed Design 18](#_Toc496710846)

[5.2 Software Detailed Design 18](#_Toc496710847)

[5.2.1 Code Impacts 18](#_Toc496710848)

[5.2.2 Source ID Definition 18](#_Toc496710849)

[5.2.3 Integration with MVI – Sample Queries / Responses 19](#_Toc496710850)

[5.3 Communications Detailed Design 19](#_Toc496710851)

[6 External Interface Design 20](#_Toc496710852)

[6.1 Interface Architecture 20](#_Toc496710853)

[6.2 Interface Detailed Design 20](#_Toc496710854)

[7 Human-Machine Interface 21](#_Toc496710855)

[7.1 Interface Design Rules 21](#_Toc496710856)

[7.2 Inputs 21](#_Toc496710857)

[7.3 Outputs 21](#_Toc496710858)

[7.4 Navigation Hierarchy 21](#_Toc496710859)

[8 System Integrity Controls 21](#_Toc496710860)

[9 Appendix A 21](#_Toc496710861)

[9.1 Abbreviations and Definitions 21](#_Toc496710862)

[9.2 Glossary 22](#_Toc496710863)

[9.3 Requirements Traceability Matrix 23](#_Toc496710864)

[9.4 Packaging and Installation 23](#_Toc496710865)

[9.5 Design Metrics 23](#_Toc496710866)

[9.6 Required Technical Documents 23](#_Toc496710867)

[10 Approval Signatures 24](#_Toc496710868)

List of Figures

[Figure: 1 DVP/MVI Interactions 5](#_Toc496710869)

[Figure: 2 Security Boundaries 9](#_Toc496710870)

[Figure: 3 DVP Generic - To Be Process (zoom to 200%) 12](#_Toc496710871)

[Figure: 4 DVP Process - To Be Initial Release (zoom to 200%) 13](#_Toc496710872)

[Figure: 5 Interface Architecture 20](#_Toc496710873)

# Introduction

The Identity Services (IdS) Master Veteran Index (MVI) and Access Services (AcS) functionality, per the VA Chief Information Officer (CIO) Identity Management Policy Directive VAIQ #7338443 mandate issued by VA Information and Technology, requires all applications within VA to comply with the VA IAM requirements. Adhering to the mandate will help ensure that references to the identity of Veterans and their beneficiaries are reflected accurately, and that appropriate levels of security protecting the identities, information, and interests of the VA stakeholders are implemented. Overall, this improves the service rendered to the Veterans and their beneficiaries throughout the VA. The Identity and Access Management (IAM) Integrated Project Team (IPT) prioritized the Digital Veterans Platform (DVP) system for FY 2017.

The purpose of this document is to provide the high-level design that the DVP team is required to implement in order to comply with the IdS requirements, in accordance with the IAM service requests: SR-985 and SR-3937.

The target audience for this design document is detailed below.

* Veterans Relationship Management (VRM) IAM IPT: This group is responsible for approving and prioritizing all development efforts for IdM within VA.
* DVP Development team: This group is responsible for fulfilling the IdM requirements within the DVP system.
* DVP Business Owner: This group is responsible for business requirement and final approval of DVP application.
* IdS MVI Development Leads: They are responsible for the implementation of MVI requirements to ensure correct integration of the DVP system with MVI approved requirements.
* Health Information Governance/Data Quality: This group is responsible for managing business requirements within Identity Management.

## Purpose

The purpose of this document is to detail the design for the integration between DVP and MVI. Within this context, DVP is said to be the “consuming application” (i.e., it is the application that requests data from MVI and not vice versa). MVI is said to be the “service” (i.e., it provides data to DVP upon request).

Integration between these two systems will require following an established set of protocols. It is the purpose of this document to detail those protocols.

## Identification

This iSDD conforms to VIP (Veteran-focused Integration Process) standards.

## Scope

The DVP solution will be designed to be flexible and enable secure seamless interoperability between VA and commercial applications, enabling advanced analytics to deliver a cohesive Veteran-centered experience both inside and outside VA. The proposed architecture contains a number of strategic, integrated components: Electronic Health Record(s), Operation Management Platforms including: financial, supply chain, and human resource systems, Customer Relationship Management system, Analytics systems, and Application Programming Interface (API) gateway, along with a number of legacy systems that support key business lines in order to provide seamless interoperability with internal and external systems.

Scope and objectives include the following:

* DVP will check the request, correlate the patient ID and retrieve the requested information and return the results to the consumer.

The MVI integration touch points for the DVP integration are the following:

* GetCorrespondingIDs

It should be noted that this integration with MVI will follow the Decentralized Hybrid Integration pattern.

For more information about this pattern, see section 6.2 of the MVI Service Description Document.

## Relationship to Other Plans

The DVP integration is based on the following:

* MVI SSD v3.4
* DVP IAM iRSD

## Methodology, Tools, and Techniques

The DVP team will operate in an Agile (SCRUM) project management methodology to guide and accomplish the goals of this integration in the most efficient way possible. Documentation of all project performance and procedures will be completed in accordance with the Veteran-focused Integration Process (VIP) documentation expectations and requirements. For tracking interface requirements, the development team will translate iRSD requirements. Further details are described in section 2.5.4.

