VistA Services Assembler Phase 2 (VSA-P2)

Requirements Specification Document



Department of Veterans Affairs

Office of Information and Technology (OI&T)

Product Development (PD)

Documentation Version 5.7

March 2016

Revision History

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

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 3/08/2016 | 5.7 | Address VA VSA feedback | Apex – M. Waterloo |
| 2/17/2016 | 5.7 | Extensive fixes to minimize exchanges between APEX and VA. Track Changes were Accepted.  APEX changed staff from Carol Jones to S. Greenacre and information was not communicated to new staff. | VA and Engility |
| 01/21/2016 | 5.7 | Technical Edit. Recreated the list of tables and figures TOC. Section A6, added content to a table.  Updated Table 3 with the missing Requirement numbers and Expected Delivery Dates.  Added S-Cure charts to section 4. | C. Perkins,  S. Greenacre |
| 01/19/2016 | 5.6 | Update to address VA comments.  Updated to current ProPath Template. | S. Greenacre |
| 11/30/2015 | 5.5 | Tech Edits   * Accepted all Track Changes and removed comments. * Re-formatted entire document for 508 compliance. * Added alt txt to all tables and figures for 508 compliance. * Updated all hyperlinks per the 508 compliance guidelines. | C. Perkins |
| 11/23/2015 | 5.4 | Tech Edits   * Updated the Delivery Column * Added contract verbiage to Introduction, Purpose and Scope | C. Jones  J. Yaple |
| 11/19/2015 | 5.3 | Tech Edits   * Replaced VSA VistA.js with VSA * Replaced SOA with SOAP | C. Jones |
| 11/06/2015 | 5.2 | Tech Edits   * Replaced VSA VistA with VSA VistA.js * Replaced SOAP with SOA * Updated footer * Added justification to deleted requirements * Deleted a number of requirements with justification that text related to a specific technical implementation that is no longer current. * Updated several requirements to reflect scope. * Replaced “VSA Wizard” with VistA.js Wizard functionality. * Noted that Figures 2, 3 and 4 are no longer current. * Corrected previous substitution of SOAP with SOA. This update will be validated with Government. * Updated delivery column values and 2.6.15 requirements apportionment. | C. Jones  J. Yaple |
| 07/28/2015 | 5.1 | Tech Edits   * Added note below Figure 8. * Corrected footer formatting in Section 8. * Renamed document to follow latest VSA naming conventions. | E. Laurel  T. Blom |
| 07/24/2015 | 5.0 | Tech Edits   * Baselined document based on latest updates and feedback from the VSA development team. * Modified styles and formatting throughout. * Added “Draft” watermark on all pages. This should be removed prior to final document approval and signoff. * Updated document for latest documentation standards. * Reviewed, added, and updated hyperlinks throughout. * Reviewed Word document for Section 508 conformance. * Reviewed and edited content for grammar, punctuation, and spelling. | T. Blom  E. Laurel |
| 07/16/2015 | 4.0 | Removed language on the use of technology specific to Java, JavaScript, etc., and other edits provided. | J. Garcia  V. Rodriguez  E. Laurel |
| 10/20/2014 | 3.0 | Tech Edits   * Final baseline document for signatures in PDF. * Extensive rewrite and reformat of entire document as per K. Cox review. | VSA Development Team |
| 10/03/2014 | 2.0 | Tech Edits   * Baselined document to Version 2.0. * Reviewed updated sections in document, since last baseline. * Reviewed document for formatting and style updates. * Updated all document references, table of contents, table of figures, etc. * Added Office of General Counsel software disclaimer statement to Section 5. * Made minor grammar, punctuation, and spelling updates as needed. * Checked document for Section 508 conformance. * Removed “Draft” and watermark from the document for final PDF and digital signatures. | T. Blom  E. Laurel |
| 10/01/2014 | 1.7 | * Added spec for “Federated” calls item 2.6.4d. * Added requirement for Federated calls, see PS2.9.8. | E. Laurel  J. Garcia |
| 09/22/2014 | 1.6 | * Updated Business Needs table in Section 2.6.15 with the most up to date “status”. * Added security requirements for PII and PHI as SS2.13.7 in Section 2.13. | E. Laurel  J. Garcia  V. Rodriguez  L. Janko |
| 09/18/2014 | 1.5 | Added numbering (e.g., 2.2a, etc.) to non-functional specs not containing tables; this will better link requirements to RRC identifiers. | E. Laurel |
| 09/10/2014 | 1.4 | Added content to Section 2.13, “Security Specifications.” | J. Garcia  E. Laurel |
| 08/26/2014 | 1.3 | * Revised scope in Section 1.2. * Added RRC requirement identifier numbers in Section 2.6; corresponds with new entries of requirements in RRC for VSA-P2. * Added Goals/Objectives table from BRD in Section 1.2.1. * Removed all comments. | E. Laurel |
| 08/14/2014 | 1.2 | Tech Edits   * Updated Figure 2, Figure 3, and Figure 4; based on new footer added to VSA Wizard screens. * Updated Figure 7 and Figure 8. * Changed WebLogic Release “12.1.2” to “12.x” in Table 17. * Added Registered trademark ® to references in Table 17 and Table 18. | T. Blom  M. Ogi |
| 08/07/2014 | 1.1 | Tech Edits   * Updated caption for Figure 7. * Updated Figure 7 and added Figure 8 based on feedback from J. Garcia. * Updated acronyms in Table 18. * Removed the moved content from “Status” column, to “OWNR” column and then deleted the “Status” column. Also, changed references from “POC” to “Phase 2” throughout the table. | T. Blom  J. Garcia |
| 08/04/2014 | 1.0 | Tech Edits   * Reviewed document for formatting and style updates. * Updated all document references, table of contents, table of figures, etc. * Made minor grammar, punctuation, and spelling updates as needed. * Checked document for Section 508 conformance. * Baselined document to Version 1.0. | T. Blom  E. Laurel |
| 07/31/2014 | 0.5 | Revised multiple sections; MultiDivisional, Performance, Scope Integration, Reliability, Usability, GUI, Communications, User Interface sections | J. Garcia  E. Laurel |
| 07/17/2014 | 0.4 | * Revised multiple sections; added comments * Accepted inserts and deleted comments | G. Smullen  J. Garcia  M. Ogi  E. Laurel |
| 06/23/2014 | 0.3 | Revised Federation of Web Services; added content for Pre and Post Action actions | E. Laurel |
| 05/28/2014 | 0.2 | Revised to modify scope and added functionality for Representational State Transfer (RESTful) services | E. Laurel |
| 04/08/2014 | 0.1 | Initial VSA 1.0.0 Phase 2 document: Based on original Prototype; Proof-of-Concept document.  Revised content to reflect requirements for Increment 1 (RESTful Services). | E. Laurel |

Table of Contents

[1. Introduction 1](#_Toc441483201)

[1.1. Purpose 2](#_Toc441483202)

[1.2. Scope 3](#_Toc441483203)

[1.3. References 6](#_Toc441483205)

[2. Overall Description 7](#_Toc441483206)

[2.1. Accessibility Specifications 7](#_Toc441483207)

[2.2. Business Rules Specification 7](#_Toc441483208)

[2.3. Design Constraints Specification 7](#_Toc441483209)

[2.4. Disaster Recovery Specification 8](#_Toc441483210)

[2.5. Documentation Specifications 8](#_Toc441483211)

[2.6. Functional Specifications 8](#_Toc441483212)

[2.6.1. VSA Wizard Functionality—User Interface 8](#_Toc441483213)

[2.6.2. VSA Wizard Functionality—Functions 23](#_Toc441483214)

[2.6.3. VSA Runtime Environment Components 25](#_Toc441483215)

[2.6.4. Federation 27](#_Toc441483216)

[2.6.5. Pre/Post Logic Processing 29](#_Toc441483217)

[2.6.6. Exception and Error Handling 31](#_Toc441483218)

[2.6.7. Architectural Principles 32](#_Toc441483219)

[2.6.8. Supporting geographically distributed VA System Topology 33](#_Toc441483220)

[2.6.9. Non-Functional Requirements 36](#_Toc441483221)

[2.6.10. Capacity, Load, and Performance 38](#_Toc441483222)

[2.6.11. Security 39](#_Toc441483223)

[2.6.12. User Identity Propagation 40](#_Toc441483224)

[2.6.13. Availability - COOP/Disaster Recovery 42](#_Toc441483225)

[2.6.14. Documentation 43](#_Toc441483226)

[2.6.15. Apportioning of Requirements 44](#_Toc441483227)

[2.7. Graphical User Interface (GUI) Specifications 44](#_Toc441483229)

[2.8. Multi-divisional Specifications 48](#_Toc441483230)

[2.9. Performance Specifications 48](#_Toc441483231)

[2.10. Quality Attributes Specification 49](#_Toc441483232)

[2.11. Reliability Specifications 49](#_Toc441483233)

[2.12. Scope Integration 49](#_Toc441483234)

[2.13. Security Specifications 49](#_Toc441483235)

[2.14. System Features 49](#_Toc441483236)

[2.15. Usability Specifications 49](#_Toc441483237)

[3. Purchased Components 49](#_Toc441483238)

[4. Estimation 49](#_Toc441483239)

[5. Approval Signatures 53](#_Toc441483241)

[Appendix A: Non-Functional Requirements 54](#_Toc441483242)

[System Performance Reporting Requirements 54](#_Toc441483243)

[Operational Environment Requirements 54](#_Toc441483244)

[Documentation Requirements 55](#_Toc441483245)

[Implementation Requirements 55](#_Toc441483246)

[Data Protection/Back-up/Archive Requirements 56](#_Toc441483247)

[Levels for Disaster Recovery 56](#_Toc441483248)

[Data Quality/Assurance Requirements 56](#_Toc441483249)

[User Access/Security Requirements 56](#_Toc441483250)

[Usability/User Interface Requirements 57](#_Toc441483251)

[Conceptual Integrity 57](#_Toc441483252)

[Availability 57](#_Toc441483253)

[Interoperability 57](#_Toc441483254)

[Manageability 58](#_Toc441483255)

[Performance 58](#_Toc441483256)

[Reliability 58](#_Toc441483257)

[Security 59](#_Toc441483258)

[Supportability 59](#_Toc441483259)

[Usability 60](#_Toc441483260)

[Documentation 60](#_Toc441483261)

[Appendix B: Acronym List and Glossary 60](#_Toc441483262)

[Acronyms 60](#_Toc441483263)

[Definitions 63](#_Toc441483264)

List of Tables

[Table 1: VSA Primary and Failover Sites 4](#_Toc441483265)

[Table 2: VSA Project Scope: Goals and Objectives 5](#_Toc441483266)

[Table 3: VSA Wizard Functionality-User Interface Requirements 8](#_Toc441483267)

[Table 4: VSA Wizard Functionality-Functions 24](#_Toc441483268)

[Table 5: VSA Runtime Environment 25](#_Toc441483269)

[Table 6: VSA Federation Requirements 27](#_Toc441483270)

[Table 7: VSA Pre/Post Logic Processing Requirements 30](#_Toc441483271)

[Table 8: VSA Exception and Error Handling Requirements 31](#_Toc441483272)

[Table 9: VSA Architectural Principles Requirements 32](#_Toc441483273)

[Table 10: VSA System Topology Requirements 33](#_Toc441483274)

[Table 11: VSA Non-Functional Requirements 36](#_Toc441483275)

[Table 12: VSA Capacity-Load and Performance Requirements 38](#_Toc441483276)

[Table 13: VSA Security Requirements 39](#_Toc441483277)

[Table 14: VSA User Identity Propagation Requirements 40](#_Toc441483278)

[Table 15: VSA Availability-Continuity of Operations (COOP)/Disaster Recovery (DR) Requirements 42](#_Toc441483279)

[Table 16: VSA Documentation Requirements 43](#_Toc441483280)

[Table 17: Levels for Disaster Recovery 56](#_Toc441483281)

[Table 18: Acronyms 60](#_Toc441483282)

[Table 19: Definitions 63](#_Toc441483283)

List of Figures

[Figure 1: VSA “Pre/Post” Logic Model 30](#_Toc441489830)

[Figure 2: VSA RPC Wizard-Select RPC 45](#_Toc441489831)

[Figure 3: VSA RPC Wizard-Edit Definition 46](#_Toc441489832)

[Figure 4: VSA API Browser-Show RESTified RPCs 46](#_Toc441489833)

[Figure 5: VSA API Browser-Select A RESTified RPC 47](#_Toc441489834)

[Figure 6: VSA API Browser-Testing The Rest Call 47](#_Toc441489835)

[Figure 7: VSA API Browser-REST Call Results 48](#_Toc441489836)

[Figure 8: Cumulative Probability (“S-curve”) Chart—Project Duration 51](#_Toc441489837)

[Figure 9. Cumulative Probability (“S-curve”) Chart—Project Effort 52](#_Toc441489838)

# Introduction

This Requirements Specifications Document (RSD) is developed for the VistA Services Assembler Phase 2 (VSA-P2) project supported by the Office of Information and Technology (OI&T)/Product Development (PD) division. The majority of Department of Veterans Affairs (VA) healthcare computing involves Veterans Health Information System and Technology Architecture (VistA) and applications integrated with VistA. The increasing need to integrate VA computing systems across business domains and with external systems, such a Department of Defense (DoD), has taken precedence in bridging the technical gap between legacy VistA and disparate technologies of Commercial-Off-The-Shelf (COTS) products. In addition, the VA healthcare Service Oriented Architecture (SOA) currently faces challenges in:

* Exposing existing and new VistA business functionality and data securely and efficiently.
* Exposing VistA functionality by hiding the distributed location of all the existing VistA systems.
* Providing services for use by other systems and technologies.
* Decoupling consuming systems from the implementation details of VistA.

OI&T/PD has defined VSA as a new design approach for providing SOA-compliant integration between VistA applications and external systems.

VSA facilitates a solution that provides the ability to:

* Expose VistA functionality as SOA-compliant Web services directly.
* Federate (route) VistA functionality requests to one or more VistA systems.
* Integrate legacy system VistA to external systems and applications.

With VistA as an SOA service provider, these services use VistA application business logic directly; thus, eliminating redundancy and leveraging existing, tested VistA application functionality. The VSA solution will provide:

* Utilities to automate the creation of VistA SOA services platform.
* VistA SOA Services Platform (runtime) environment necessary to host SOA services and execute the corresponding VistA functionality to one or more VistA systems.

VSA addresses SOA objectives and provides the ability for VistA applications and services to become:

* Economically extensible
* Maintainable
* Individually replaceable
* Fully compliant

The VSA design pattern has been identified as the preferred architectural design for SOA-based systems going forward. The intent is to address the functional requirements of connectivity designs (e.g., Medical Domain Web Services [MDWS], VistA Integration Adaptor [VIA], Clinical Data Services [CDS], etc.) for the integration of VistA with external systems/applications. As such, VSA will need to satisfy the consumer needs currently supplied by those approaches, as well as significantly enhance the following:

* VistA security
* System performance
* Sustainability

The VSA solution has adopted a phased, iterative approach to requirements definition, development, and implementation of new capabilities. This approach allows VA to:

* Finalize a discrete set of requirements.
* Implement VSA software capabilities.
* Roll out utilities and software functionalities.

This document contains a list and description of requirements that depict the functionality within the iteration of planning, analysis, design, and software development for the VSA Phase 2 project. The requirements in this document can be used to construct, configure, and test the software to support the VSA solution.

## Purpose

The purpose of this RSD is to:

* Define the requirements for developing the VSA solution, which includes:
* Framework functional requirements
* Technical requirements
* Security requirements
* System requirements
* Provide traceability between the VSA product features and business needs with guidance from the Business Requirements Document (BRD).
* Function as the VSA master requirements specification, providing a baseline to support testing of the proposed toolset, which includes the VSA Wizard functionality and its associated runtime environment used for service execution.

It is important to note that these specifications will be revised as needed for subsequent iterations of this document, and will be published as part of the Project Management Accountability System (PMAS) planning phase for the next increments of the VSA project.

