# VistA Integration Adapter (VIA)

# 1.0

# User Guide

June 2016
Department of Veterans Affairs
Office of Information and Technology (OI&T)

## Revision History

| Date | Revision | Description | Author |
|---|---|---|---|
| 06/28/16 | 1.0 | Initial VIP Version | Engility |

## Artifact Rationale

Per the Veteran-focused Integrated Process (VIP) Guide, the User's Guide is required to be completed  prior to Critical Decision Point #2 (CD2), with the expectation that it will be updated as needed. A User Guide is a technical communication document intended to give assistance to people using a particular system, such as VistA end users. It is usually written by a technical writer, although it can also be written by programmers, product or project managers, or other technical staff. Most user guides contain both a written guide and the associated images. In the case of computer applications, it is usual to include screenshots of the human-machine interfaces, and hardware manuals often include clear, simplified diagrams. The language used is matched to the intended audience, with jargon kept to a minimum or explained thoroughly. The User Guide is a mandatory, build-level document, and should be updated to reflect the contents of the most recently deployed build. The sections documented herein are required, if applicable to your product.

# Table of Contents

# Tables

# Figures

# 1. Introduction

Due to increased requests for external access to the Veterans Health Information Systems and Technology Architecture (VistA) system, the Department of Veterans Affairs (VA) has defined a need for a set of web services to facilitate those requests. Two Medical Domain Web Services (MDWS) applications were deployed to meet the need. However, these already-fielded products now require unanticipated corrective maintenance and do not meet current Information Technology (IT) supportability requirements in terms of security, stringent performance, scalability, and maintainability.

Therefore, a plan to replace MDWS is underway with the VistA Integration Adapter (VIA) middleware application. It is a solution that can be leveraged across the enterprise in support of the VA. VIA is designed to correct the aforementioned deficiencies and re-engineer the MDWS applications to a Java Enterprise Edition (JEE) platform, according to the Performance Work Statement (PWS) and Business Requirements Document (BRD).

Additionally, VIA facilitates the convergence of JMeadows and MDWS services into a common service platform, thereby providing the existing functionality of MDWS to client applications. VIA creates a set of common web services that allow consuming applications to securely retrieve data from VistA systems. It is intended to perform as an adapter that abstracts consuming applications from the technical details of the VistA computing environment. VIA utilizes a modularized design approach that separates generic-infrastructure functionality from business-oriented functionality, and thereby allows consuming applications to connect to VistA systems. VIA exposes the business interfaces using the Simple Object Access Protocol (SOAP) standard. This common service platform establishes a shared infrastructure for accessing a diverse set of information across geographic locations and spanning multiple technologies, and ultimately establishes a set of data access services within the VA's extensive application landscape.

## 1.1. Purpose

The purpose of the VIA User Guide is to provide technical information to system administrators, IT support staff, and other authorized users. It will be updated as needed in subsequent releases.

## 1.2. Document Orientation

The document orientation is shown below in Sections 1.2.1 through 1.2.6.

### 1.2.1. Organization of the Manual

The major sections of the User Guide are shown in the following table:

**Table 1: Organization of User Guide**

| Section | Description |
|---|---|
| 1.0 | Introduction |
| 2.0 | System Summary |
| 3.0 | Getting Started |

| Section | Description |
|---|---|
| 4.0 | Using the Software |
| 5.0 | Troubleshooting |
| 6.0 | Acronyms and Abbreviations |

The target audience for this guide includes system administrators, IT support staff, and other authorized users.

### 1.2.2. Assumptions

This guide was written with the assumption that there will be no direct users because VIA is middleware.

### 1.2.3. Coordination

VIA code is released through Austin Information Technology Center (AITC). The production code is identified at AITC as VIA. Coordination is done by the use of AITC standard request practices. A Request for Change (RFC) and a communication to the team is done to get on the AITC release calendar.

The following table shows the current sustainment team supporting VIA:

**Table 2: VIA Sustainment Team**

| Name | Role(s) |
|---|---|
| REDACTED | Application Manager |
| REDACTED | Build Manager (Release Manager) |
| REDACTED | Database Administrator |
| REDACTED | Linux System Administrator |

### 1.2.4. Disclaimers

The VIA software and documentation disclaimers are shown below in Sections 1.2.4.1 and 1.2.4.2.

