VistA Services Assembler (VSA) Phase 2 Business Requirements Document (BRD)

Department of Veterans Affairs

Office of Information and Technology (OI&T)

Product Development (PD)

Document Version 5.0

March 2015

Revision History

Table 1. Document Revision History

Date	Version	Description	Author
03/16/2015	5.0	Baselined document: • All review edits accepted and comments removed. • Document PDF to be sent out for signoff signatures. • Checked document styles and formatting; adjusted as needed. • Performed Section 508 review and updates.	VSA Development Team
03/16/2015	4.1	Final Peer Review.	VSA Development Team Travis Hilton Engility
03/11/2015	4.1	Revised Original BRD to reflect vista.js changes.	Travis Hilton Engility

Contents

Figures and Tables

Figures

Tables

1 Purpose

The VistA Services Assembler (VSA) Business Requirements Document (BRD) is authored by the business community for the purpose of capturing and describing the business needs of the customer/business owner. The BRD does the following:

- Provides insight into the “AS-IS” and “TO-BE” business areas.
- Identifies stakeholders.
- Profiles primary and secondary user communities.
- Identifies what capabilities the stakeholders and the target users need and why these needs exist.
- Provides a focused overview of the request requirements, constraints, and other considerations identified.

This document is a business case and does *not* mandate a development methodology; however, the requirements are written using agile methodology terminology. The intended audience for this document is the Office of Information and Technology (OI&T) to facilitate project planning when the project is approved and funded. These requirements are *not* documented at a level sufficient for development.

2 Overview

The Department of Veterans Affair (VA) computing origins consist of legacy systems that functioned in a largely autonomous manner. VA healthcare Information Technology (IT) automation, now known as Veterans Health Information Systems and Technology Architecture (VistA), began with a collection of decentralized systems capable of serving the needs of a single medical center. In the course of the following, it has become necessary to identify a new design approach for providing Service Oriented Architecture (SOA)-compliant integration between VistA applications and external systems:

- IT application modernization
- Architectural revisions for achieving Service Oriented Architecture (SOA) implementation
- Integration of external systems
- Diverse technologies
- Commercial-Off-The-Shelf (COTS) products

Implementing SOA design in the VA has presented the challenge of bridging the technical gap between the mainframe, procedural language environment of legacy VistA, and disparate (often object-oriented) technologies of COTS products and external systems. In addition to the need for identifying a technical solution, past efforts to implement SOA in the VA have illuminated an organizational gap as well; wherein, the software development staff involved with VistA generally lacked the technical skills necessary to efficiently develop integration solutions involving both VistA and other system technologies.

Early integration methods have used the following to transfer data between systems:

- HL7 messaging
- File transfers
- Mail messages

Previous attempts to implement SOA style integration usually followed one of two approaches:

- Replacement (“re-hosting”) of VistA applications (e.g., HealtheVet).
- Encasement solutions that attempt to compensate for the technical/design differences of VistA through the use of connectivity adaptors and intermediary databases.

The “encasement” solutions have been problematic in terms of:

- System Performance—Result in less than optimal system performance, because interaction from the adaptor to VistA tends to be “fine grained” resulting in overall chattiness between the “consuming applications” and the adaptor, and between the adaptor and VistA.
- Software redundancy—Often accomplish compensation for VistA characteristics through the use of redundantly deployed business logic, which can become out of synch with the logic in corresponding VistA “provider applications.”
- Overall sustainability—Treat VistA as a single system rather than a collection of applications that may require individual replacement.

As an alternative, the VistA Services Assembler (VSA) Phase 2 provides the ability for VistA to expose SOA-compliant services directly, or more appropriately, through proxies exposed on an Enterprise Service Bus (ESB)/Enterprise Messaging Infrastructure (eMI).

The VSA solution:

- Leverages existing VistA application functions and data exposure methods.
- Eliminates redundantly deployed business logic.
- Facilitates individual “plug and play” replacement of VistA applications.
- Ensure all corresponding business logic is positioned within the authoritative services for the “provider application”.
- Exposes SOA services directly and individually from “provider application.”
- Provides the ability to create “coarse grained” services reducing the overall chattiness (and eliminating interaction with the middle tier entirely).

The basic concept of the VSA solution is:

- Provisioning of software and hardware elements (including both “design-time” and “run-time” components that facilitate the exposure of VistA application logic as SOA-compatible services.
- Bridging the “orientation gap” between the traditional M development environment and SOA services (e.g., web services) deployment technologies.

An overview of this solution involves several areas including the following topics:

- Solution components
- “Design-time”
- “Run-time”
- Software deployment

It should be noted that the platforms that the VistA SOA services will be deployed to and the services generation wizard will be a “one-time” rollout (with subsequent maintenance updates as needed); whereas the distribution of VistA SOA services created and supported for these platforms would occur on an ongoing basis as a part of application software enhancements. It is important to recognize the distinction between the service generation wizard and associated platforms and business application “services” created for the platforms.

At the heart of the VSA solution is the concept that a service generation wizard will be provided that allows traditional M environment software developers or system integrators to auto-generate web services based on descriptive information specified by the user of the service generation wizard. This would produce the *non*-production environment software elements used in advancing the “service” through the established Software Development Lifecycle (SDLC). The service generation wizard would only be deployed and used in development (*non*-production) environments as a part of the “service” product development process.

Initial rollout of the VSA solution will involve the distribution of the service generation wizard to *non*-production environments, as well as production platforms that facilitate exposing VistA M logic as web services.

At “run-time,” application “services” that have been deployed for production use will be engaged as web services via ESB/eMI proxies. VistA system federating logic would provide a mechanism to route requests to appropriate VistA systems and aggregate returned results. VistA systems processing these requests would use platforms to expose web services and ultimately use VistA M-based application logic to execute actions necessary to respond to the “service” request.

3 Scope

The scope of VistA Services Assembler (VSA) comprises of two parts. First major scope element is to create the functionality necessary to automate the creation of Veterans Health Information Systems and Technology Architecture (VistA) Service Oriented Architecture (SOA) services and provide the platforms necessary to support and operate those services. This set of utilities is envisioned to include the service generation wizard and assorted components that support the following:

- Storage of service descriptors.
- Web service interaction with the traditional M computing environment.
- Federation of queries and responses across VistA systems.

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

- 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 VistA SOA services (e.g., the Service Registry, Enterprise Service Bus [ESB]/Enterprise Messaging Infrastructure [eMI], federating service platform, VistA, etc.).

In the initial VSA development effort, 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 demonstrate the functionality of platforms and service generation wizard. 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 platforms and service generation wizard to produce VistA based services.

Use of the platforms 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 Adapter [VIA], Clinical Data Services [CDS], etc.) for the integration of VistA with external systems/applications. As such, VSA needs to satisfy the consumer needs currently supplied by those approaches, as well as significantly enhancing:

- VistA security
- System performance
- Sustainability

4 Customer and Primary Stakeholders

Larry Clark, representing Department of Veterans Affairs (VA), is the primary stakeholder for this request.

Mike Davis, representing Veterans Point of Service (VPS), is also the primary stakeholder of this request.

REF: For the complete list of primary and secondary stakeholders, see Appendix C—Stakeholders, Users, and Workgroups.

