



Interagency Program Office Program Management Support

*IPO iEHR Volume 2 SOA Service Governance
10312013*



IPO PM Support
Department of Defense / Department of Veterans Affairs Interagency Program Office

Document Number: Volume 2 SOA Service Governance
Release/Revision Status: Version 1.3
Release/Revision Date: October 31, 2013
File Name: IPO_iEHR_Volume_2_SOA_Service_Governance_10312013.doc

Unclassified

This page intentionally left blank.

DRAFT

Approved By:

<Name>
<Title>, <Organization>

Date

DRAFT

Record of Changes

Date	Authors	Version	Change Reference
02/15/2013	Dr. Arunava Chatterjee, William Sweet, Joe Diliberto, Jr., Raju Vemulamanda, Raju Prasannappa	1.0	Initial Version
05/15/2013	Dr. Arunava Chatterjee, William Sweet, Raju Prasannappa, Kim Galway	1.1	Updated Integration governance, compliance
06/06/2013	Raju Prasannappa	1.2	Removed the signature page
08/08/2013	Dr. Arunava Chatterjee	1.3	<ol style="list-style-type: none"> 1. Divided volume into separate parts with own references section, wherever applicable 2. Updated SLM Section Workflows 3. Added Error Codes 4. Added Message Priorities 5. Updated Service Taxonomy 6. Added Service Metadata 7. Updated Business Rules Section 8. Removed Ontology Engineering Tutorial 9. Removed Transactions Tutorial. 10. Consolidated sections and removed some inconsistencies

This page intentionally left blank.

DRAFT

Table of Contents

Part 1	SOA Service Governance.....	1
1-1.	Introduction.....	1
1-2.	Purpose.....	2
1-3.	Scope.....	3
1-4.	Target Audience.....	4
1-5.	Service Lifecycle Management.....	5
1-5.1	Roles and Overall Flow.....	6
1-5.2	Inception.....	7
1-5.2.1	Inception Workflow.....	7
1-5.2.2	Inception Phase Policies.....	8
1-5.2.3	Service Registration.....	9
1-5.3	Design.....	10
1-5.3.1	Design Workflow.....	10
1-5.3.2	Design Phase Policies.....	11
1-5.4	Construction.....	13
1-5.4.1	Construction Workflow.....	13
1-5.4.2	Construction -time Testing.....	14
1-5.5	Testing.....	15
1-5.5.1	Testing Workflow.....	15
1-5.5.2	Testing Policies.....	17
1-5.6	Deployment.....	17
1-5.6.1	Deployment Workflow.....	17
1-5.6.2	Deployment Policies.....	19
1-5.6.3	Release Management Guidance.....	19
1-5.7	Operation.....	19
1-5.7.1	Operation Workflow.....	19
1-5.7.2	Operation Policies.....	21
1-5.8	Deprecation.....	21
1-5.9	Retirement.....	23
1-5.10	Service Life Cycle Management (SLM) Waivers.....	23
1-5.11	Service Review Checkpoints.....	24
1-5.12	References.....	25
Part 2	Common Service Harvesting.....	26
Part 3	Service Documentation.....	28
Part 4	Architecture Guidance.....	29
4-1.	Architecture Approach.....	29
4-1.1	Architecture Documentation.....	30
4-1.2	Reference Architecture.....	30
4-1.2.1	Conceptual RA.....	30

4-1.2.2	iEHR Future State Architecture	31
4-1.2.3	Transition Architecture	32
4-1.2.4	Platform Specific Model	36
4-2.	References	43
Part 5	Messaging	44
5-1.	Message Model Standards	44
5-2.	Message Protocol Standards.....	45
5-3.	Message Criticality and Patient Safety Frameworks.....	46
5-4.	Service Taxonomy.....	47
5-4.1	Taxonomy Attributes.....	47
5-4.2	Documenting Taxonomies.....	48
Part 6	Registry and Repository Guidelines.....	50
6-1.	Working with the Repository	50
6-1.1	Definition and Disambiguation.....	50
6-1.2	Design-Time Repository Guidelines	51
6-1.3	Design Time Repository Artifacts	52
6-1.3.1	Non-Versioned Artifacts	52
6-1.3.2	Versioned Artifacts.....	52
6-1.4	Run-Time Repository Guidelines.....	52
6-1.5	Run-Time Repository Artifacts.....	53
6-1.5.1	Non-Versioned Artifacts	53
6-1.5.2	Versioned Artifacts.....	53
Part 7	Architectural Patterns	55
7-1.	Supported Web Service Standards	55
7-2.	Interface Management Standards	56
7-3.	Enterprise Integration Software Application Patterns	57
7-4.	SOA Service Patterns.....	58
7-5.	Real-time Access.....	59
7-6.	Integration Governance	60
7-6.1	Integration Pattern Governance Process.....	60
7-6.2	Integration Component Certification Process:	60
7-7.	Message Exchange Patterns.....	62
7-8.	Solution Integration Patterns	63
7-9.	Component Integration Patterns.....	65
7-10.	Implementation Guidance.....	66
7-10.1	Message Broker Implementation Patterns	66
7-10.2	Interoperability Guidelines	66
7-11.	Business Rules Governance	67
7-12.	References	68
Part 8	Naming Conventions	69
8-1.	Service Naming	69

8-2. Namespaces..... 69

Part 9 Coding Conventions 70

Part 10 Logging, Auditing and Error Handling 71

 10-1. Logging..... 71

 10-2. Auditing Records 74

 10-3. Error Codes 78

 10-4. References 79

Part 11 Versioning..... 80

 11-1. Versioning Policies 80

Part 12 Security..... 82

 12-1. Security Considerations..... 82

 12-1.1 Additional Access Control Topics 82

 12-2. Security Policies 84

 12-3. References 85

Part 13 External Governance Dependencies 86

 13-1. Business Governance 86

 13-2. Information Governance 87

 13-3. Infrastructure Governance 88

Part 14 iEHR SOA Compliance Criteria 89

 14-1. Business Dimension: Componentized Business Provides and Consumes Services..... 89

 14-2. Organization and Governance Dimension: Emerging SOE Governance 90

 14-3. Method Dimension: Service-Oriented Modeling 91

 14-4. Application Dimension: Service Design 92

 14-5. Architecture Dimension: Emerging SOA..... 93

 14-6. Information Dimension: Information as a Service 94

 14-7. Infrastructure and Management Dimension: Project-based SOA Environment..... 95

 14-8. References 96

Part 15 SLM Entry and Exit Criteria 97

Part 16 SLM Checklists 98

Part 17 SLM Waiver Form..... 105

Part 18 Service Documentation Template..... 107

 18-1. Service Documentation - Blank 107

 18-2. Service Documentation - Completed 110

Part 19 Taxonomy Registration Template 115

Part 20 Namespace Registration Template..... 116

Part 21 Roles and Responsibilities..... 117

Part 22 OSIMM..... 118

 22-1. OSIMM Level 4: Service 118

 22-2. OSIMM 7 Dimensions..... 119

 22-2.1 Business..... 119

 22-2.2 Organization & Governance 119

22-2.3	Method	119
22-2.4	Application.....	119
22-2.5	Architecture	119
22-2.6	Information	119
22-2.7	Infrastructure & Management.....	119
Part 23	PMAS Milestones	120
Part 24	iEHR Service Metadata	121
Part 25	Additional References.....	126

DRAFT

PART 1 SOA SERVICE GOVERNANCE

1-1. Introduction

The Department of Defense (DoD) and Department of Veterans Affairs (VA) integrated Electronic Health Record (iEHR) initiative is in the process of making Service Oriented Architecture (SOA) its primary architectural paradigm. Throughout this document, “iEHR” will be used to represent the current initiative.

In defining the complete implementation, the SOA Suite Integrated Project Team (IPT) has decided to create three subsections of the SOA across the enterprise:

- **Service Oriented Enterprise (SOE):** the SOE implies a consistent, enterprise-wide approach to service orientation, including necessary organizational structures and an enterprise roadmap. This information is available in Volume 1: SOE Governance
- **Service Oriented Architecture (SOA):** the SOA implies an implementation of the SOA paradigm to include policies and practices for the governance of services. This document, covers the SOA.
- **Service Oriented Infrastructure (SOI):** the SOI implies the hardware, network, virtualized servers, and operating systems necessary to enable the SOA. The SOI is covered in Volume 2: SOI Governance.

This volume focuses on SOA. It provides a rudimentary discussion of the architecture as well as guidance regarding design and development.

This document should be considered a living document and subject to modification and refinement based on input from stakeholders.

1-2. Purpose

The purpose of this document is to establish a set of processes and policies as well as provide overall guidance regarding the implementation of SOA.

DRAFT

1-3. Scope

This document discusses the following topics.

- SOA Services Lifecycle
- Services Architecture Standards and Policies
- Services Development Standards and Policies
- Release Management Standards and Policies
- Run-time Governance

DRAFT

1-4. Target Audience

The intended audience of this volume is: the divisions within the Office of the Chief Information Officer (OCIO), TRICARE Management Activity (TMA), the Interagency Program Office (IPO), the Office of the Chief Technology Officer (OCTO), the Service Military Medical departments, the VA Office of Information (OI) Architecture Strategy and Design (ASD) and Service Delivery and Engineering (SDE), and other iEHR stakeholders, as appropriate. This SOE strategy document is intended to be refined in a collaborative manner with input from all stakeholders.

DRAFT

1-5. Service Lifecycle Management

The Service Lifecycle is an integral part of Service Portfolio Governance where Services and their respective Service Consumers are managed from Service Inception to Deprecation and ultimately Service Retirement.

The Service Lifecycle is the collection of events that Services undergo as part of their use. Typically, these events can be broken down into several phases.

1. **Inception** – Services are conceptualized based on business needs.
2. **Design** – Services are architected to fulfill specific business functions. The design must comply with service design principles as provided by the governance body, the SOA Center of Excellence (CoE).
3. **Construction** – Services are created using multiple technologies. The construction of the Services must comply with service construction principles as provided by the CoE.
4. **Testing** – Services undergo test procedures to verify that they exhibit appropriate functional and nonfunctional behavior. The testing of the Services must meet CoE guidelines and standards. The Services must pass CoE mandated criteria.
5. **Deployment** – The Services are introduced into the SOA ecosystem. The deployment of the Services must take into account necessary configuration changes to the Services and SOA such that they perform in the SOA production environment without impeding the behavior of the SOA ecosystem. Documentation is appropriately changed and entered in to the service registry
6. **Operation** – In this phase, the services are running in the SOA production environment. During operation, the Services are monitored for Service Level Agreement (SLA) conformance and compliance with business policies. As needed, the SOA and the Services are adjusted such that all run-time governance criteria are met. Updates, patches, and the presence of multiple versions of the Services must be accounted for and managed such that the SOA production environment is not adversely affected.
7. **Deprecation** – Services eventually approach the end of their Lifecycle. Actors impacted by the service must be informed that the services are approaching end of life. Information regarding how the services will be replaced is provided, and the consumers are given a time window during which they can prepare for the service end-of-life.
8. **Retirement** – Services are removed from the SOA ecosystem. As before, this must occur in a manner that does not impede SOA from performing according to SLAs.

Phases 1 – 6 map to the Service Lifecycle definition defined in the Open Group Standards [3]. Phases 7 and 8 are additional phases added for iEHR.

The state of a particular Service is defined by its phase in the Service Lifecycle (e.g., Inception, Construction, etc.).

The Service Lifecycle can have multiple entry points but the execution of the Service Lifecycle must take into account the needs of various Actors. In particular, the Service Consumer-Service Provider interaction must be considered. For example, the SLA between the service consumer and provider must be defined and enforced during design time and run-time.

In addition, successful execution of the activities in these phases requires a set of overarching “Governing Processes”: Compliance, Dispensation, and Communication.

- Compliance ensures that the organization’s policies and standard processes are adhered to at each step of the Service Portfolio Governance process.
- Dispensation allows for deviations from the processes to be evaluated and take appropriate action such as waivers, appeals, or trigger re-evaluation of the process, etc.

- Communication is the process by which the organization’s policies and decisions are communicated to the Providers, Consumers of the Services, as well as other actors with an interest in the iEHR SOA and its implementation.

Compliance will be implemented by enforcing check points. Any exceptions will be handled by Dispensation processes defined by the CoE, and communicating the results of these two Governance processes will be the responsibility of the CoE in cooperation with the iEHR IPO.

The Service Lifecycle phases are described in detail in the rest of this section.

1-5.1 Roles and Overall Flow

The Service Lifecycle phases are described in detail in the rest of this section. As an overall view the entire flow is shown in Figure 1. The Roles are described in Part 21.

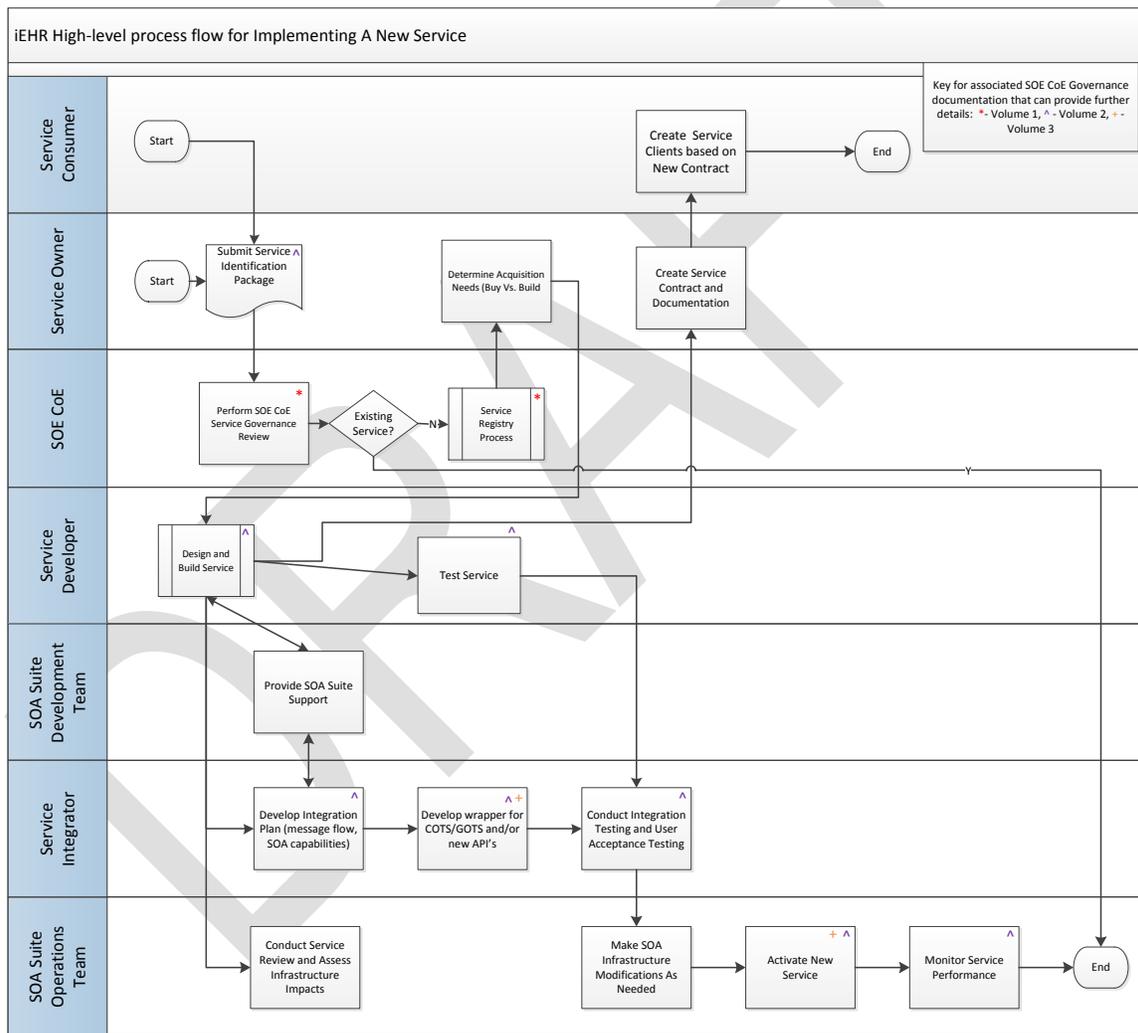


Figure 1 - Roles and Activities for Service Integration

1-5.2 Inception

A request for functionality or change to an existing Service or set of Services is initiated by Stakeholders such as Service Consumers or Service Owners. This information will be used by the Business Working Group (BWG) to ensure that a macro-business evaluation is taken. This approach takes into consideration how other business owners could use the same functions and address future needs. The goal is to provide opportunities for reuse and flexibility to allow the business to grow with minimal impact to the infrastructure.