The functional system *shall* cover the comprehensive, scope for VSA. It is anticipated that as part of each project increment, a detailed requirements analysis will be performed. This analysis will include a detailed elaboration of the requirements as they pertain to the respective scope of each increment. Updated versions of the RSD will be published to address the defined scope for each respective increment and to include detailed use cases. These detailed use cases will be developed incrementally under each requirements analysis phase of the project.

To gain an explicit understanding of requirements needed to support VSA, this document is intended for use by the following:

* VHA business representatives
* Program and project management
* Architects
* Development
* Software Quality Assurance (SQA) analysts
* Other project stakeholders

## Scope

The scope for VSA Phase 2 is to:

* Transition the Platform prototype to a deployable initial operating capability (IOC) pilot package status; rollout the IOC package to up to 11 pilot sites; and provide enhancements, source code, installation procedures, documentation and training. In order to transition the Platform prototype to a deployable IOC pilot package, the Contractor shall further extend the prototype to meet VA Enterprise-grade requirements, 508 compliance, patient safety mandates, as well as undergo extensive performance and user acceptance testing given the clinical nature of the solution involving direct patient care and safety.
* Create a software development tool, VistA Services Assembler Wizard, to simplify the creation of VistA SOA services. The type of services that the Wizard generates and the types of services that the runtime environment can host and run are Representational State Transfer (REST) and Simple Object Access Protocol (SOAP) service.
* Create, model (in runtime), develop, and deploy a VistA SOA Services Platform for IOC national deployment to host and execute VistA SOA services using Government provided test and development infrastructure(s). The platform includes a Federating Platform and corresponding service execution components that allows aggregation of functionality in multiple VistA instances. The platform components and service execution provide a simple view that exposes functionality in multiple VistA systems as individual VistA SOA services.

Identify and produce multiple sample VistA SOA business services as “reference implementations.” In the near term, VSA will use virtual integration environments provided by the Enterprise Development Environment (EDE) teams. However, to include development and testing environments for building, integrating, and testing VSA components, VSA -P2 will progress to environments in the following locations:

**IOC Sites:[[1]](#footnote-1)**

* Boise, ID
* Hudson Valley, NY
* Memphis, TN
* Hampton Roads, VA
* South Texas (San Antonio), TX

**For Federation:**

Table : VSA Primary and Failover Sites

| Primary Sites | Failover Sites |
| --- | --- |
| Austin Information Technology Center (AITC) | Philadelphia Information Technology Center (PITC) |

As VSA implements connectivity and software distribution models that are not currently mainstream in the VA, the second area of scope involves:

* Documentation of policy, process, and technical user guidance regarding the optimal definition of VSA services.
* Distribution of such services to the various systems and technical environments necessary for the function of VSA services. For example:
* Enterprise Messaging Infrastructure (eMI)
* VSA Run-Time
* HyperText Transfer Protocol (HTTP) to VistA MUMPS Binding/ Integration with patient lookup service
* Identity and Access Management (IAM)

The VSA project team will collaborate with business representatives and software development initiatives to identify sample VistA SOA business services as a “reference implementation” to critically test and to demonstrate the functionality of VSA utilities. However it is not the scope or intent of the VSA effort to identify and produce a collection of “services” for subsequent organizational use. This activity would be performed by software development initiative teams that use VSA utilities to produce VistA-based services.

The business benefits of a full-featured VSA include, but are not limited to, the following:

* Provides complete compliance and integration with VA Enterprise Design Patterns for SOA architecture and infrastructure.
* Produces VistA SOA services that are authoritative and non-duplicated.
* Enhances system maintainability, sustainability, and application replace ability.
* Produces technically standardized and highly maintainable SOA services.
* Cost effective, low organizational impact, with minimal staff orientation and training.
* Enables rapid assembly of existing VistA functionality as SOA-compliant services by non-programmers.
* Facilitates rapid VistA SOA service development through an incremental approach.
* Allows for incremental deployment and consumption of VistA SOA services.
* Addresses SOA objectives with enhanced security and system performance (once IAM Security Assertion Mark-up Language (SAML) token process is implemented).
* Provides immediate value before full eMI infrastructure is in place.
* Alleviates “vendor dependence” concerns while exponentially expanding VistA extensibility and “open source” product development opportunities.
* Significantly expands VistA extensibility and system integration opportunities.

Table : VSA Project Scope: Goals and Objectives

| Goal/Objective | Desired Outcome | Measurement | Impact |
| --- | --- | --- | --- |
| Provide utilities for automating the creation of VistA SOA services and supporting their use. | VistA SOA Services Platform utilities that support automated VistA SOA service creation and execution. | VistA SOA services are consumable by external systems/applications.  **NOTE:** Web services will be managed by the Enterprise Messaging Infrastructure (eMI), which will serve as a Web servicer registry and repository. | VistA can be easily integrated with external systems/applications through SOA-compliant architecture. |
| Enable the exposure of existing VistA data and methods as SOA-compatible Web services. | Use of VSA utilities to create business services that provide business value to VA mission and objectives. | Current VA development and system integration initiatives are consumers of VistA SOA services implemented by VSA. | Advanced state of SOA architecture implementation in the VA and associated software functionality and sustainability benefits. |
| Abstract VistA system developers and integrators from multiple technology orientation. | Consuming systems / applications can be integrated without extensive knowledge of VistA technology. | VA software development staff members are able to rapidly create and deploy VistA SOA Services with minimal re-training. | Improved use of existing staff skills, leveraging of previously implemented VistA application functionality. |
| Improve VistA development “time to market” and overall system sustainability. | Automated, standardized production of VistA SOA service components. | Development time for VistA related services is notably reduced supporting efficient software delivery. | Provides cost effective methodology for the integration of VistA with external systems. |

## References

The content of this document is based, in part, on the PMAS Requirements Specification Document (RSD) template. Additional documents referenced in the creation of this requirements specification include:

* ProPath SharePoint site
* Project Management Accountability System (PMAS) Program Management Document Repository
* VistA Services Assembler Business Requirements Document (BRD)
* Authentication, Authorization, and Audit Design Pattern: Internal User Identity Authentication

VA Enterprise Design Patterns:

* Documents Design Patterns SOA
* VA Enterprise Shared Services (ESS) Family of Services/Initiative Management (FoSIM)
* Enterprise Shared Services (ESS) Service Oriented Architecture (SOA) Methodology / VA Family of Services Initiatives Management Fosim
* VA Handbook 6500
* VA Directive 6500
* FIPS PUB 199, Standards for Security Categorization of Federal Information and information Systems

National Institute of Standards and Technology (NIST) Special Publication (SP):

* [National Institute of Standards and Technology Special Publication: 800-30](http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800_30_r1.pdf) rev 1
* [National Institute of Standards and Technology Special Publication: 800-37](http://csrc.nist.gov/publications/nistpubs/800-37-rev1/sp800-37-rev1-final.pdf)
* [National Institute of Standards and Technology Special Publication: 800-53](http://csrc.nist.gov/publications/PubsSPs.html)
* [National Institute of Standards and Technology Special Publication: 800-60 rev 1](http://csrc.nist.gov/publications/nistpubs/800-60-rev1/SP800-60_Vol1-Rev1.pdf)
* [National Institute of Standards and Technology Special Publication: 800-53 rev 4 (supersedes Revision 3)](http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53Ar4.pdf)
* [National Institute of Standards and Technology Special Publication: 800-111](http://csrc.nist.gov/publications/nistpubs/800-111/SP800-111.pdf)
* [Fast Healthcare Interoperability Resources (FHIR) Specification](https://www.hl7.org/fhir/overview.html)
* VA Acronyms

# Overall Description

The following specifications delineate the requirements necessary for the development of the VSA components, detailing the overall specifications for the utilities and infrastructure elements that support the auto-generation and execution of Web services. These components cover a small range of VSA business needs, functional and non-functional requirements, and will serve as input to subsequent system development lifecycle activities and artifacts, such as architecture definition, system design, and system test plans. VSA provides an easy to use and administer VSA Wizard functionality with the goal to provide authorized information and legacy services to VA consumers requesting access to Veteran data. Many of the requirements are applicable to multiple sections of the Requirements Specification document (RSD).

The requirements for the VSA project are entered and managed into the VA Rational Team Concert (RTC) tool via Rational Requirements Composer (RRC) and Rational Quality Manager (RQM). They will be used to create and maintain full traceability between new/updated requirements and the associated change requests from initial discovery to resolution. Artifacts can be found at:

* Rational Requirements Management - VSA (RM)
* Rational Quality Manager - VSA (QM)
* Rational Change and Configuration Management - VSA (CM)

The VA VSA Product Development (PD) SharePoint site will be used as a repository and knowledge management tool for project-related documents, such as the VSA Project Charter, Project Management Plan, VSA architectural and technical artifacts can be found on the VA VSA PD Project SharePoint site.

## Accessibility Specifications

Refer to requirements 2.6.9.433 and 2.6.9.434 in Section 2.6.9 for Section 508 compliance.

## Business Rules Specification

Business rules are implemented by the remote procedure calls (RPCs) themselves. VSA responds with error messages to the requesting client application. Implementation of organizational business rule logic allows client and provider application developers to make requests to VSA *without* requiring knowledge of rules that govern access to RPCs, querying RPCs, and aggregating and parsing RPC results. It is important to note that consuming applications are responsible for their own adherence to all VA business rules.

## Design Constraints Specification

The VSA design is informed and driven by architectural design patterns and artifacts. Each section of the architecture document derives conceptual application, data, and infrastructure designs and an overview describing the full functionality of the utilities to be provided. Flowcharts, screen shots, swim lanes, pattern sketches, and other graphic depictions are used to show how the VSA solution supports the overall design of the VSA Wizard functionality and infrastructure components.

Technical overview documents are found on the VSA’s Project SharePoint site.

## Disaster Recovery Specification

Refer to Section 2.6.13.

## Documentation Specifications

Refer to Section 2.6.14.

## Functional Specifications

The VSA Wizard functionality and its runtime infrastructure components allow service developers to choose the VistA functionality that are exposed as a Web service.

1. Service developer using the VSA Wizard functionality:
2. Selects the RPC.
3. Creates a service definition file.
4. Generates a service for a specific runtime.
5. Developer tests the service on its specific runtime.

Original requirements are in normal text. *Modified or new requirements are in* *italics*. **Justifications** explain any changes to the requirements set.

“Expected Delivery” indicates the planned delivery Phase, Increment, and Sprint. Every effort will be made to maintain this, but as Agile development dictates, priorities may be shifted by the customer needs and by any dependencies with development or outside interfaces as needed. For the most accurate representation of the planned delivery, please use VA JAZZ and the Requirements Traceability Matrix (RTM) in the VA SharePoint site.

### VSA Wizard Functionality—User Interface

The VSA Wizard functionality enables service developers to either identify or create new VistA functionality as a Web service.