5 Goals/Objectives and Outcome Measures

Table 2. Goals/Objectives and Outcome Measures

Goal/Objective and Desired Outcome	Impact/Benefit	Measurement
Suggest changing this to: Provide platforms and service generation wizard for automating the creation and execution of Veterans Health Information Systems and Technology Architecture (VistA) Service Oriented Architecture (SOA) services and supporting their use	*VistA can be easily integrated with external systems/applications through SOA-compliant architecture.*	*VistA SOA services are manageable through the ESB/eMI and consumable by external systems/applications.*
Enable the exposure of existing VistA data and methods as SOA-compatible web services and use of platforms and service generation wizard to create business services that provide business value to VA mission and objectives.	Advanced state of SOA architecture implementation in the VA and associated software functionality and sustainability benefits.	Current VA development and system integration initiatives are consumers of VistA SOA Services implemented by VSA.
Abstract VistA system developers and integrators from multiple technology orientation and consuming systems / applications can be integrated without extensive knowledge of VistA technology.	Improved use of existing staff skills, leveraging of previously implemented VistA application functionality.	VA software development staff is able to rapidly create and deploy VistA SOA Services with minimal re-training.
Improve VistA development “time to market” and overall system and automated, standardized production of VistA SOA Service components.	Provides cost effective methodology for the integration of VistA with external systems.	Development time for VistA-related services is notably reduced supporting efficient software delivery.

6 Enterprise Need/Justification

In recent years, an increasing priority has been placed on the following:

- Modernization of Department of Veterans Affairs (VA) computing environments.
- Integration of VA computing systems across the business domains (e.g., healthcare, benefits, and memorial affairs).
- Using COTS products that extend business functionality and interface with external systems (e.g., Department of Defense [DoD]).
- Provide shared collaboration space for other government agencies and outside vendors.

Previous approaches for Veterans Health Information Systems and Technology Architecture (VistA) Service Oriented Architecture (SOA) implementation have included:

- Replacement ("re-hosting") of VistA applications (e.g., HealtheVet).
- Encasement solutions that attempt to compensate for the technical/design differences of VistA through the use of connectivity adaptors and intermediary databases.

These approaches have been less than cost-effective, and in the case of the "encasement" set of designs, have been problematic in terms of:

- System performance
- Software redundancy
- Security
- Overall sustainability

There is a recognized need for a VistA approach that does the following:

- Provides a simplified (yet architecturally correct) technical solution.
- Addresses the organizational challenge of performing application development across multiple technical environments (e.g., traditional M computing and modern object-oriented web services design).

The importance of resolving this challenge is illuminated by an increasing number of high-priority efforts involving the integration of disparate systems suited to the Service Oriented Architecture (SOA) approach.

Both HL7 messaging and web services are regarded as methods for creating SOA-compliant services. And while VistA has implemented and used HL7 messaging for several decades, the ability to efficiently develop integration solutions that interact directly with VistA in the form of web services has been elusive.

There is also a need for a solution that:

- Alleviates the need for external system developers to have an extensive orientation to the technical functions internal to legacy VistA.
- Provides an automated VistA SOA service assembly methodology that minimizes, to the extent possible, further development burden on VistA applications by exposing methods that currently exist in the form of Remote Procedure Calls (RPCs) and Application Programming Interfaces (APIs).

In addition to bridging the technology gap between traditional M computing and the object-oriented/web-services-oriented design, the solution will need to:

- Facilitate the location of patient data across the approximately 130 decentralized VistA systems.
- Aggregate query responses returned to requesting systems.
- Combine fine-grained VistA methods into more coarsely-grained or composite services that reduce network “chattiness” between VistA SOA Services and consuming applications.

7 Business Requirements

7.1 Themes, Epics (Needs), and User Narratives (Business Requirements)

Themes, epics, user narratives, user stories, and acceptance criteria will be captured in the Requirements Traceability Matrix (RTM). Table 3 provides a list of the epics that are detailed in the RTM for the VistA Services Assembler (VSA) project. The RTM is stored as a separate document and can be accessed via the Requirements Traceability Link located in the New Service Request Database.

NOTE: A link to the Requirements Traceability Matrix will be inserted here once the Requirements Specifications Document (RSD) is completed and approved and will be used for the creation of RTM.

NOTE: Italicized font in Table 3 depicts changed requirements.

Table 3. VSA Phase 2 Requirements

Identifier	Epic
BN1	Adhere to Enterprise Requirements
BN2	**VistA Services Assembler (VSA) Phase 2 design-time utilities** ***Provide service generation wizard design-time utilities.*** **Justification for Change:** Removed VSA branding. Branding is not part of the business need. Use the general service generation wizard term.
BN2.1	Provide a Veterans Health Information Systems and Technology Architecture (VistA) Service Oriented Architecture (SOA) service generation Wizard—a user facing open source-compatible application to automate creation of VistA SOA Services. *Provide a Veterans Health Information Systems and Technology Architecture (VistA) Service Oriented Architecture (SOA) service generation* ***wizard****—a user facing open source-compatible application to automate creation of VistA SOA Services.* **Justification:** Use the general wizard term instead of a proper noun.
BN2.11	Facilitate the ability to expose existing VistA application business logic M routines (associated with Remote Procedure Calls [RPCs], Application Programming Interfaces [APIs], VA FileMan, etc.) as Intranet web services.
BN2.12	Facilitate the creation of VistA SOA Services for data and methods from a variety of VistA business domains (e.g., clinical, administrative, financial, etc.).
BN2.13	Provide the ability to ensure VistA SOA Services are created with standardized technical characteristics.
BN2.14	Provide the ability to define the creation of services that reference a single VistA system or federate the routing of edits, queries, and aggregation of responses across one, many, or all VistA systems.

Identifier	Epic
BN2.15	Provide the ability to generate services that are SOA-compliant and can be fully integrated with organizational SOA infrastructure (e.g., Enterprise Service Bus [ESB]/ Enterprise Messaging Infrastructure [eMI], repository, etc.).
BN2.16	Provide VistA SOA service generation utilities capable of creating both Simple Object Access Protocol (SOAP) and Representational State Transfer *(*REST)-style Web services. *Provide VistA SOA service generation utilities capable of creating REST-style Web services.* **Justification:** SOAP is no longer required for integration. REST is the new minimum requirement at the enterprise level and enables best performance.
BN2.17	Provide the ability to create a stored description that contains the "service descriptors" used by the VSA Wizard to auto generate (or regenerate) VistA SOA Services. *Provide the ability to create a service descriptor for each service generated by the service generation wizard that will enable the revision, regeneration, source control and other capabilities to take place.* **Justification**: Do not need VSA Wizard branding and clarified wording.
BN2.18	Provide the ability to store the service descriptors used in the generation of VistA SOA Services for future reference, revision, re-generation, source control, etc. *Provide the ability to store the service descriptors used in the generation of VistA SOA Services for future reference, revision, re-generation, etc. in source control.* **Justification**: Specifying source control as the target storage to align with enterprise standards for configuration management.
BN2.19	Provide integration between the Wizard and the VSA service descriptors repository. *Provide the ability for the service generation wizard to load, modify and save service descriptors. These service descriptors are delivered to and retrieved from source control.* **Justification:** Source control is now used to store and retrieve service descriptors. The service generation wizard loads them from disk and writes them to disk. A standard source control application is to deliver to and retrieve from source control.
BN2.2	Ensure that services assembled by the wizard are deployed only to the non-production, development environment. **Justification for Removal:** This is a constraint and will be moved to CO4 in Section 9.4, Constraints.
BN2.21	Facilitate the ability to check if M routines defined in as "service descriptors" in the VSA Wizard actually exist in the corresponding VistA M environments referenced. *Facilitate the ability to check if M routines defined in "service descriptors" and used by the service generation wizard actually exist in the corresponding VistA M environments referenced.* **Justification:** Clarified wording/text.
BN2.22	Facilitate the ability to define namespacing, versioning, payload transformations, etc. as "service descriptor" elements reflected in generated service deployment package. *Provide the ability define name spacing, versioning, payload transformations, etc. within the service generation wizard and to reflect that information as appropriate in each service descriptor and implement in each deployment package.* **Justification:** Clarified wording/text.