## Policies, Directives, and Procedures

* VIP (Veteran-focused Integration Process) is a Lean-Agile framework that services the interest of Veterans through the efficient streamlining of activities that occur within the enterprise. The VIP Guide can be found at: <https://www.voa.va.gov/DocumentView.aspx?DocumentID=4371>
* National Institute of Standards and Technology (NIST) Special Publication 800-60 standards.
* HL7v3.0 XML over Simple Object Access Protocol (SOAP) Web Services per MVI-SDD.

## Constraints

* If there is a significant change in volume of the current messages as noted in Section 2.5.2, it is expected that DVP stakeholders will notify MVI management before new volume increases.
* This integration's code progression and testing progression will begin in the MVI development environment, and advances to SQA, UAT, Pre-Production, and Production. Change control and quality requirements will be met and approved by IdM subject matter experts in each environment before DVP is allowed to connect to the next environment

Environments and associated mappings (Note: VA Authentication Federation Infrastructure (VAAFI) proxies communication with MVI):

|  |  |  |  |
| --- | --- | --- | --- |
| **MVI** | **VAAFI** | **DVP** | **Comments** |
| DEV | INT | QA |  |
| SQA/ Stage 1A | SQA | Staging-Lab  Lab |  |
| Pre-Prod | Pre-Prod | Staging | MVI Pre-Prod environment contains production data and is PII certified |
| Prod | Prod | Prod | MVI Prod environment contains production data and is PII certified |

## Design Trade-offs

Not Applicable

## User Characteristics

The unique integration Sender ID for DVP is 200DVPG. This identifier SHALL be passed in the asAgent stanza to reflect that DVP is proxying the request for the sender. DVP SHALL also provide the DVP consumer ID.

### User Problem Statement

DVP will call MVI’s GetCorrespondingIDs operation using the identifier provided by the caller to obtain the MVI correlated identifier(s) of the record necessary for retrieval of the required information.

### User Objectives

The integration touch points with MVI shall ensure the following:

* DVP will perform a MVI GetCorrespondingIDs operation to obtain the target corresponding identifier, (e.g., ICN / EDIPI) associated with the provided identifier ((e.g., EDIPI / ICN), needed for searching internally within the VA or externally to correlated systems.

# Background

DVP will be the VA’s data access service, responsible for obtaining and transferring patient data between internal VA and external producers and consumers. The DVP API will provide VA internal and external clients efficient access to up-to-date patient data.

## Overview of the System

## Overview of the Business Process

DVP provides the capacity to locate and discover patient data. The first release of DVP will use the Station Numbers and DFN provided by MVI to obtain VistA data located at VA sites and will provide this data to internal and external consumers as defined by their MOUs.

The interactions between DVP and MVI are depicted in below.