Table : VSA Wizard Functionality-User Interface Requirements

| Req. # | Requirement | Expected Delivery |
| --- | --- | --- |
| 2.6.1.1 | *The VSA Wizard functionality shall include a secure VA Intranet Web interface used by VA MUMPS (M) developers or system integrators at design-time.*  (Original) The VSA Wizard web application *shall* include a secure VA Intranet web interface to be used by VA MUMPS (M) developers or system integrators at design-time.  **Justification for Revision:** Added text for clarity | P2.2.2 |
| 2.6.1.1.1 | The VSA Wizard web application shall include a secure VA Intranet web interface to be used by VA MUMPS (M) developers or system integrators at design-time to comply with the VistA Authentication “M4A” minimum attributes.  **Justification for Removal:** Combined and addressed by 2.6.1.3 | Removed |
| 2.6.1.1.2 | The VSA Wizard web application shall include a secure VA Intranet web interface to be used by VA MUMPS (M) developers or system integrators at design-time to comply with the VistA Authentication using SAML tokens.  **Justification for Removal:** Combined and addressed by 2.6.1.1.3 | Removed |
| 2.6.1.1.3 | *The VSA wizard functionality shall include a secure VA Intranet web interface to be used by developers or system integrators at design-time that complies with VA 6500 security requirements and uses VA ESS IAM authentication/authorization.*  (Original) The VSA Wizard web application *shall* include a secure VA Intranet web interface to be used by VA MUMPS (M) developers or system integrators at design-time to comply with the VistA Authentication via Transport Layer Security (TLS) connections.  **Justification for Revision:** Reworded for clarity | P2.3 |
| 2.6.1.1.4 | *The VSA Wizard functionality shall include a secure VA Intranet Web interface to be used by VA MUMPS (M) developers or system integrators at design-time via eMI connections.*  (Original) The VSA Wizard web application *shall* include a secure VA Intranet web interface to be used by VA MUMPS (M) developers or system integrators at design-time to comply with the VistA Authentication via Enterprise Service bus (ESB) connections.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.1.2 | *VSA shall provide a Graphical User Interface (GUI).*  (Original) VSA *shall* provide VSA Wizard Graphical User Interface (GUI).  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.1.3 | The VSA Wizard service assembler shall implement user authentication for the use of the VSA Wizard. This is consistent with 2.6.1.1.3 | P2.2.2 |
| 2.6.1.4 | *The VSA Wizard functionality shall enable service developers to identify/define existing VistA logic as the business logic for a Web service.*  (Original) The VSA Wizard main page *shall* enable service developers to identify/define existing VistA logic as the business logic for a web service.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.1.5 | *The VSA Wizard functionality shall allow a user to search through a list of existing Service Descriptors for editing.*  (Original) The VSA Wizard main page *shall* allow a user to search through the file system of the Federating Platform to select an existing Service Descriptor file.  **Justification for Revision:** Reworded for clarity | P2.3 |
| 2.6.1.6 | *The VSA Wizard main page shall provide the user with the ability to:*   * *create a new Service Descriptor* * *browse through service descriptors to allow selection* * *enter the name of the service descriptor manually* * *select the correct service descriptor and click on it for editing*   (Original) The VSA Wizard main page *shall* provide the user with the ability to create a new Service Descriptor file.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.1.7 | The VSA Wizard point and click interface shall allow navigation through the file system to allow selection of the correct file path.  **Justification for Removal:** Addressed by 2.6.1.6 | Removed |
| 2.6.1.8 | The VSA Wizard point and click interface shall allow manual entry of the path to the file.  **Justification for Removal:** Addressed by 2.6.1.6 | Removed |
| 2.6.1.9 | The VSA Wizard point and click interface shall allow the user to select the correct Extended Markup Language (XML) file and click on the file to start editing.  **Justification for Removal:** Addressed by 2.6.1.6 | Removed |
| 2.6.1.10 | The system shall display a “Text Description Box” detailing Simple Object Access Protocol (SOAP) Service Information fields from the Service Descriptor. Includes:   * Service Name * Version * Runtime Namespace * Service Namespace | P2.4 |
| 2.6.1.11 | The system shall display a “Text Description Box” detailing Representational State Transfer (REST) Service Information from the Service Descriptor. Includes:   * Service Name * Version * Runtime Namespace * URL Path * Life Cycle * Produces * Consumes | P2.2.2 |
| 2.6.1.12 | The system shall provide a Service Descriptor form to define the service. | P2.2.2 |
| 2.6.1.13 | The system shall provide the user with the ability to select a type of service (SOAP or REST).  **Justification for Removal:** This requirement was combined with 2.6.1.14 | Removed |
| 2.6.1.14 | The system shall display service types to select from, at the time of selecting a Service Descriptor the system should show the correct service type associated with the service (when in Edit mode):   * SOAP * REST | P2.5 |
| 2.6.1.15 | *The VSA Wizard selection field shall provide the ability to input the following SOAP information:*   * *Web Service Name.* * *Web Service Version* * *Service Namespace (prevent name collisions)* * *Run-time Namespace (prevent name collisions)*   (Original) The VSA Wizard selection field *shall* provide the ability to input SOAP information for a Service Name.  **Justification for Revision:** Reworded for clarity | P2.5 |
| 2.6.1.16 | The VSA Wizard selection field shall provide the ability to input SOAP information for the Version of the web service.  **Justification for Removal:** This requirement was combined with 2.6.1.15 | Removed |
| 2.6.1.17 | The VSA Wizard selection field shall provide the ability to input SOAP information for the Runtime Namespace.  **Justification for Removal:** This requirement was combined with 2.6.1.15 | Removed |
| 2.6.1.18 | The system shall display SOAP-specific fields on the VSA Wizard Service Descriptor form. | P2.5 |
| 2.6.1.19 | The VSA Wizard selection field shall provide the ability to input SOAP information for a Service Namespace to allow multiple applications to use the VSA Wizard without naming collisions.  **Justification for Removal:** This requirement was combined with 2.6.1.15 | Removed |
| 2.6.1.20 | The system shall display REST-specific fields on the VSA Wizard Service Descriptor form. | P2.3 |
| 2.6.1.21 | The VSA Wizard selection field shall provide the ability to input REST information for a Uniform Resource Locator (URL) path to the resource of the data. | P2.2.2 |
| 2.6.1.22 | The system shall provide the user with the ability to select a REST life cycle entry from a drop-down list. | P2.3 |
| 2.6.1.23 | The system shall provide the user with the ability to select a REST “Produces” entry from a drop-down list. | P2.3 |
| 2.6.1.24 | The system shall provide the user with the ability to select a REST “Consumes” entry from a drop-down list. | P2.3 |
| 2.6.1.25 | The system shall provide the user with the ability to search for an existing RPC based on an RPC Name. | P2.2.2 |
| 2.6.1.26 | The system shall provide the user with the ability to input information in the following field:   * RPC Name | P2.2.2 |
| 2.6.1.27 | The system shall provide a drop-down list for the user to select from a list of RPCs returned from VistA. | P2.2.2 |
| 2.6.1.28 | The system shall provide the user with the ability to view the details of a selected RPC. | P2.2.2 |
| 2.6.1.29 | The system shall provide the user with the ability to auto-generate an operation from a selected RPC. | P2.2.2 |
| 2.6.1.30 | The system shall provide the user with the ability to modify the auto-generated SOAP operation. | P2.5 |
| 2.6.1.31 | The system shall provide the user with the ability to add one or more SOAP-specific operations to one RPC. | P2.5 |
| 2.6.1.32 | The system shall provide the user with the ability to add one SOAP operation to multiple RPCs (aka “Service Composition”). | P2.5 |
| 2.6.1.33 | The system shall provide the user with the ability to input SOAP information for an RPC Name. | P2.5 |
| 2.6.1.34 | The system shall provide the user with the ability to input SOAP information for an Operation Name. | P2.5 |
| 2.6.1.35 | The system shall provide the user with the ability to select a SOAP Response Type from a drop-down list correlating a return type value of the data response from VistA that can be represented in any of the following formats:   * String * JavaScript Object Notation (JSON) * List * Map | P2.5 |
| 2.6.1.36 | The system shall provide the user with the ability to edit a SOAP-specific operation. | P2.5 |
| 2.6.1.37 | The system shall provide the user with the ability to delete a SOAP-specific operation, and all its associated input parameters. | P2.5 |
| 2.6.1.38 | The system shall display to the user a warning message confirming a deletion of a SOAP operation. | P2.5 |
| 2.6.1.39 | The system shall provide the user with the ability to add one to multiple SOAP-specific parameters if necessary that define the input parameters for an operation. | P2.5 |
| 2.6.1.40 | The system shall provide the user with the ability to input SOAP information for a Name of a parameter. | P2.5 |
| 2.6.1.41 | The system shall provide the user with the ability to input SOAP information for a Type of parameter. | P2.5 |
| 2.6.1.42 | The system shall provide the user with the ability to select a SOAP-specific Parameter Type from a drop-down list that defines the type of parameter. | P2.5 |
| 2.6.1.43 | The system shall provide the user with the ability to delete a SOAP-specific input parameter. | P2.5 |
| 2.6.1.44 | The system shall display to the user a warning message confirming a deletion of a SOAP-specific input parameter. | P2.5 |
| 2.6.1.45 | The system shall provide the user with the ability to edit a SOAP-specific input parameter. | P2.5 |
| 2.6.1.46 | The system shall provide the user with the ability to expand or collapse the SOAP-specific input parameters information for any operation. | P2.5 |
| 2.6.1.47 | The system shall provide the user with the ability to add one to multiple REST-specific operations that correspond to a VistA remote procedure to be invoked. | P2.2.2 |
| 2.6.1.48 | The system shall provide the ability to display input REST Operation information for an RPC Name. | P2.2.2 |
| 2.6.1.49 | The system shall provide the ability to display input REST Operation information for an Operation Name. | P2.2.2 |
| 2.6.1.50 | The system shall provide the ability to display input REST Operation information for a Response Type. | P2.2.2 |
| 2.6.1.51 | The system shall provide the ability to display input REST Operation information for a HTTP Method. | P2.3 |
| 2.6.1.52 | The system shall provide the ability to display input REST Operation information for a URL Path. | P2.2.2 |
| 2.6.1.53 | The system shall provide the ability to display input REST Operation information for a Consumes entry. | P2.3 |
| 2.6.1.54 | The system shall provide the user with the ability to display input REST Operation information for a Produces entry, e.g. JSON. | P2.2.2 |
| 2.6.1.55 | The system shall provide the ability to enter the name of the REST-specific RPC associated with this operation. | P2.3 |
| 2.6.1.56 | The system shall provide the ability to enter a unique Operation Name for this REST-specific operation. | P2.3 |
| 2.6.1.57 | The REST-specific Operation Name will correspond to the default name of the generated Runtime operation. | P2.3 |
| 2.6.1.58 | *The system shall provide the user with the ability to select a REST-specific Response Type from a drop-down list correlating a return type value of the data response from VistA that can be represented in any of the following formats:*   * *String* * *JavaScript Object Notation (JSON)*   (Original) The system shall provide the user with the ability to select a REST-specific Response Type from a drop-down list correlating a return type value of the data response from VistA that can be represented in any of the following formats:   * String * JavaScript Object Notation (JSON) * List * Map   **Justification for Revision:** String and JSON are minimum formats for REST, therefore, List and MAP are not needed here. | P2.3 |
| 2.6.1.59 | The system shall provide the user with the ability to select a REST-specific HTTP Method from a drop-down list correlating a method definition value that can be represented as:   * GET * POST * PUT * DELETE * HEAD | P2.3 |
| 2.6.1.60 | *The system shall provide the user with the ability to enter the relative Uniform Resource Locator (URL) path of the RESTful operation.*  (Original)The system *shall* provide the user with the ability to enter the relative Uniform Resource Locator (URL) path of the RESTful operation. This corresponds to the @Path annotation of the generated Runtime operation.  **Justification for Revision:** Deleted text for clarity | P2.3 |
| 2.6.1.61 | *The system shall provide the user with the ability to select a REST-specific Consumes entry from a drop-down list to specify the Multimedia Internet Mail Extensions (MIME) media type that can be consumed by the operation. Examples of media types are:*   * *“text/plain”* * *“test/html”* * *“application/xml”* * *“application/h-www-form-urlencoded”* * *“application/json”*   (Original)The system *shall* provide the user with the ability to select a REST-specific Consumes entry from a drop-down list to specify the Multimedia Internet Mail Extensions (MIME) media type that can be consumed by the operation. Examples of media types are:   * “text/plain” * “test/html” * “application/xml” * “application/h-www-form-urlencoded” * application/json”   This corresponds to the @Consumes annotation of the generated Runtime operation.  **Justification for Revision:** Deleted text for clarity. | P2.3 |
| 2.6.1.62 | *The system shall provide the user with the ability to select a REST-specific Produces entry from a drop-down list to specify the MIME media type of the response that the operation can produce and send back to the client. Examples of media types are:*   * *“text/plain”* * *“test/html”* * *“application/xml”* * *“application/h-www-form-urlencoded”* * *“application/json”*   (Original) The system *shall* provide the user with the ability to select a REST-specific Produces entry from a drop-down list to specify the MIME media type of the response that the operation can produce and send back to the client. This corresponds to the @Produces annotation of the generated Runtime operation. Examples of media types are:   * “text/plain” * “test/html” * “application/xml” * “application/h-www-form-urlencoded” * “application/json”   **Justification for Revision:** Deleted text that is implementation specific | P2.3 |
| 2.6.1.63 | The system shall provide the user with the ability to edit a REST-specific operation. | P2.4 |
| 2.6.1.64 | The system shall provide the user with the ability to delete a REST-specific operation, and all its associated input parameters. | P2.4 |
| 2.6.1.65 | The system shall display to the user a warning message confirming a deletion of a REST-specific operation. | P2.2.2 |
| 2.6.1.66 | The system shall provide the user with the ability to add one to multiple REST-specific parameters if necessary that define the input parameters for an operation. | P2.4 |
| 2.6.1.67 | The system shall provide the ability to display input REST-specific information for a Name of a REST parameter. | P2.4 |
| 2.6.1.68 | The system shall provide the ability to display input information for a Name of a REST parameter. | P2.4 |
| 2.6.1.69 | The system shall provide the ability to display input information for a Type of a REST parameter. | P2.4 |
| 2.6.1.70 | The system shall provide the ability to display input information for a Param Type of a REST parameter. | P2.4 |
| 2.6.1.71 | The system shall provide the ability to display input information for a Param Name of a REST parameter. | P2.4 |
| 2.6.1.72 | The system shall provide the ability to display input information for a Default Name of a REST parameter. | P2.4 |
| 2.6.1.73 | *The system shall require the user to enter a unique Name for the REST-specific input parameter.*  (Original)The system *shall* provide the user with the ability to enter a unique Name for the REST-specific input parameter.  **Justification for Revision:** Reworded for clarity | P2.4 |
| 2.6.1.74 | The unique parameter name corresponds to the name of the REST-specific input variable used in the generated Runtime operation. | P2.4 |
| 2.6.1.75 | The system shall provide the user with the ability to select Type from a drop-down list that defines the type of the REST-specific input parameter (e.g., string, ref, list, and map). | P2.2.2 |
| 2.6.1.76 | The system shall provide the user with the ability to select a REST-specific Param Type, which indicates how the parameter will be sent to the resource method. Possible values can include:   * PathParam * QueryParam * MatrixParam * FormParam * HeaderParam * CookieParam * Context   This corresponds to the annotation of the parameter in the resource method signature. | P2.4 |
| 2.6.1.77 | The system shall provide the user with the ability to enter the REST-specific name of the parameter. This corresponds to the argument to be used in the following annotation of the input parameter of the generated Runtime operation:   * @PathParam * @QueryParam * @MatrixParam * @FormParam * @HeaderParam * @CookieParam * @Context   **Justification for Removal:** This is addressed by 2.6.1.76 | Removed |
| 2.6.1.78 | The system shall provide the user with the ability to specify a default value for this input parameter if one is not passed. | P2.4 |
| 2.6.1.79 | The default value corresponds to the @DefaultValue annotation of the REST-specific input parameter of the generated Runtime operation.  **Justification for Removal:** This is addressed by 2.6.1.78 | Removed |
| 2.6.1.80 | The system shall provide the user with the ability to delete a REST input parameter. | P2.4 |
| 2.6.1.81 | The system shall display to the user a warning message confirming a deletion of a REST input parameter. | P2.2.2 |
| 2.6.1.82 | The system shall provide the user with the ability to edit a REST-specific input parameter. | P2.2.2 |
| 2.6.1.83 | The system shall provide the user with the ability to hide or show the REST-specific input parameters for any operation. | P2.4 |
| 2.6.1.84 | *The system shall provide the user with the ability to save a Service Descriptor.*  (Original) The system *shall* provide the user with the ability to save a Service Descriptor file in XML format.  **Justification for Revision:** Removed XML format - implementation specific | P2.2.2 |
| 2.6.1.85 | *The system shall display the contents of a newly generated service descriptor.*  (Original) The system *shall* display the contents of a newly generated service descriptor file.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.1.86 | The system shall not allow the user to generate a service from partially filled Service Descriptor form. | P2.4 |
| 2.6.1.87 | The system shall label fields in the Wizard that are required. | P2.2.2 |
| 2.6.1.88 | *The system shall provide the user with the ability to auto-generate a service descriptor into a deployable runtime package.*  (Original) The system *shall* provide the user with the ability to auto-generate a service descriptor XML file into a deployable runtime package.  **Justification for Revision:** Removed XML format - implementation specific | P2.4 |
| 2.6.1.89 | The system shall provide the user with the ability to deploy and publish the Web service runtime package to the non-Production Federating Platform. | P2.4 |
| 2.6.1.90 | The system shall display the URL of the service to be tested. | P2.4 |
| 2.6.1.91 | The system shall provide the user with the ability to test and execute a deployed Web service. | P2.2.2 |
| 2.6.1.92 | The GUI shall consist of a structured point-and-click interface, and with free-text fields on a Service Descriptor form for the user to enter information. | P2.2.2 |
| 2.6.1.93 | The system shall provide text fields used to accept alphanumeric entries. Specific text fields will also accept special characters. | P2.2.2 |
| 2.6.1.94 | The system shall provide drop-down selection menus used to present longer lists of responses but permit only a single response (e.g., RPCs). | P2.2.2 |
| 2.6.1.95 | The system shall provide drop-down selection menus used when a short list of responses allows only a single answer (e.g., a String, JSON, list, or map response). | P2.2.2 |
| 2.6.1.96 | The system shall provide buttons used to search, delete, insert, or hide information about an RPC operation or parameter. | P.2.3 |
| 2.6.1.97 | *In development, VSA Wizard shall enforce VistA user security to its corresponding development VistA instance, by meeting the requirements of the VSA Run-Time as a consumer application (the Wizard as the consumer).*  (Original) VSA Wizard *shall* enforce VistA user security to its corresponding development VistA instance, by meeting the requirements of the VSA Run-Time as a consumer application (the Wizard as the consumer).  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.1.98 | *In development, the VSA Wizard shall allow the user to specify its VistA development environment and to provide the necessary credentials through Access and Verify codes to connect securely with VistA.*  (Original) If the VSA Wizard is deployed in a centralized and shared environment, the VSA Wizard *shall* allow the user to specify its VistA development environment and to provide the necessary credentials through Access and Verify codes to connect securely with VistA.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.1.99 | The VSA Phase 2 software shall allow the user (e.g., developer) to use a VA Intranet Web browser to access the VSA Wizard on the runtime environment. | P2.2.2 |
| 2.6.1.100 | The system shall provide the ability for the user to modify existing or create new Service Descriptor, which are used to create VSA Web services (runtime packages). | P2.3 |
| 2.6.1.101 | The system shall display Service Descriptor information to allow the user to select it for edit. | P2.2.2 |
| 2.6.1.102 | The VSA Phase 2 software shall create the following display screens and output relative to the user:   * VSA Wizard main page (Service Descriptor front-end) * VSA Wizard Service Descriptor forms for SOAP and REST; Includes display of any VSA system messages. | P2.3 |
| 2.6.1.103 | *The VSA Phase 2 software shall create the following display screens:*   * *Service Descriptor, which is used to define a Web service and to generate the runtime package.*   (Original) The VSA Phase 2 software *shall* create the following display screens and output file:  Service Descriptor XML file, which is used to define a web service and to generate the runtime package.  **Justification for Revision:** Removed XML format - implementation specific | P2.3 |
| 2.6.1.104 | The VSA Phase 2 software shall create the following display screens and output file:   * A deployable Web service runtime package. | P2.3 |
| 2.6.1.105 | *When the form is displayed, the VSA Wizard functionality shall enable a user to click or tab to each field when entering data.*  (Original) When the form is displayed, the VSA Wizard *shall* enable a user to click or tab to each field when entering data.  **Justification for Revision:** Reworded for clarity | P2.3 |
| 2.6.1.106 | *When a user saves a form, the VSA Wizard functionality shall display a message as the form is being saved.*  (Original) When a user saves a form, the VSA Wizard *shall* display a message as the form is being saved.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.1.107 | *When a user submits a form, the VSA Wizard functionality shall provide an indication that the form has been saved by displaying the contents of the newly created service descriptor*  (Original) When a user submits a form, the VSA Wizard *shall* provide an indication that the form has been saved by displaying the contents of the newly created service descriptor into an XML document.  **Justification for Revision:** Removed XML format - implementation specific | P2.3 |
| 2.6.1.108 | When a user selects to create or deploy a Web service runtime package, the VSA Wizard functionality shall display a message as the service is being created and deployed. | P2.3 |
| 2.6.1.109 | *VSA runtime platform shall provide an administrator user interface for configuring URL endpoints, user names and passwords in accordance with VA security requirements.*  (Original) The system *shall* provide an administrator user interface for configuring VSA on the runtime system to specify URL endpoints, user names and passwords.  **Justification for Revision:** Reworded for clarity | P2.6 |
| 2.6.1.110 | The system shall provide a configuration file for configuring the VSA application server on the runtime system to specify URL endpoints, user names and passwords.  **Justification for Removal:** Addressed in 2.6.1.109 | Removed |
| 2.6.1.111 | *The VSA Wizard functionality shall allow the user to configure and choose the VistA instance.*  (Original) The VSA Wizard *shall* allow the user to configure and choose the needed development VistA instance.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.1.112 | *The application shall conform to look and feel standards as established by Section 508 of the Rehabilitation Act (29 U.S.C. § 794d).*  (Original) The application *shall* conform to look and feel standards as established by the VA’s current and *future* healthcare systems’ core specifications for re-hosting.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.1.113 | A GUI application shall be developed for the creation of service descriptors. | P2.2.2 |
| 2.6.1.114 | *The VSA Wizard functionality shall allow the user to create deployable runtime package*  (Original) The Wizard GUI *shall* allow the user to create deployable runtime package from the created service descriptors.  **Justification for Revision:** Reworded for clarity | P2.3 |
| 2.6.1.115 | *The solution shall provide a point and click interface with free text fields from the VSA Wizard functionality.*  (Original) The solution *shall* provide a point and click interface with free text fields from the Wizard form.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.1.116 | *During creation of a service descriptor, part of the GUI’s functionality shall interrogate the VistA for available RPCs and retrieve all the details of a given RPC to pre-populate the interface fields.*  (Original) During creation of a service descriptor, part of the GUI’s functionality *shall* interrogate the development VistA for available RPCs and retrieve all the details of a given RPC to pre-populate the interface fields.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.1.117 | (NEW) *The VSA solution shall provide a UI for documentation about the server-side APIs (including exposed VistA SOA services [SOAP or REST]).* | P2.2.2 |

### VSA Wizard Functionality—Functions

The VSA Wizard functionality provides for the auto-generation of VistA-based services.