It is the responsibility of the Service Developer to suggest how this functionality is to be enabled (i.e., new Service(s), update to existing Service(s), composition of Service(s), Business Processes etc.). The suggestion is subsequently verified by the CoE Technical Working Group (TWG). The TWG decides whether or not to accept the suggestion, offer a different solution, or request a revision of the suggestion. It is the responsibility of the CoE TWG to search the registry and repository for an existing service for reuse and reject the request for new service if a similar service already exists. If an existing service meets the business need with no changes, this is communicated to the Stakeholder at this time. If changes are needed to an existing service, then an Engineering Change Proposal (ECP) is generated.

If a new service is required, the Service Owner must create a service definition document that includes a description of the service, service version, service policy, and the service level contracts for the consumers. A template for Service Documentation is provided in Part 19. The CoE team will assign an owning organization to the service if the request is approved.

1-5.2.1 Inception Workflow

Figure 2 - Inception Process FlowFigure 2 shows the steps and the actors involved in this phase. Here, Service Portfolio Management is an actor within the BWG that is responsible for maintaining cost and risk related information of Services..

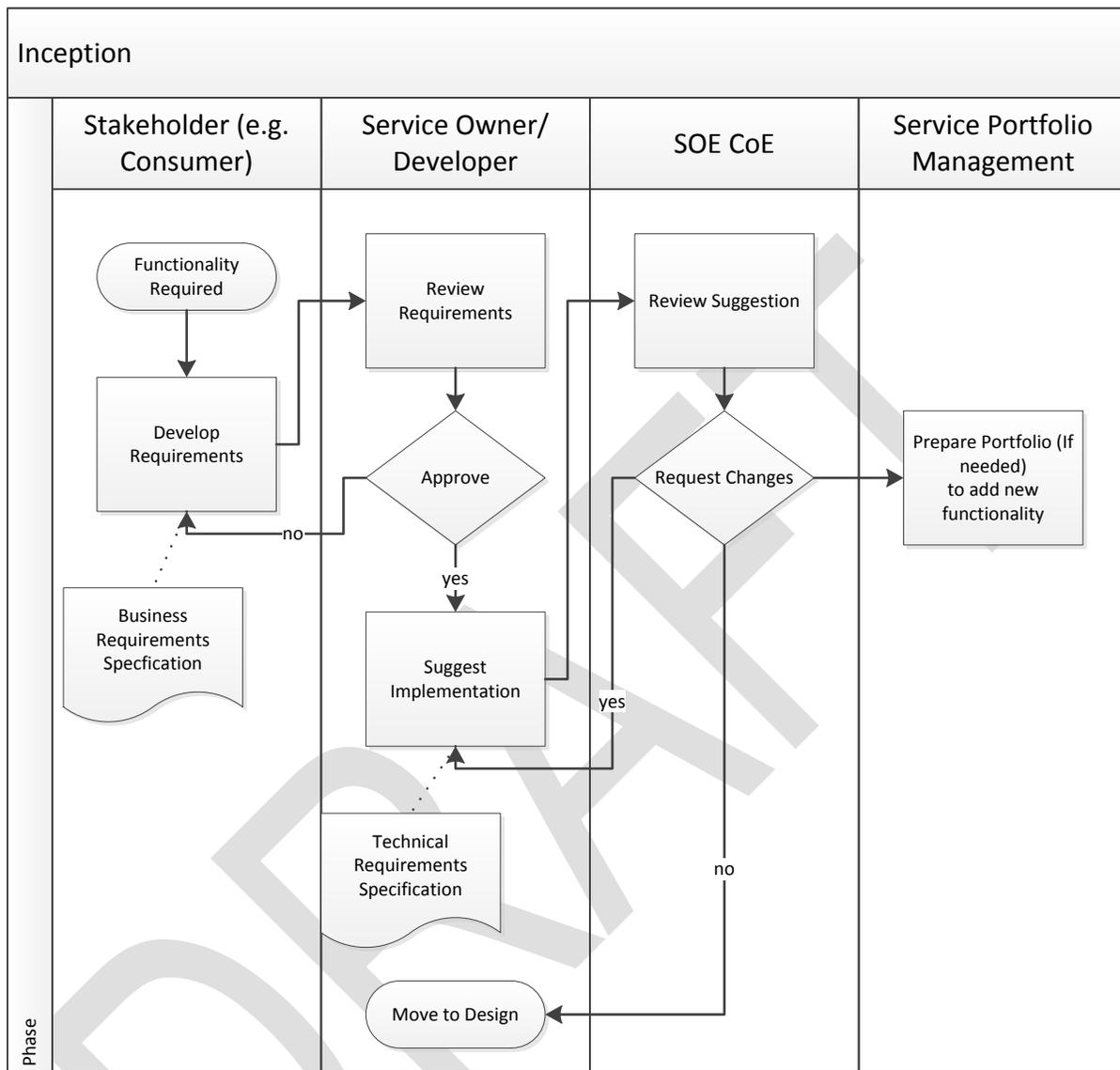


Figure 2 - Inception Process Flow

1-5.2.2 Inception Phase Policies

The Inception Phase shall include analysis of opportunities to leverage existing services in the Service Catalog. Services Harvesting (discussed in Section 6) will also be considered during the Inception Phase.

- All candidate Services shall be approved prior to entering the Design Phase.
- Proposed services that duplicate functionality of operational services shall be approved for development only if the CoE finds a compelling reason to do so. For example, consideration may include but is not limited to the following:
 - There is significant enhancement of Quality of Service (QoS) factors in the proposed service.
 - The operational service is nearing the end of its lifecycle.

- Current Consumers of the operational service can migrate to the new service with minimal effort.
- When considering a service for inclusion in the Service Catalog, the potential for reuse by additional Consumers shall be a primary consideration.

1-5.2.3 Service Registration

Service Registration is the addition of a SOA Service to the Registry. Prior to Service Registration the Service’s Lifecycle Phase, Namespace, and Taxonomy must be determined. The Namespace and Taxonomy determination should involve the Service Librarian or Information Architect. Following this, minimally an abstract WSDL should be provided and the Service Documentation Template completed. The process is described in Figure 3.

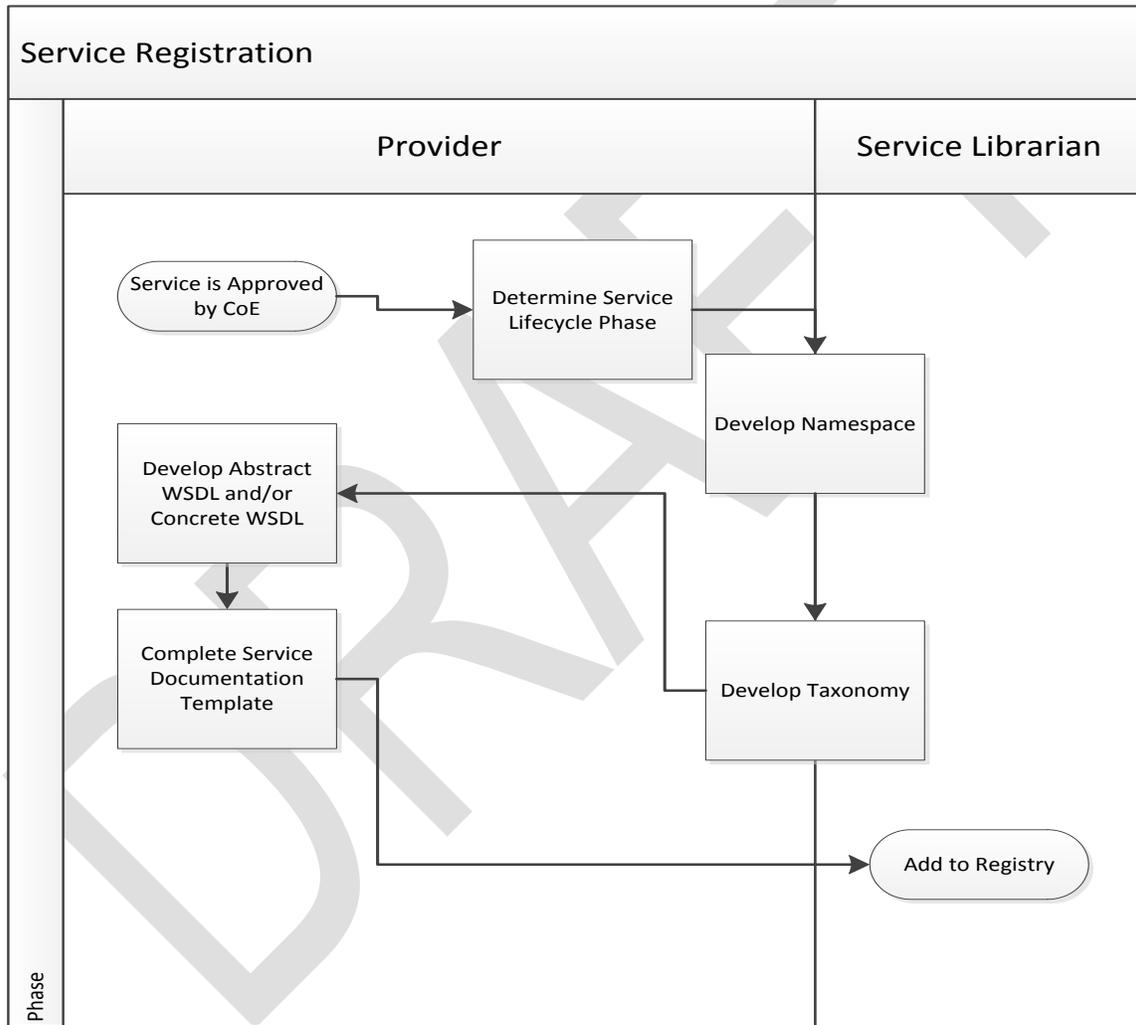


Figure 3 - Service Registration

1-5.3 Design

During the Design Phase the Service Developer creates a detailed design of the proposed functionality. The design is submitted for review to the TWG and the Operations team for review. If there are issues of concern from the TWG or the Operations team the Waiver Sub-process is invoked (see Part 18 for the template). Otherwise, the Consumer is informed of the design changes, and activities move to the Construction Phase.

1-5.3.1 Design Workflow

Figure 4 shows the steps and the actors involved in this phase.

DRAFT

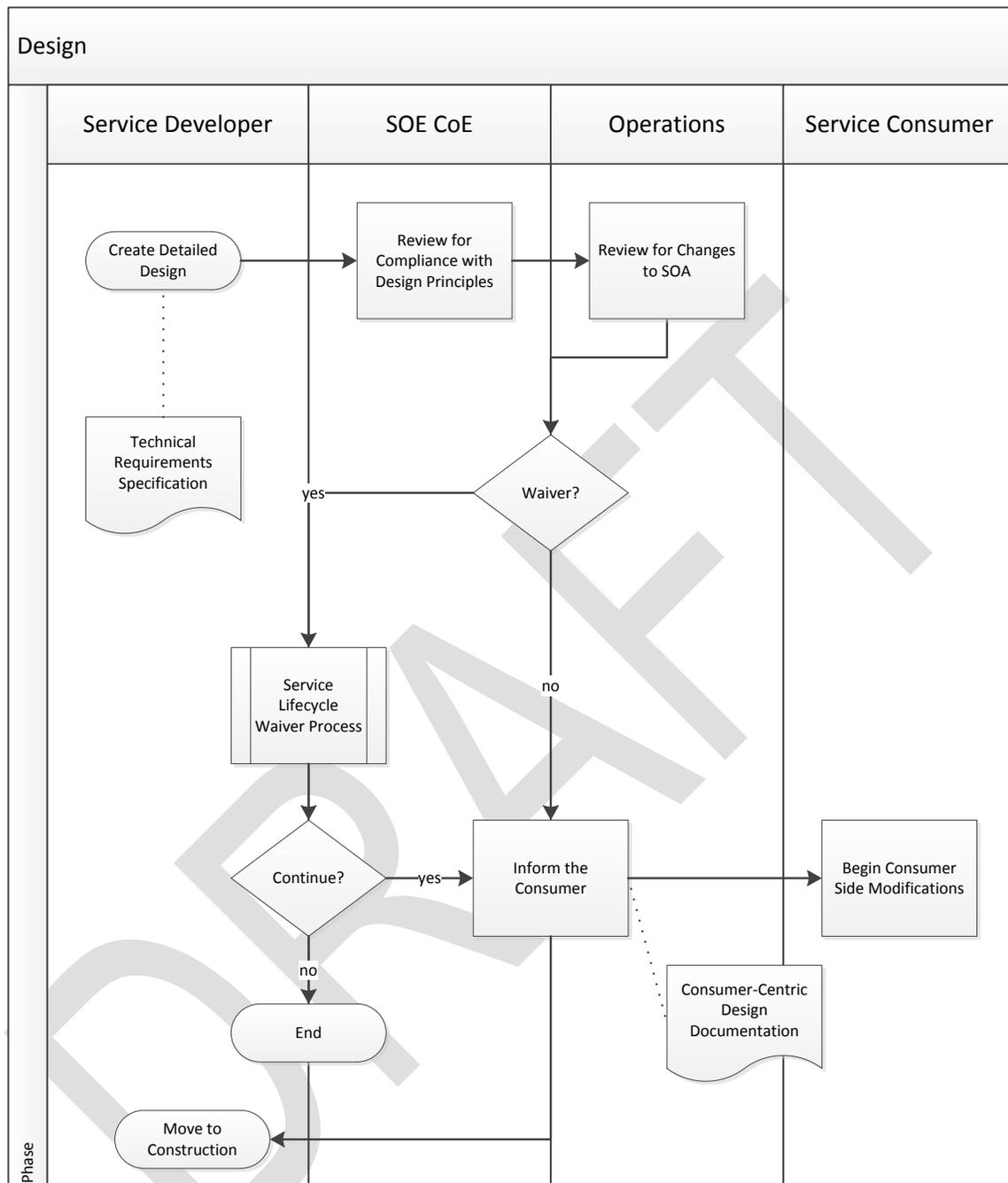


Figure 4 - Design Phase Flow

1-5.3.2 Design Phase Policies

- Services shall be designed according to a Technical Service Contract and a negotiated Service Level Agreement (SLA) which together comprise the Service Contract.
- In order to avoid collision between Service Owners, a unique XML namespaces shall be registered for each Service Owner. All XML documents for a Service Owner shall use the unique namespace registered for that Service Owner.