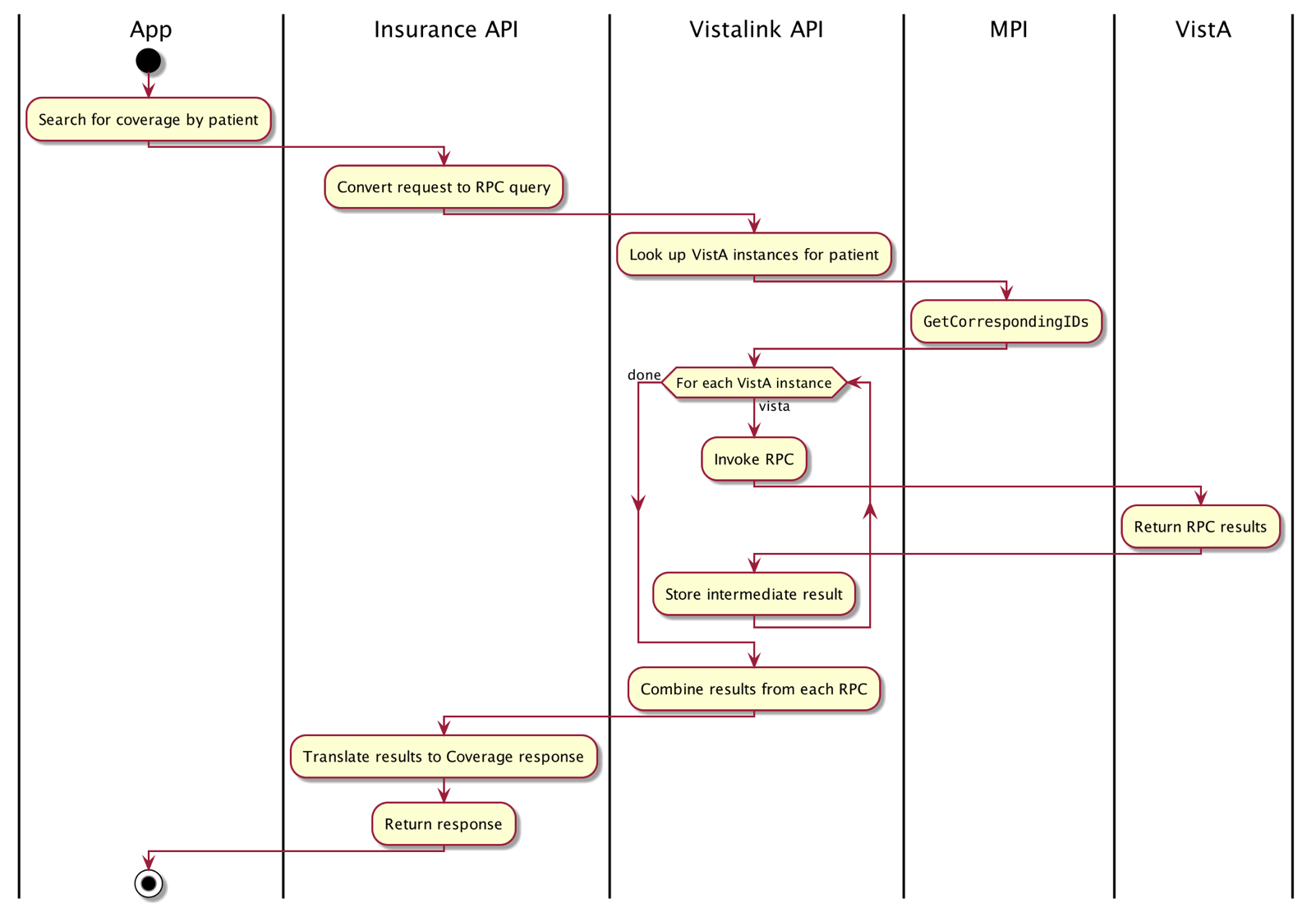


Figure: 1 DVP/MVI Interactions

DVP is architected to be a high performance, high availability, load balanced and failover ready, clustered gateway between the VA and external enterprises.

Additional information is provided in the DVP IAM iRSD.

**DVP Integration Requirements**

DVP includes one call:

* A 1309 call will be made to retrieve the target identifier of the system of record (e.g., Station/DFN for VistA sites, EDIPI for DoD, ICN for VA Enterprise). The DVP request to the MVI GetCorrespondingIDs operation shall contain a fully qualified identifier.

1. Use MVI GetCorrespondingIDs

**Requirements**:

1. DVP, when necessary, shall call MVI’s GetCorrespondingIDs operation to retrieve identifiers associated to a person record.
2. DVP shall follow the business rules defined by IAM Subject Matter Experts (SMEs) when performing a GetCorrespondingIDs operation. (See References Section)

## Business Benefits

The primary business benefit of this MVI integration is for DVP to be able to acquire the identifiers required to obtain the latest Veteran data, so that it can be shared with the appropriate parties to provide effective and efficient care and support to our Veterans.

## Assumptions and Constraints

### Design Assumptions

The key assumptions that have influenced the design of this integration include the following:

* The modifications necessary to facilitate this integration will be on the side of the consuming application. This is because MVI currently exposes the necessary service operations. The consuming application needs to invoke the appropriate operations with the necessary parameters and parse the returned result.
* Components will be kept consistent with simple, precise functionality that is easy to test. Where possible, a test mechanism that can be used repeatedly as needed will be built for each component.
* The modules, functionality, and business components used by DVP are capable of invoking Web Services and parsing the results.
* The client team will engage the VA Authentication Federation Infrastructure (VAAFI) team to coordinate establishing a secure two-way SSL connection for the Web Service calls to take place. (See section 2.5.5 for more information)
* The consuming application will use DVP to initiate the MVI services that have been identified to be in scope, and will therefore use DVP as a proxy to MVI Web Services. DVP requests will therefore use the appropriate identifiers in the AsAgent element, within the sender element of the MVI Web Service request.

<sender typeCode="SND">

<device classCode="DEV" determinerCode="INSTANCE">

<id root="2.16.840.1.113883.3.42.10001.100001.12" extension="***200DVPE***" />

<asAgent classCode="AGNT">

<representedOrganization classCode="ORG" determinerCode="INSTANCE">

<id root="2.16.840.1.113883.4.349" extension="**200DVPG**" />

</representedOrganization>

</asAgent>

</device>

</sender>

* DVP will provide a unique, non-duplicated message identifier for each Web Service request to MVI (the extension attribute of the initial id element in each message) and versionCode value 3.5:

Example:

<PRPA\_IN201309UV02 xmlns="urn:hl7-org:v3" xmlns:ps="http://vaww.oed.oit.va.gov"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="urn:hl7-org:v3 ../../schema/HL7V3/NE2008/multicacheschemas/PRPA\_IN201309UV02.xsd" ITSVersion="XML\_1.0">

<id root="1.2.840.114350.1.13.0.1.7.1.1" extension="***MCID-DVP\_12345***"/>

<creationTime value="20151209150301"/>

<versionCode code="3.5"/>

. . .

* DVP will include an appropriate dataEnterer element in each MVI Web Service request, as described in the MVI SDD that will reflect the user or entity initiating the request.

Format:

<dataEnterer contextControlCode="AP" typeCode="ENT">

    <assignedPerson classCode="ASSIGNED">

          <id extension="*200DVPG*" root="2.16.840.1.113883.4.349">

          </id>

          <assignedPerson determinerCode="INSTANCE" classCode="PSN">

            <name>

              <given>***First Name of User or Entity***</given>

              <family>***Last Name of User or Entity***</family>

            </name>

          </assignedPerson>

    </assignedPerson>

</dataEnterer>

* Other assumptions and considerations as described in the associated iRSD.