Identifier	Epic
BN2.23	Provide the ability to generate individual VistA SOA services packaged independently and configured by the VSA Wizard. *Provide the ability to generate individual VistA SOA services packaged independently and configured by the service generation wizard.* **Justification:** Clarified wording/text.
BN2.24	Provide the ability to generate multiple versions of a single (logical) service that can co-exist concurrently in the SOA environment.
BN2.25	Provide the ability to populate the development environment Service Registry and ESB/eMI with information that facilitates efficient testing of new services and requires minimal developer orientation to the Service Registry and ESB/eMI (*not* necessarily an automated functionality).
BN2.26	Configuration of ESB/eMI and Service Registry utility usage *shall* be strictly modularized so it can be reconfigured or omitted by other organizations as needed to correspond with available SOA infrastructure (or the absence thereof).
BN2.27	Facilitate the deployment of VSA services to the VistA federating platforms associated with corresponding M systems of the *non*-production development environment. *Provide processes and governance for the deployment of VistA SOA Services to the VistA SOA Service federating platforms associated with corresponding M systems of the non-production development environment.* **Justification:** Clarified wording/text.
BN2.28	VSA Wizard operations and associated “service descriptor” data *shall* include the ability to define VistA SOA service name, version, parameters and transformations, response and transformation, service operation names, mapping to M routines, input parameter descriptions and payload definitions, response description and payload definition, transformation definitions from the service operation parameter types to the M routine parameter types, M routine response type to service operation response type transformation, etc. *Service generation wizard operations and associated “service descriptor” data shall include the ability to define VistA SOA Service name, version, parameters and transformations, response and transformation, service operation names, mapping to M routines, input parameter descriptions and payload definitions, response description and payload definition, transformation definitions from the service operation parameter types to the M routine parameter types, M routine response type to service operation response type transformation, etc.* **Justification:** Replaced VSA Wizard branding.
BN2.29	VSA service descriptors *shall* include information related to specific components (e.g., payload transformation utilities, etc.) used in service descriptions where indicated. *VistA SOA Service - service descriptors shall include information related to specific components (e.g., payload transformation utilities, etc.) used in service descriptions where indicated.* **Justification:** Used consistent VistA SOA service wording.

Identifier	Epic
BN2.3	VSA Wizard *shall* provide for configurable federation of one, many, all VistA sites (subject to any site selection limitations associated with an individual developer). *Service generation wizard shall provide for configurable federation of one, many, all VistA sites (subject to any site selection limitations associated with an individual developer).* **Justification:** Replaced VSA Wizard branding.
BN2.31	The VSA Wizard will generate an “instruction packet template” for a given operation and store it in the service descriptor associated with that operation. When the operation is executed on the service, it will instantiate the instruction template and execute associated parameters. *The Service generation wizard will generate an “instruction packet template” for a given operation and store it in the service descriptor associated with that operation. When the operation is executed on the service, it will instantiate the instruction template and execute associated parameters.* **Justification:** Replaced VSA Wizard branding.
BN2.32	The VSA Wizard *shall* produce technically standardized, highly maintainable SOA services that conform to Enterprise Shared Service (ESS) standards. *Service generation wizard shall produce technically standardized, highly maintainable SOA services that conform to Enterprise Shared Services (ESS) standards.* **Justification:** Replaced VSA Wizard branding.
BN2.33	In addition to “service descriptors,” the VSA Wizard *shall* record VistA SOA service generation details (including originating developer, date/time, etc.). *In addition to “service descriptors,” the service generation wizard shall record VistA SOA service generation details (including originating developer, date/time, etc.).* **Justification:** Replaced VSA Wizard branding.
BN2.34	The VSA Wizard will validate that M routines specified in service descriptors actually exist in the development environment VistA systems referenced by the service being defined. **Justification for Removal:** Replaced by 3.27.
BN2.35	In its fully developed state, the VSA wizard functionality *shall* include a selection of an aggregation strategy, caching strategy, run-time data cleansing strategy, de-duplication/de-confliction strategy, partial data handling strategy, terminology mapping integration strategy (run-time terminology resolution and can include something with design time), multi-threading strategy, regional federating service location strategy, exception handling strategy, and security integration strategy. *Provide the ability for the service generation wizard, in its fully developed state, to include a selection of an aggregation strategy, caching strategy, run-time data cleansing strategy, de-duplication/de-confliction strategy, partial data handling strategy, terminology mapping integration strategy (run- time terminology resolution and can include something with design time), multi-threading strategy, exception handling strategy, and security integration strategy.* **Justification:** Clarified wording/text.

Identifier	Epic
BN3	**VSA infrastructure utilities for design-time activities and run-time VistA SOA service execution** ***Platform for design-time activities and run-time VistA SOA Service execution.*** **Justification:** Clarifying to use platform concept.
BN3.1	Provide VSA infrastructure utilities necessary to execute VistA SOA Services. *Ensure that the platform provided shall be adequate and necessary to execute VistA SOA Services.* **Justification:** Clarifying to use platform concept.
BN3.11	Provide and implement VistA SOA Federating Service Platforms at the regional computing level for the deployment of the VSA Wizard, service descriptors database, service federating routing/aggregation logic and individual VSA generated VistA SOA services. *Provide and implement VistA SOA Services federating service platforms at the regional computing level for the deployment of the service federating routing/aggregation logic and generated VistA SOA services.* **Justification:** Removed branding.
BN3.12	Create and deploy logic and components as needed to facilitate federated routing of queries and aggregation of responses to and from multiple VistA systems.
BN3.13	Ability to configure the use of external Identity and Access Management (IAM)/patient identity management services to associate patients with relevant system (e.g., VistA) activity.
BN3.14	Create and deploy logic and components as needed to provide for the definition and configuration of “site sets” with which VistA SOA services can be associated (one, many, all VistA sites).
BN3.15	VSA Federating Service Platforms *shall* employ standard clustering, etc. techniques to ensure scalability. *Provide VistA SOA Services federating service platforms that shall employ standard clustering, etc. techniques to ensure scalability.* **Justification:** Removed branding.
BN3.16	VSA utilities and components *shall not* contain VistA application-specific business logic other than that needed to facilitate the federation of queries and responses *VSA platforms shall not contain VistA application-specific business logic. The authoritative VistA SOA Services and authoritative local VistA services deployed to the platforms on the other hand may contain business logic as approved by their authoritative providers.* **Justification:** Changed to use the federating service platforms concept instead of VSA utilities and components.
BN3.17	Provide a VistA M Routine Calling Service (VMRCS) as a Web service deployed to each VistA instance that facilitates the use of M routines as the basis for VistA SOA service application business logic. *Provide a local VistA services platform that can be deployed to each VistA instance that provides for the hosting of local VistA services that encapsulate local VistA application business logic.* **Justification:** VMRCS has been replaced with the local VistA services platform and local VistA services.