#### 1.2.4.1. Software Disclaimer

This software was developed at the Department of Veterans Affairs (VA) by employees of the Federal Government in the course of their official duties. Pursuant to title 17 Section 105 of the United States Code this software is not subject to copyright protection and is in the public domain. VA assumes no responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. We would appreciate acknowledgement if the software is used. This software can be redistributed and/or modified freely if any derivative works bear some notice that they are derived from it, and any modified versions bear some notice that they have been modified.

#### 1.2.4.2. Documentation Disclaimer

The appearance of external hyperlink references in this manual does not constitute endorsement by the Department of Veterans Affairs (VA) of the Web site or the information, products, or services contained therein. The VA does not exercise any editorial control over the information you may find at these locations. Such links are provided and are consistent with the stated purpose of the VA.

### 1.2.5. Documentation Conventions

This section is not applicable.

### 1.2.6. References and Resources

The following are references and resources for the VIA User Guide:

- VIA System Design Document
- VIA Business Requirements Document
- VIA Interface Control Document
- VIA Requirements Specification Document

## 1.3. National Service Desk and Organizational Contacts

*REDACTED*

# 2. System Summary

The VIA system provides a secure, standards-based platform for delivery of Electronic Health Record (EHR) data from VistA and other sources to the requesting systems. VIA accomplishes this by retrieving data from numerous distributed data sources, aggregating the data, and providing it to the calling application. The existing services may be designed initially for a specific client system, but they are also used by other client applications with similar data needs.

This is made possible by establishing multiple, independent, de-coupled web services for the data retrieval capability and for providing core business functionality. The web services are deployed in a standards-based, consistent fashion and are deployed in a JEE container. The JEE container specified in the Task Order response (Oracle WebLogic) is highly scalable, widely used within VHA, and provides high performance suitable for enterprise-class application needs.

## 2.1. System Configuration

The relevant services and the interfaces associated with VIA services are shown in Figure 1 below. These include the VIA environment, service requestor systems, and data source systems from a logical perspective.