### Design Constraints

The primary constraint concerning the design of this integration is that requests sent from the consuming application to the service (i.e., MVI) must conform to the protocol that MVI expects (i.e., an XML document conforming to the HL7 v3.0 schema). See the MVI Service Description Document for more information.

As mentioned previously, custom DVP work will be required to complete this integration. It is unlikely that any coding will need to be performed on the MVI side as MVI already exposes the necessary operation (GetCorrespondingIDs) as a publicly available service operation.

This design will use Web Services. In this case, the consuming application (in this case, DVP) will need to send SOAP (Simple Object Access Protocol) messages, which conform to the HL7v3 schema as documented in the MVI SDD.

## Overview of the Significant Requirements

The design requires that modifications to DVP conform to the Repository Integration pattern.

### Overview of the Significant Functional Requirements

See section 2.5.1 for a complete overview of the functional requirements. The following table provides a high-level overview of DVP business requirements. Note that the ID section will need to be completed by DVP developers/project managers as specific configuration management deltas materialize.

|  |  |  |
| --- | --- | --- |
| **ID** | **Specific Requirement/Synopsis** | **Requirement** |
| 1 | GetCorrespondingIDs | DVP shall have the capability of performing Web Service calls to MVI’s GetCorrespondingIDs operation using the identifier provided (e.g., ICN, EDIPI) to obtain the associated target identifier(s) (e.g., EDIPI, ICN). |

### Functional Workload and Functional Performance Requirements

DVP estimates the following:

* Estimated daily total number of Get Corresponding IDs requests: 15,000 daily (initially)
* Estimated monthly total number of Get Corresponding IDs requests: 450,000 monthly (initially)
* Standard hours of operation:
  + Peak usage times: 24/7/365
* Bulk Processes: No
* Estimated growth rate: 275,000 requests daily within six months

.

### Operational Requirements

Not applicable

### Overview of the Technical Requirements

This integration requires the use of MVI Web Services by DVP, conforming to the HL7v3 schema, as described in the MVI SDD.

The response from the request will conform to the HL7v3 schema, as also described in the MVI SDD. The application must then parse the response and render output/processing accordingly.

### Overview of the Security or Privacy Requirements

The interface will be designed to use HTTPS with bi-directional certificate validation.

This interface will use Mutual Transport Layer Security (TLS) Authentication with VA-issued certificates to identify and authorize server-to-server communications to a specific endpoint identified by a URL where the services reside. TLS also provides the message’s confidentiality and integrity between the endpoints. TLS is set at TLS 1.1+, and Ciphers are set to HIGH, client communication over SSLv3 or TLS1.0 is not supported.

DVP will use VAAFI to communicate to MVI. This data transmission interface will comply with Federal Information Processing Standards (FIPS) Publication 199. The following confidentiality, integrity, and availability categorizations have been laid out with regard to the transfer and handling of data germane to this interface:

* Confidentiality of data: High
* Integrity of data: High
* Availability of data: Moderate

These categories also mirror similar National Institute of Standards and Technology (NIST) Special Publication 800-60 standards.

DVP will utilize TLS authentication with VAAFI servers as provided by the VAMF MVI Common Client. The certificates that will be used to trust and establish the connectivity in the various environments are as follows.

The client servers will use VAAFI to communicate to MVI as shown in Figure 2.



Figure: 2 Security Boundaries

| **Server/Certificate Identification** | |
| --- | --- |
| DVP Production | Root Certificate:   * VA-Internal-S2-RCA1-v1 * VA-Internal-S2-ICA4   Server Certificates:   * production-vistalink.lighthouse.va.gov |
| DVP Staging | Root Certificate:   * VA-Internal-S2-RCA1-v1 * VA-Internal-S2-ICA4   Server Certificates:   * staging-vistalink.lighthouse.va.gov |
| DVP Lab | Root Certificate:   * VA-Internal-S2-RCA1-v1 * VA-Internal-S2-ICA4   Server Certificates:   * lab-vistalink.lighthouse.va.gov |
| DVP Staging-Lab | Root Certificate:   * VA-Internal-S2-RCA1-v1 * VA-Internal-S2-ICA4   Server Certificates:   * staging-lab-vistalink.lighthouse.va.gov |
| DVP QA | Root Certificate:   * TBD   Server Certificates:   * qa-vistalink.lighthouse.va.gov |
| VAAFI Production | Root Certificates:   * ORC ECA Root CA 2 * ORC ECA SW3 |
| VAAFI Lab | Root Certificates:   * ORC ECA Root CA 2 * ORC ECA SW3 |

### System Criticality and High Availability Requirements

DVP is deemed “mission critical” for Patient Safety and Care. MVI services are deemed “Mission Critical.” As a result, the MVI system and the resulting link between the systems will follow the appropriate standards required.

DVP accepts the response during unlikely downtime is handled differently from a Not Found (NF).