Identifier	Epic
BN3.18	Ensure that the VMRCS component is a private service that can only be consumed by VSA services and employs two-way SSL connectivity. *Provide the ability to ensure that the local VistA services deployed the local VistA services platform at each VistA instance are private services that can only be consumed by VistA SOA Services deployed to the VistA SOA Services federating service platforms and employs enterprise approved security for transport.* **Justification:** Reworded to align with enterprise security guidance.
BN3.19	Create VMRCS logic and components for REST/SOAP Web services to be deployed to each site that delegates requests to run M routines to the VistA M Routine Calling Adaptor (VMRCA). **Justification for Removal:** It is satisfied by 3.17.
BN3.2	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. *Provide a local VistA services platform to support local VistA services that can run M routines.* **Justification:** VMRCS and VRMCA have been replaced with local VistA services.
BN3.21	Provide VMRCA functionality that ensures Kernel authentication, execution of M routines, transformation of payloads between M routine-compatible syntax and various external formats (e.g., XML and JavaScript Object Notation [JSON]), filtering (e.g., 20 fields to 3 fields, filter rows by a criteria), type conversions (e.g., VA FileMan date to Greenwich Mean Time [GMT]), and JSON removal of escape characters, etc. *Provide a local VistA services platform to support local VistA services that ensure enterprise security including IAM/Kernel security, execution of M routines, transformation of payloads between M routine-compatible syntax and various external formats (e.g., XML and JavaScript Object Notation [JSON]), filtering (e.g., 20 fields to 3 fields, filter rows by a criteria), type conversions (e.g., VA FileMan date to Greenwich Mean Time [GMT]), and JSON removal of escape characters, etc.* **Justification:** Kernel security and local capabilities to VistA are now provided by the local VistA services platform and the new local VistA services to be deployed to the platform.
BN3.22	Detailed service deployment design *shall* include the ability to decouple the physical endpoints, logical systems, and alternatives for MVI. Each of these would be supported in the platform, but abstracted from the service implementation so they are separately configurable (*not* hard coded).
BN3.23	VSA infrastructure utilities *shall* resolve RPC Broker security and capacity limitations (no encryption, 32k payload). *Provide a local VistA services platform to support local VistA services that shall resolve RPC Broker security and capacity limitations (no encryption, 32k payload).* **Justification:** Clarified wording/text.
BN3.24	VMRCA *shall* facilitate VistA “environmental shell” logic that ensures consuming application and user authentication has occurred and other required VistA system variables and checks are present. *Ensure that the local VistA services platform incorporates required enterprise security integration including runtime Kernel security environment integration.* **Justification:** VRMCA and security integration is replaced with local VistA service platform security integration.

Identifier	Epic
BN3.25	VMRCA *shall* provide a mechanism that determines/relays information regarding exceptions and errors encountered during the execution of M routines. *Provide a local VistA services platform to support local VistA services that implement ESS exception handling guidelines including communicating and handling of exceptions thrown by M routines.* **Justification:** VRMCA and exception handling integration is replaced with local VistA service platform exception handling integration.
BN3.26	Each VSA component *shall* include “markers” that facilitate elective performance monitoring of each component (*not* just end-to-end performance of a service transaction). *Provide a local VistA services platform to support local VistA services that include performance monitoring hooks and instrumentation to allow for component level performance monitoring.* **Justification:** Performance monitoring hooks previously defined with the VSA component wording have been replaced with performance monitoring hooks to be provided by the local VistA services platform and local VistA services.
BN3.27	Provide the runtime ability to verify that all dependencies of a VistA SOA Service are deployed and available in any given environment (i.e., M routines, MVI, local HTTP web services, etc.).
BN4	**“Reference implementation” VistA SOA Service examples** **Justification for Removal:** Reference implementation has been provided by the VPS team as part of the innovations lab.
BN4.1	Evaluate the “service needs” across major initiatives (e.g., VistA Evolution, mobile development and Connected Health). **Justification for Removal:** Reference implementation has been provided by the VPS team as part of the innovations lab.
BN4.11	Organize candidate use cases according to VSA component requirements (e.g., federated queries) to identify “real life” example test/demo services that can be produced with available VSA functionality. **Justification for Removal:** Reference implementation has been provided by the VPS team as part of the innovations lab.
BN4.12	Analyze the “service needs” to identify examples of functionality for which VistA application business logic already exists, so as to minimize/avoid the requirements elicitation and coding of VistA application business logic to the extent possible for VSA service test/demo examples. **Justification for Removal:** Reference implementation has been provided by the VPS team as part of the innovations lab.
BN4.13	Create several “reference implementation” service test/demo examples that demonstrate the viability of the VSA Wizard/run-time infrastructure utilities and a “pattern” of success relative to “read” and “write” services, federation of queries, etc. (e.g., MVI-facilitated queries using more than 5 sites, a “read” service, writing to VistA for functionality (e.g., progress notes, multiple service deployment, example of “paging”/aggregating data, exceptions, etc.). **Justification for Removal:** Reference implementation has been provided by the VPS team as part of the innovations lab.

Identifier	Epic
BN4.14	Coordinate with initiatives as indicated to arrange for critical testing and consumption of VistA SOA services generated and executed by the VSA Wizard and infrastructure utilities. **Justification for Removal:** Reference implementation has been provided by the VPS team as part of the innovations lab.
BN4.15	Provide both “read” and “write” VistA SOA service test/demo examples generated by the VSA Wizard and implemented in the Java Spring Framework that demonstrate the viability of the VSA service generation and execution methodology. **Justification for Removal:** Reference implementation has been provided by the VPS team as part of the innovations lab.
BN4.16	Facilitate creation of “authoritative” VistA SOA services in a manner such that all business logic necessary to perform application specific functionality is drawn from the VistA application of jurisdiction. This will generally be VistA M logic but can also include systems, applications and utilities by which VistA applications are extended. **Justification for Removal:** Reference implementation has been provided by the VPS team as part of the innovations lab.
BN4.17	Identify and document the process necessary to record the VistA SOA services in the Service Registry and expose them via the ESB/eMI (this is likely to be a manual process, at least in production), reference ability to decouple endpoints if ESB/eMI is available. **Justification for Removal:** Reference implementation has been provided by the VPS team as part of the innovations lab.
BN5	**VSA software development/distribution policy and process definition** ***Provide the ability for the distribution processes to distribute VistA SOA Services federating and local VistA services platforms and VistA SOA Services to the “open source” community, the VHA Innovations Sandbox, etc.*** **Justification:** Now using platforms.
BN5.1	Define the design of a standard VSA development environment that can be implemented via a replicable virtual machine configuration. *Provide the ability to define the design of a standard VistA SOA Services development environment that can be implemented via a replicable virtual machine configuration.* **Justification:** Clarified wording/text.
BN5.11	Define/document/provide the software distribution process and utilities necessary for the deployment of the VSA Wizard, the service descriptors database and VSA infrastructure utilities. *Provide the ability to define/document/provide the software distribution process and utilities necessary for the deployment of the service generation wizard, the service descriptors management in source control and service platforms.* **Justification:** Now using source control and platforms.
BN5.12	Define/document/provide the software distribution process and utilities necessary for the deployment of VistA SOA Services, including the ability to check for system dependencies (e.g., routines referenced by the “service” descriptor definition, etc.