- Programs shall re-use an existing namespace whenever possible.
- Any conflicts between programs over namespaces shall be resolved by the CoE TWG.
- XML documents shall be specified using XML schemas and the XML Schema Definition (XSD) shall comply with the XML Schema Language. Use of Document Type Definitions (DTDs) or sample XML is not acceptable.
- All XSD schemas shall be validated with a static analysis tool that does not allow content modification.
- The use of wild-cards, unstructured, or CDATA in schemas should be avoided.
- Types shall be specified for all schema constructs.
- Services design shall be loosely-coupled to the service interface.
- The service interface is the sole entry point into service logic and resources. Services shall be accessed only via the exposed, published interfaces.
- All service interfaces shall be defined using a Technical Service Contract. Simple Object Access Protocol (SOAP) services shall include a Web Service Definition Language (WSDL) definition, one or more XML schema definitions, and Web Services (WS) Policy definitions as required.
- Services shall have an interface that expresses a well-defined functional boundary that does not overlap with other services.
- Each service shall accept a single document as an input and return a single document. The input and output messages will be validated against a schema at design-time that represents the data required to complete the business function.
- The message schema for web services shall reside in the Registry and Repository in an XML schema (e.g. associated with the WSDL), and shall not reside in the method signature (e.g. WSDL) of the service.
- Services shall be designed so that they can be tested and monitored to determine whether services become unavailable. This sort of failure checking should leverage the CA LISA tool.
- Services shall be designed so that they can be tested and monitored to determine whether a service has a detectable security fault. This sort of failure checking should be able to leverage the CA LISA tool.
- Services shall be designed so that they can be tested and monitored to determine whether factors specified in the SLA portion of the Service Contract are out of the permitted range, including but not limited to, resource utilization and the fault behaviors and performance metrics identified in the WSRR taxonomy.
- The Service Contract shall be approved by the CoE.
- The Service Contract shall contain agreed upon functional and non-functional requirements. These non-functional requirements shall include, but are not limited to:
 - Security constraints
 - Quality of Service
 - Service Level Agreement
 - Service semantics
- Static (e.g., hard coded) service addresses shall not be used. Dynamic addressing is preferred for the purpose of location transparency and failover.
- The service logic exposed by the service shall handle concurrent access without deadlock or loss of data integrity.

- Services shall be designed to minimize efferent and afferent coupling.
- Services shall be implemented in a manner that does not require consumers to use a specific language (e.g., Java only) to access the service.
- Services shall be categorized according to the taxonomies described in the Registry and Repository so that they may be appropriately registered.
- In the event of exceptions, services shall provide fault content to the consumer and the audit log, without compromising security, which shall include sufficient information for consumer recovery.

1-5.4 Construction

During this phase the Services are built. Depending on the technologies used, certain policies may apply such as standards, frameworks, patterns for integration, naming conventions, comments, and programming idioms. Documentation detailing the Services for use by the Consumer (e.g., taxonomic classification, keywords, business rules, WSDL, etc.) for inclusion into the Registry and Repository are developed. In general, the TWG may review the functionality being constructed to verify that the Provider is adhering to the agreed upon design and construction policies. The activities of the Construction phase are described in Figure 5.

1-5.4.1 Construction Workflow

Figure 5 describes the Construction Workflow.

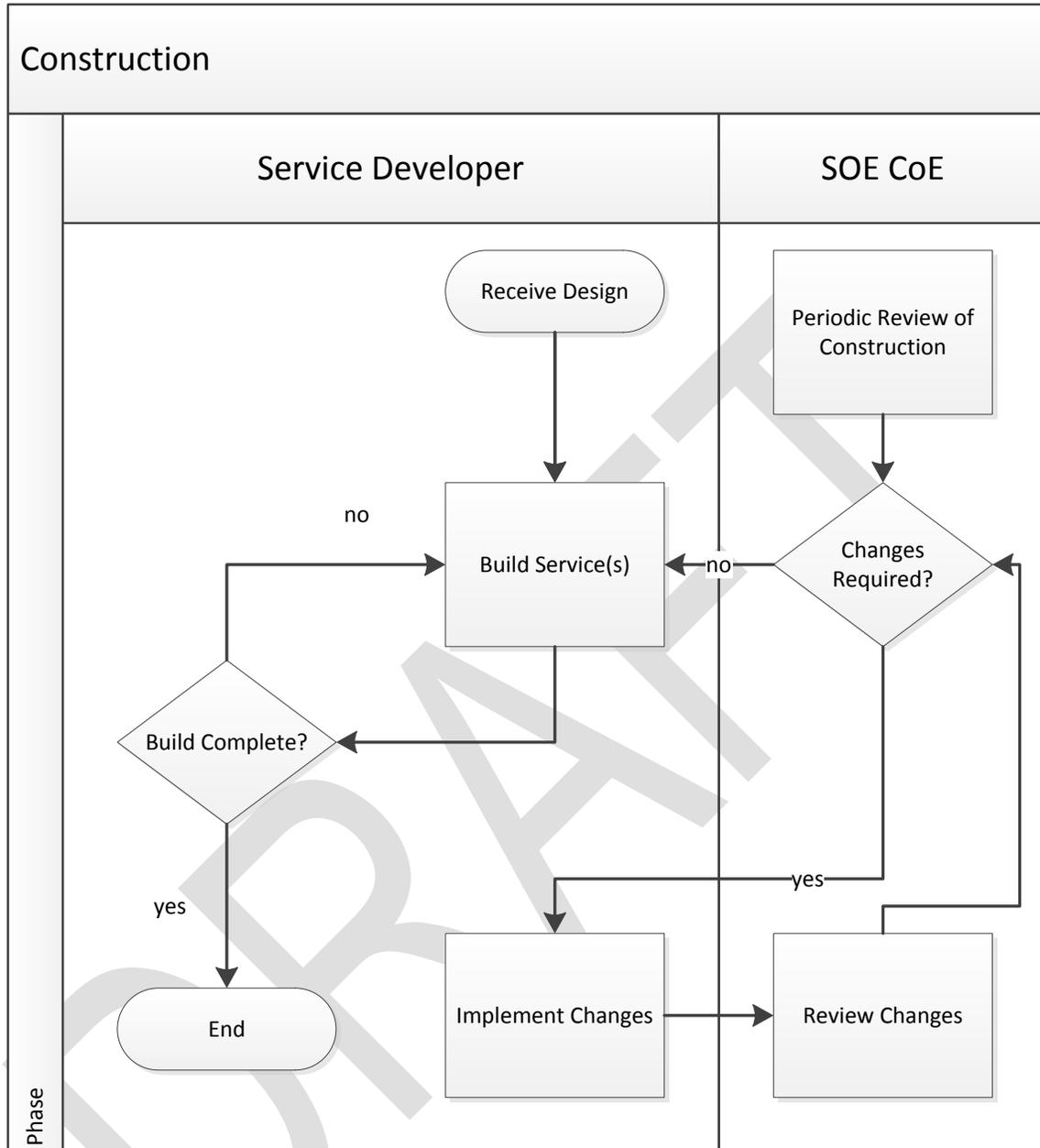


Figure 5 - Construction Workflow

1-5.4.2 Construction -time Testing

All developers building custom services that are deployed in an iEHR environment will undergo the following tests:

- Code level unit testing (e.g., JUNit, NUnit)
- Code level security testing (e.g., using Fortify)
- Code level quality testing (e.g., using FindBugs)
- Service level testing using CA LISA

1-5.5 Testing

Prior to entering the production environment, Services must be tested to verify that functional and non-functional requirements are being met.

The activities for Testing are shown in the next section.

1-5.5.1 Testing Workflow

Figure 6 shows the steps and the actors involved in this phase.

DRAFT

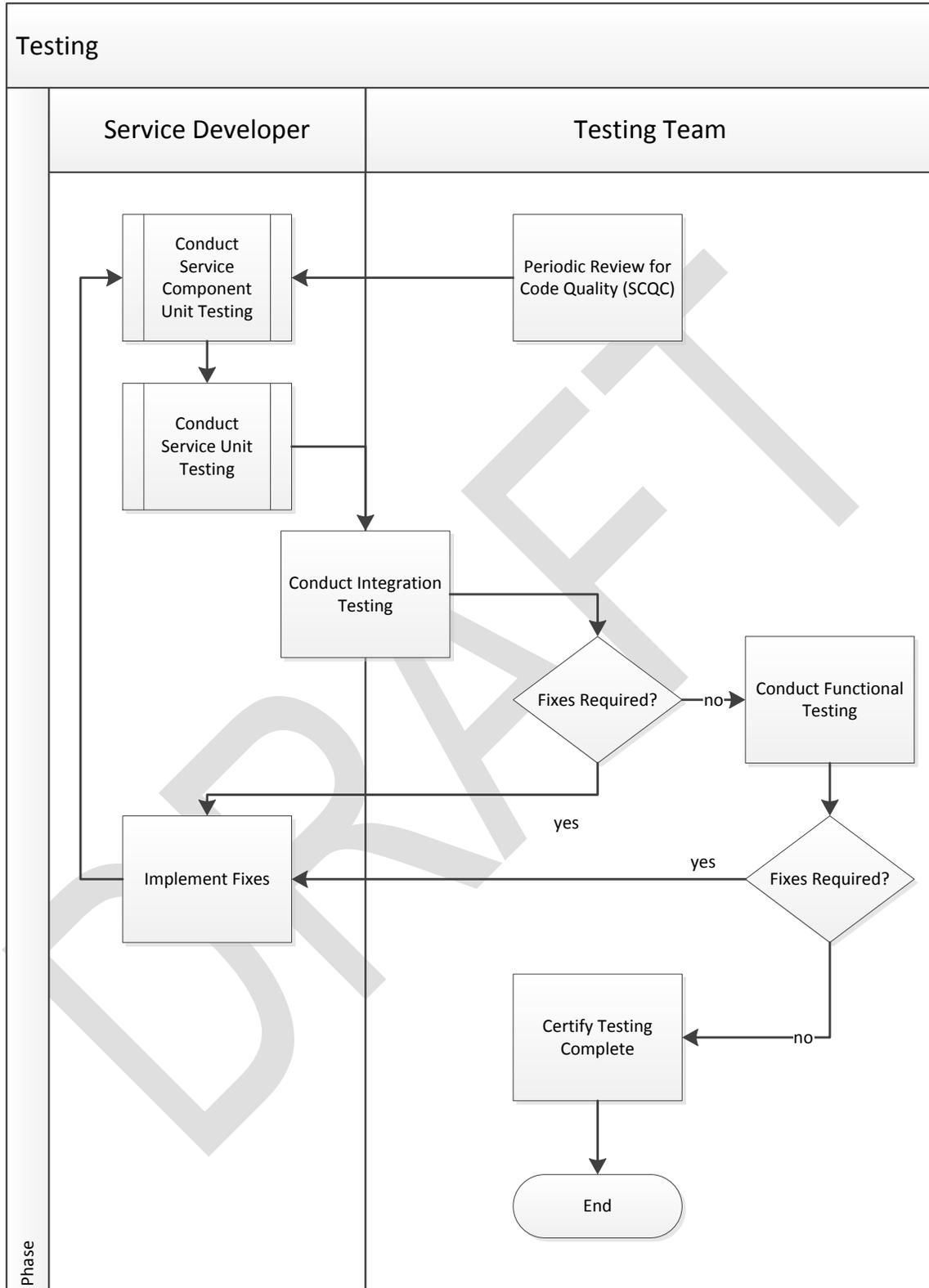


Figure 6 - Testing Workflow

1-5.5.2 Testing Policies

- **Component Unit testing** – The components comprising the Services shall be tested against the functional and non-functional requirements. Proper behavior under exception and error conditions shall be also verified. Unit testing frameworks shall be used as available.
- **Service Unit testing** – The Services comprising the functionality shall be tested in a test harness (e.g., CA LISA). The Services shall be verified to meet functional requirements, SLAs, exception and error conditions.
- **Software Code Quality Checking** – The Services and Service components shall be periodically tested against software code quality and security metrics by the Testing Team.
- **Integration testing** – The Services shall be checked to interoperate with the SOA ecosystem such that the behavior of the Services and the SOA ecosystem are within acceptable parameters.
- **Functional testing** – The Services shall be tested for compliance with functional requirements using test cases in an integrated environment with the SOA ecosystem.
- **Performance testing** – The Services shall be tested to assure that they can handle the workload in production while maintaining SLAs of the rest of the SOA.
- **Certification** – The Services must meet "Ready for Deployment" activities prior to proceeding to the Deployment phase.

1-5.6 Deployment

The Services are transferred to the Operations Team along with explicit instructions regarding configuration. Additionally, documentation detailing the Services for use by the Consumer (e.g., taxonomic classification, keywords, business rules, WSDL, etc.), developed during the Construction phase are provided to the Operations Team for updating the Registry and Repository. The activities associated with Deployment are as follows.

1-5.6.1 Deployment Workflow

Figure 7 shows the steps and the actors involved in this phase.