DVP and MVI services use the reliability specifications outlined in the following table.

| **Service Availability Level 4** | |
| --- | --- |
| **Description** | Mission Critical Information |
| **Minimum Availability** | 99.9% |
| **Maximum Downtime Per Month** | 43 minutes |
| **Business Value** | Essential to fundamental business operations; outage seriously impairs functioning of business. |
| **Operational Hours** | Required 24 hours a day, 7 days a week. |
| **Significant Outage** | More than 5 minutes of downtime at any time is considered significant. |
| **Outage Impact** | Interruption of service may result in severe financial, regulatory, safety, patient health, or other business issues. |
| **Scheduled Maintenance** | Scheduled maintenance must provide continuity of service availability with minimal interruption. |
| **Availability Design Target** | 99.9% |
| **Component Availability Target** | 99.9% |

### Special Device Requirements

Not applicable

## Legacy System Retirement

Not applicable

# Conceptual Design

## Conceptual Application Design

This design assumes current load levels will remain fairly constant with slow, steady growth over time. If there is a significant change in system use, it is expected that DVP stakeholders will notify MVI management, as outlined in section 1.7.

### Application Context

**DVP** is the VA’s Data Access Service, responsible for transport and temporary storage (caching) of structured and non-structured data between internal VA and external producers and consumers. DVP provides a common access mechanism for Veteran and Service Member electronic record information stored in and outside the VA. DVP removes the risk of using separate systems. Instead, it provides a new platform or modernized system that allows for Veteran and employee increased transparency. The solution will eliminate multiple silos. VADP will integrate with other VA systems including VBMS, EDW, CDW, VistALink, Corporate, and a number of legacy systems.

### High-Level Application Design

DVP calls MVI’s GetCorrespondingIDs operation to retrieve all identifiers associated with an MVI person record.

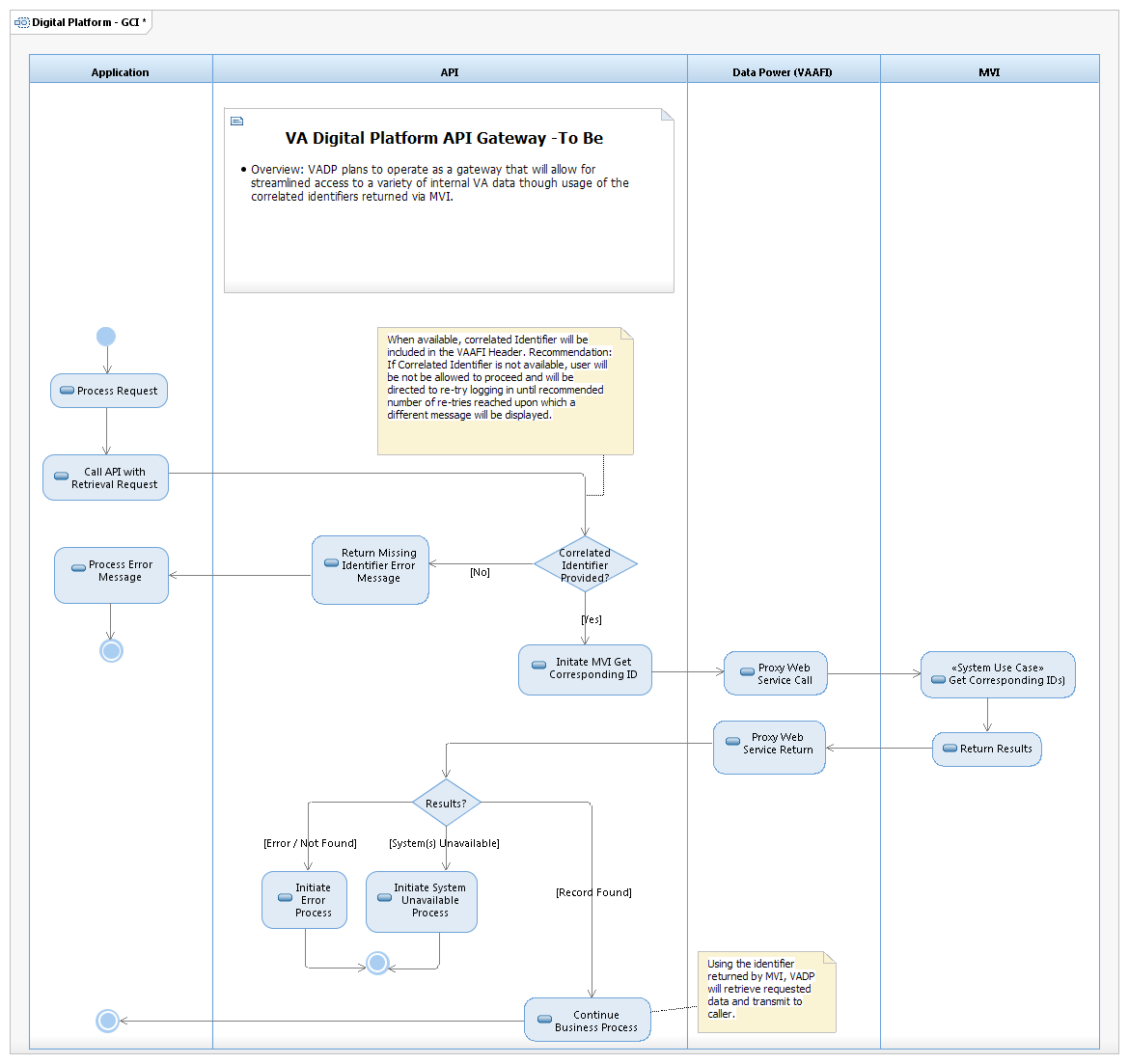


Figure: 3 DVP Generic - To Be Process (zoom to 200%)