Identifier	Epic
BN5.13	Defined VSA software and services distribution processes *shall* be consistent with organizationally established Software Development Lifecycle (SDLC) principles and industry best practices (including internal testing, User Acceptance Testing (UAT), promotion across *non*-production and production environments, etc.). *Provide the ability for defined VistA SOA Services and platforms software and services distribution processes to be consistent with organizationally established Software Development Lifecycle (SDLC) principles and industry best practices (including internal testing, User Acceptance Testing (UAT), promotion across non-production and production environments, etc.).* **Justification:** Now using platforms.
BN5.14	VistA SOA Service distribution process and utilities *shall* include smoke test functionality that verifies network connectivity, connection to endpoints and the existence of routines specified by VistA SOA services on all VistA systems referenced. *VistA SOA Service distribution process and platforms shall include smoke test functionality that verifies network connectivity, connection to endpoints and the existence of routines specified by VistA SOA services on all VistA systems referenced.* **Justification:** Now using platforms.
BN5.15	Define/document policy and process relative to developer access (including “local” developers) to the VSA Wizard, the “Federating Service Platform” and allowable site endpoint selections. *Provide the ability to define/document policy and process relative to developer access (including “local” developers) to the service generation wizard, the “Federating Service Platform” and allowable site endpoint selections.* **Justification:** Now using platforms.
BN5.16	Define/document policy and process relative to “local” developer access to regional/*non*-national distribution of VistA SOA services (e.g., “Class II” development). *Provide the ability to define/document policy and process relative to “local” developer access to regional/non-national distribution of VistA SOA services (e.g., “Class II” development).* **Justification:** Clarified wording/text.
BN5.17	Define/document policy and process relative to the distribution and governance of SSL certificates for VistA SOA services. Administration of certificates between the Federating Service Platform and VistA can be system-specific, between a service and VSA certificates can be handled by development teams that create services, etc. *Provide the ability to define/document policy and process relative to the distribution and governance of SSL certificates for VistA SOA services. Provide for consumer/producer enterprise approved system access control and its management.* **Justification:** Now using enterprise approved access control.
BH5.18	The distribution processes *shall* facilitate the ability to distribute VSA infrastructure utilities and VistA SOA Services to the “open source” community, the VHA Innovations Sandbox, etc. **Justification for Removal:** This was moved to 5 and will be removed from here
BN5.19	Define/document initial target deployment topology design.

Identifier	Epic
BN5.2	Define/document initial target deployment rollout process design.
BN5.21	Define/document policy and process for developer compliance with organizational ESB/eMI and Service Registry participation.
BN5.22	Define/document policy and process for the namespacing and versioning of VistA SOA service components.
BN5.23	Define/document policy and process related to “local” developer participation in the ESB/eMI, Service Registry, service generated wizard and Federating Service Platform.
BN6	**VSA General Characteristics** ***VistA SOA Services Platforms and Services General Characteristics*** **Justification:** Now using platforms.
BN6.1	VSA generated VistA SOA services *shall* provide complete compliance and integration with organizationally established SOA architecture and infrastructure, including the ability to decouple physical endpoints via ESB/eMI proxy abstraction, etc. *Generated VistA SOA services shall provide complete compliance and integration with organizationally established SOA architecture and infrastructure, including the ability to decouple physical endpoints via ESB/eMI proxy abstraction, etc.* **Justification:** Clarified wording/text.
BN6.11	The VSA infrastructure utilities and generated services *shall* facilitate environmental configuration that accommodates the characteristics of VistA systems of organizations other than the VA. *The VistA SOA services platforms and generated services shall provide environmental configuration that accommodates the characteristics of VistA systems of organizations other than the VA.* **Justification:** Now using platforms.
BN6.12	VistA SOA Services *shall* be authoritative and *non*-redundant (i.e., use VistA application M-based business logic). **Justification for Removal:** Uniqueness of services is handled by the ESS service design and evaluation process as defined by PMAS and ProPath. Depreciate 6.12 and don't reuse.
BN6.13	VistA SOA Services *shall* be manageable through a Service Registry/ESB/eMI (or consumable directly from the “Federating Service Platform” by organizations that do *not* have or do *not* wish to use a Service Registry/ESB/eMI). *VistA SOA Services shall be manageable through a Service Registry/ESB/eMI (or consumable directly from the VistA SOA services federating platforms by organizations that do not have or do not wish to use a Service Registry/ESB/eMI).* **Justification:** Now using platforms.
BN6.14	Creation of VistA SOA Services *shall* employ principles that use (or create as needed) “umbrella” logic in the VistA M-based application environment to implement “coarse-grained” and/or composite services to minimize network “chattiness” to the extent possible.

Identifier	Epic
BN6.15	The VSA infrastructure utilities *shall* be completely implementable with “open source” products and technology to the extent possible. *The VistA SOA services platforms shall be completely implementable with “open source” products and technology to the extent possible.* **Justification:** Now using platforms.
BN6.16	VSA solution development *shall* be closely coordinated with capacity and performance staff to ensure optimal tuning and avoidance of performance issues. *VistA SOA services platform solution development shall be closely coordinated with capacity and performance staff to ensure optimal tuning and avoidance of performance issues.* **Justification:** Now using platforms.
BN6.17	The VSA Wizard shall abstract M developers from the details of Web service creation and technology; as well as abstract system integrators from the details of VistA environment and technology. *The service generation wizard shall abstract M developers from the details of web service creation and technology; as well as abstract system integrators from the details of VistA environment and technology.* **Justification:** Clarified wording/text.
BN6.18	The VSA service deployment process *shall* integrate and comply with established organizational governance processes (e.g., ICR, ESS, etc.). *The VistA SOA services deployment process shall integrate and comply with established organizational governance processes (e.g., ICR, ESS, etc.).* **Justification:** Clarified wording/text.
BN6.19	The VSA effort *shall* provide for communications with OI&T and “open source” communities including announcements, presentations, training materials, etc. *The VistA SOA services platform effort shall provide for communications with OI&T and “open source” communities including announcements, presentations, training materials, etc.* **Justification:** Clarified wording/text.
BN6.2	VSA services *shall* provide system performance that meets/exceeds Enterprise-level demand and expectations, performance monitoring “markers” *shall* be implemented that can electively be activated to evaluate the performance of individual VSA components (*not* just end-to-end). *VistA SOA services platforms shall provide system performance that meets/exceeds Enterprise-level demand and expectations, performance monitoring “markers” shall be implemented that can electively be activated to evaluate the performance of individual VistA SOA services platform components (not just end-to-end).* **Justification:** Now using platforms.

Identifier	Epic
BN6.21	VSA utilities and services *shall* ensure that “environmental shell” information, variables, etc. are present at the time of “service” execution, including the communication of accurate user identity information to “provider” systems (e.g., VistA). *VistA SOA services platforms and services shall ensure that “environmental shell” information, variables, etc. are present at the time of “service” execution, including the communication of accurate user identity information to “provider” systems (e.g., VistA).* **Justification:** Now using platforms.
BN6.22	Compliance with all VA security standards related to the transport of data, etc.
BN6.23	Modular design able to use external person identity management services to add, obtain, and modify accurate user identity information where standardized person identity values are available.
BN6.24	No Personally Identifiable Information (PII) or Protected Health Information (PHI) *shall* be permanently stored by VSA infrastructure utilities (with the exception of service creation audit information, such as, the developer that created a service), caching may be necessary at times. *No Personally Identifiable Information (PII) or Protected Health Information (PHI) shall be permanently stored by VistA SOA services platforms (with the exception of service creation audit information, such as, the developer that created a service), caching may be necessary at times.* **Justification:** Now using platforms.
BN6.25	Implementation of industry standard security mechanisms as needed to protect PHI, PII and other sensitive VA data in motion and at rest.