Table : VSA Wizard Functionality-Functions

| Req. # | Requirement | Delivery |
| --- | --- | --- |
| 2.6.2.117 | The VSA Wizard service descriptor shall allow to Locate and select for edit an existing Service Descriptor. | P2.2.2 |
| 2.6.2.118 | The VSA Wizard shall Create a Service Descriptor. | P2.2.2 |
| 2.6.2.119 | The VSA Wizard functionality shall Search for RPCs to be called from VistA. | P2.2.2 |
| 2.6.2.120 | The VSA Wizard functionality shall Retrieve and add operations and parameter mappings. | P2.4 |
| 2.6.2.121 | The VSA Wizard shall Save and store service descriptor. | P2.2.2 |
| 2.6.2.122 | The VSA Wizard functionality shall Create and deploy runtime package (on the development system) that are necessary in defining and publishing an RPC as a Web service. | P2.2.2 |
| 2.6.2.123 | *The VSA Wizard shall provide a link to test the deployed VistA SOA service.*  (Original) The VSA Wizard *shall* provide a link to test the deployed web service.  **Justification for Revision:** Reworded for clarity | P2.5 |
| 2.6.2.124 | *The VSA Wizard shall provide a method for reviewing and verifying the newly auto-generated VistA SOA service.*  (Original) The VSA Wizard *shall* provide a workflow for reviewing and verifying the newly auto-generated Web service.  **Justification for Revision:** Reworded for clarity | P2.5 |
| 2.6.2.125 | The system shall provide the ability to store Service Descriptors and runtime packages. | P2.2.2 |
| 2.6.2.126 | The VSA Wizard functionality shall provide RPC Service for Wizard—An internal VSA Web service that the VSA Wizard functionality will use to interrogate VistA for available RPCs and to retrieve all the details of a given RPC to automatically generate a Web service operation that invokes that RPC. | P2.2.2 |
| 2.6.2.127 | *The VSA Wizard shall provide Service Code Generator—An internal VSA utility that parses the service definition generates the necessary runtime components and artifacts that will be compiled and combined with other libraries into a deployable runtime package.*  (Original) The VSA Wizard *shall* provide Service Code Generator—An internal VSA utility that parses the service definition file in XML format and generates the necessary runtime components and artifacts that later will be compiled and combined with other libraries into a deployable runtime package by the Service Assembler.  **Justification for Revision:** Removed XML and reworded for clarity | P2.3 |
| 2.6.2.128 | The VSA Wizard functionality service assembler shall aggregate all the runtime components that define a service. | P2.3 |
| 2.6.2.129 | The VSA Wizard functionality service assembler shall combine runtime components with other necessary dependencies. | P2.3 |
| 2.6.2.130 | The VSA Wizard functionality service assembler shall produce a deployable runtime package. | P2.3 |
| 2.6.2.131 | The VSA Wizard functionality shall provide a service deployer for deploying generated service files to the development platform. | P2.3 |
| 2.6.2.132 | The VSA Wizard functionality shall utilize a runtime environment for testing federated services. | P2.3 |
| 2.6.2.133 | The VSA Wizard runtime environment shall provide a repository for storing Service Descriptor and runtime packages. | P2.2.2 |
| 2.6.2.134 | *The service descriptor shall contain service definitions detailing the service, application package, operation, expected input parameters, and data types.*  (Original) The service descriptor XML file *shall* contain service definitions detailing the service, application package, operation, expected input parameters, and data types.  **Justification for Revision:** Removed XML | P2.2.2 |
| 2.6.2.135 | The Service Descriptor generated by the Wizard shall be stored on the file system of the non-Production runtime environment. | P2.4 |
| 2.6.2.136 | Create Web Service runtime package: A copy of runtime component libraries shall be packaged in each deployable Web Service artifact. | P2.4 |
| 2.6.2.137 | *The VSA Wizard shall validate a form when the user saves a service descriptor to prevent runtime errors. Mandatory fields shall have a visual indicator.*  (Original) The VSA Wizard *shall* validate an incomplete form when the user saves a service descriptor.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.2.138 | *The VSA Wizard functionality shall use standard and secure communication protocols (Hypertext Transfer Protocol Secure [HTTPS]). Using Government-provided PKI certificates.*  (Original) The VSA Wizard *shall* provide communication to development VistA instances and use standard and secure communication protocols (Hypertext Transfer Protocol Secure [HTTPS]).  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.2.139 | (NEW) *The VSA solution shall include a Web-based Wizard to expose VistA as SOA services (SOAP and REST*). | P2.2.2 |

### VSA Runtime Environment Components

Additional VSA functionality as follows:

Table : VSA Runtime Environment

| Req. # | Requirement | Delivery |
| --- | --- | --- |
| 2.6.3.300 | *The VSA runtime environment shall enable the Wizard to interact with one development VistA system.*  (Original) The VSA runtime environment *shall* provide for the hosting of the VSA Wizard to enable the Wizard to interact with one development VistA system.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.3.301 | The VSA runtime environment shall facilitate the VSA Wizard to store Service Descriptor and runtime packages generated by the VSA Wizard. | P2.2.2 |
| 2.6.3.302 | The VSA runtime environment logic shall perform the routing (federation) of queries from provider and consumer “service” requests to and from VistA. | P2.2.2 |
| 2.6.3.303 | The VSA runtime environment logic shall facilitate “run-time” execution of generated VistA-based Web services. | P2.2.2 |
| 2.6.3.304 | *The VSA VistA runtime environment shall provide Web connectivity to VistA as an HTTP to VistA MUMPS binding that conveys ‘service requests’ to one or more VistA systems in the standard and secured HTTPS protocol. Using Government-provided PKI certificates.*  (Original)The VSA runtime environment *shall* provide web connectivity to VistA as an HTTP to VistA MUMPS binding that conveys ‘service requests’ to one or more VistA systems in the standard and secured HTTPS protocol.  **Justification for Revision:** Added text requirement | P2.2.2 |
| 2.6.3.305 | VSA shall provide a VistA “listener,” known as the VistA M Routine Calling Service (VMRCS).  **Justification for Removal:** Remove, combined and satisfied by 2.6.3.304. | Removed |
| 2.6.3.306 | VSA shall provide Web based (HTTP) connectivity to the VistA functionality and data such as RPCs, routines, or database access. | P2.2.2 |
| 2.6.3.307 | The VSA VistA “listener” (VMRCS) shall be responsible for transferring requests to run M routines to the VistA M Routine Calling Adaptor (VMRCA) and the legacy M environment.  **Justification for Removal:** Remove, this is covered by 304 thru 306 which will be moved to the SDD. | Removed |
| 2.6.3.308 | VSA shall provide a VistA M Routine Calling Adaptor (VMRCA) that integrates the VMRCS component with the traditional M computing environment and provides for the invocation of M routines*.*  **Justification for Removal:** Remove, this is covered by 304 thru 306 which will be moved to the SDD. | Removed |
| 2.6.3.309 | A VSA generated service shall contain one or more operations. | P2.2.2 |
| 2.6.3.310 | VSA generated services shall support the execution of a single RPC or routine API. | P2.2.2 |
| 2.6.3.311 | A single VSA generated service shall support the sequential execution of multiple “chained” RPCs or APIs. | P2.2.2 |
| 2.6.3.312 | VSA services that execute multiple “chained” RPCs or APIs may include “stateful” RPCs that rely on the results of a preceding RPC in the execution sequence. | P2.2.2 |
| 2.6.3.313 | VSA shall validate the presence of required information in ‘service requests’ including user identity, input parameters, federation routing, etc. | P2.2.2 |
| 2.6.3.314 | VSA shall establish VistA background information for a session consistent with ASD security patterns and organizationally established ‘user identity propagation’ guidelines. | P2.2.2 |
| 2.6.3.315 | The VSA Web connectivity to VistA shall not allow connections or accept ‘service requests’ from any other application or middleware, except VSA. | P2.4 |
| 2.6.3.317 | VSA shall verify that all “service requests” include user identity attributes, the “consuming application” identity and routing information. | P2.4 |
| 2.6.3.442 | VSA shall facilitate the logging of ‘service request’ transactions. | P2.2.2 |
| 2.6.3.443 | VSA transaction logging data shall include date/time of transaction, ‘consuming application’, service invoked and destination VistA systems. | P2.2.2 |
| 2.6.3.444 | (NEW) *The VSA solution shall facilitate sending and receiving of an industry standard C32 Continuity of Care Document (CCD) transaction between VA and external parties.*  [*HITSP Summary Documents Using HL7 Continuity of Care Document (CCD) Component*](http://vaww.oed.portal.va.gov/projects/sts/Library/STS%20Development%20Documents/Data%20Analysis/HITSP_V2.5_2009_C32_-_Summary_Documents_Using_CCD.pdf) | P2.6 |

### Federation

The VSA Runtime Environment provides for federated routing of “service requests” to multiple VistA systems and the aggregation of responses returned.

Table : VSA Federation Requirements

| Req. # | Requirement | Delivery |
| --- | --- | --- |
| 2.6.4.318 | The VSA shall provide federated aggregation of responses from provider and consumer “service” requests to Veterans Health Information Systems and Technology Architecture (VistA). | P2.2.2 |
| 2.6.4.319 | The VSA Federating Platform shall facilitate federated routing of queries across multiple VistA systems and aggregation of returned results. | P2.2.2 |
| 2.6.4.320 | VSA federation functionality shall facilitate routing of “service requests” to a single specified VistA system. | P2.2.2 |
| 2.6.4.321 | VSA federation functionality shall facilitate routing of “service requests” to a specified list of VistA systems. | P2.2.2 |
| 2.6.4.322 | VSA federation functionality shall facilitate routing of “service requests” to all VistA systems. | P2.2.2 |
| 2.6.4.323 | VSA federation functionality shall facilitate routing of “service requests” to all VistA systems to which a specified patient is known (“treating facilities”). | P2.5 |
| 2.6.4.324 | VSA federation functionality shall facilitate routing of “service requests” to all VistA systems to which a specified VistA user is known. | P2.5 |
| 2.6.4.325 | VSA federated calls to multiple VistA systems shall be made asynchronously and in parallel to optimize performance of returned results. | P2.2.2 |
| 2.6.4.326 | The results of VSA federated calls to multiple VistA systems shall be aggregated into a single response for return to the “consuming application.” | P2.2.2 |
| 2.6.4.327 | VSA shall use the MVI to determine the list of “treating facilities” for the routing of ‘service requests’ to each VistA system that has a specified patient on record. | P2.5 |
| 2.6.4.328 | VSA shall facilitate the return of aggregated “service request” responses from multiple VistA systems that are incomplete due to time-out, non-responsiveness, etc. of one or more VistA systems. | P2.5 |
| 2.6.4.329 | Aggregated “service request” responses from multiple VistA systems that are incomplete due to time-out, non-responsiveness, etc. of one or more VistA systems shall include an exception message noting the incomplete nature of the response. | P2.2.2 |
| 2.6.4.330 | Exception messages included with aggregated “service request” responses from multiple VistA systems that include an incomplete response shall include a list of VistA systems to which the “service request” was routed but from which no response was received. | P2.2.2 |
| 2.6.4.331 | With the exception of federation (e.g., MVI, etc.), all business logic that is the basis for VSA generated services shall be in the VistA M environment in the form of RPCs or APIs. | P2.2.2 |
| 2.6.4.332 | The core logic for VSA utilities shall be packaged and distributed separately from service-specific business logic (e.g., RPCs). | P2.2.2 |
| 2.6.4.333 | Stability of the core logic for VSA utilities shall be maintained by keeping VSA core logic generic and non-service-specific. | P2.2.2 |
| 2.6.4.334 | (NEW) *The VSA solution shall provide a services Federation capability that provides access to Enterprise entities including:*   1. *Instances of VistA,* 2. *Medication Image Library (MIL),* 3. *Master Veteran Index (MVI)* | P2.6 |
| 2.6.4.335 | (NEW) *The VSA solution shall ensure access to Enterprise entities is federated through a dynamic, data-driven, configurable scoping mechanism that allows system administrators to control runtime routing in production. A specific example of this is the integration of a SOAP service.* | P2.6 |

### Pre/Post Logic Processing

VSA provides the ability to extend the behavior of VSA-generated services through the configuration of “pre” and “post” processing logic. The following describes the process flow:

1. “Consuming application” obtains a user identity SAML token from IAM.
2. “Consuming application” sends “service request” (including SAML token) to eMI.
3. eMI authenticates the “consuming application,” authorizes execution of a specific service, validates the SAML token.
4. eMI routes ‘service request’ to VSA.
5. VSA (optionally) invokes “Pre” logic in the eMI environment.
6. VSA uses VistA logic to establish VistA background representing the user.
7. VSA (optionally) invokes “Pre” logic in the VistA M environment.
8. VSA invokes VistA RPC.
9. VistA RPC returns result to VSA.
10. VSA (optionally) invokes “Post” logic in the VistA M environment.
11. VSA (optionally) invokes “Post” logic in the eMI environment.
12. VSA returns the “service response” to the eMI.
13. eMI returns “service response” to the “consuming application”.