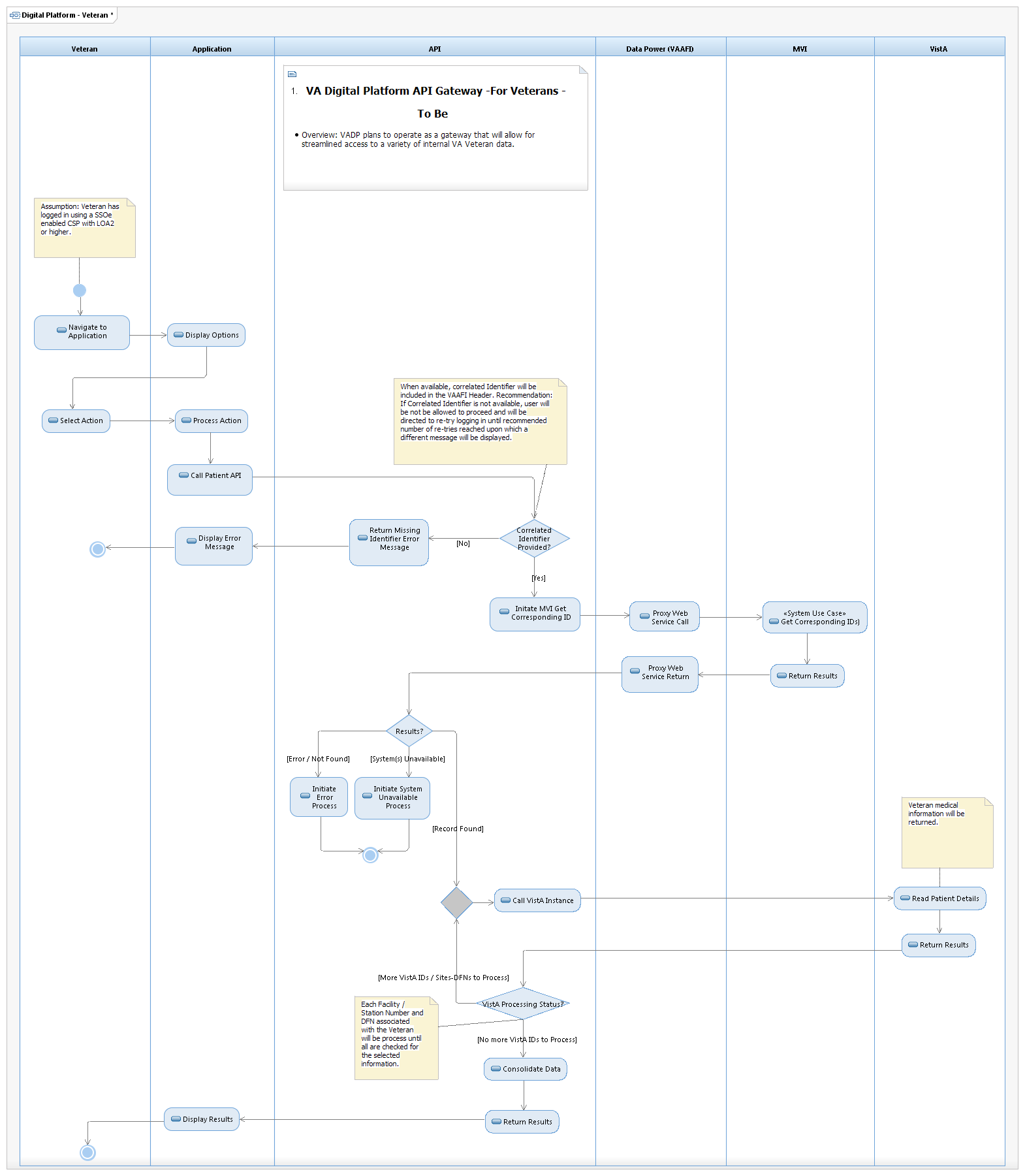


Figure: 4 DVP Process - To Be Initial Release (zoom to 200%)

### Application Locations

DVP servers are hosted by Amazon’s AWS GovCloud.

MVI servers are hosted in Austin, Texas with a failover site in Hines, Illinois.

The use of failover sites should be transparent to end-users. Certain administrative tasks may be necessary, but the fulfillment of these tasks will not impact the design of this system.

## Conceptual Data Design

### Project Conceptual Data Model

This is not applicable, as no data model changes will be required for this integration.

### Database Information

This is not applicable, as no schema changes will be required for this integration.

## Conceptual Infrastructure Design

This is not applicable, as the MVI integration will not require a change to the existing infrastructure.

### System Criticality and High Availability

See section 2.5.6 for information on system criticality and high availability.

### Special Technology

This is not applicable.