7.2 User Access Levels

Table 4. User Access Levels

User Level	Role	Responsibilities	Access Level
Primary User (e.g., VPS, API 2.0)	Developers	Creation and deployment of Clinical Reconciliation Services.	Administrative for *non*-production environments
Primary User (e.g., eHMP)	Application Users	Will consume some components for integration into the VistA exchange platforms.	Administrative
Secondary User	VistA SOA services consumers (consuming applications)	Consume deployed VistA SOA Services.	Registered Services
Local VistA System Administrator	Administrators	Deploy services to local environments/platforms.	Administrative

User Level	Role	Responsibilities	Access Level
Federation VistA System Administrator	Administrators	Deploy services to federated environments/platforms.	Administrative

7.3 Known Interfaces and Data Sources

Figure 1. Known Interfaces and Data Sources

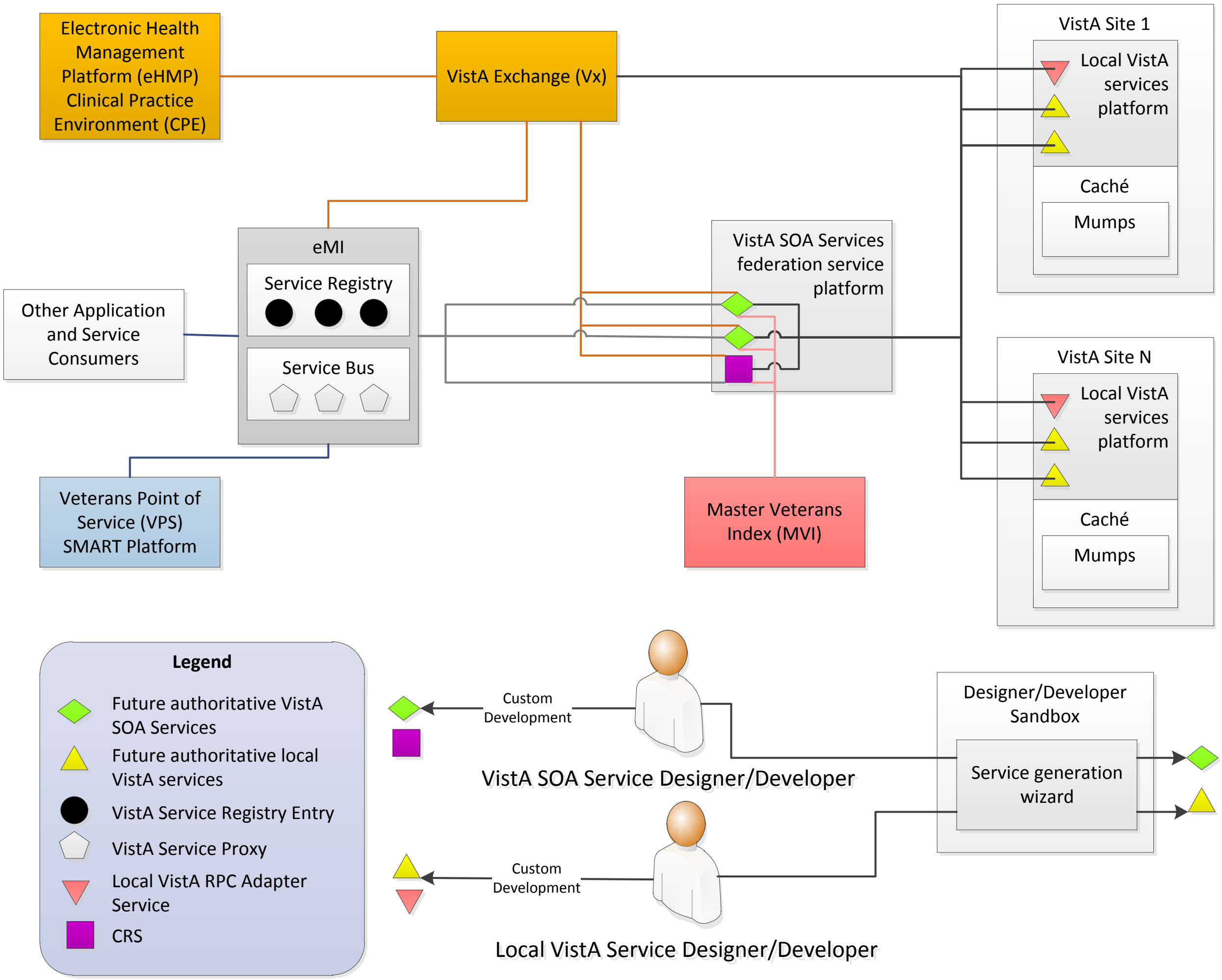
Electronic Health Management Platform (eHMP) Clinical Practice Environment (CPE)
VistA Exchange (Vx)
VistA Site 1
Local VistA services platform
Caché
Mumps
eMI
Service Registry
Service Bus
Other Application and Service Consumers
VistA SOA Services federation service platform
VistA Site N
Local VistA services platform
Caché
Mumps
Veterans Point of Service (VPS) SMART Platform
Master Veterans Index (MVI)
Legend
Future authoritative VistA SOA Services
Future authoritative local VistA services
VistA Service Registry Entry
VistA Service Proxy
Local VistA RPC Adapter Service
CRS
Custom Development
VistA SOA Service Designer/Developer
Designer/Developer Sandbox
Service generation wizard
Custom Development
Local VistA Service Designer/Developer

Table 5. Known Interfaces and Data Sources

Name of Application	Description of Current Application	Interface Type
Master Veterans Index (MVI)	Source of VistA site location information.	Outbound
VistA SOA Services federating service platform	VistA SOA services are hosted on federating service platforms provided by the solution.	Inbound/Outbound
Local VistA services platform	A local VistA service platform is deployed to each VistA instance. Local VistA services are deployed to these platforms.	Inbound/Outbound
Future authoritative VistA SOA Services	Authoritative VistA SOA Services will be developed and deployed to this solution's platforms to read and write data to local VistA services.	Inbound/Outbound
Future authoritative local VistA services	Authoritative local VistA services will be developed and deployed to this solution's platforms deployed at each VistA site. These services will read and write data to VistA.	Inbound/Outbound
Local VistA RPC Adapter Service	This local VistA service will be provided by the platform to enable one VistA Service operation to one RPC call integration.	Inbound/Outbound
Veterans Point of Service (VPS) Clinical Reconciliation Service (CRS)	This SOA service will be deployed to the federation platforms initially.	Inbound
VPS SMART Platform	This platform hosts many applications supporting Veterans Point of Service including Medication Reconciliation and Allergy Review MRAR.	Inbound/Outbound
VistA Exchange	The VistA Exchange platform will integrate components from this solution to write back to VistA.	Inbound/Outbound
Other Applications and Service Consumers	Many other applications and services will consume the VistA SOA Services through the eMI.	Inbound/Outbound
VistA	VistA will be written to and read from by authoritative local VistA services developed and deployed to this solution's platforms deployed at each VistA site.	Inbound/Outbound
Services generation wizard	Developers may use the services generation wizard to generate VistA SOA services and local VistA services.	Outbound
Electronic Messaging Infrastructure (eMI)	VistA SOA services are registered on the eMI services registry and proxies for these services are implemented on the eMI Service Bus. This capability provides for access control, endpoint routing, discovery, etc.	Inbound/Outbound

7.4 Related Projects or Work Efforts

Department of Veterans Affairs (VA) has explored a variety of system integration/connectivity solutions including intermediary databases (e.g., Corporate Data Warehouse [CDW]), as well as compensating middleware approaches (e.g., Medical Domain Web Services [MDWS], JMeadows, Clinical Data Service [CDS] etc.). Although these approaches provide a variety of services to consuming systems/applications, the combined “service needs” and urgency of current and emerging software development initiatives exceeds the ability of those solutions to provision Veterans Health Information Systems and Technology Architecture (VistA) services within the timeframes desired.