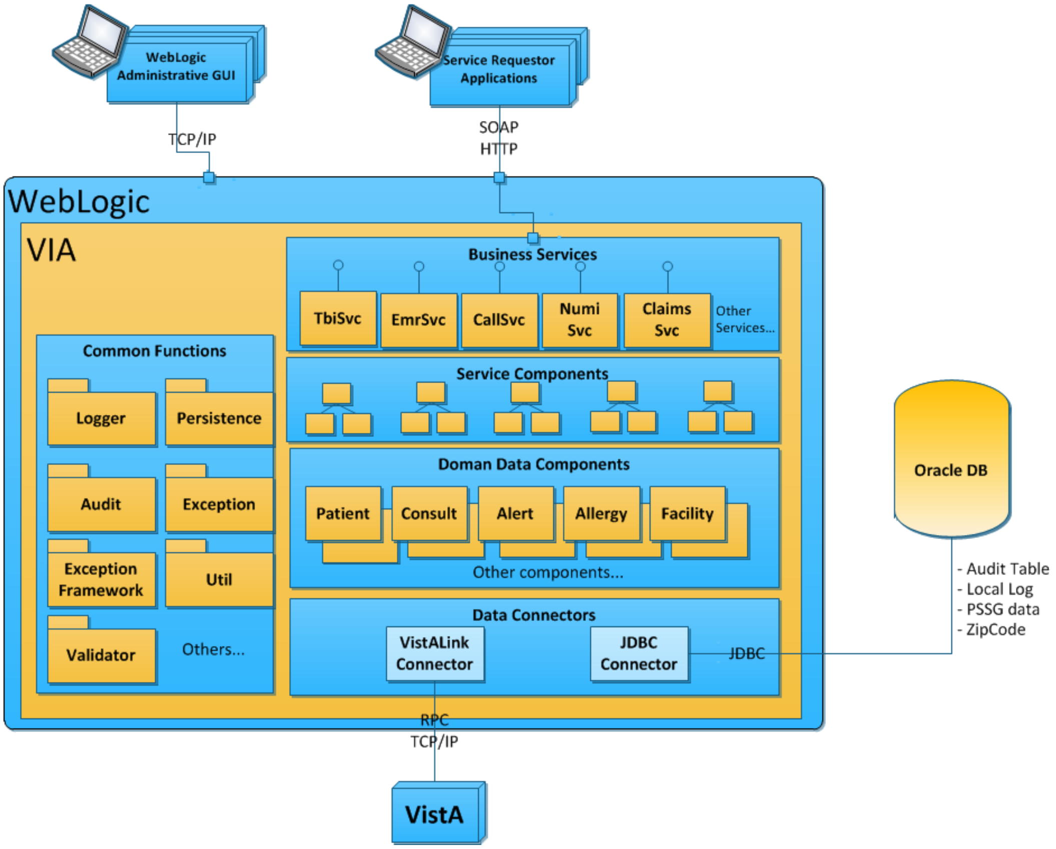

**Figure 1: VIA High-Level Application Design**

The To-Be environment utilizes a local DB (database) for internal application use, (e.g., logging and auditing). The DB must be available at all times for the system to continue operations successfully. If the DB becomes unavailable for any reason, the application services using the DB cannot continue to operate. System availability will be impacted because the existing DB must be taken offline for routine maintenance and system upgrades.

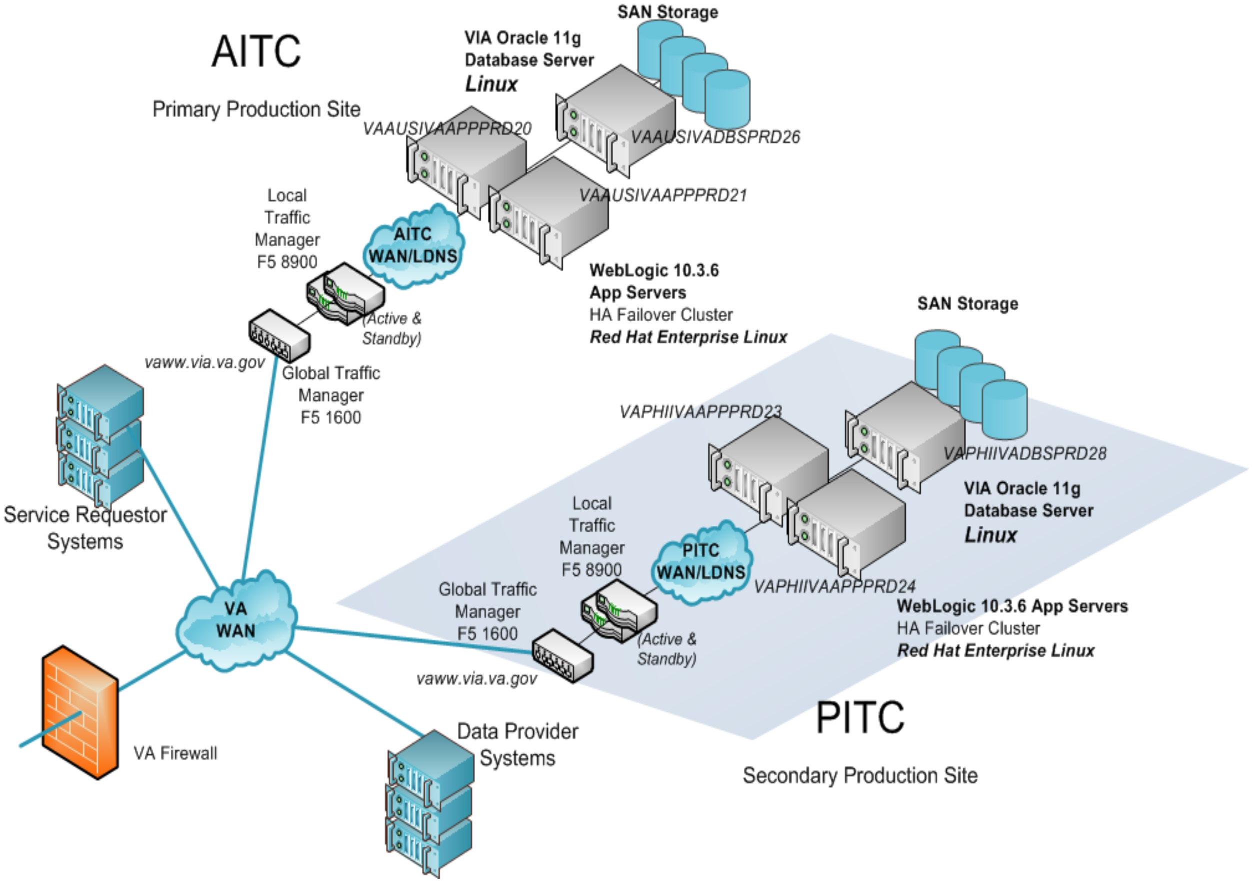

**Figure 2: VIA Production Hardware Configuration**

There are three primary scenarios that the production configuration must be able to handle:

1. Failure of a single application server – rules configured in the LTMs will remove the Internet Protocol (IP) address from the pool of available servers. The remaining active application servers will continue processing.
2. Failure of a single DB server – Since there is a single DB server at each site, and the DB server must be available for processing, then VIA processing for the site cannot continue. Rules configured in the Global Traffic Manager (GTM) will disallow all VIA processing for that site by removing the IP address from the pool of available servers. The remaining active application servers will continue processing
3. Failure of all application servers at one data center – rules established in the GTM will disallow VIA processing for that site.

## 2.2. Data Flows

Figure 3, below, illustrates the interface data flow between systems involved in the VIA service interactions. The interfaces can be categorized as 1) Front-End Systems, which act as clients who request data and 2) Back-End systems, which act as data provider systems.

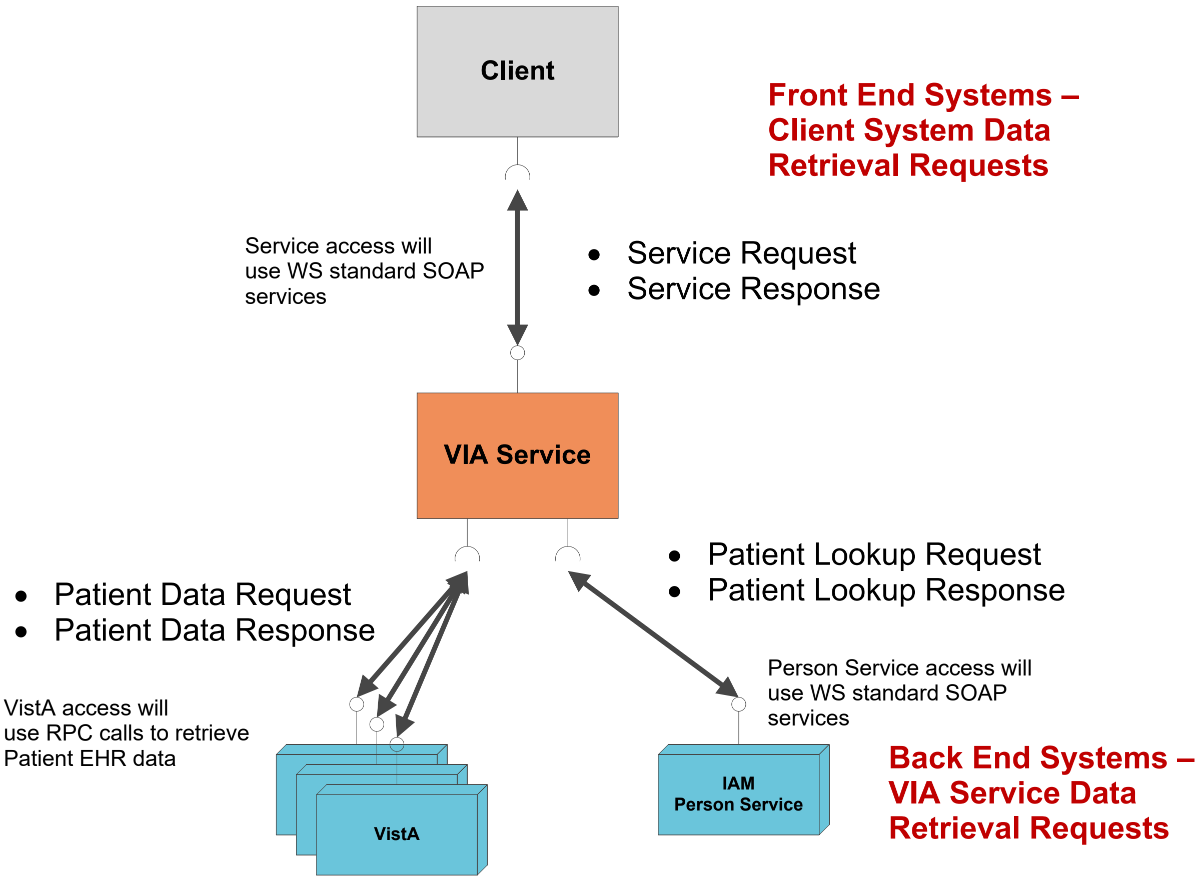
Client
Front End Systems – Client System Data Retrieval Requests
Service access will use WS standard SOAP services
Service Request
Service Response
VIA Service
Patient Data Request
Patient Data Response
Patient Lookup Request
Patient Lookup Response
VistA access will use RPC calls to retrieve Patient EHR data
Person Service access will use WS standard SOAP services
VistA
IAM Person Service
Back End Systems – VIA Service Data Retrieval Requests