### Technology Locations

The DVP app is housed within the Amazon GovCloud.

| **Technology Component** | **Location** | **Usage** |
| --- | --- | --- |
| Development Environments | Amazon GovCloud Dev environment | Development Environment established to continue development in a stabilized on-premise environment more closely reflecting the planned production configuration |
| Production Environment | Amazon GovCloud production environment | Professionally hosted, production-ready deployment environment. |
| Workstations | At all participating VA Facilities nationwide and external | Production end user stations |
| Special Hardware | N/A |  |
| Interface Processors | N/A |  |
| Legacy Mainframe | N/A |  |
| Legacy Application Server | N/A |  |
| Legacy Databases | N/A |  |

MVI servers are hosted in Austin, Texas with a failover site in Hines, Illinois.

| **Technology Component** | **Location** | **Usage** |
| --- | --- | --- |
| Development Environments | AIDE-managed infrastructure at Verizon NAP of the Capital Region, Culpeper, VA | Initial development environments established to allow maximum accessibility to development efforts and customer inspection. Environments include: Development (Dev), Development Test (DevTest), Software Quality Assurance (SQA), User Acceptance Test (UAT), Demo, and ProdDevTest. |
| Production Environment | Verizon NAP of the Capital Region, Culpeper, VA | Professionally hosted, production-ready deployment environment. |
| Workstations | At all participating VA Facilities nationwide | Production end user stations |
| Special Hardware | N/A |  |
| Interface Processors | N/A |  |
| Legacy Mainframe | N/A |  |
| Legacy Application Server | N/A |  |
| Legacy Databases | N/A |  |

### Conceptual Infrastructure Diagram

DVP initiates communications to the MVI application server in Austin, Texas via SOAP using web services.

In the event that MVI in Austin, Texas is unable to respond, failover redirects the request to the MVI data center in Hines, Illinois.



## Hardware Architecture

DVP runs on Amazon Web Services (AWS) GovCloud, and therefore exists solely on virtualized hardware running in the cloud. The AWS product that provides virtualized hardware is called the Elastic Compute Cloud, or EC2.

DVP integration’s code and testing progression begins in the MVI development environment and advances to the MVI SQA environment, and MVI production environment. Change control and quality requirements are met and approved by IdM SMEs in each environment before DVP is allowed to advance/connect to the next environment.

## Software Architecture

It should be noted that the DVP software architecture is service-oriented architecture (SOA) compliant.

## Communications Architecture

The DVP /MVI integration will be handled via MVI Web Service. The operation of concern here is:

* GetCorrespondingIDs

It should be noted that the communications between DVP and MVI will need to be handled using a secure channel - specifically, the channel requires VAAFI. For more information, see section 2.5.5.

The unique system ID that will be used by this integration process is 200DVPE with 200DVPG as the AsAgent, which identifies the Web Service requesting entity to MVI.

# Data Design

Database design changes will not be required for this integration.

## Database Management System Files

Not applicable to DVP

## Non-Database Management System Files

Not applicable to DVP

# Detailed Design

## Hardware Detailed Design

No changes to the existing hardware will be required.

## Software Detailed Design

As mentioned previously, DVP will use Web Services to initiate communication with MVI, via the Common Client proxy. In so doing, the protocol used for both the request and response will be SOAP.

Included in the SOAP message (sometimes called the SOAP envelope) is what is commonly referred to as the “payload.” The next sections discuss the expected payload for the required operations: [1309-GetCorrespondingIDs.]

The request/response pair that will be used in this DVP-MVI integration is the 1309/1310 pair, GetCorrespondingIDs operation.

For more information about these operations, including detailed field descriptions, see the MVI Service Description document.

### Code Impacts

None.

### Source ID Definition

Message requests to the MVI Web Service will be made by using a fully qualified (meaning four-part) Source ID. Fully qualified Source ID can be found in the extension attribute of the ID element. Per HL7v3 specifications, the Source ID components are as follows:

If Integration Control Number (ICN) implementation:

* ICN – (Ex: 1001179396V504445)
* Identifier Type – Value will be “NI”
* Assigning Authority (optional) –Value will be “USVHA”
* Assigning Facility (optional) – Value will be “200M”

If Electronic Data Interchange Person Identification (EDIPI) Source ID implementation:

* EDIPI
* Identifier Type – Value will be “NI”
* Assigning Authority – Value will be “USDOD”
* Assigning Facility – Value will be “200DOD”

If Security Identifier (SEC ID) (long-term solution) Source ID implementation:

* SEC ID
* Identifier Type – Value will be “PN”
* Assigning Authority – Value will be “USDVA”
* Assigning Facility – Value will be “200PROV”

Other identifier types may be used, as described in the MVI SDD.

### Integration with MVI – Sample Queries / Responses

Sample queries and responses for applicable MVI operations will be provided separately. Additionally, the MVI Service Description Document should be referenced for complete explanations for every field in the GetCorrespondingIDs Message.

## Communications Detailed Design

Communications between DVP and MVI will use HL7v3 messaging, as previously mentioned.

#### MCCI\_IN000002UV01 – Response Sample

The response is one of those listed in the following table.

|  |  |  |
| --- | --- | --- |
| **Acknowledgment**  **typeCode**  **code attribute** | **queryAck**  **queryResponseCode**  **code attribute** | **Meaning** |
| AA | OK | Application Acknowledgement/Results returned (happy path) |
| AE | AE | Application Error //Unknown Key Identifier Not Found (0 results) |
| AE | AE | Application Error |

The following responses will be returned under the specified conditions.

|  |  |
| --- | --- |
| **Condition** | **Response** |
| MVI Down | HTTP Error 404 |
| MVI Database Down | AE Acknowledgment typeCode code attribute |

# External Interface Design

The interface between DVP and MVI is a software service (Web Service interface). The Web Service calls will use the SOAP protected by TLS between the servers after mutprual authentication. Data transferred under this agreement will include Veteran’s personal identity information.