Figure 1 depicts the process flow:

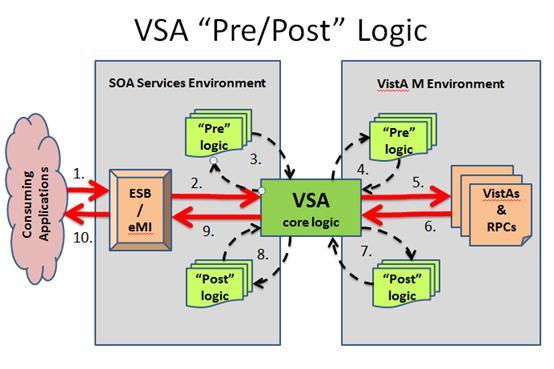


Figure : VSA “Pre/Post” Logic Model

Table : VSA Pre/Post Logic Processing Requirements

| Req. # | | Requirement | Delivery |
| --- | --- | --- | --- |
| 2.6.5.335 | VSA shall facilitate the processing of “Pre” actions that precede the execution of the service business logic (RPC or API). | | P2.2.2 |
| 2.6.5.336 | VSA shall facilitate the processing of “Post” actions that follow the execution of the service business logic (RPC or API). | | P2.2.2 |
| 2.6.5.337 | VSA shall facilitate “Pre” and “Post” actions that allow the execution of other SOA services. | | P2.2.2 |
| 2.6.5.338 | VSA shall facilitate “Pre” and “Post” actions that allow the execution of other VistA logic. | | P2.2.2 |
| 2.6.5.339 | The core logic for VSA utilities shall not contain “Pre/Post” logic elements. | | P2.2.2 |
| 2.6.5.340 | The core logic for VSA utilities shall be packaged and distributed separately from “Pre/Post” logic elements. | | P2.2.2 |
| 2.6.5.341 | VSA “Pre/Post” processing shall allow the execution of services and/or M logic created by other sources (e.g., “open source”). | | P2.2.2 |
| 2.6.5.342 | | VSA “Pre/Post” action processing shall facilitate the following ‘to be’ process flow:   1. “Consuming application” obtains a user identity SAML token from IAM. 2. “Consuming application” sends “service request” (including SAML token) to eMI. 3. eMI authenticates the “consuming application,” authorizes execution of a specific service, validates the SAML token. 4. eMI routes ‘service request’ to VSA. 5. VSA (optionally) invokes “Pre” logic in the eMI environment. 6. VSA uses VistA logic to establish VistA background representing the user. 7. VSA (optionally) invokes “Pre” logic in the VistA M environment. 8. VSA invokes VistA RPC. 9. VistA RPC returns result to VSA. 10. VSA (optionally) invokes “Post” logic in the VistA M environment. 11. VSA (optionally) invokes “Post” logic in the eMI environment. 12. VSA returns the “service response” to the eMI. 13. eMI returns “service response” to the “consuming application.” | P2.6 |

### Exception and Error Handling

VSA provides standardized exception and error handling.

Table : VSA Exception and Error Handling Requirements

| Req. # | Requirement | Delivery |
| --- | --- | --- |
| 2.6.6.343 | If required information is not present in a “service request”, VSA shall refuse the request and return an error response as indicated. | P2.2.2 |
| 2.6.6.344 | VSA shall implement standardized VSA error codes based on ESS SOA Design Guidelines and standards. | P2.4 |
| 2.6.6.345 | VSA shall return standard HTTP error codes as related to Representational State Transfer (REST) services. | P2.2.2 |
| 2.6.6.346 | VSA shall return VistA error information when exceptions occur in that environment. | P2.2.2 |

### Architectural Principles

VSA provides utilities for the auto-generation of VistA-based services and the integration of external systems with VistA.

Table : VSA Architectural Principles Requirements

| Req. # | Requirement | Delivery |
| --- | --- | --- |
| 2.6.7.347 | VSA shall be fully compliant with guidelines and standards as established by VA Enterprise Architecture and Architecture, Strategy and Design (ASD). | P2.2.2 |
| 2.6.7.348 | All components and technologies of the VSA product shall be compliant with the VA Enterprise Architecture Technical Reference Model (TRM). | P2.2.2 |
| 2.6.7.350 | The VSA solution shall comply with established VA standards and conventions for software development. | P2.2.2 |
| 2.6.7.351 | VSA shall strictly implement relevant architectural design patterns as established by ASD, including those related to SOA security and “user identity propagation.” | P2.2.2 |
| 2.6.7.352 | VSA shall provide generic utilities which support the creation of VistA based services which are consumable by all ‘consuming applications’ in the SOA services environment. | P2.2.2 |
| 2.6.7.353 | VSA generated Web services shall be consumable from the Enterprise Messaging Infrastructure (eMI) in a technology agnostic manner. | P2.3 |
| 2.6.7.354 | VSA shall support the creation of authoritative, non-redundant VistA based services. | P2.2.2 |
| 2.6.7.355 | VSA utilities shall not contain or replicate business logic that exists in VistA applications. | P2.2.2 |
| 2.6.7.356 | All VSA generated services shall be defined/exposed on the Enterprise Messaging Infrastructure (eMI). | P2.3 |
| 2.6.7.357 | VSA services shall be fully documented in the eMI Service Registry. | P2.6 |
| 2.6.7.358 | VSA shall implement technology that leverages ‘open source’ compatibility. | P2.2.2 |
| 2.6.7.359 | (NEW) *The VSA solution shall be SMART-compliant (Substitutable Medical Applications, Reusable Technologies Platform), both as a consumer and a producer of FHIR-based Web services (Fast Healthcare Interoperability Resources).*  [*SMART Health IT Technology Platform*](http://smarthealthit.org/about/) | P2.5 |
| 2.6.7.360 | (NEW) *The VSA solution shall support use of structured FHIR data to coordinate interactions between mobile and connected devices.* | P2.6 |
| 2.6.7.361 | (NEW) *The VSA solution shall support VA alignment with HL7 FHIR standard policies*. | P2.6 |
| 2.6.7.362 | (NEW) *The solution shall be designed to synchronize with other Enterprise service registries, to include six VISN Registries and the Enterprise Registry, to include use of the Enterprise Messaging Infrastructure (eMI).* | P2.6 |

### Supporting geographically distributed VA System Topology

The VSA Runtime Environment (Federating platforms connecting VistA systems with eMI/SOA) will support geographically distributed VA system topology. This includes infrastructure throughout Continental United States (CONUS), including Enterprise and Regional Data Processing Centers.

Table : VSA System Topology Requirements

| Req. # | Requirement | Delivery |
| --- | --- | --- |
| 2.6.8.359 | *VSA shall facilitate deployment of the VSA Federating Platform to accommodate the geographically distributed VA system topology (infrastructure/servers).*  (Original) VSA *shall* implement system topology (servers) that facilitate deployment of the VSA Federating Platform.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.8.359.1 | VSA shall implement system topology (servers) that facilitate deployment of the VSA Federating Platform (decentralized at the Remote Data Processing Centers [RDPCs]). | P2.6 |
| 2.6.8.360 | *VSA shall rely on VA eMI to authenticate the ‘consuming application’, consumption of specific services and validated via the VA IAM SAML token--and shall not authenticate or validate ‘consuming application’ consumption of services*  (Original) VSA *shall* rely on the ESB to have authenticated the ‘consuming application’, authorized the consumption of specific services and validated the IAM SAML token--and *shall* *not* perform these activities independently or redundantly.  **Justification for Revision:** Reworded for clarity | P2.5 |
| 2.6.8.361 | VSA shall be integrated with the Enterprise Messaging Interface (eMI) consistent with ASD design patterns. | P2.4 |
| 2.6.8.362 | *VSA shall rely on the eMI exclusively to accept “service requests” from ‘external applications’ and shall not accept “service requests” directly.*  (Original) VSA *shall* *not* accept “service requests” directly from external applications—except as communicated through the ESB.  **Justification for Revision:** Reworded for clarity | P2.6 |
| 2.6.8.363 | VSA shall coordinate with SDE to identify specifications for initial server implementation. | P2.4 |
| 2.6.8.364 | *VSA shall coordinate with SDE to determine optimal server specifications for the “to be” implementation based on implementation and testing of the initial implementation.*  (Original) Information gathered as a result of implementation and testing of the “initial” implementation *shall* be used to determine optimal server specifications for the “to be” implementation.  **Justification for Revision:** Reworded for clarity | P2.6 |
| 2.6.8.365 | VSA servers *shall* comply with National Institute of Standards and Technology (NIST) and Federal Desktop Core Configuration (FDCC) guidelines regarding build-outs and security settings. | P2.2.2 |
| 2.6.8.366 | Servers *shall* use standard protocols and ports for communication and network settings. | P2.2.2 |
| 2.6.8.367 | VSA systems *shall* comply with VA Directive 6102 relative to VA Internet and Intranet standards for systems operation. | P2.2.2 |
| 2.6.8.368 | *VSA shall implement an “initial implementation” that facilitates rapid early deployment to production in the geographically distributed VA system topology (infrastructure/servers).*  (Original) VSA *shall* implement an interim topology that facilitates rapid early deployment to production.  **Justification for Revision:** Reworded for clarity | P2.3 |
| 2.6.8.369 | *The VSA “initial implementation” shall support separate environments for development in the geographically distributed VA system topology (infrastructure/servers).*  (Original) The VSA “initial implementation” topology *shall* provide separate environments for development.  **Justification for Revision:** Reworded for clarity | P2.3 |
| 2.6.8.370 | *The VSA “initial implementation” shall support separate environments for integration testing in the geographically distributed VA system topology (infrastructure/servers).*  (Original) The VSA “initial implementation” topology *shall* provide separate environments for integration testing.  **Justification for Revision:** Reworded for clarity | P2.3 |
| 2.6.8.371 | *The VSA “initial implementation” shall support separate environments for user functional and acceptance testing by SQA in the geographically distributed VA system topology (infrastructure/servers).*  (Original) The VSA “initial implementation” topology *shall* provide separate environments for user functional and acceptance testing.  **Justification for Revision:** Reworded for clarity | P2.3 |
| 2.6.8.372 | The VSA “initial implementation” shall support separate environments for SQA testing in the geographically distributed VA system topology (infrastructure/servers).  **Justification for removal:** Addressed by 2.6.8.371 | Removed |
| 2.6.8.373 | *The VSA “initial implementation” shall support separate environments for capacity/load/performance testing by VA ETS in the geographically distributed VA system topology (infrastructure/servers).*  (Original) The VSA “initial implementation” topology *shall* provide separate environments for capacity/load/performance testing  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.8.374 | *The VSA “initial implementation” shall support separate environments for security testing by VA OIS in the geographically distributed VA system topology (infrastructure/servers).*  (Original)The VSA “initial implementation” topology *shall* provide separate environments for security testing.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.8.375 | *The VSA “initial implementation” shall support separate environments for pre-production in the geographically distributed VA system topology (infrastructure/servers).*  (Original) The VSA “initial implementation” topology *shall* provide separate environments for pre-production.  **Justification for Revision:** Reworded for clarity | P2.3 |
| 2.6.8.376 | *The VSA “initial implementation” shall support separate environments for production in the geographically distributed VA system topology (infrastructure/servers).*  (Original) The VSA “initial implementation” topology *shall* provide separate environments for production.  **Justification for Revision:** Reworded for clarity | P2.3 |
| 2.6.8.377 | *The VSA “initial implementation” shall provide multiple instances of the VSA Federating Platform to facilitate development of federated ‘service request’ routing functionality in the geographically distributed VA system topology (infrastructure/servers).*  (Original) The VSA “initial implementation” topology *shall* provide multiple instances of the VSA Federating Platform to facilitate development of federated ‘service request’ routing functionality.  **Justification for Revision:** Reworded for clarity | P2.3 |
| 2.6.8.378 | *The VSA “initial implementation’ shall support network communications with other SOA components (e.g., eMI, VistA systems, etc.).*  (Original) The VSA “initial implementation’ topology *shall* facilitate network connectivity with other SOA components (e.g., ESB, VistA systems, etc.).  **Justification for Revision:** Reworded for clarity | P2.3 |
| 2.6.8.379 | *The VSA “initial implementation” topology shall provide multiple instances of the VSA Federating Platform to support development of federation and fail-over logic in the geographically distributed VA system topology (infrastructure/servers).*  (Original) The VSA “initial implementation” topology *shall* provide multiple instances of the VSA Federating Platform to support development of federation and fail-over logic.  **Justification for Revision:** Added text requirement for clarity | P2.3 |
| 2.6.8.380 | Testing results and “lessons learned” from the VSA ‘initial implementation’ topology shall be used to document recommended optimal specifications and topology for the VSA ‘to be’ implementation.  **Justification for Removal**: Addressed in 2.6.8.364 | Removed |
| 2.6.8.381 | *VSA shall support multiple instances of the VSA Federating Platform to support “to be” deployment of VSA systems to RDPCs or AITCs as indicated by VA system specification for the ‘to be’ implementation.*  (Original) VSA *shall* implement multiple instances of the VSA Federating Platform to support “to be” deployment of VSA systems to RDPCs or ITCs as indicated by determined system specification for the ‘to be’ implementation.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.8.382 | *VSA shall coordinate with SDE to determine optimal “to be” topology for system security, reliability and performance*.  (Original) VSA *shall* implement “to be” topology that optimizes system security, reliability and performance.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.8.383 | *VSA shall coordinate with SDE to finalize “to be” implementation system specifications for the “to be” implementation based on “initial implementation” testing results and “lessons learned.”*  (Original) VSA *shall* finalize “to be” implementation system specifications based on VSA ‘initial implementation’ testing results and “lessons learned.”  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.8.384 | *VSA shall coordinate with SDE to determine final “to be” implementation topology based on “initial implementation” testing results and “lessons learned.”*  (Original) VSA *shall* finalize “to be” implementation topology based on VSA ‘initial implementation’ testing results and “lessons learned.”  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.8.385 | *VSA shall be designed to facilitate “scale up” or “scale out” to accommodate expanding usage, capacity and performance needs.*  (Original) VSA system and topology *shall* be designed to facilitate “scale up” or “scale out” to accommodate expanding usage, capacity and performance needs.  **Justification for Revision:** Reworded for clarity | P2.5 |

### Non-Functional Requirements

Table : VSA Non-Functional Requirements

| Req. # | Requirement | Delivery |
| --- | --- | --- |
| 2.6.9.386 | *The contractor shall use VA-approved source control utilities and techniques for the storage and management of all VSA artifacts and deliverables.*  (Original) VSA *shall* use source control utilities and techniques for the storage and management of all VSA artifacts and deliverables.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.9.387 | *VSA deliverables shall be designed in the context of* VA Enterprise Shared Services (ESS) Family of Services/Initiative Management (FoSIM) *solution and service planning activities to ensure architectural models are consistent.*  (Original) VSA utilities *shall* be designed and modeled using a web-based, multi-tier architecture.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.9.388 | Architecturally significant elements (dealing with scalability, stability, flexibility, interoperability, etc.) shall be designed in the Unified Modeling Language (UML). | P2.2.2 |
| 2.6.9.389 | *Software elements that will exist outside of the legacy VistA environment shall be developed using VA Enterprise Architecture design patterns and methods.*  (Original) Software elements that will exist outside of the legacy VistA environment *shall* be developed using Object-Oriented (OO) technologies.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.9.390 | An object-oriented analysis approach shall be used to map the problem domain into an extensible object model. | P2.2.2 |
| 2.6.9.391 | A traceability matrix shall be defined which maps relationships between BRD requirements, RSD requirements, SDD design elements and test scripts. | P2.2.2 |
| 2.6.9.392 | User interface characteristics shall be designed for optimal usability.  **Justification for Removal**: Addressed by 2.6.9.434 | Removed |
| 2.6.9.393 | VSA documentation shall include the following VA Office of General Counsel (OGC) 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 provided that any derivative works bear some notice that they are derived from it, and any modified versions bear some notice that they have been modified.” | P2.2.2 |
| 2.6.9.433 | The Wizard functionality GUI interface shall comply with all Section 508 legal and organizational expectations. | P2.5 |
| 2.6.9.434 | *The contractor will provide certification to the Section 508 Office that contracted deliverables meet Section 508 requirements.*  (Original) The VSA project team will provide certification to the Section 508 Office that the VSA product meets all Section 508 requirements.  **Justification for Revision:** Reworded for clarity | P2.5 |
| 2.6.9.435 | VSA shall comply with VA Web development standards. | P2.2.2 |