**Figure 3: VIA Interface Architecture**

## 2.3. User Access Levels

Access to manipulate DB structure and storage will be restricted to administrative-level users. These users will be the security operations staff and DB administrators, who will either make use of the security information or perform maintenance support of the DB infrastructure.

All application connectivity to the audit logs will be through a single user's profile that will be configured to read and write data privileges. The DB user will have no permission to update or permanently remove data from the audit logs. This approach aligns with VA Directive 6500 on the access to audit data.

All application connectivity to the exception logs will be through a single user who will have full read, write, update, and delete access to the exception logs. The exception logs are for the sole purpose of monitoring and debugging middleware software behavior, so the requirement to reset the exception logs will occur with fair regularity.

## 2.4. Continuity of Operation

VIA is middleware and will be hosted at AITC. VIA will adopt all of AITC's contingencies in the event of emergency, disaster, or accident.

## 2.5. Getting Started

To configure and install the supporting software for VIA, please refer to the following:

- VIA Installation Manual
- VIA Interface Control Document
- VIA Interface Control Document Annex A
- VIA Interface Control Document Annex B
- VIA Interface Control Document Annex C
- VIA Interface Control Document Annex D
- VIA Interface Control Document Annex E
- VIA Interface Control Document Annex F
- VIA Interface Control Document Annex G

If your application is currently stateful, then the following documents should be helpful in the conversion from Stateful to Stateless:

VIA Stateful to Stateless Conversion Guide

## 2.6. Logging On

VIA is middleware; therefore, the consuming application (i.e., CPRS) is the source for log on.

## 2.7. System Menu

This section is not applicable.

## 2.8. Changing User ID and Password

VIA is middleware; therefore, the consuming application is the source for log on. Although there is no password needed for VIA specifically, VIA will pass the authorization used to log into the consuming application. Changes to a user id or password will be handled by the consuming application.

## 2.9. Exit System

When the user exits the consuming application, VIA is also exited.

## 2.10. Caveats and Exceptions

This section is not applicable.

# 3. Using the Software

You must assign to users and Clinical Application Coordinators (CACs) the following secondary menu option. This option provides access to the VIA context created in the patch.  If this option is not assigned, a user will not be able to utilize VIA. The VIA project team recommends working with your site's IT staff to determine which users should receive the menu option.

The VIAB WEB SERVICES OPTION provides access to all RPC calls that VIA uses. Therefore, work with your site's CACs to gather a list of VIA users and determine which views each user needs. Only assign the views that each user needs.

1. Log in to VistA.
2. At the Select OPTION NAME prompt, type EVE and then press the <Enter> key.
3. At the Choose 1-5 prompt, type the number 1 (for EVE Systems Manager Menu) and press the <Enter> key.
4. At the Select Systems Manager Menu Option prompt, type User Management and press the <Enter> key.
5. At the Select User Management Option prompt, type edit (for Edit an Existing User) and press the <Enter> key.
6. At the Select NEW PERSON NAME prompt, type the user's name using the following format: lastname, firstname. Press the <Enter> key.
7. Press the <Down Arrow> key to highlight the Select SECONDARY MENU OPTIONS field. (Type a question mark (?) to see a list of the secondary options that are currently assigned to the user.)
8. In the SECONDARY MENU OPTIONS field, type VIAB WEB SERVICES OPTION, and then press the <Enter> key.
9. At the Are you adding …as a new SECONDARY MENU OPTIONS (the…for this NEW PERSON)? No// prompt, type Yes and press the <Enter> key.
10. Press the <Enter> key again to accept this new option.
11. In the SYNONYM field, type a synonym for the option (optional). Press the <Enter> key.
12. Press the <Enter> key to close the COMMAND field and return to the Select SECONDARY MENU OPTIONS field.
13. Press the <Down Arrow> key to move through the Edit Existing User dialog. At the end of each page, type the letter n in the COMMAND field to enter the following page.
14. Stop on page 3.
15. Check the user's person class, which appears on page 3, to make sure the user's person class is active.
16. If the user's person class is not active, select an active person class for the user.
17. When you have entered all applicable secondary menu options and verified that the user has an active person class, type the word Exit in the COMMAND field.
18. At the Save changes before leaving form (Y/N)? prompt, type the letter Y and press the <Enter> key.