Although the VistA Services Assembler (VSA) integration pattern is regarded as the desired architectural approach for future system integration, it is expected in the near-term that multiple system integration/connectivity solutions will continue to coexist. However, long-term evaluation of these solutions is expected to occur once VSA is operational.

8 Service Level Requirements

8.1 Availability

Table 6: Availability

Service Level Requirement (SLR) Question	SLR Criteria	Description
1. How much time should the system be available (and how much down time is acceptable due to incident [unexpected] outage)?	99.9% (8.76 hours down time)	Many systems will depend on the services deployed to these platforms. They need to remain online to the extent that can be supported by the enterprise.
2. When should the system be available (what will be the core operating hours of the system)?	24x7	Many systems will depend on the services deployed to these platforms. They need to remain online to the extent that can be supported by the enterprise.
3. How soon should the system fully recover from an outage? (Includes Mean Time to Restore)	2 minutes	Many systems will depend on the services deployed to these platforms. They need to remain online to the extent that can be supported by the enterprise.
4. How much data will be restored when outage is recovered?	100% (continuous back-up)	Many systems will depend on the services deployed to these platforms. They need to remain online to the extent that can be supported by the enterprise.
5. What time period should be considered for maintenance periods?	Maintenance should be done concurrently. It should never be down for maintenance.	Many systems will depend on the services deployed to these platforms. They need to remain online to the extent that can be supported by the enterprise.
6. What standard time zone will the system operate in?	All time zones	Systems across the nation will depend on these platforms. The local VistA services platform deployed to each VistA instance can operate in the same time zone as the VistA it is connected to. Regionally deployed federation platforms operate in the time zone of the region. Federation platforms deployed to the AITC operate in the time zone of the AITC, etc.

8.2 Capacity and Performance

Table 7. Capacity and Performance

SLR Question	SLR Criteria	Description
1. How many users will be on the system hourly?	>1000	Many systems and applications will depend on the services deployed to these platforms.
2. How many transactions will each average user perform each hour?	>10	Many systems and applications will depend on the services deployed to these platforms.
3. What are the anticipated peak user times during the day?	Other (specify)	Day time health care systems and nightly processing systems. Daytime clinical users, Veterans and night time automated systems will continually drive the applications and processes that use the services deployed to the solution's platforms.
4. What is the anticipated peak transaction load (when do you think that there will be the most transactions being performed on the system) during the day?	*Other (specify)*	Day time health care systems and nightly processing systems. Load is expected at all times. Peak load patterns cannot be accurately predicted at this time.
5. How many new users will be added in one year?	*>1000*	The VPS Medication Reconciliation & Allergy Review (MRAR) system is the initial consuming system of the Clinical Reconciliation Service (CRS) initially deployed to the platforms. It is Veteran facing and will have wide adoption.
6. How many more (if any) transactions will be added in one year?	*>10*	More services will be developed and deployed to the platform in one year.
7. What kind of information will be stored (specify average of each kind per month)?	*Text based computable payloads (average TBD)*	Text based computable payloads will be stored and retrieved to and from VistA by the services deployed to the platform.
8. What kind of search capacity is required?	*Heavy (greater than 1,000 per hour)*	Assuming that searches imply reads. There is no specific "search" capability required for the platforms, but the services deployed to the platforms will read and write.
9. What type of system(s) is/are required?	*Intranet (All VA)*	Deployments will be at AITC, Regional data centers and all production VHA VistA instances.

SLR Question	SLR Criteria	Description
10. Is there a need for heavy application reporting? If yes, when?	*Other (specify)*	The platforms need to support Audit reporting capabilities that the services deployed to them can integrate with. Audit reports are expected as per enterprise audit reporting requirements.

8.3 Interfaces and Security

Table 8. Interfaces and Security

SLR Question	SLR Criteria	Description
1. Does this system interact with other existing systems?	Yes	Interacts with production VHA VistA instances and MVI.
2. Will this system require additional monitoring for Information Technology system metrics?	Yes	Production monitoring of the platforms and the services deployed to them shall integrate with enterprise production monitoring systems.
3. Will this system contain personally identifiable information, Protected Health Information, Health Insurance Portability and Accountability Act (HIPAA) information, or other confidential/regulated data?	Yes	Audit log entries for the services deployed to the platforms may, by necessity contain PHI and PII if the services deployed to them transmit PHI or PII.
4. Who will be the anticipated users of this system?	• Regional • VA • VA Central Office (VACO) • Enterprise Services applications	Data center administrative staff will deploy and maintain the systems. Project teams and business personnel may interact with monitoring and reporting sub-systems.

9 Other Considerations

9.1 Alternatives

The similar solution examples described herein do *not* represent the full set of Veterans Health Information Systems and Technology Architecture (VistA)-to-web-services approaches that are known to exist. Other organizations (e.g., Indian Health Service [IHS], Open VistA, and the vendor DSS) have created similar solutions; however, these are primarily middleware approaches rather than the perspective of exposing authoritative Service Oriented Architecture (SOA) services directly from VistA.

9.2 Assumptions

The following assumptions (AS*n*) have been identified:

- **AS1:** Sufficient resources will be made available to perform a “Phase 2” OI&T internal development effort that demonstrates the viability of the VSA design and produces near-term results that support the “service” needs of major initiatives. Additional development phases will be required to implement all features and functionality desired.
- **AS2:** Sufficient technologies and utilities are available and allowed (or will be allowed) by the VA Technical Reference Model (TRM), necessary to expose VistA M logic as Web services.
- **AS3:** The set of infrastructure utilities necessary to implement the VSA design are a straightforward and moderate level of effort that will allow the VSA solution to be developed without requiring a protracted timeframe or significant resources, relatively speaking.

9.3 Dependencies

The following VistA Services Assembler (VSA) dependencies (DE*n*) have been identified:

- **DE1:** Availability of stakeholders, SMEs, etc. necessary for the elicitation of detailed requirements and clarification of objectives.
- **DE2:** Availability of hardware platforms, licenses, etc. as necessary to deploy the initial set of VSA functionalities.

9.4 Constraints

The following VSA constraints (CO*n*) have been identified:

- **CO1:** Solution *must* comply with VA standards and conventions.
- **CO2:** The solution *must* provide satisfactory Enterprise-level capacity and performance.
- **CO3:** The solution *must* protect security, privacy, and confidentiality of data transmitted.
- **CO3:** The solution *must* provide Enterprise-level Continuity of Operations (COOP)/Disaster Recovery (DR) commensurate with the business criticality of data and processes supported by VSA services.
- **CO4:** The solution *must* ensure that the service generation wizard cannot deploy a service to production.

9.5 Business Risks and Mitigation

The VSA business risks (RI*n*) and suggested mitigation that has been identified as follows:

Table 9. VSA Business Risks and Mitigation

Business Risks	Mitigation
RI1: If insufficient resources are made available to support the development necessary to facilitate the creation of VSA-generated VistA SOA services, then near-term objectives to support major initiatives may *not* be met.	Coordinate with senior management and infrastructure development staff to communicate development needs and status as indicated.
RI2: If changing organizational priorities impacts resource availability, then development timelines, project functional scope, and quality can be adversely affected.	Communicate and coordinate the relationship of the VSA integration solution value relative to ongoing organizational system integration objectives.
RI3: If the VSA solution does *not* meet the needs of existing consuming applications and projected future integration needs, then consuming applications can continue using existing solution examples or seek other methods of system integration.	Work closely with stakeholders and consuming applications to understand requirements, examine full range of similar solution examples to identify design approaches, weaknesses, strengths, functional capacities, etc., which can be combined to create an effective VSA Enterprise-level solution.