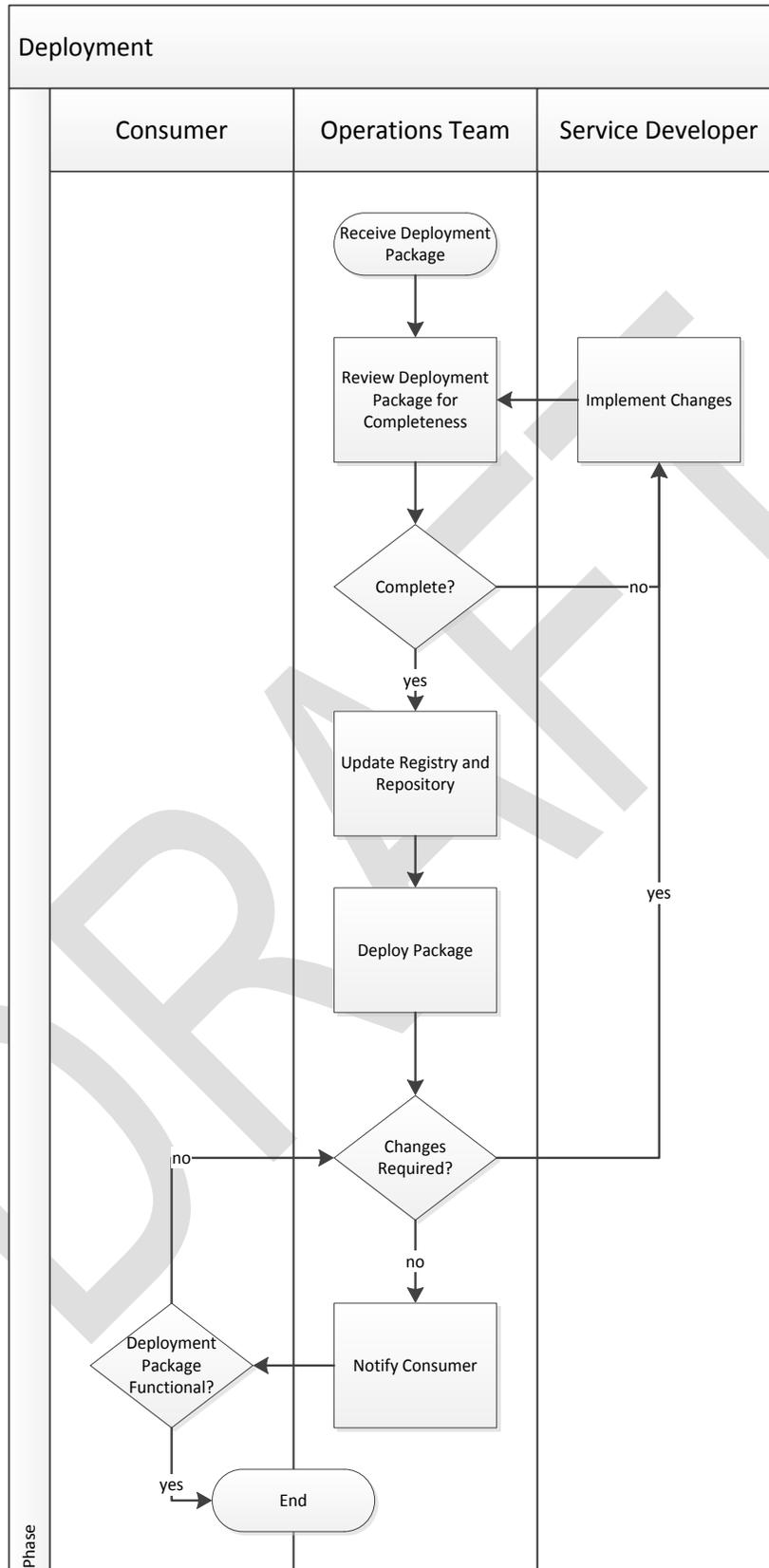


Figure 7 - Deployment Workflow

1-5.6.2 Deployment Policies

- The Service Developer shall provide a Deployment Plan.
- The Service Developer shall provide production and operational Support documentation.
- The Service Developer shall provide service configuration documentation.
- The Operations Team shall configure runtime management tools as appropriate.
- The Service Registrar shall update the Service Registry with operational data.
- The Service Developer shall ensure that versioning/change management is up to date.
- Service Consumers shall be informed of Service Deployment.

1-5.6.3 Release Management Guidance

Release Management primarily involves SLM Deployment Phase processes and activities as described in Part 2 Service Lifecycle Management (SLM). The SLM Deployment Phase Checklist provided in Part 16 incorporates key aspects of Release Management. Prior to release, all iEHR SOA services must be fully documented with any changes approved and appropriate versioning policies applied. This section contains additional Release Management guidance.

1-5.7 Operation

The Services are introduced into the production environment. The Services are now under the purview of run-time governance policies, change and configuration management (e.g., ITIL). Operational Monitoring of SLAs is described as follows.

1-5.7.1 Operation Workflow

Figure 8 shows the steps and the actors involved in this phase.

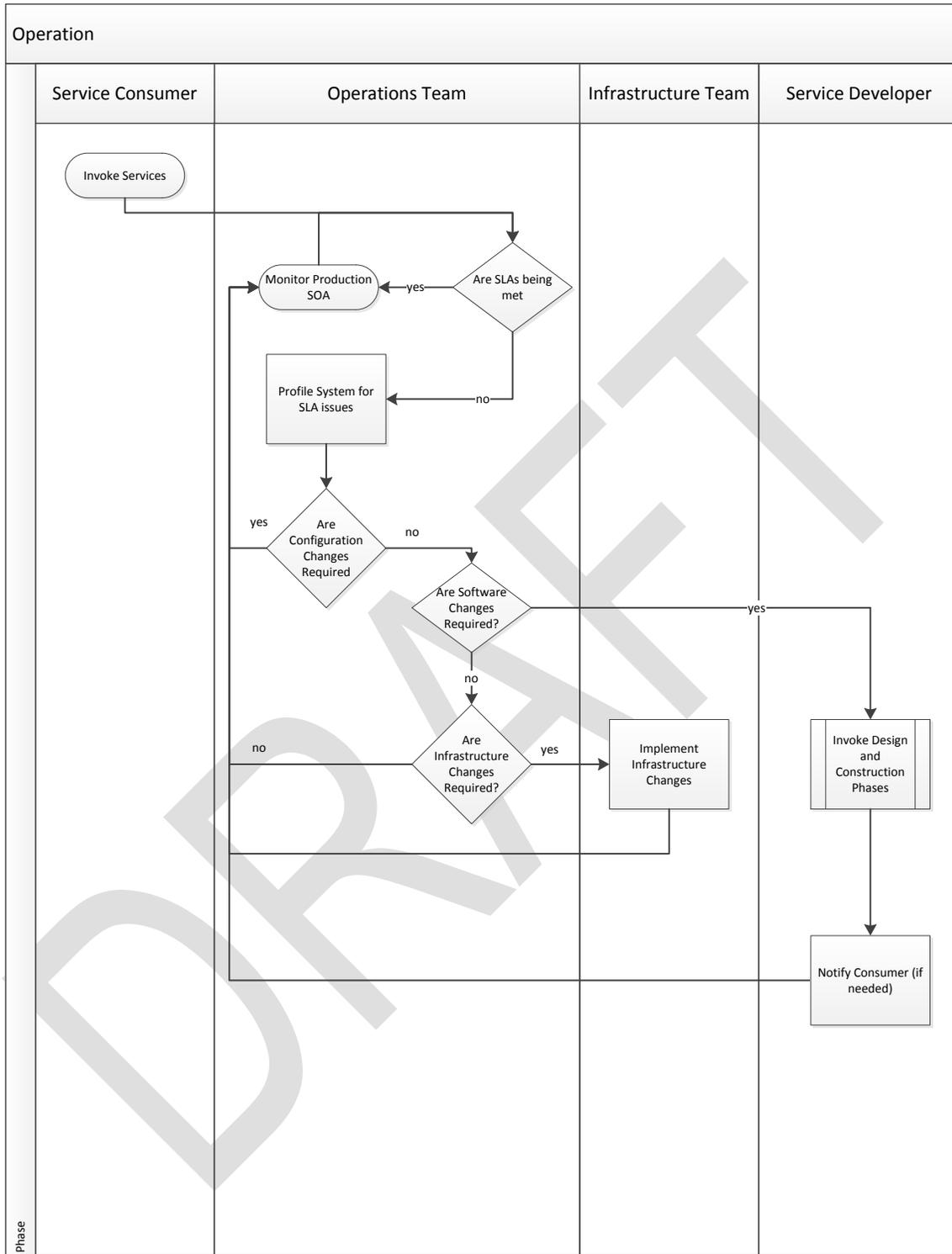


Figure 8 - Operations Workflow

1-5.7.2 Operation Policies

- SOA Suite Operations Team shall monitor services to determine whether services become unavailable.
- SOA Suite Operations Team shall monitor services to determine whether a service experiences a detectable security fault.
- SOA Suite Operations Team shall monitor services to determine if factors specified in SLAs are out of the permitted range, including but not limited to resource utilization, and the fault behaviors and performance metrics identified in the documentation.
- SOA Suite Operations Team shall be responsible for ensuring that services are monitored.
- All abnormal conditions that cannot be corrected automatically shall send an alert through the enterprise service management infrastructure, allowing the Operations Team to correct the problem in a timely manner.
- All alerts that may be sent to the enterprise service management infrastructure shall have documented escalation procedures and, if possible, the process to address the abnormal condition.
- The Provider-to-Consumer data-stream messaging service shall provide for a Provider/Consumer failover mechanism to recover messages and to reconnect to another available message broker as deemed necessary.
- Monitoring and SLA Enforcement will use a combination of administrative intervention in conjunction with SOA Suite's CA Application Program Monitor (APM), WebSphere Registry and Repository (WSRR), and CA Wiley Introscope. Administrators will use these to manage the health of the SOA. Developers on the SOA Suite are expected to provide SLAs through human readable documentation as well as in terms of WS-I standards such as WS-Policy when applicable. Developers will support these SLAs through the SOA Suite functionality or from within the Service as required.

1-5.8 Deprecation

Prior to Retirement, the Services are deprecated. All associated Actors are provided a deprecation timeline and information regarding the functionality that will replace the deprecated Services. The activities associated with Deprecation are shown in Figure 9.

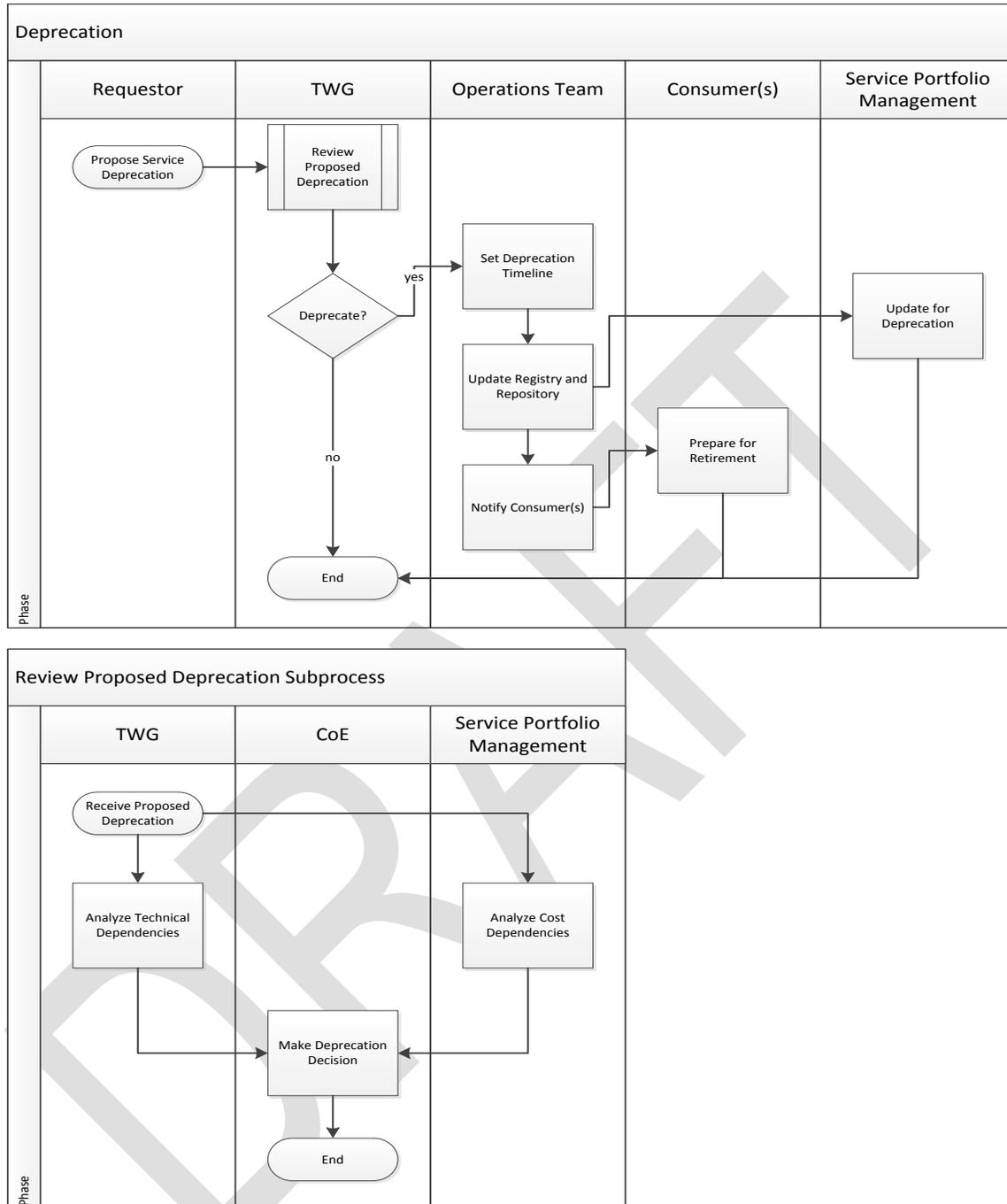


Figure 9 - Deprecation Workflow

1-5.9 Retirement

As updates to the SOA ecosystem occur, Service may be removed from the SOA. Prior to this event the Services have been depreciated as part of the Deprecation phase of the lifecycle. Services must be removed from the SOA without adversely affecting the production environment. The activities associated with Retirement are shown in Figure 10.