## Interface Architecture

The diagram in Figure 5 displays a detailed description of the protocols used in inter-tier communication.

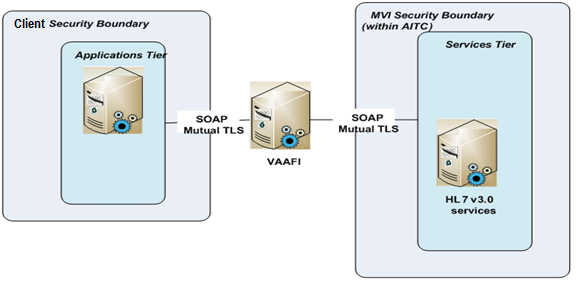


Figure: 5 Interface Architecture

## Interface Detailed Design

It is anticipated and expected that DVP will provide multiple testing environments. These include Development, SQA, and Pre-Production (i.e. UAT) environments.

This integration’s code and testing progression will go from MVI DEV, to MVI SQA, to MVI UAT, then to MVI Production. Change control and quality requirements will be met and approved by IdM subject matter experts in each environment before DVP is allowed to connect to the next environment.

# Human-Machine Interface

## Interface Design Rules

Not Applicable

## Inputs

DVP callers will be able to provide inputs for any of the acceptable input types accepted by the MVI GetCorrespondingIDs operation.

## Outputs

The GetCorrespondingIDs operation will return responses as described in the MVI SDD, and sample requests/responses as provided separately to DVP.

## Navigation Hierarchy

For the DVP Navigation hierarchy, please see figure 1 DVP/MVI Interactions.

# System Integrity Controls

There is no change from existing DVP architecture.

# Appendix A

## Abbreviations and Definitions

| **Abbreviation** | **Definition** |
| --- | --- |
| ADR | Administrative Data Repository |
| API | Application Program Interface |
| CCPC | Consolidated Co-payment Processing Center |
| CCR | Code Change Request |
| Common Client | VAMF common Web Service for MVI access |
| COTS | Commercial Off-the-Shelf |
| CRUD | Create, Retrieve, Update, Delete |
| DOB | Date of Birth |
| ECA | External Certification Authority |
| EDIPI | Electronic Data Interchange Person Identification (EDIPI) |
| FIPS | Federal Information Processing Standards |
| GUI | Graphical User Interface |
| HEC | Health Eligibility Center |
| IAM | Identity and Access Management |
| ICN | Integration Control Number |
| IdM | Identity Management |
| IdS | Identity Services |
| IPT | Integrated Project Team |
| JDOC | Javadoc |
| MPI | Master Patient Index |
| MVI | Master Veteran Index |
| NIST | National Institute of Standards and Technology |
| ORC | Office of Regulatory Compliance |
| PMAS | Project Management Accountability System |
| SEC ID | Security Identifier |
| SME | Subject Matter Expert |
| SOA | Service-Oriented Architecture |
| SOAP | Simple Object Access Protocol |
| SR | Service Request |
| SSN | Social Security Number |
| TLS | Transport Layer Security |
| UI | User Interface |
| VA | Veterans Affairs |
| VAAFI | VA Authentication Federation Infrastructure |
| VAMC | Veteran Affairs Medical Center |
| VHA | Veterans Health Administration |
| VRM | Veterans Relationship Management |
| ES | Enrollment System |

## Glossary

| **Term** | **Definition** |
| --- | --- |
| HL7 | Health Level 7 - A message standard used to support hospital workflows |
| IdM System | Provides a system that creates and maintains an enterprise-wide unique identity for all persons of interest for VHA |
| Veteran | Any person who served honorably on active duty in the armed forces of the United States |

## Requirements Traceability Matrix

The latest version of the DVP RTM will be maintained by MVI Analyst Team. This is located at the following link: <https://wiki.mobilehealth.va.gov/pages/viewpage.action?pageId=47449183>

## Packaging and Installation

The solution components will be managed within the Stash Repository established for the Connected Care environment. The DVP repository is located at the following link: [https://coderepo.mobilehealth.va.gov/projects/VADVP](https://coderepo.mobilehealth.va.gov/projects/VAGDX)

## Design Metrics

DVP adheres to VA’s agile SDLC in the development of the application. The development effort is driven by the creation and implementation of user stories and user story epics. Each user story is assigned story points which indicate the relative complexity of the tasks involved.

For example a story or epic may be assigned story points of 1, 3, 5, 8, or 13 ranked from lowest to highest in terms of relative complexity. As each sprint is carried out, progress is tracked by the number of user stories and story points completed. This provides a rolling metric which quickly shows business owners, managers, and team members the progress of the project in terms of work completed and work that is still outstanding.

## Required Technical Documents

There is no change from existing DVP technical documents.

# Approval Signatures

REVIEW DATE: <date>

SCRIBE: <name>

Signed:

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

Tim Robinson, DVP Project Manager Date

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

Danny Reed, Enterprise Architecture, IdS Date

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

Jeff Podolec, IAM Program Manager Date

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

Mike Mims, Integrated Product Team (IPT) Chair Date