### Capacity, Load, and Performance

Table : VSA Capacity-Load and Performance Requirements

| Req. # | Requirement | Delivery |
| --- | --- | --- |
| 2.6.10.394 | The VSA “run time” functionality shall provide sub-second response time consistent with Enterprise-level capacity and performance for critical business applications. | P2.4 |
| 2.6.10.395 | *The VSA “design time” functionality shall provide response time consistent with software development utility user expectations.*  (Original) The VSA “design time” functionality *shall* provide sub-second response time consistent with software development utility user expectations.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.10.396 | VSA capacity shall support thousands of consumer initiated connections received via the ESB.  **Justification for Removal:** Addressed by 3.6.10.398. | Removed |
| 2.6.10.397 | VSA load shall support millions of consumer initiated transactions (“service requests”) received via the ESB from thousands of concurrent users.  **Justification for Removal:** Addressed by 3.6.10.398 | Removed |
| 2.6.10.398 | VSA shall support loads and peak utilization periods similar to CPRS and Medical Domain Web Services (MDWS). | P2.5 |
| 2.6.10.399 | VSA shall support multiple concurrent connections and thousands of concurrent users.  **Justification for Removal:** Addressed by 3.6.10.398. | Removed |
| 2.6.10.400 | VSA shall accommodate transaction sizes commensurate to the limitations of the eMI. | P2.6 |
| 2.6.10.401 | Failure rate of component systems shall be less than 10% in a rolling 12 month period for VSA Federating Platforms*.*  **Justification for Removal:** Addressed by 2.6.10.402. | Removed |
| 2.6.10.402 | VSA components shall provide full operational capacity 99.99% of the time, 24/7/365. | P2.4 |
| 2.6.10.403 | *VSA contractor shall support Government capacity, load and performance testing effort.*  (Original) VSA *shall* execute capacity, load and performance testing.  **Justification for Revision:** Text change for greater clarity of roles. | P2.6 |
| 2.6.10.404 | VSA shall support independent capacity, load and performance testing. | P2.2.2 |
| 2.6.10.405 | VSA capacity, load and performance testing shall evaluate the full SOA stack (from the “consuming application” to VistA and back).  **Justification for Removal:** Addressed by 2.6.10.404. | Removed |
| 2.6.10.406 | VSA capacity, load and performance testing shall measure and evaluate the latency of each application component and network segment the full VSA supported SOA stack.  **Justification for Removal:** Addressed by 2.6.10.404. | Removed |
| 2.6.10.407 | VSA capacity, load and performance testing environments shall support evaluation of SOA stack component distribution across the network (Wide Area Network [WAN]).  **Justification for Removal:** Addressed by 2.6.10.404. | Removed |
| 2.6.10.408 | Testing performed in the VSA interim topology shall be used to determine/modify/validate the optimal topology for the VSA “to be” implementation. | P2.4 |
| 2.6.10.409 | *VSA will use standard VA monitoring tools or real-time monitoring and on-demand evaluation of system performance during normal operation or when technical issues/problems occur that may require a remediation.*  (Original) VSA *shall* provide a system administration user interface for real-time monitoring and on-demand evaluation of system performance during normal operation or when technical issues/problems occur that may require a remediation.  **Justification for Revision:** Reworded for clarity and focus on leveraging/reuse of current monitoring approach. | P2.3 |
| 2.6.10.410 | (NEW) *The VSA solution shall allow the number of external applications accessing VistA to scale up based on VistA system licensing limitations.* | P2.6 |

### Security

Table : VSA Security Requirements

| Req. # | Requirement | Delivery |
| --- | --- | --- |
| 2.6.11.409 | *The VSA solution shall comply with VA 6500 and related security program requirements.*  (Original) The VSA solution *shall* comply with all organizationally established security, privacy, and confidentiality criteria related to sensitive data and business processes.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.11.410 | *No Personally Identifiable Information (PII) or Personal Health Identifiers (PHI) shall be stored VistA SOA Services Platform utility components.*  (Original) No Personally Identifiable Information (PII) or Personal Health Identifiers (PHI) *shall* be stored VSA infrastructure utility components.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.11.411 | *The internal connectivity and communications of the VSA platforms shall be encrypted and exclusive to VSA, eMI, and its specific VistA system(s) wherever VSA will be installed.*  (Original) Two-way Secure Socket Layer (SSL) connectivity shall be implemented between the VSA VistA “listener” (VMRCS) and the VSA Federating Platform to provide a private service that can only be consumed/connected to by the VSA Federating Platform.  **Justification for Revision:** VMRCS portion was removed, is implementation specific. Retain & re-word to preserve the intent (private consumption of VSA services). | P2.2.2 |
| 2.6.11.412 | Secure Socket Layer (SSL/TLS) authentication shall be implemented between the VSA Federating Platform and the eMI to provide a trusted connection that can only be used by the VSA Federating Platform and the eMI. | P2.6 |
| 2.6.11.413 | VSA network connectivity from the eMI to VSA shall employ SSL/TLS to implement payload encryption and authentication of both client and server systems. | P2.2.2 |
| 2.6.11.414 | VSA network connectivity from VSA VistA systems shall employ SSL/TLS to implement payload encryption and authentication of both VSA and VistA systems. | P2.2.2 |
| 2.6.11.430 | *VSA shall support Government security evaluation/testing of deliverables.*  (Original) VSA *shall* engage independent security evaluation/testing of VSA components.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.11.431 | VSA shall engage independent security evaluation/testing of the VSA supported SOA stack.  **Justification for Removal:** Addressed by 2.6.10.409 | Removed |
| 2.6.11.432 | VSA shall provide a suitable security testing environment for independent evaluation/testing of VSA components and the SOA stack.  **Justification for Removal:** Addressed by 2.6.10.430 | Removed |

### User Identity Propagation

VSA provides the ability to communicate information necessary to ensure user identity propagation across architectural tiers.

Table : VSA User Identity Propagation Requirements

| Req. # | Requirement | Delivery |
| --- | --- | --- |
| 2.6.12.415 | *The VSA solution shall comply with VA 6500 and related security program requirements to ensure that User identity Information is propagated across all the layers in the solution using VA approved solutions such as the IAM STS SAML token or M4A.*  (Original) VSA *shall* implement functionality that supports user identity propagation across all architectural tiers of the SOA stack--from the “consuming application” to the “service provider” (VistA).  **Justification for Revision:** Update provided by VA. | P2.3 |
| 2.6.12.416 | The VSA shall accommodate the communication of “consuming application” user identity in the form of the “interim approach” minimum four attributes (M4A). | P2.4 |
| 2.6.12.417 | M4A elements shall include Subject Organization, Subject Organization ID, Unique User ID and Subject ID. | P2.4 |
| 2.6.12.418 | For “interim approach” service requests, VSA shall accept an “Authorization” HTTP header with the minimum four attributes (M4A). For example:  Authorization: VistAid {“subject name“:“One Xuuser,”subject unique ID”:“728”,“organization name“:“San Francisco VAMC”,“organization unique ID”:“662”}. | P2.4 |
| 2.6.12.419 | For “service requests” from “consuming applications” that have integrated with IAM user authentication, VSA shall accept ‘service requests’ that include the IAM SAML token. | P2.3 |
| 2.6.12.420 | For SOAP services, VSA shall accept the SAML token in WS-Security SOAP header. | P2.4 |
| 2.6.12.421 | *For REST services, VSA shall accept an “Authorization” HTTP header with a SAML token in accordance with RFC 2617. For example:*  *Authorization: <trust:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0</trust:TokenType> -<trust:RequestedSecurityToken>… …</trust:RequestedSecurityToken>*  (Original) For REST services, VSA *shall* accept an “Authorization” HTTP header with a SAML token. For example:  Authorization: <trust:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0</trust:TokenType> -<trust:RequestedSecurityToken>… …</trust:RequestedSecurityToken>  **Justification for Revision:** Added text for clarity | P2.4 |
| 2.6.12.422 | The VSA shall accommodate the communication of “consuming application” user identity in the form of the IAM ‘to be’ solution SAML token. | P2.4 |
| 2.6.12.423 | VSA shall not honor or process “service requests” that do not contain user identity attributes in the form of the IAM SAML token or M4A. | P2.4 |
| 2.6.12.424 | An error response shall be returned to the “consuming application” relative to “service requests” that have been rejected due to missing or incomplete user identity attributes. | P2.2.2 |
| 2.6.12.425 | VSA functionality shall facilitate the processing of either the SAML token or the “interim approach” M4A attributes (both variations) to allow organizational transition to the IAM “to be” solution without requiring revision of VSA logic or custom configuration relative to “consuming applications.” | P2.4 |
| 2.6.12.426 | VSA shall not manipulate (change) the consuming application’s user identity or authorization attributes. | P2.2.2 |
| 2.6.12.427 | VSA shall not transport or apply RPC “context option” information to determine user authorization to execute an RPC. | P2.2.2 |
| 2.6.12.428 | VSA shall provide VistA M environment functionality to execute RPC logic without the need for a “context option.” | P2.2.2 |
| 2.6.12.429 | When available in the “service request,” VSA shall apply the IAM SAML token as the single authoritative source of user identity information. | P2.3 |

### Availability - COOP/Disaster Recovery

Table : VSA Availability-Continuity of Operations (COOP)/Disaster Recovery (DR) Requirements

| Req. # | Requirement | Delivery |
| --- | --- | --- |
| 2.6.13.436 | *VSA will contribute VSA updates to the VA contingency plan which describes system availability and Continuity of Operations/Disaster Recovery (COOP/DR) elements including failover, failsoft, backup and restore procedures, etc.*  (Original) VSA *shall* document a contingency plan which describes system availability and Continuity of Operations/Disaster Recovery (COOP/DR) elements including failover, failsoft, backup and restore procedures, etc.  **Justification for Revision:** Clarify roles of VSA and VA in contingency planning | P2.2.2 |
| 2.6.13.437 | VSA shall provide 99.99% uptime by implementing failover across multiple VSA Federating Platforms. | P2.5 |
| 2.6.13.438 | The VSA production implementation shall provide Enterprise-level COOP/DR determined appropriate to system support for critical business processes and data (e.g., provision of healthcare).  **Justification for Removal:** Addressed by 2.6.13.436 | Removed |
| 2.6.13.439 | VSA non-production environments shall implement COOP/DR as appropriate for software development and testing environments. | P2.2.2 |
| 2.6.13.440 | *Failover functionality shall be used to allow functionality to be repaired/replaced/upgraded without adversely impacting the 99.99% uptime of the remaining functionality.*  (Original) Failover functionality *shall* be used to allow production servers to be repaired/replaced/upgraded without adversely impacting the 99.99% uptime of the remaining servers.  **Justification for Revision:** Clarified to focus on delivered functionality not servers (physical or virtual infrastructure). | P2.5 |
| 2.6.13.441 | Each VSA Federating Platform shall facilitate failover to other VSA Federating platforms so that a failure may occur at one platform and workload shall be transferred to the other Federating Platforms that remain operational. | P2.4 |

### Documentation

Table : VSA Documentation Requirements

| Req. # | Requirement | Delivery |
| --- | --- | --- |
| 2.6.14.444 | VSA shall provide relevant documentation for each software release. | P2.2.2 |
| 2.6.14.445 | VSA shall provide release notes for each software release. | P2.2.2 |
| 2.6.14.446 | VSA shall provide a user guide for each software release. | P2.2.2 |
| 2.6.14.447 | VSA shall provide an installation guide for each software release. | P2.2.2 |
| 2.6.14.448 | *VSA shall draft guidelines for the distribution of VSA utilities and the distribution of VSA-generated services.*  (Original) VSA *shall* produce separate guide~~s~~ for the distribution of VSA utilities and the distribution of VSA-generated services.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.14.449 | *VSA shall draft the process for distributing VSA utilities as a combined application package (both M and VSA runtime components).*  (Original) VSA *shall* define and document the process for distributing VSA utilities as a combined application package (both M and VSA runtime components).  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.14.450 | *VSA shall draft the process for distributing VSA generated services*.  (Original) VSA *shall* define and document the process for distributing VSA generated services.  **Justification for Revision:** Reworded for clarity | P2.2.2 |
| 2.6.14.451 | VSA shall draft recommended organizational policy regarding the implementation of VSA generated services. | P2.2.2 |
| 2.6.14.452 | VSA shall draft recommended organizational policy regarding the use of VSA utilities. | P2.6 |

### Apportioning of Requirements

VSA requirements are apportioned according the designation in the “Delivery” column of the tables appearing in Sections 2.6.1 through 2.6.14. All requirements are represented in an “x.y.z” nomenclature

x = Phase (As determined by PMAS)

y = Increment (A cycle of sprints with planned capabilities lasting either 3 or 6 months

Increment 1 & 2 are 6 month increments as determind by PMAS

Increments 3-6 will be 3 month increments as determined by VIP

z = Delivered sprint for requirement fulfillment

Sprint value “z” will only be populated when delivery is planned. Increment values for future work are planned placeholders. The determination of increment and sprint is made as part of an Agile process, and user functionality can be reprioritized to best meet user needs, or work with dependencies in the development cycle.

Requirements marked as removed are either redundant requirements (duplicated in other requirements, and met there), or they have been determined thru the Agile process to be no longer relevant to the work at hand. This latter outcome is a natural part of Agile learning, where discovery thru development allows the capabilities to expand to existing needs not yet discovered.

## Graphical User Interface (GUI) Specifications

Refer to Section 2.6.1 for detailed requirements.

A service developer user interacts with a Web application interface similar to the GUI shown in Figure 2, Figure 3, and Figure 4.