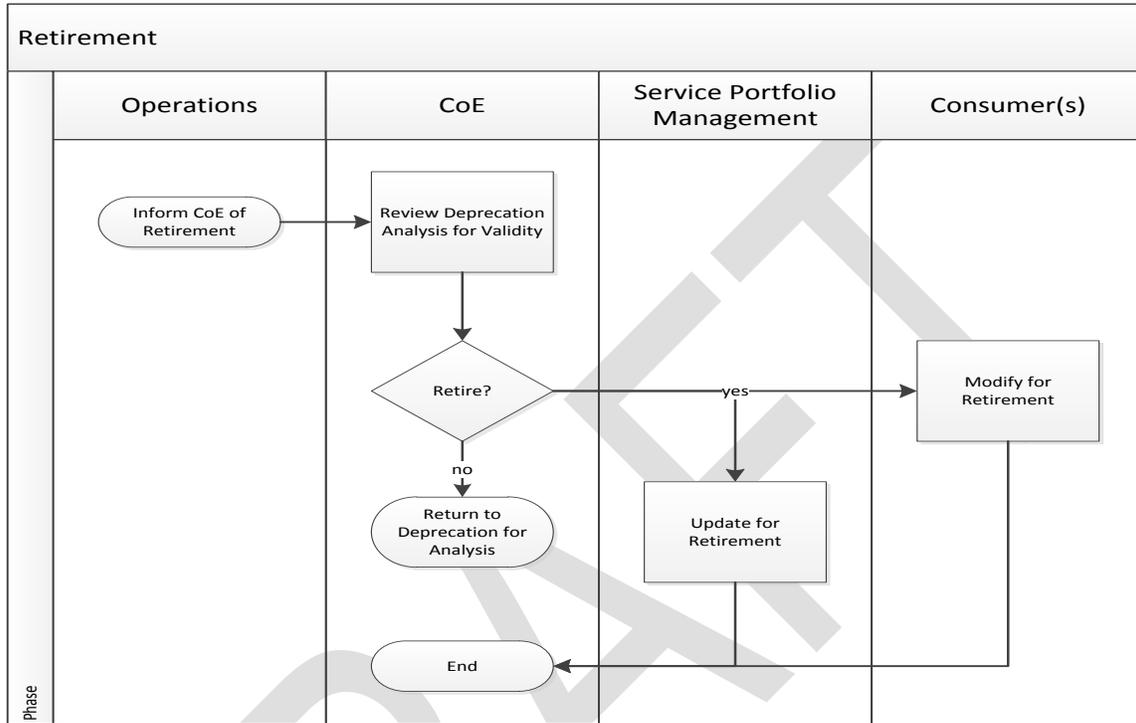


Figure 10 - Retirement Workflow

1-5.10 Service Life Cycle Management (SLM) Waivers

The Service Owner is the organizational entity responsible for creating the Service. The Service Developer is responsible for adhering to the SLM process and documenting any exceptions by following the SLM waiver process.

Figure 11 describes the waiver process activities that are performed by various Actors.

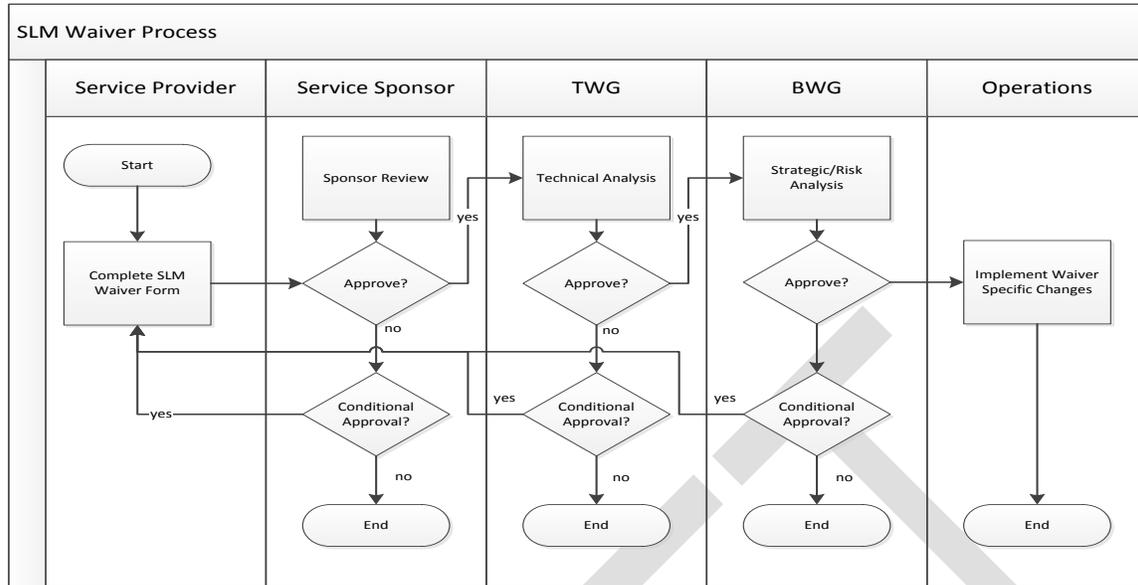


Figure 11 - Service Waiver Process

1-5.11 Service Review Checkpoints

Projects must be reviewed for compliance with architecture, design, and development guidelines. In order to conduct project reviews, checkpoints along the project lifecycle must be established. These checkpoints may correspond to:

- SRR – Specification Readiness Review
- PDR – Preliminary Design Review
- CDR – Critical Design Review
- Checkpoints during the development lifecycle (e.g., at the beginning or end of Agile Sprints)
- TRR – Test Readiness Review

Figure 12 shows an example of SOA Audit checkpoints.

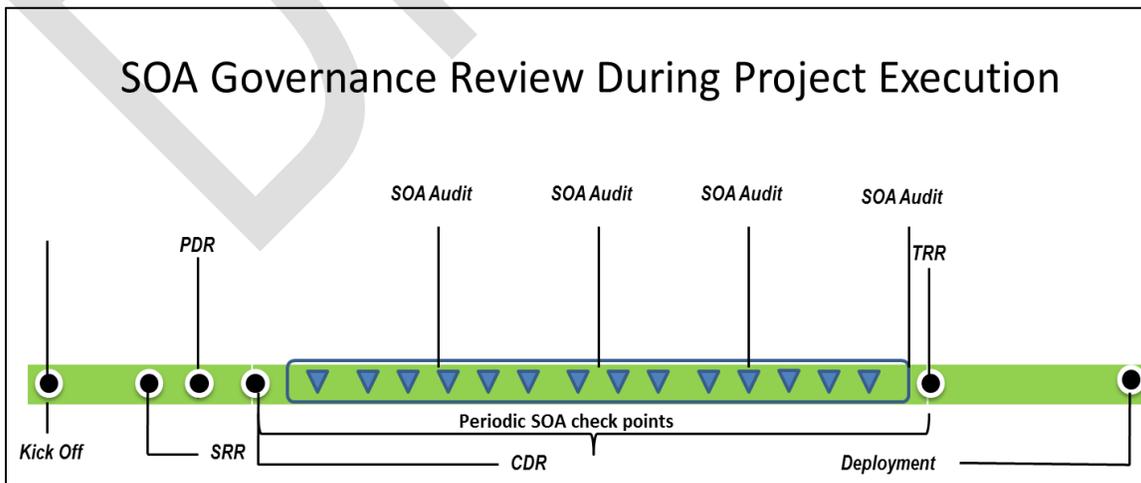


Figure 12 - SOA Governance Checkpoints

1-5.12 References

1. SOA Governance Framework, The Open Group, 2009.
2. IPO iEHR SOE Strategy Volume 1 SOE Roadmap and CoE ConOps DRAFT10312013.docx
3. IPO iEHR SOE Strategy Volume 3 SOI Governance DRAFT.docx

DRAFT

PART 2 COMMON SERVICE HARVESTING

During the Service lifecycle, the Actors associated with the lifecycle may decide that there are Services that warrant consideration for inclusion in the Common Services Repository. Common Services are Services that contain functionality leveraged by multiple business domains within the SOA. Services corresponding to the HL7 RLUS Specification are examples of Common Services.

Services are identified for inclusion in the Common Services Repository based on the number of domains requesting the similar functionality. If a custom Service has been built, it is reviewed by the TWG. If changes are required, the Service is delegated to the Service Developer for the necessary changes. The process for Common Service Harvesting is as shown in Figure 13.

DRAFT

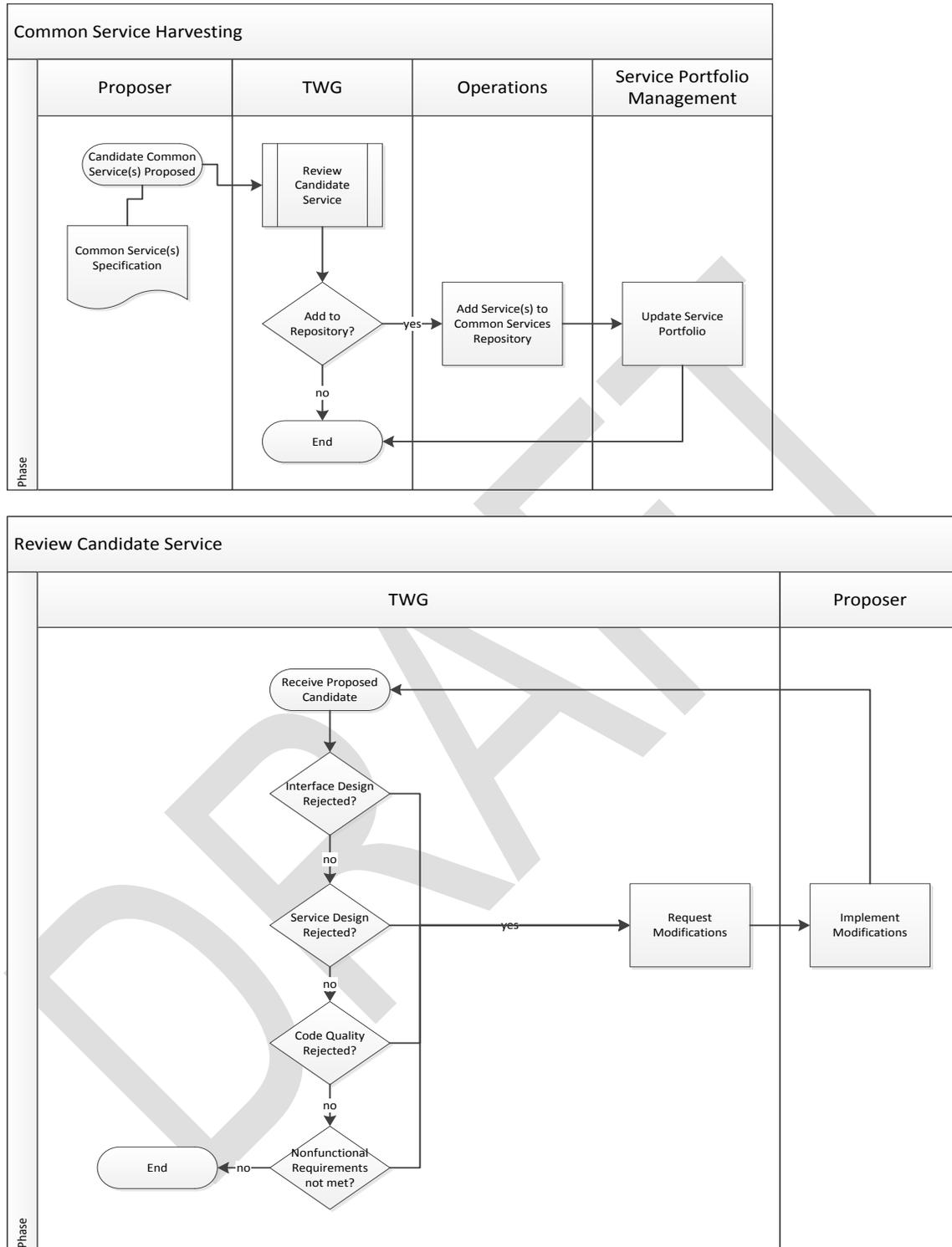


Figure 13 - Common Services Harvesting Workflow

PART 3 SERVICE DOCUMENTATION

Service documentation is the specification of all metadata around a Service. Since all Services will be included in the Service Registry and Repository (SRR), all Service documentation should minimally leverage the Service Documentation template specified in Part 18. The CoE may require additional documentation, as necessary.

DRAFT

PART 4 ARCHITECTURE GUIDANCE

This section provides an overview of the architectural that should be used in working with the SOA. A description of a Reference Architecture (RA) is provided followed by a brief discussion of the Platform Independent Model (PIM) and the Platform Specific Model (PSM). A Transition Architecture along with aspects of Data Federation and Identity Management are also discussed.

4-1. Architecture Approach

The Architecture approach used at the IPO is Krutchten’s 4+1 methodology as described in Figure 14 below [1].

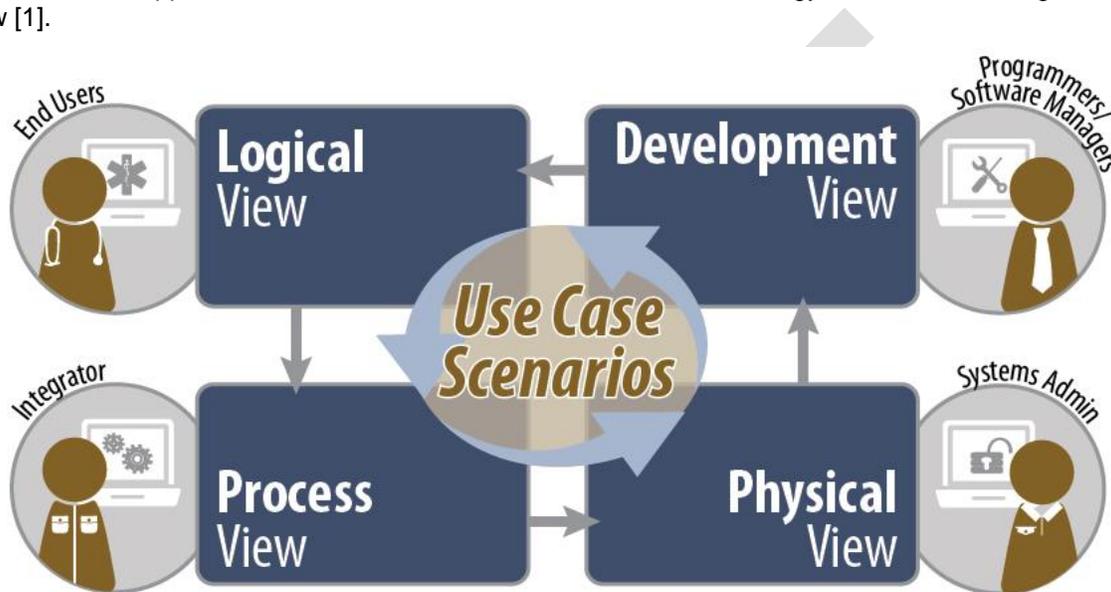


Figure 14 - "4+1" Architectural View

The justification for using the 4+1 methodology in illustrating the design is to provide a holistic and contextualized set of views such that each stakeholder will have a view that is specific to their areas of concern. Each separate set of views is given context through use case scenarios, and linked together by those same scenarios to form the overall architectural blueprint.

The 4+1 views that should be used to design and document solutions are described as follows in Table 1.