10 Appendix A—References

- VA Handbook 6500 – Information Security Program: http://vaww.va.gov/vapubs/viewPublication.asp?Pub_ID=786&FType=2
- VA IT Strategic Plan goals for implementation of Service Oriented Architecture (SOA): http://vaww.vha.vaco.portal.va.gov/sites/ADUSH/ADUSH%20Web%20Files/VA_2011-2015_Strategic_Plan_Refresh.pdf
- VA Software Document Library (VDL): http://www4.va.gov/vdl/
- VHA Section 508 checklists (1194.21, 1194.22, 1194.24, 1194.31 and 1194.41): http://www.ehealth.va.gov/508/resources_508.html
- Office of Information and Technology (OI&T) Master Glossary: http://vaww.oed.wss.va.gov/process/OIT%20Master%20Glossary/Home.aspx
- VA Acronym Lookup: http://vaww1.va.gov/Acronyms/index.cfm

11 Appendix B—Models

VistA Services Assembler (VSA) Phase 2 design and process execution models to be created as a part of the detailed requirements elicitation and system design definition.

12 Appendix C—Stakeholders, Users, and Workgroups

Table 10. Stakeholder Support Team (BRD Development)

Type of Stakeholder	Description	Responsibilities
Requester	Dave Waltman, Senior Advisor to USH, VistA Evolution Triad	Submitted request. Submits business requirements. Monitors progress of request. Contributes to BRD development.
Endorser	Dave Peters Acting DCIO for Product Development, OI&T	Endorsed this request. Provides strategic direction to the program. Elicits executive support and funding. Monitors the progress and time lines.
Business Owner(s)/Program Office(s)	Larry J Clark, Supervisory Management & Program Analyst, OI&T	Provides final acceptance of BRD with sign-off authority. Provides strategic direction to the program. Elicits executive support and funding. Monitors the progress and time lines.
Business Owner(s)/Program Office(s)	Mike Davis, Director VPS, CBO	Provides final acceptance of BRD with sign-off authority. Provides strategic direction to the program. Elicits executive support and funding. Monitors the progress and time lines.
Business Subject Matter Expert(s) (SME)	Terry Luedtke (VA)/Travis Hilton (CTR) Technical Lead, ASD	Provide background on current system and processes. Describe features of current systems, including known problems. Identify features of enhancement.
Technical SME(s)	Neena Mathew (CTR), Technical SME, Apex	Provide technical background information about the current software and requested enhancements.
User SME(s)	Kevin Seiter (CTR), Requirements Analyst, VPS/Vecna	Ensure that the enhancements will account for current business processes and existing software capabilities.
Security Requirements SME(s)	Sheila Ervin, ISO, VA CSP	Responsible for determining and providing guidance on compliance with HIPAA.
Service Coordination SME(s)	Florence Ingram, VPS Project Manager, CBO	Responsible for ensuring all aspects of non-functional requirements have been accurately recorded for this request.
Business Liaison Staff	Lori Warren, Analyst, OI&T PD	Serve as the liaison between the Program Office (Business Owner) and Product Development throughout the lifecycle.

Type of Stakeholder	Description	Responsibilities
Requirements Analyst(s)	Victor Rodriguez/ Elaine Laurel Lead SQA Test Analyst/ Functional Analyst, OI&T PD	Responsible for working with all stakeholders to ensure the business requirements have been accurately recorded for this request.

13 Appendix D—User Interface/User-Centered Design Principles

For user interface and user-centered design, see the VistA Services Assembler (VSA) System Design Document (SDD).

14 Appendix E—Acronyms and Abbreviations

Table 11. Acronyms and Abbreviations

Term	Definition
ADC	Application and Development Competency
API	Application Programming Interface
API 2.0	VistA Application Programming Interface (API) Exposure 2.0 (Project)
ASD	Architecture Strategy and Design
BN	Business Need
BRD	Business Requirements Document
BRE	Business Rules Engine
CDS	Clinical Data Services
CDW	Corporate Data Warehouse
COOP/DR	Continuity of Operations/Disaster Recovery
COTS	Commercial-Off-The-Shelf (software or product)
DoD	Department of Defense
eHMP	Enterprise Health Management Platform
eMI	Enterprise Messaging Infrastructure
EMR	Electronic Medical Record
ERR	Enterprise Requirements Repository
ESB	Enterprise Service Bus
ESS	Enterprise Shared Services
FIPS	Federal Information Processing Standard
GMT	Greenwich Mean Time
GUI	Graphical User Interface
HL7	Health Level 7
IAM	Identity and Access Management
ICR	Integration Control Registration
iEHR	integrated Electronic Health Record
IHS	Indian Health Service
IT	Information Technology
JMeadows	Joint VA/DOD system integration middleware (Java based MDWS)
JSON	JavaScript Object Notation
M (MUMPS)	Massachusetts General Hospital Utility Multi-Programming System

Term	Definition
MDWS	Medical Domain Web Services (“Meadows”)
NIST	National Institute of Standards and Technology
NSR	New Service Request
O&M	Operations and Maintenance
OI&T	Office of Information and Technology
OWNR	Owner Requirement
PD	Product Development
PHR	Patient Health Record
REST	Representational State Transfer *(*REST)
RPC	Remote Procedure Call
SDLC	Software Development Lifecycle
SLA	Service Level Agreements
SME	Subject Matter Expert
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
TRM	Technical Reference Model
UAT	User Acceptance Testing
VA	Department of Veterans Affairs
VACO	VA Central Office
VAMC	VA Medical Center
VDL	VA Software Document Library
VHA	Veterans Health Administration
VistA	Veterans Health Information Systems and Technology Architecture
VMRCA	VistA M Routine Calling Adaptor
VMRCS	VistA M Routine Calling Service
VPS	Veterans Point of Service
VSA	VistA Services Assembler
XML	Extensible Mark-up Language

15 Appendix F—Acceptance Signatures

The requirements defined in this document are the high level business requirements necessary to meet the strategic goals and operational plans of the Application and Development Competency (ADC). Further elaboration to these requirements may be done in more detailed artifacts.

Business Owner

Signifies that the customer accepts the documented requirements, that they adequately represent the customers desired needs, and that the customer agrees with the defined scope.

Signed: ____

Larry J. Clark, Supervisory Management & Program Analyst

VistA Services Assembler Phase 2(VSA-P2) Business Sponsor Date

Business Owner/ Business Liaison

Signifies that the customer accepts the documented requirements, that they adequately represent the customers desired needs, and that the customer agrees with the defined scope.

Signifies appropriate identification and engagement of necessary stakeholders, and the confirmation and commitment to quality assurance and communication of business requirements to meet stakeholder expectations.

Signed: ____

Mike Davis, Director VPS, CBO

VistA Services Assembler Phase 2(VSA-P2) Business Sponsor Date

Customer Advocate

Confirms that the request merits consideration and review by the Business Intake Review Board.

Signed: ____

Lien Q. Dinh

Director Product Engineering, OI&T Date

Office of Information and Technology

Indicates agreement that the requirements have been received, are clear, understandable, and are documented sufficiently to facilitate project planning when the project is approved and funded. It is understood that negotiations may need to occur with the Business Owner during project planning as a result of technical reviews and feasibility.

Signed: ____

Steven Oster, OI&T-Product Development (PD) Project Manager

VSA-P2

VSA-P2 Integrated Project Team (IPT) Co-Chairperson   Date