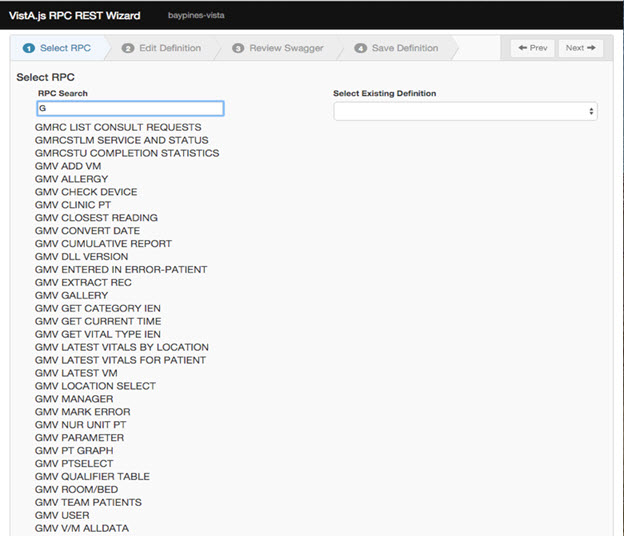


Figure : VSA RPC Wizard-Select RPC

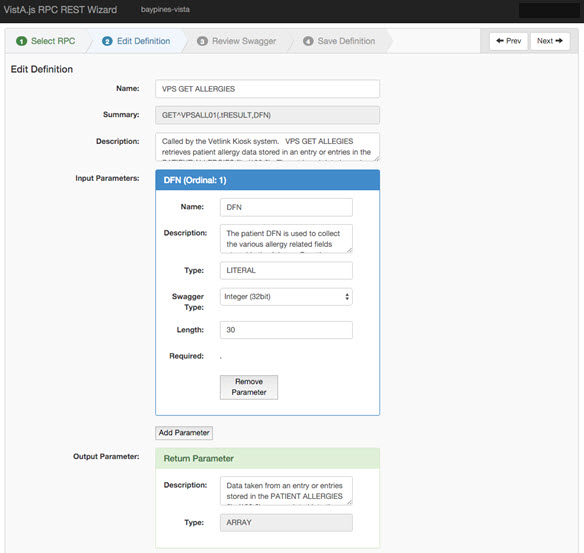


Figure : VSA RPC Wizard-Edit Definition

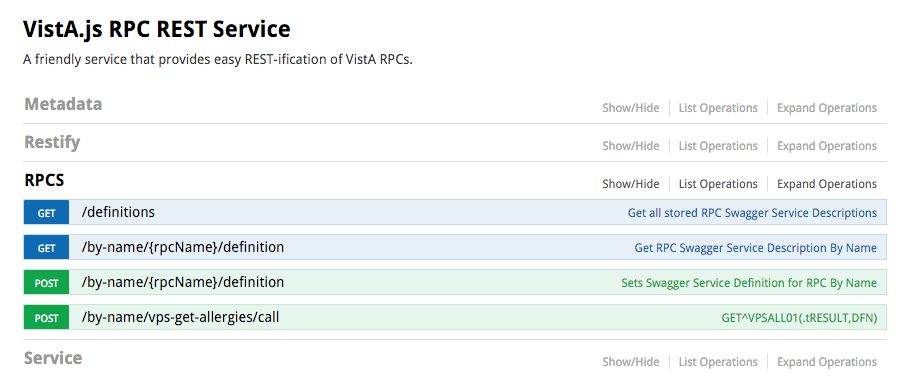


Figure : VSA API Browser-Show RESTified RPCs

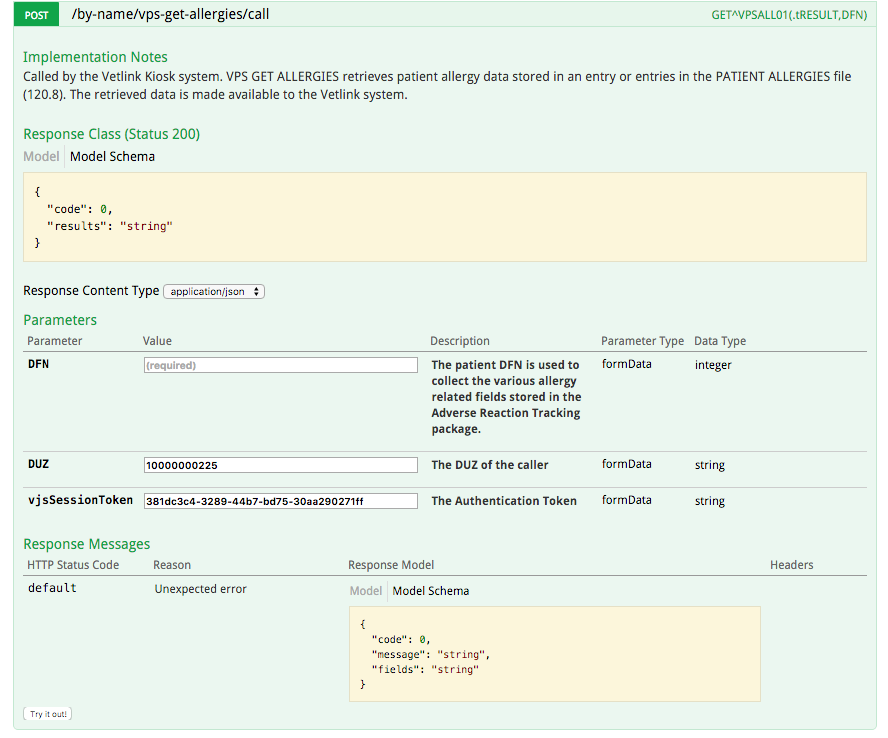


Figure : VSA API Browser-Select a RESTified RPC

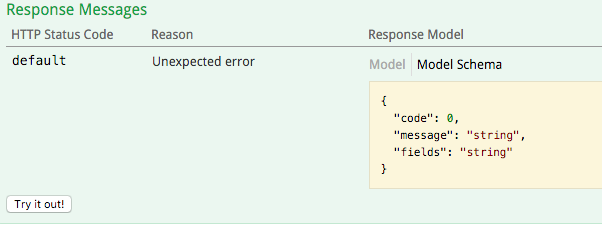


Figure : VSA API Browser-Testing the Rest Call

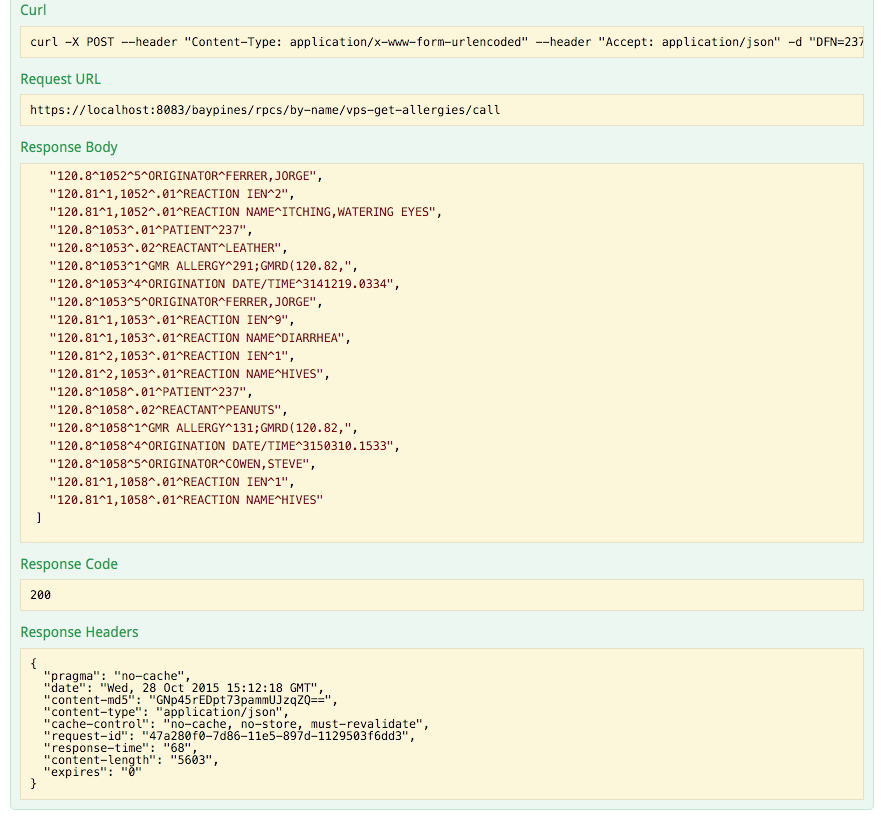


Figure : VSA API Browser-REST Call Results

## Multi-divisional Specifications

Not Applicable.

## Performance Specifications

Refer to Section 2.6.10 for detailed requirements.

VSA will execute performance, capacity, and independent testing of the VSA product, and as part of SQA analysis and testing. Web services exposing VistA functionality will be implemented in various layers (eMI/eMI, security validation, etc.). Some of these Web services can access one or more VistA systems concurrently. Performance will depend on the type of Web service (Federated for multiple VistA systems and Non-Federated for a single VistA).

**NOTE:** Requirements are subject to change after true performance testing is conducted at the Bay Pines Test Lab.

## Quality Attributes Specification

Refer to Section 2.6.9.

## Reliability Specifications

Refer to Section 2.6.13.

## Scope Integration

Refer to section 2.6, “Functional Specifications.”

VSA is not an application. VSA is a VistA SOA Services Platform that host services for exposing VistA functionality. Applications use VSA as a tool for generating services and as a runtime platform for hosting services. VSA integrates with all VistA systems with MVI and eMI. Therefore, scope integration for VSA is VistA SOA Services Platform integration.

## Security Specifications

Refer to Section 2.6.11.

## System Features

Refer to Section 2.6.3.

## Usability Specifications

Refer to Section 2.6.3.

# Purchased Components

Specifications for purchased components for the VistA Services Assembler (VSA) product will be determined as the project team collaborates with the systems/network teams.

# Estimation

Project Software Functional Size and Size-Based Effort and Duration Estimate

Application

| Item | A | B | C | D | E | Total |
| --- | --- | --- | --- | --- | --- | --- |
| **Counted Function Points** | 65 | N/A | N/A | N/A | N/A | 65 |
| **Estimated Scope Growth** | N/A | N/A | N/A | N/A | N/A | N/A |
| **Estimated Size at Release** | 49-85 | N/A | N/A | N/A | N/A | 49-85 |

| Size-Based Effort Estimates | Labor Hours | Probability |
| --- | --- | --- |
| **Low-Effort Estimate – With indicated probability, project will consume no more than:** | 1030 | 50% |
| **High-Effort Estimate – With indicated probability, project will consume no more than:** | 1920 | 75% |

| Size-Based Duration Estimates | Work Days | Probability |
| --- | --- | --- |
| **Low-Duration Estimate – With indicated probability, project will consume no more than:** | 68 | 50% |
| **High-Duration Estimate -- With indicated probability, project will consume no more than:** | 115 | 75% |

**NOTE:** Based on the minor changes to the RSD, a new Function Point Analysis (FPA) project estimation count is not warranted at this time.

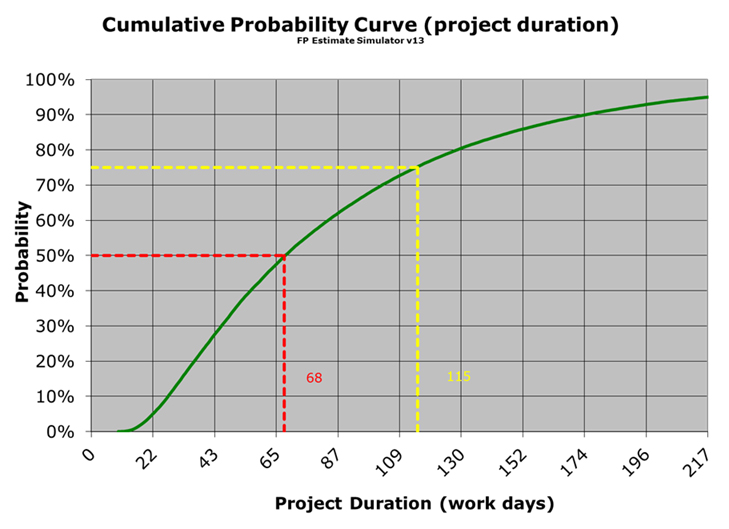


Figure : Cumulative Probability (“S-curve”) Chart—Project Duration

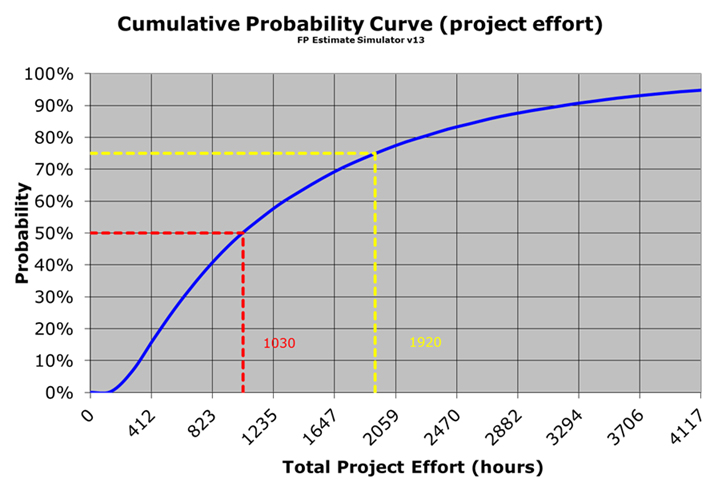


Figure : Cumulative Probability (“S-curve”) Chart—Project Effort

# Approval Signatures

REVIEW DATE:

SCRIBE: Elaine Laurel

Signed:

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

Steven Oster, Integrated Project Team (IPT) Chair Date

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

Mike Davis, Business Sponsor Date

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

Russell Holt, IT Program Manager Date

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

Lori Warren, Project Manager Date

Appendix A: Non-Functional Requirements

The following non-functional requirements should be reviewed and assessed while developing the requirements for the project.

Refer to Section [2.6.9](#_Non-Functional_Requirements) for detailed requirements.

System Performance Reporting Requirements

(Note: Each system developed by the Department of Veterans Affairs (VA) Office of Information and Technology (OI&T) must comply with the following mandatory requirements.)

1. Include instrumentation to measure all performance metrics specified in the Non-Functional Requirements section of the Requirements Traceability Matrix (RTM). At a minimum, systems will have the ability to measure reporting requirements for Responsiveness, Capacity, and Availability as defined in the non-functional requirements section of the RTM.
2. Make the performance measurements available to the Information Technology (IT) Performance Dashboard to enable display of “actual” system metrics to customers and IT staff.

Operational Environment Requirements

1. System response times and page load times shall be consistent with \_\_\_CPRS\_\_\_\_ standards (for example, My HealtheVet or HealtheVet). (Comment: There may be different expectations for an external display vs. a query. Need to address these different uses. Also indicate if this information is unknown).
2. Maintenance, including maintenance of externally developed software incorporated into the \_\_VSA\_\_\_application(s), shall be scheduled during off peak hours or in conjunction with relevant maintenance schedules. The business owner should provide specific requirements for establishing system maintenance windows when planned service disruptions can occur in support of periodic maintenance.
3. Information about response time degradation resulting from unscheduled system outages and other events that degrade system functionality and/or performance shall be disseminated to the user community within 30 minutes of the occurrence. The notification shall include the information described in the current Automated Notification Reporting (ANR) template maintained by the VA Service Desk. The specific business impact must be noted in order for OIT to provide accurate data in the service impact notice of the ANR.
4. Provide a real-time monitoring solution to report agreed/identified critical system performance parameters.
5. Critical business performance parameters shall be identified e.g., transaction speed, response time for screen display/refresh, data retrieval, etc. in a manner that data capture can occur to support metric reporting and support the OI&T performance dashboard display. If no such performance metrics are required or provided there will be no program specific Service Level Agreements (SLA) created, nor shall there be any active/real time monitoring through OI&T Performance Dashboard to provide the business owners any performance metrics.
6. Notification of scheduled maintenance periods that require the service to be offline or that may degrade system performance shall be disseminated to the business user community a minimum of 48 hours prior to the scheduled event.

Documentation Requirements

1. The training curriculum shall state the expected training time for primary users and secondary users to become proficient at using the \_\_\_\_\_\_\_\_\_\_\_\_ application(s).
2. All training curricula, user manuals and other training tools shall be developed/updated by \_\_\_\_\_\_ <<insert name of Program Office>> and delivered to all levels of users \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. If known, insert how much time in advance the training tools will be delivered and via what mechanism(s); for example, 2-4 weeks in advance of the release of the enhancement through nationwide conference calls and PowerPoint presentations). The curricula shall include all aspects of the enhanced \_\_\_\_\_\_\_\_ application(s) and all changes to processes and procedures.
3. The training curriculum developed by the Program Office shall state the expected task completion time for primary and secondary users.
4. User manuals and training tools shall be developed. If they already exist, updates shall be made, as necessary, to them and they shall be delivered to all levels of users.
5. IT will provide the level of documentation required to support the system and maintain operations and continuity. Documentation shall represent minimal programmatic and lifecycle operations support documentation artifacts as defined by VA standards in ProPath and as required by the VA Enterprise System Engineering Lifecycle and Release Management office for sustained operations, maintenance, and support (http://URL/lifecycle/default.aspx) prior to approval by any VA change control board and release into production.

Implementation Requirements

1. Technical Help Desk support for the application shall be provided for users to obtain assistance with \_\_\_\_\_\_\_\_\_\_\_.
2. The IT solution shall be designed to comply with the applicable approved Enterprise SLA.
3. The implementation must be complete by \_\_\_\_\_\_\_\_\_\_. (Enter date - dd-mm-yyyy)

Data Protection/Back-up/Archive Requirements

1. Based upon the criticality of the system, provide a back-up and data recovery process for when the system is brought off-line for maintenance or technical issues/problems.
2. Data protection measures, such as back-up intervals and redundancy shall be consistent with systems categorized as routine (30 day restoration), mission essential (72 hour restoration), or mission critical (12 hour restoration).
3. Business owners are required to state the mission criticality of the IT services required in order to assist the planners and developers in determining best strategies for engineering an IT solution to meet their business objectives/needs. The business owner needs to state the criticality of the data and the impact to the business during a service disruption so appropriate technologies can be considered.

Levels for Disaster Recovery

Table : Levels for Disaster Recovery

| Classification | Recovery Time Objective | Recovery Point |
| --- | --- | --- |
| Objective Routine | 30 day restoration | TBD |
| Mission Essential | 72 hour restoration | 24 hours |
| Mission Critical | 12 hour restoration | 2 hours |

Recovery Time Objective (RTO) – RTO defines the maximum amount of time that a system resource can remain unavailable before there is an unacceptable impact on other system resources, supported mission/business processes, and the MTD.

Maximum Tolerable Downtime (MTD) - The MTD represents the total amount of time the system owner/authorizing official is willing to accept for a mission/business process outage or disruption and includes all impact considerations.

Recovery Point Objective (RPO) - The RPO represents the point in time, prior to a disruption or system outage, to which mission/business process data can be recovered (given the most recent backup copy of the data) after an outage.

Data Quality/Assurance Requirements

A monitoring process shall be provided to ensure that data is accurate and up-to-date and provides accurate alerts for malfunctions while minimizing false alarms.

User Access/Security Requirements

Ensure the proposed solution meets all Veterans Health Administration (VHA) Security, Privacy, and Identity Management requirements including VA Handbook 6500 (see the Enterprise Requirements section of the RTM).