Table 1 – Detailed "4+1" Views

View Type	Audience	Benefits	Related Artifacts
Logical View	Designers	This view is most commonly used to communicate design aspects to end user stakeholders. As an example, this view set would provide Lab staffers a look into the data and the flow of that data with respect to their most important business functions.	Class diagram, Communication diagram, Sequence diagram
Process View	Integrators	This view can provide systems integrators with insight into how multiple processes that are sometimes concurrent operate, and how they may support, interact, or conflict with one another, such as in Pharmacy and Lab operations (user or system based).	Activity diagram
Development View	Programmers	This set of views provides developers with an understanding of the software components that are needed to implement the system at its various layers. One example of such a component would	Component diagram, Package diagram

View Type	Audience	Benefits	Related Artifacts
		be a common user interface for facility staff members at the application layer that allows users to interact with the data based on user roles and contexts.	
Physical View	Deployment managers	Regarding the topology of a Military Treatment Facility / Veterans Affairs Medical Center (MTF/VAMC), this set of views is crucial in understanding how the architectural components will interact, particularly in determining the interaction patterns between local data storage and distributed data storage, as one example.	Deployment diagram
Use Case Scenarios View	All Stakeholders	This view set is the tie-in component for the rest of the architectural views. It identifies architectural elements and provides validation and verification for the realized blueprint. The IPO and iEHR efforts can use the use case view to determine common grounds for identifying business processes and data object interactions from a user perspective that will drive out further architectural views as described towards the realizable blueprint.	Use case document, Use case model

4-1.1 Architecture Documentation

Architecture documentation will prefer Unified Modeling Language (UML 2.0), Business Process Model Notation (BPMN 2.0). Process flows, layer diagrams, and other representations are permitted as needed.

This section discusses the SOA Architecture and provides rudimentary guidance associated with the usage of the SOA Suite.

4-1.2 Reference Architecture

A comprehensive Reference Architecture is provided in the Enterprise Technical Architecture [2]. This architecture provides a detailed discussion of architecture using an example implementation leveraging Open Source technology. Since this document is focused on Governance rather than architecture, a more rudimentary discussion of the architecture is provided here. The discussion leverages the Open Group Conceptual Model for SOA and maps it to representations of the iEHR SOA as well as the SOA Suite.

4-1.2.1 Conceptual RA

From a Reference Architecture (RA) perspective, any number of industry RAs can be leveraged. Generally, these divide the architecture in terms of concepts that separate concerns addressed by different technical features of the SOA. Figure 15 shows a conceptual SOA using a modified Open Group high level model of SOA [3].

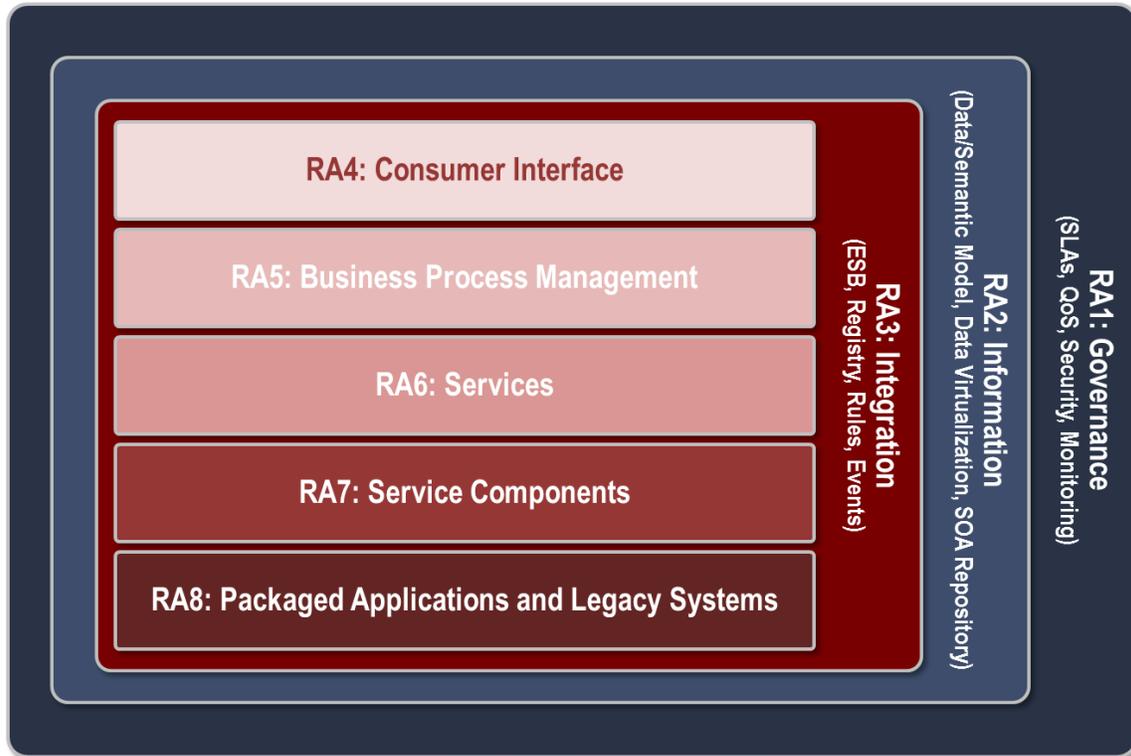


Figure 15 – Open Group Reference Architecture

4-1.2.2 iEHR Future State Architecture

Given the general framework for a SOA RA defined by The Open Group model, this section maps the RA described above in terms of a PIM. Since technological components can provide either a subset or a superset of the concepts described in the RA, the mapping is not isomorphic. Multiple technologies may contribute to a single concept such as Information (RA2) or a single technology may contribute to more than one concept such as SOA Registry and Repository (RA2 and RA3). Therefore, the technologies that correspond to a particular capability (RA1-RA8) are highlighted in yellow in the diagrams throughout this section. Additionally, items such as Clinical Context Management (CCM), which are not covered in RA1-RA8 but are necessary for the iEHR implementation, are shown. The PIM should be considered notional and will evolve as the iEHR architecture matures.

Figure 16 shows a future state for the iEHR SOA.

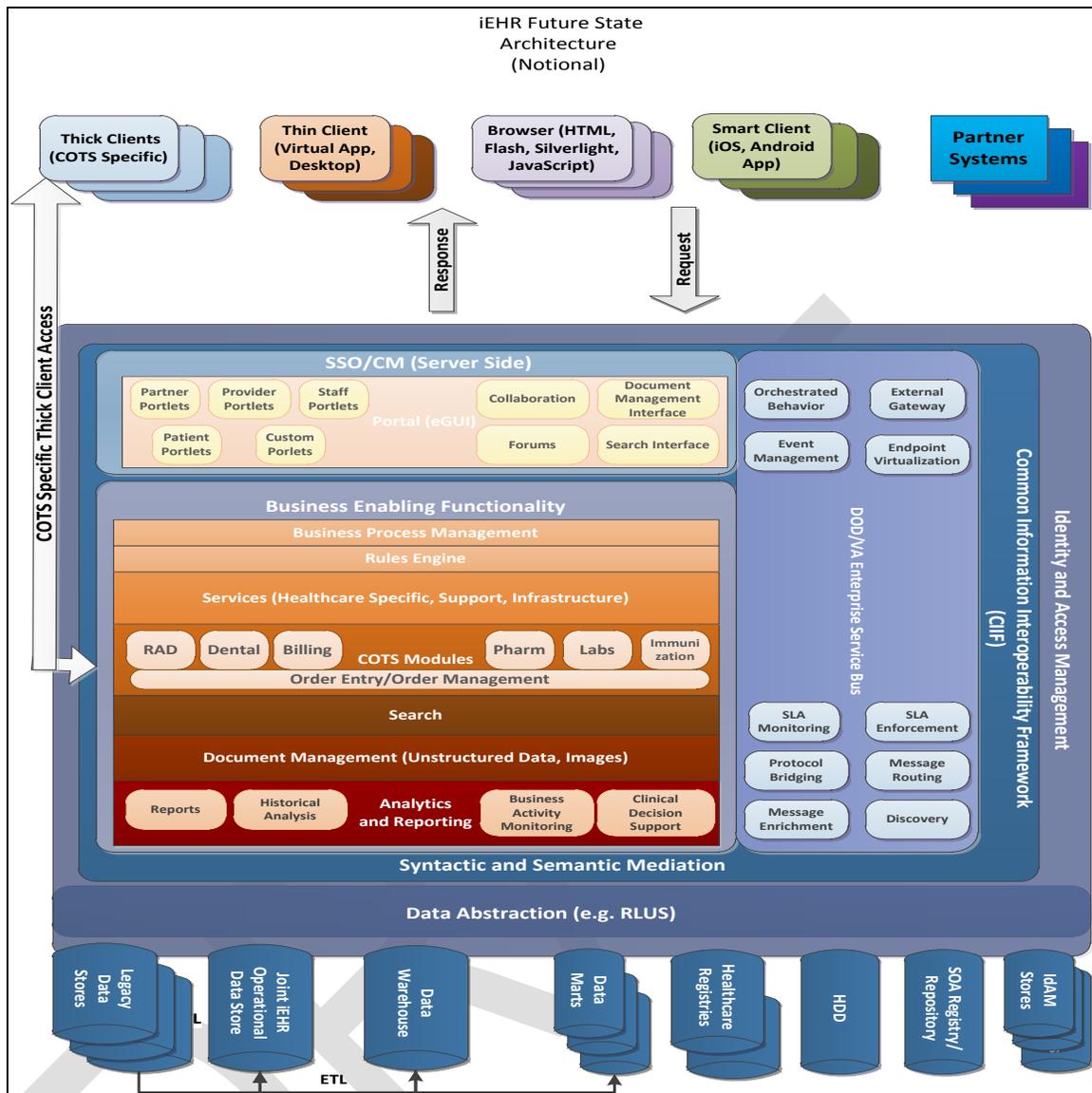


Figure 16 – iEHR Future State PIM

4-1.2.3 Transition Architecture

The implementation of the iEHR has been split into multiple increments such that legacy systems are replaced in phases. Consequently, there must be an architecture where components of the future state and the current state co-exist. This transition architecture is depicted in Figure 17.

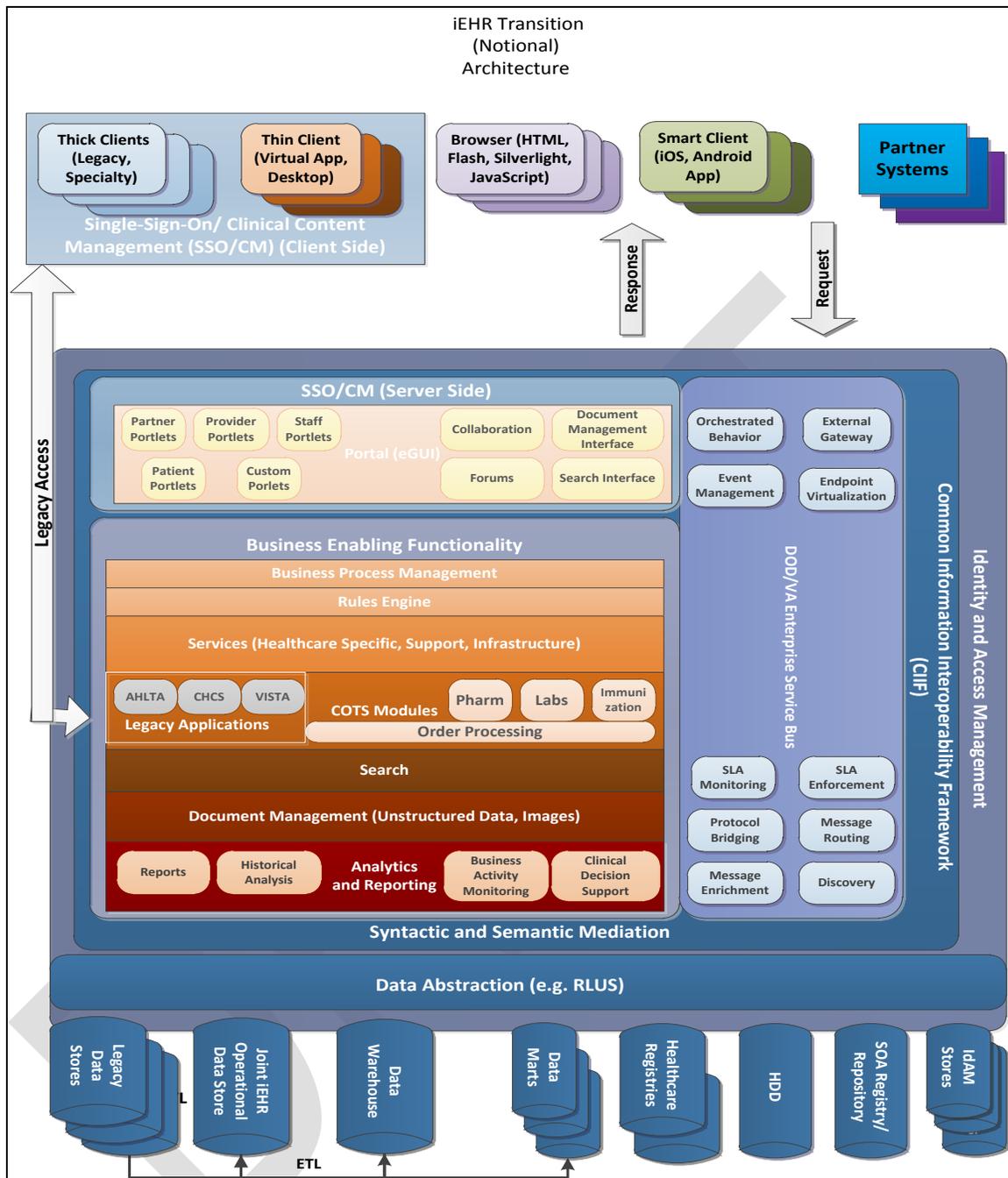


Figure 17 – iEHR Transition Architecture Indicating the Presence of Legacy Thick Clients and Components

The Transition Architecture accounts for the presence of legacy systems (See Legacy Applications in Figure 17) and new COTS systems in tandem. The Transition Architecture supports Client Side Single-Sign On/Context Management (SSO/CM) as necessary on the Desktop and Server Side SSO/CM. Additionally, access to Legacy Applications from Thick Clients is shown. In future, Specialty Clients may be necessary to access COTS modules. In this case, the Legacy Access Pathway (See Legacy Access in Figure 17) would serve as a Specialty Client Pathway. It is expected that as COTS applications become more SOA aware, the Special Client Pathway would not be necessary in the long term. Further Client Side SSO/CM would be replaced by Server Side SSO/CM.

- **Transition Architecture Details** – No one diagram can depict all aspects of the Transition Architecture. Therefore, this section describes some of the components of the architecture not shown explicitly in the previous representations.
- **Identity and Access Management Stores** – The Identity and Access Management (IdAM) Stores encapsulate the particulars of DoD and VA Identity and Access Management. This includes Credentialing Services, Correlation Services, Active Directory (AD), Defense Enrollment Eligibility Reporting System (DEERS), or other IdAM Repository. Figure 18 shows an exploded view of the IdAM Stores.