# 4. Troubleshooting

REDACTED.

## 4.1. Special Instructions for Error Correction

This section describes the exception handling policies for VIA services. VIA exposes data from multiple data sources in the form of services provided to external systems (service requestors). These service requestors have their own policies for handling errors.

VIA is responsible for fulfilling the service request as long as the system is operational.  The general error handling policies are provided below.

- Web Services (WS) standard SOAP exception handling will be used for communicating exceptions back to the service requestor. This will make use of SOAP faults to provide information regarding the exception.
- Standard Java exception handling will be used within service application code.
- When critical errors occur that place the system in an unrecoverable and unstable state, the application will terminate. These types of errors include code errors and missing resources (e.g., internal DB, configuration file, and runtime dependency). The rationale behind taking the extreme step of terminating the application is that if a critical error is raised and users are allowed to continue, all of the subsequent processing could be compromised. This could result in incorrect processing and corrupted data being stored in the DB.
- An error that occurs because of a data source being unavailable is a special condition. When this happens, the situation should be considered a non-fatal error and processing should continue so that the service request is fulfilled, with any available data successfully retrieved along with information specifying which data sources were unavailable. A use-case scenario for this would be retrieving EHR data for a patient with records at four VHA VistA sites. If one of the four sites was unavailable and the connection was to time out, the service would provide data from the three sites and indicate which site was unavailable.
- Exceptions will be caught and handled at the highest-level controlling entity (typically the Controller classes) in the method call chain. At this top level, exceptions will be logged using the Log4J logging mechanism. Encapsulating the logging functions at the highest level within the code structure is intended to avoid duplicate log entries. No other methods within the call chain should catch the error, unless it could add information relevant to resolving the error.
- Within the service or framework, catch blocks must not “eat” the exception. This refers to code where the exception is caught with an empty block or a block that simply displays information and allows the application to continue. This hides the error and could potentially leave the application in an unreliable state.
- Error information presented to the user must be meaningful to the user and must be information on which the user can act. If the user can take no corrective action, a message will be displayed indicating that the user should contact the Help Desk for Tier 1 support.
- All exceptions will be logged using VIA’s common Log4J logging framework.

- The Fail-Fast principle will be applied in all application code. The Fail-Fast principle states that errors will be identified as close to the source of the problem as possible and a suitable exception will be thrown. The rationale behind this principle is that when an error condition occurs within the code, it should be identified as soon as possible so that: 1) errors are not propagated to other parts of the code, and 2) the cause can be identifies and resolved quickly. If errors are not handled when they occur, the system may continue to process them and they can become more difficult to resolve.

# 5. Acronyms and Abbreviations

Below is a table of acronyms and/or abbreviations used in this document and the meaning of each. Additional acronyms and definitions can be found in the OIT Master Glossary.

**Table 3: Acronyms and Abbreviations**

| Term | Definition |
|---|---|
| **AITC** | Austin Information Technology Center |
| **BRD** | Business Requirements Document |
| **CPRS** | Computerized Patient Record System |
| **CAC** | Clinical Application Coordinator |
| **DB** | database |
| **EHR** | Electronic Health Record |
| **GTM** | Global Traffic Manager |
| **IAM** | Identity and Access Management |
| **ICD** | Interface Control Document |
| **IT** | Information Technology |
| **JEE** | Java Enterprise Edition |
| **MDWS** | Medical Domain Web Services |
| **MS** | Milestone |
| **PITC** | Philadelphia Information Technology Center |
| **PWS** | Performance Work Statement |
| **SOAP** | Simple Object Access Protocol |
| **RESTful** | Representational State Transfer |
| **RFC** | Request for Change |
| **RPC** | Remote Procedure Call |
| **SOAP** | Simple Object Access Protocol |
| **TRM** | Technical Reference Manual |
| **VHA** | Veterans Health Administration |

| Term | Definition |
|---|---|
| **VIA** | VistA Integration Adapter |
| **VISTA** | Veterans Health Information Systems and Technology Architecture |

# 6. Appendix

This section is not applicable.

# 7. Index

This section is not applicable.