Usability/User Interface Requirements

Adhere to good User Interface/User Centered Design (UI/UCD) principles as outlined in the Usability Appendix of the BRD.

Conceptual Integrity

Provide standards based messaging and middleware infrastructure needed to support both Legacy Veterans Health Information Systems Technology Architecture (VistA) and future VistA 4 deployments.

Availability

1. Maintenance window, including maintenance of externally developed software incorporated into the VistA 4 application(s), will be by mutual agreement between OI&T and the VHA Point of Contact (POC) for the affected facility (ies). VHA will provide POCs for each facility.
2. VistA application unavailability due to an unplanned outage or planned outages that exceed the defined maintenance window will not exceed 8.76 hours per year and will not exceed 43.8 minutes per month (99.9% availability).
3. The application shall be available 24 hours a day, seven days a week, with an uptime of 99.9%.
4. All system updates and scheduled maintenance should occur between the hours of 1800 and 0600 (per local time zone), when clinical usage would be lightest.

Interoperability

1. The system shall support all recognized health system standards i.e., Health Level 7 (HL7), Fast Healthcare Interoperability Resources (FHIR).
2. Systems must be heterogeneous and agnostic for operating systems and code bases.

Provide the ability to securely transfer large files (of 4-8 gigabyte) from an external source to VA systems.

1. Provide access to the system over a remote access solution.

Manageability

1. Provide Service Desk/Incident and Problem Management tracking related to maintenance events of patient care systems with priority over non-patient care systems.
2. Provide data related to maintenance events, both routine and exceptional, including key metadata:

* Predicted routine work
* Occurrences where maintenance is completed, including restart from down time
* Identity of the organization performing maintenance
* User performing maintenance (if available)
* Identity of the system
* Date/time, physical location
* Systems impacted
* Does it affect patient care
* Non-urgent or emergent

1. Provide audit capabilities for system access and usage with settings that are configurable to support internal and external audits based on federal and VHA mandates.
2. The system must comply with VA Directive 6300 Records and Information Management and with VHA Records Control Schedule (RCS) 10-1, in general and specifically with Electronic Final Version of Health Record: Destroy/Delete 75 years after last episode of patient care, or longer (if specified).

Performance

1. Provide an Infobutton Query Responder on all platforms with a response time of less than .5 seconds.
2. The system shall recognize, report, and retransmit data lost, with less than 0-1% chance of incomplete patient records.
3. Provide patient data (for data within the system) transactions (e.g., capture, search, request for data) within .5 seconds.
4. Mouse or key-based UI controls, e.g., menus, checkboxes shall provide instantaneous responsiveness (<90ms).
5. Part-screen refreshes after user action shall complete within a pro-rated interval between 200 ms and 1200 ms times a percentage of the screen area being refreshed. For example, a component 10% of the screen area would refresh in (1200 – 200) \* 0.10 + 200 = 300 ms.

Reliability

1. Provide system reliability:

* Threshold = 99.9%
* Objective = 99.99% system and application

1. Provide system reliability:

* Level 1 severity =<1 failure per month
* Level 2 severity =<2 failures per month
* Level 3 severity =<3 failures per month

Security

Provide management of electronic attestation of information including the retention of the signature of attestation (or certificate of authenticity) associated with incoming or outgoing information.

Supportability

1. Provide alerts (that extend beyond system messages to external systems like mobile devices) for malfunctions, while preventing false alarms for local, regional, and national evaluations in real time.
2. Provide reports on performance metrics as specified in the VistA 4 Effectiveness and Value / Benefits Framework on a bi-weekly basis.
3. Provide national, regional, and local reports on performance metrics as specified in the VistA 4 Effectiveness and Value / Benefits Framework.
4. Provide performance metrics (from request for information to receipt of information on the screen) monitored by the system and system administrators so they know what the user experience is like without users having to call them and tell them the system is running very slow.
5. Provide the ability for VHA and IT staff to create standard and ad-hoc reports of usage, bandwidth, response time, login time, and other variables with a verification process for measuring the capabilities of the system.
6. Provide end-user training on how to generate the various system performance reports (e.g., in standard file formats such as Comma Separated Values [CSV], Portable Document Format [PDF], or Excel) depending on the user's needs.
7. Provide the ability to view system statistics (e.g., information on the specific network environment) and identify areas that are having issues or are beyond capacity, in near-real-time (to be quantified at a later time).
8. Technical Help Desk support for the application via instant message, on-line, phone, and remote desktop access support, shall be provided for users to obtain assistance 24/7.
9. The IT solution shall be designed to comply with the applicable approved Enterprise SLAs.
10. Data protection measures, such as back-up intervals and redundancy shall be consistent with systems categorized as mission critical (1hr restoration, 2hrs backup recovery). Impact of system failure must be monitored on a near real time basis.
11. Provide the ability to set thresholds and notification type (e.g., email or text alerts) when alerting the user about response time degradation and unscheduled outages.
12. Disaster Recovery Plans (DRP) and Continuity of Operations Plan (COOP) will be updated and tested semi-annually to address the VistA 4 product (see National Security and Homeland Security Presidential Directive: National Continuity Policy. NSPD-51/HSPD-20, May 9, 2007 <http://www.fas.org/irp/offdocs/nspd/nspd-51.htm>)

Usability

1. Provide view ability/usability of VistA 4 applications on mobile devices.
2. User prompts and screen help shall be embedded into the system to guide use of the solution.

Documentation

1. The training curriculum shall be provided in two hours or more of training time for primary users and secondary users to become proficient at using the VistA 4 application(s).
2. All training curricula, user manuals and other training tools shall be developed/updated by the VE Program Office and delivered to all levels of users 4 weeks in advance of the release of the enhancement through mediums that will best support the sharing of information to all affected staff.
3. Provide follow-up training classes tailored to VHA workflow 4 weeks after the users have begun to use the system

Appendix B: Acronym List and Glossary

This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the VistA Services Assembler (VSA) Phase 2.

Acronyms

In addition to the acronyms defined below, refer to the Office of Information and Technology (OI&T) Master glossary.

Table : Acronyms

| Term | Description |
| --- | --- |
| AITC | Austin Information Technology Center |
| API | Application Program Interface |
| BRD | Business Requirements Document |
| CDS | Clinical Data Services |
| CIO | Chief Information Office |
| CONUS | Continental United States |
| COOP | Continuity of Operations |
| COTS | Commercial-Off-The-Shelf |
| CSP | Caché Server Pages |
| CVS | Conformance Validation Statement |
| DFN | Data File Number |
| DoD | Department of Defense |
| DR | Disaster Recovery |
| EDE | Enterprise Development Department |
| eMI | Enterprise Messaging Infrastructure |
| ESB | Enterprise Service Bus |
| ESS | Enterprise Shared Services |
| EWD | Enterprise Web Development |
| FDCC | Federal Desktop Core Configuration |
| GUI | Graphical User Interface |
| HTTPS | Hypertext Transfer Protocol Secure |
| IAM | Identity and Access Management |
| iEHR | Integrated Electronic Health Record |
| IHS | Indian Health Service |
| IT | Information Technology |
| IV&V | Independent Verification and Validation |
| JAX-WS | Java API for XML Web Services |
| JSON | JavaScript Object Notation |
| M (MUMPS) | Massachusetts General Hospital Utility Multiprogramming System (now known as “M”) |
| MDWS | Medical Domain Web Services |
| MPI | Master Patient Index |
| MVI | Master Veteran Index |
| NIST | National Institute of Standards and Technology |
| OED | Office of Enterprise Development |
| OI&T | Office of Information and Technology |
| OSEHRA | Open Source Electronic Health Record Agent |
| PD | Product Development |
| PHI | Personal Health Identifiers |
| PII | Personally Identifiable Information |
| POAM | Plan of Action and Milestones |
| POC | Proof of Concept |
| RDC | Regional Data Center |
| REST | Representational State Transfer |
| RPC | Remote Procedure Calls |
| RSD | Requirements Specification Document |
| RTM | Requirements Traceability Matrix |
| SAML | Security Assertion Mark-up Language |
| SDD | Systems Design Document |
| SDLC | System Development Life Cycle |
| SEDR | Systems Engineering and Design Review |
| SME | Subject Matter Expert |
| SOA | Service Oriented Architecture |
| SOAP | Simple Object Access Protocol |
| SSL | Secure Socket Layer |
| TCP/IP | Transmission Control Protocol/Internet Protocol |
| TLS | Transport Layer Security |
| TRM | Technical Reference Model |
| URI | Uniform Resource Identifiers |
| URL | Uniform Resource Locator |
| VA | Department of Veterans Affairs |
| VAMC | VA Medical Center |
| VE | VistA Evolution |
| VHA | Veterans Health Administration |
| VistA | Veterans Health Information System and Technology Architecture |
| VM | Virtual Machine |
| VSA | VistA Services Assembler |
| WAR | Web Application ARchive |
| WMB | WebSphere Message Broker |
| WSDL | Web Services Description Language |
| WSRR | WebSphere Registry and Repository |
| XML | Extended Markup Language |

Definitions

Table : Definitions

| Term | Definition |
| --- | --- |
| Caché | An “M”-based product (by InterSystems) that has been selected as the next generation Veterans Health Information Systems and Technology Architecture (VistA) platform. Caché stores data in multidimensional arrays capable of carrying hierarchically structured data. These are the same “global” data structures used by the [MUMPS](http://en.wikipedia.org/wiki/MUMPS) programming language, which influenced the design of Caché, and are similar to those used by [MultiValue](http://en.wikipedia.org/wiki/MultiValue) (aka PICK) systems. In most applications; however, [object](http://en.wikipedia.org/wiki/Object_(computer_science)) and/or [SQL](http://en.wikipedia.org/wiki/SQL) access methods are used.  [Caché ObjectScript](http://en.wikipedia.org/wiki/Cach%C3%A9_ObjectScript), [Caché Basic](http://en.wikipedia.org/wiki/Cach%C3%A9_Basic) or [T-SQL](http://en.wikipedia.org/wiki/T-SQL) can be used to develop application business logic. External interfaces include native object binding for [C++](http://en.wikipedia.org/wiki/C%2B%2B), [Java](http://en.wikipedia.org/wiki/Java_programming_language), [EJB](http://en.wikipedia.org/wiki/EJB), [ActiveX](http://en.wikipedia.org/wiki/ActiveX), and [NET](http://en.wikipedia.org/wiki/.NET_Framework). Caché supports [JDBC](http://en.wikipedia.org/wiki/JDBC) and [ODBC](http://en.wikipedia.org/wiki/ODBC) for relational access. [XML](http://en.wikipedia.org/wiki/XML) and [Web Services](http://en.wikipedia.org/wiki/Web_Services) are also supported. |
| Connectivity | Connectivity provides connection to the Master Patient Index (MPI), Structured Query Language (SQL), and XML. |
| Federating Services Platform | Serves as the registry and repository for storing Service Descriptor and runtime packages (Web services). It federates routing of queries from provider and consumer “service” requests to and from Veterans Health Information Systems and Technology Architecture (VistA). It also provides the security interface that controls single sign-on (SSO) access, so a user’s authentication token is trusted across multiple VA Information Technology (IT) systems or organizations. |
| GUI | The Graphical User Interface application that is developed for the client workstation. |
| JSON | JavaScript Object Notation is an [open standard](http://en.wikipedia.org/wiki/Open_standard) format that uses [human-readable](http://en.wikipedia.org/wiki/Human-readable) text to transmit data objects consisting of [attribute–value pairs](http://en.wikipedia.org/wiki/Attribute%E2%80%93value_pair). It is used primarily to transmit data between a server and Web application, as an alternative to [XML](http://en.wikipedia.org/wiki/XML).[[2]](#footnote-2) |
| M Hosting Platform | A runtime system supporting ANSI standard M such as InterSystems Caché or GT.M. |
| MPI | | The Master Patient Index (MPI) is located at the Austin Information Technology Center (AITC). It is composed of a unique list of patients, each with an assigned Integration Control Number (ICN), and an associated list of VAMCs (Veterans Affairs Medical Centers) and other systems of interest where each patient has been seen. This enables the sharing of patient data between operationally diverse systems. Each patient record (or index entry) on the MPI contains multiple demographic fields which are updated to the Primary View of the MPI. |
| MVI | The Master Veteran Index returns identity data for person matching the criteria provided. Additionally, MVI will provide scores indicating a degree of certainty for each person. MVI also will provide a threshold, indicating which scores indicate a valid match. For the matched record, ICN and its identifiers in other correlated systems will be returned. |
| SOA | Service Oriented Architecture is a flexible set of design principles used during the phases of systems development and integration. A deployed SOA-based architecture will provide a loosely-integrated suite of services that can be used within multiple business domains. |
| SSL | A cryptographic protocol designed to provide communication security over the Internet. |
| VistA Services Assembler Wizard | A Web application tool which auto generates VistA SOA Services. |
| VistA SOA Federating Services Platform | A server for hosting Web services, to which VistA SOA Services are deployed. |
| VistA SOA Service | An SOA compliant service generated by the VistA Services Assembler Wizard with federating capabilities. Implemented in JAX-WS. |
| VistA Services Assembler SOA Service Descriptors | Meta data XML document created by the VistA Services Assembler Wizard used to auto generate VistA SOA Services. |
| VistA SOA Service Registry Entry | Entry in the WebSphere Registry and Repository (WSRR) used to govern a specific VistA SOA Service. |
| VistA SOA Service Proxy | Proxy on the Enterprise Messaging Infrastructure (eMI) used to abstract the service endpoint. |
| VistA | The Veterans Health Information Systems and Technology Architecture (VistA) – A rich automated environment that supports day-to-day operations at local VA health care facilities. VistA is built on a client-server architecture, which ties together workstations and personal computers with graphical user interfaces at VA facilities, as well as software developed by local medical facility staff. VistA also includes the links that allow commercial off-the-shelf software and products to be used with existing and future technologies. |
| WSDL | Web Service Description Language (WSDL) is a document that provides a common language to describe:   * What the Web service does. * What functionality it can provide. * What data it can deliver.   A developer can click the WSDL and generate the code automatically. |
| XML | XML is a [markup language](http://en.wikipedia.org/wiki/Markup_language) that defines a set of rules for encoding documents in a [format](http://en.wikipedia.org/wiki/File_format) that is both [human-readable](http://en.wikipedia.org/wiki/Human-readable_medium) and [machine-readable](http://en.wikipedia.org/wiki/Machine-readable_data). |

Template Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| September 2015 | 1.7 | Updated Headings and spacing to conform with latest OIT Documentation Standards guidelines | Process Management |
| June 2015 | 1.6 | Updated to conform with latest Section 508 guidelines and remediated with Common Look Office tool | Process Management |
| May 2015 | 1.5 | Revised by the PMAS Process Improvement Lockdown Team | PMAS Process Improvement Lockdown Team |
| December 2014 | 1.4 | Updated to conform with latest Section 508 guidelines and remediated with Common Look Office tool | Process Management |
| May 2014 | 1.3 | Reordered cover page to enhance search capabilities | Process Management |
| May 2013 | 1.2 | Add Appendix for acronyms and glossary | Process Management |
| March 2013 | 1.1 | Formatted to current ProPath documentation standards and edited to conform with latest Alternative Text (Section 508) guidelines | Process Management |
| January 2013 | 1.0 | Initial Version | PMAS Business Office |

1. The initial plans for IOC contained multiple sites across regions, with a large Federator footprint: The IOC focus for VistA.js is a much smaller footprint. Below is the original list of planned sites, if needed for historic purposes

   * Region 1: Boise, Puget Sound and Cheyenne
   * Region 2: Milwaukee, South Texas (San Antonio), and Minneapolis
   * Region 3: Richmond VA, Hampton VA, and Memphis TN
   * Region 4: Hudson Valley, Philadelphia, and Washington DC
   * Enterprise: Austin Information Technology Center and Philadelphia Information Technology Center
   * Regional Data Centers: Sacramento CA, St. Louis MO, Warner Robbins GA, Philadelphia PA, Denver CO, and Brooklyn NY

   [↑](#footnote-ref-1)
2. Definitions linked to Wikipedia.org Website: [Wikipedia](http://en.wikipedia.org/) [↑](#footnote-ref-2)