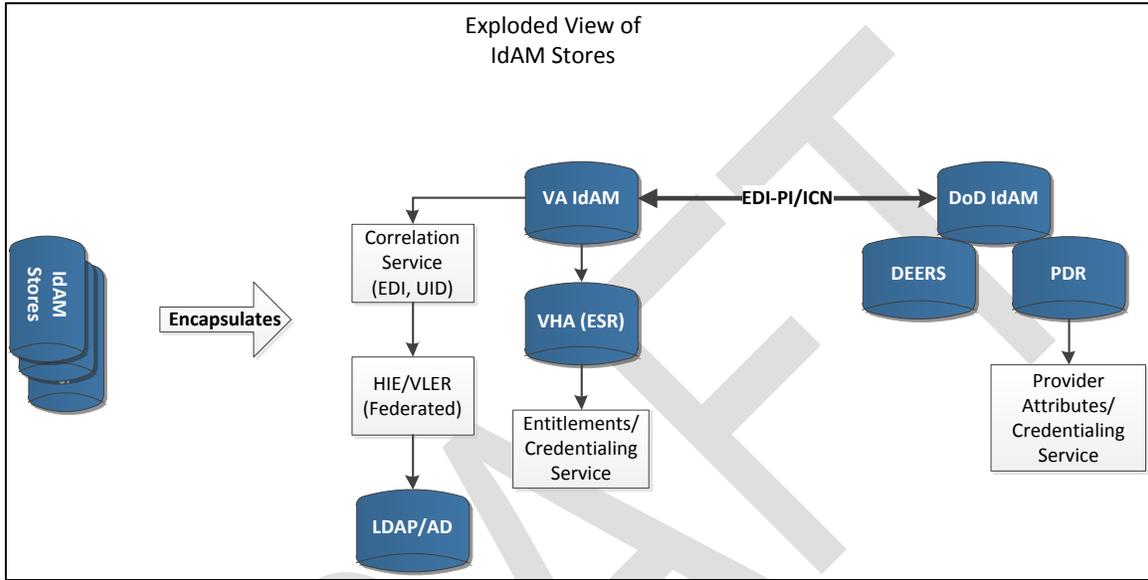


Figure 18 – IdAM Stores in the PIM Encapsulates a Network of Identity Management Systems

The IdAM Stores are therefore shorthand for a complex network of Identity and Access Controls involving both data stores and services. As indicated in Figure 18, DEERS, Virtual Lifetime Electronic Record (VLER), Veterans Health Administration (VHA) as well as Lightweight Directory Access Protocol (LDAP) and AD are involved.

- **Partner Systems** – Partner Systems implies any external system that requires access to the iEHR such as NwHIN or NwHIN Direct Consumers. Figure 19 shows an exploded view of Partner Systems.

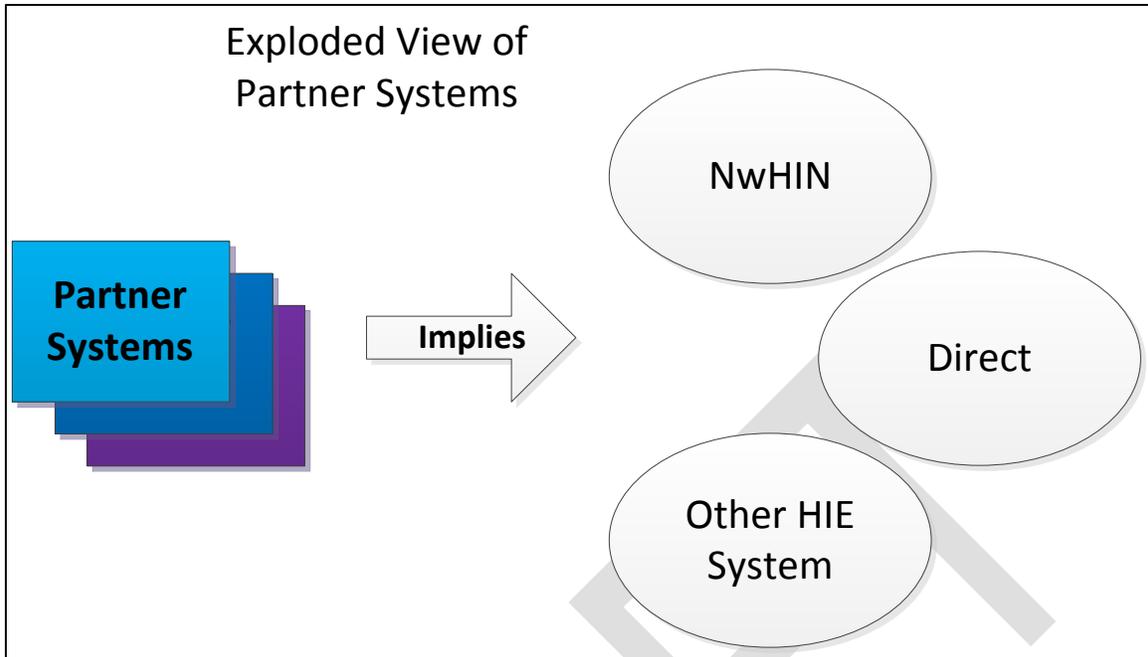


Figure 19 – Partner Systems Encapsulates All External Systems Accessing the iEHR

- Data Virtualization** – Data Virtualization implies any Data Services or tools that act as facades for data stores and data access in general. This can include Retrieve Locate Update Services (RLUS) or other services such as Auditing, Service Repository Access, or IdAM Store access. Figure 20 shows an exploded view of the Data Abstraction Layer.

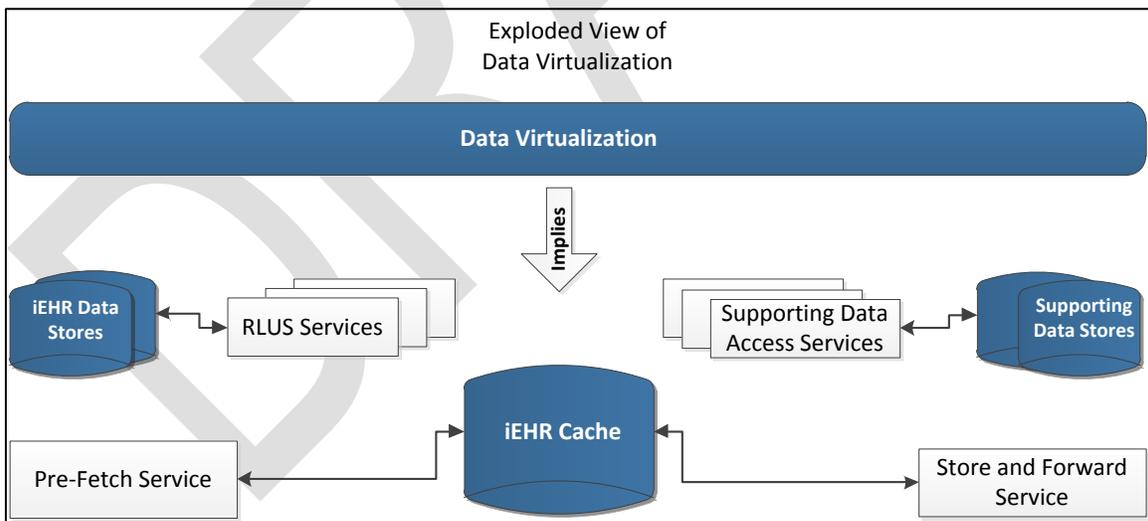


Figure 20 – The Data Abstraction Layer Encapsulates RLUS, Pre-Fetch, and Store/Forward Services

As discussed for IdAM stores, the Data Virtualization Layer is short-hand for multiple pieces of functionality that enable RLUS and other data services such as logging and auditing, as well as controls for intermittent connectivity and data federation such as Pre-Fetch and Store/Forward Capabilities.

- Transition Architecture Summary** – In summary, a detailed representation of the Transition Architecture is depicted in Figure 21.

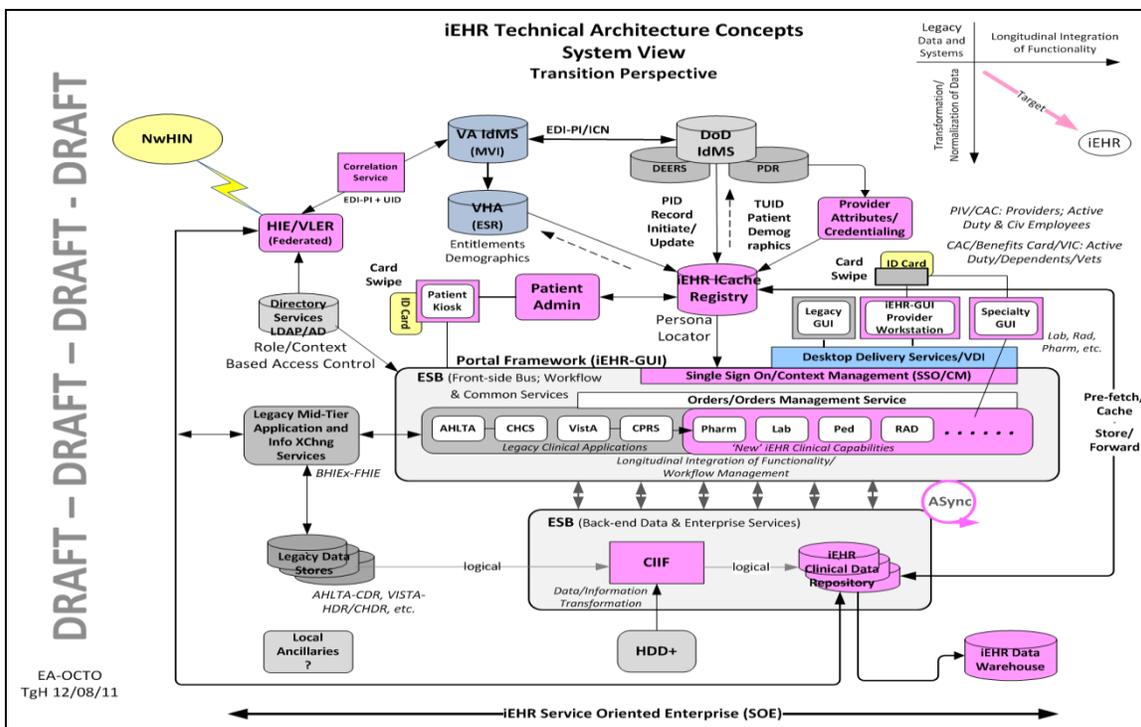


Figure 21 – Detailed View of the iEHR Transition Architecture

4-1.2.4 Platform Specific Model

The PSM describes the specific technologies used to implement the architectures described in Section 4-1.2.2, Figure 16.

In order to implement the solutions described in the PIM of Figure 16, the MHS and VA have jointly acquired a SOA Suite. The SOA Suite uses a combination of IBM, Layer 7, CA, and Open Source technologies to establish the underlying structure to support the functionality described in the RA and the PIM. The core components of this underlying structure are described in Table 2.

Table 2 – Technologies Comprising the SOA Suite

Component	Description	Recommended Usage
IBM WebSphere Message Broker	The primary Enterprise Service Bus (ESB) used at data centers of Joint DoD/VA Enterprise.	This component should be used to do message routing, simple data mediation, protocol switching and orchestration.
IBM WebSphere Transformation Extenders	Used in conjunction with the ESB to address mediation between data formats. Transformation Extenders include among others, SAP, Siebel, and Health Insurance Portability and Accountability Act (HIPAA) Electronic Data Interchange (EDI).	This component should be used in conjunction with Message Broker to conduct complex data mediation, beyond the scope of data mediation done within Message Broker.
IBM WebSphere Process Server/Business Process Manager	The primary engine for business process and workflow management.	This component should be used for scenarios where Message Broker orchestration is insufficient (e.g. long running workflows that may involve human interaction).
IBM WebSphere iLog Rules Engine	The Rules Engine to decouple business and clinical decision support rules at all levels of	This component will be used only for rules based routing of

Component	Description	Recommended Usage
	the architecture.	messages. In the long term, it MAY be used for business rules, but unless otherwise indicated by the SOE CoE, its usage will be limited to rules based routing.
IBM WebSphere MQ/MQ File Transfer Edition	Message Queuing Service used to enable assured message delivery.	This component should be used for asynchronous assured delivery of messages. The component supports transactionality, durability, and security. It should be used where a more lightweight solution such as HTTP/S is insufficient.
IBM WebSphere Registry and Repository	The SOA Services Registry for Common Services. The registry and repository will contain templates, rules and additional information associated with SOA Services.	This component should be used for design time discovery of services. At run time it may be used for discovery and may also be used to connect with monitoring tools. It should capture not only Services in various states of the Service Lifecycle but also all necessary documentation associated with the service. This includes SLAs and any policies associated with the service.
IBM WebSphere Message Broker Connectivity Pack for Healthcare	Healthcare specific framework to accelerate integration between COTS products and enable construction of Common Services.	This component should be used to convert between different versions of HL7 as part of mediation by Message Broker
Layer 7 XML Gateway	Gateway to address Security and XML processing at network boundaries.	This component should be used for authentication and authorization. It will evaluate Security Assertion Markup Language (SAML) Assertions and execute Extensible Access Control Markup Language (XACML) policies.
Mirth Open Source HL7 Broker	An Open Source lightweight Message Interchange Broker for Healthcare. The Mirth Broker will be used at locations that require a small footprint SOA implementation.	This component should be used at the Local Site level to create or convert HL7 messages and communicate with local applications such as Composite Health Care System (CHCS).
CA Application Performance Management (APM) for SOA platforms and WebSphere	An enterprise application performance management solution that enables monitoring of SOA and web applications. Used for problem detection and collaborative resolution.	This component will be used by administrators to monitor Application health and identify problems.
CA Capacity Management and Performance Suite	An infrastructure analysis and prediction tool used to optimize operations and supports ongoing planning for new enterprise application deployments and changes in a virtualized environment.	This component will be used by administrators for the analysis of infrastructure on performance management.
CA Wily Introscope	A performance monitoring and inspection	This component will be used by

Component	Description	Recommended Usage
	tool used to monitor the current state and health of the SOA solution during runtime.	administrators to profile the SOA and check for performance related issues.
iTKO LISA	A SOA testing tool that allows endpoint emulation of services and run-time detection of service behavior.	This component will be used for testing both by developers and testing staff. It should be used to test a Service or collection of Services.
WebSphere Application Server	A general purpose application hosting context.	This component will support other WebSphere Components and is generally not intended for use by developers. However, upon request, instances of the Application Server may be made available for purposes of project-specific application execution.
IBM DB2 Database	A general purpose relational database.	This component will not be made available to developers. It is intended to support other WebSphere Components.

These technologies are mapped to the Capabilities of the RA and the Components of the PIM as follows:

- Governance** – In terms of the acquired SOA Suite (the PSM), Governance is provided by WebSphere Message Broker, WebSphere Registry and Repository, and CA Capacity Management, CA Wily Introscope, and Layer 7 XML Gateway. The mapping from the SOA Suite to the PIM to the RA is depicted in Figure 22.

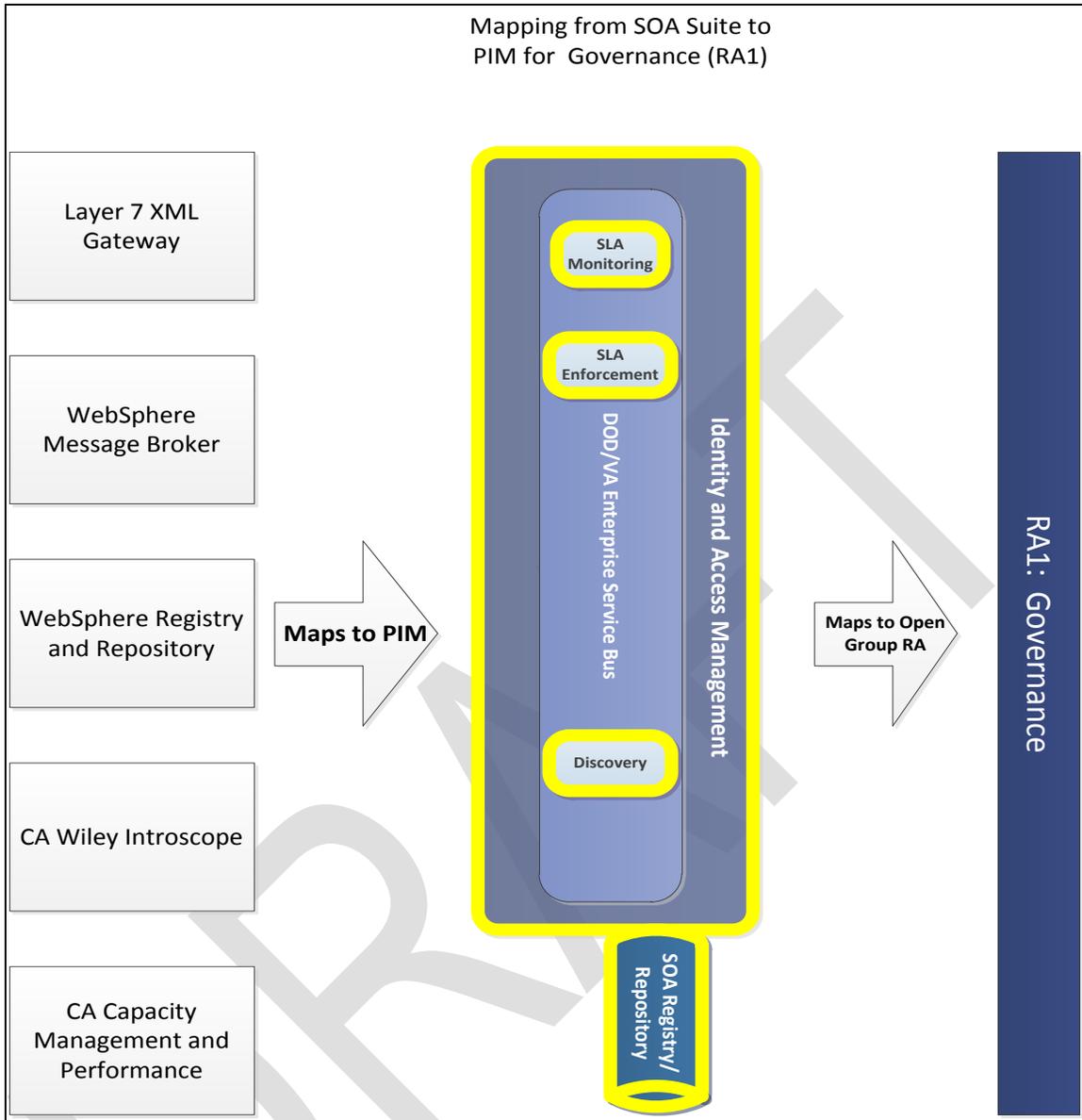


Figure 22 – The SOA Suite Components Associated with Runtime Governance Mapped to the PIM and the RA

- Information** – Not all of the Information Capabilities of the RA and the PIM are captured by the SOA Suite. Only some of the mediation portions are provided. Figure 23 depicts the technologies in the SOA Suite that support the Information Capability.

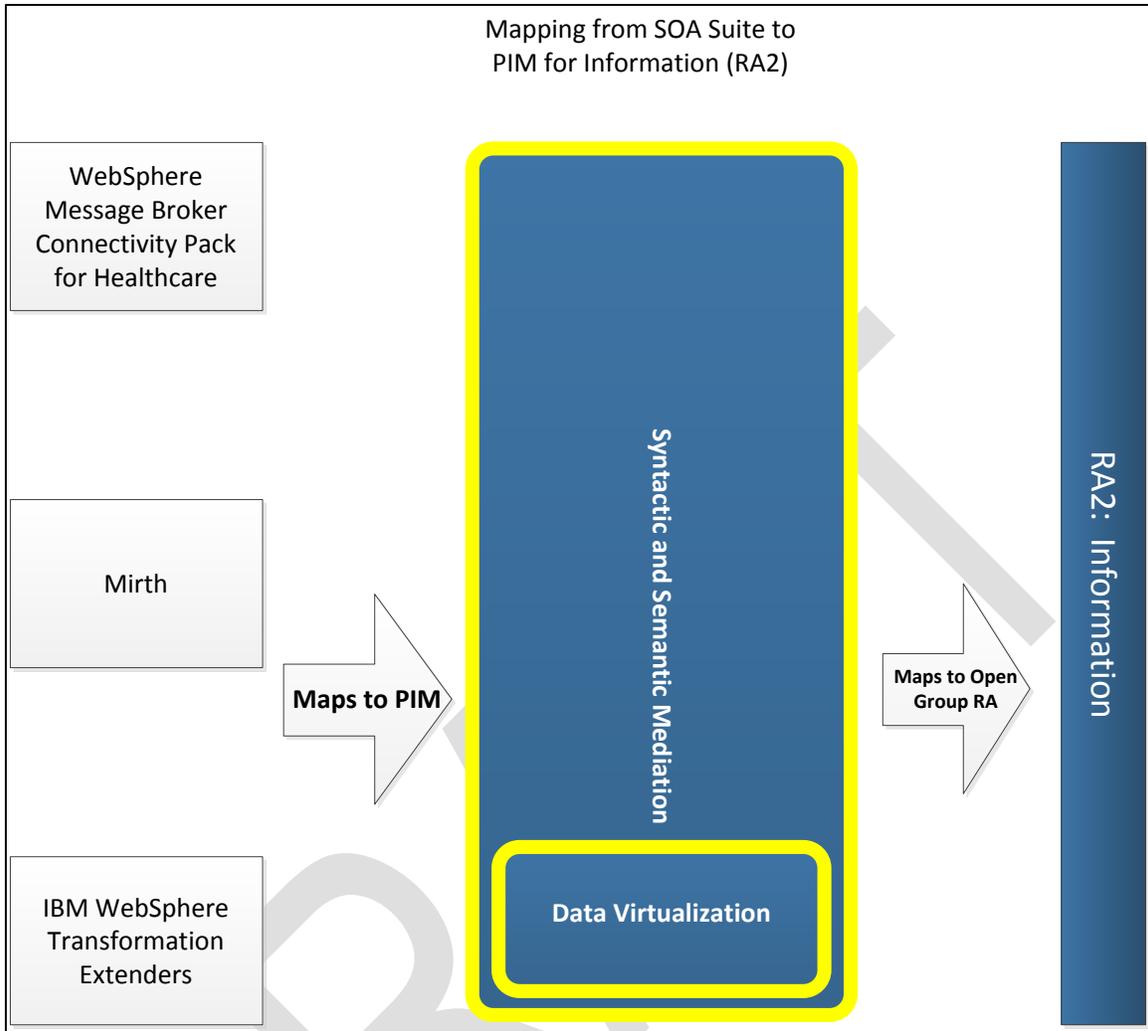


Figure 23 – The SOA Suite Components Associated with the Information Capability Mapped to the PIM and the RA

- Integration** – The Integration Capability is accounted for by multiple elements in the SOA Suite. The Suite contains more than one Messaging Engine. It contains the Layer 7 SOA Gateway, the IBM Message Broker, the Open Source Mirth Engine, as well as IBM’s Message Queuing technologies. Figure 24 depicts the mapping from the SOA Suite to RA3.

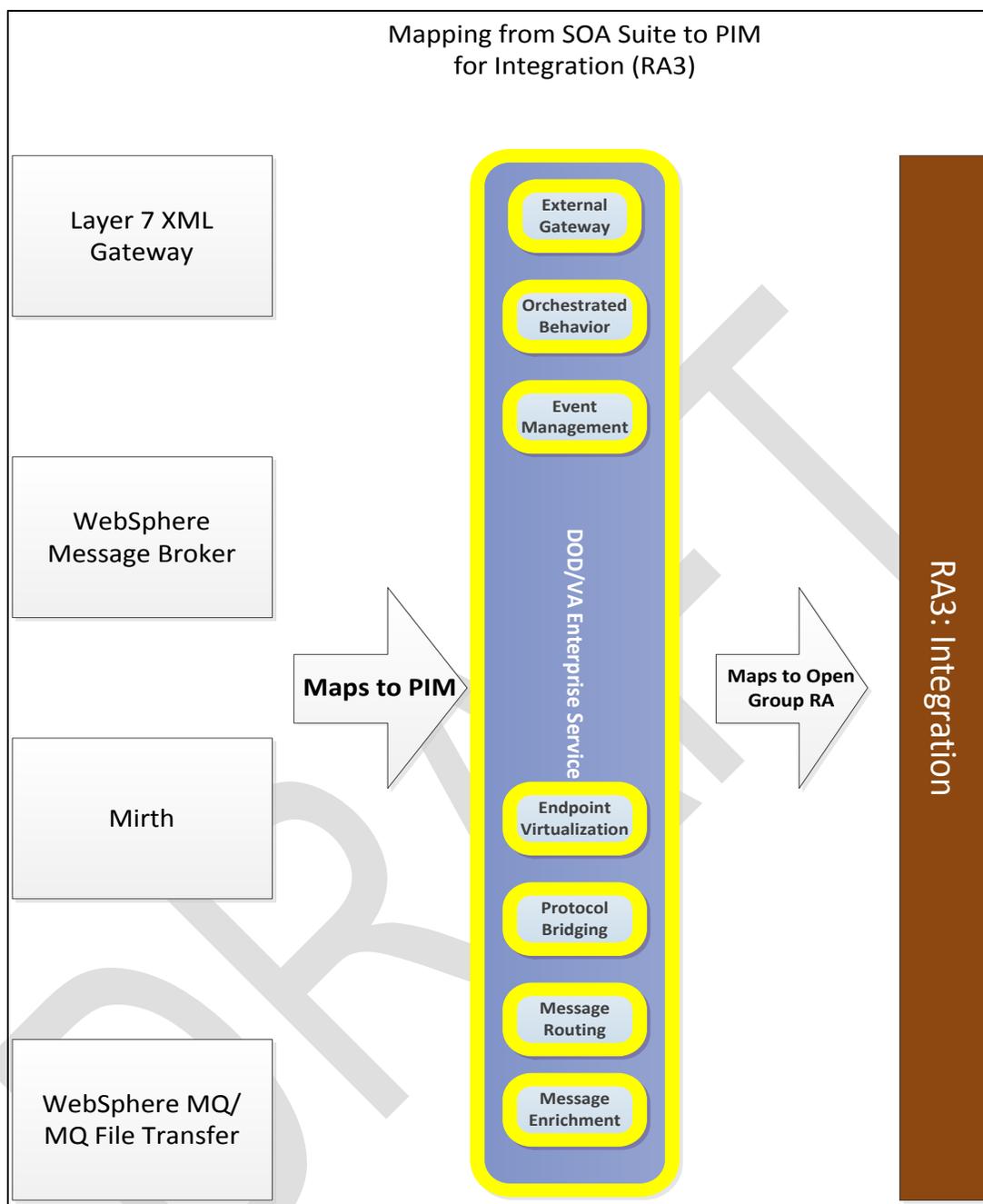


Figure 24 – The SOA Suite Components Associated with the Integration Capability Mapped to PIM and RA

- Consumer Interfaces** – The SOA Suite provides interfaces for development and administrative purposes. Since it is middleware, it does not provide interfaces for the business user. However, more generally, Consumer Interfaces encompasses Business-to-Business (B2B) interactions as well. The Layer 7 Gateway may be considered an interface for B2B transactions.
- Business Process Management** – Business Process Management from the RA, unlike other components in the RA, maps monotonically to the PIM as well as the SOA Suite. This functionality is provided by WebSphere Process Server described in Table 2. Additionally, the Suite provides the ILog Rules Engine. This directs to the Rules Engine section of the SOA RA, and can be used by the Business Process Management Segment (RA5) or the Services and Service Component (RA6/RA7).

- **Testing** – Testing of Services on the SOA Suite uses the CA LISA tool. The LISA tool enables end-to-end testing of SOA functionality by creating a test harness in which the SOA Stack runs. It uses SOA endpoint emulation to simulate SOA Consumers.

DRAFT

4-2. References

1. P. Kruchten, "The 4+1 View Model of Architecture," *IEEE Software*, vol. 12 (6), pp. 45-50, 1995. DOI: [10.1109/52.469759](https://doi.org/10.1109/52.469759)
2. MHS Enterprise Technical Architecture.
3. Open Group SOA Reference Architecture, The Open Group, 2011.

DRAFT

PART 5 MESSAGING

5-1. Message Model Standards

By default, Messages will use Fast Health Interoperability Resource (FHIR 0.9), HL7 2.x and 3.0 standards. As necessary, messages will leverage standards as follows:

- ANSI/HL7 V3 RCL, R1-2003: HL7 Version 3 Standard: Refinement, Constraint, and Localization to Version 3 Messages, Release 1. ANSI/HL7 V3 RIM, R1-2003: HL7 Version 3 Standard: Reference Information Model, Release 1 5. ANSI/HL7 V3 XMLITSDT, R1-2004: HL7 Version 3 Standard: XML Implementation Technology Specification - Data Types, Release 1
- ANSI/HL7 V3 COMT, R1-2004: ANSI/HL7 V3 XMLITSDT, R1-2004
- ANSI/HL7 CDA, R2-2005 (R2010): HL7 Version 3 Standard: Clinical Document Architecture (CDA), Release 2
- ANSI/HL7 V3 RCMR, R1-2006: HL7 Version 3 Standard: Medical Records, Release 1
- ANSI/HL7 V3 RBAC, R1-2008: HL7 Version 3 Standard: Role-based Access Control Healthcare Permission Catalog, Release 1
- ANSI/HL7 V3 MRDACM, R1-2008: HL7 Version 3 Standard: Medical Records; Data Access Consent, Release 1

Messages within a business process or between services where HL7 is not required can use XML or Java Script Object Notation (JSON) formats. However, these must adhere to the canonical semantic and syntactic models. For XML formats, XSDs and Examples must be provided as part of Service Documentation. JSON implementations must provide documentation and examples as well.

5-2. Message Protocol Standards

- Services shall be implemented using SOAP/HTTP(S), XML/HTTP(S), SOAP/Java Message Service (JMS), XML/JMS.
- The preferred data exchange format shall be FHIR when technically feasible. SOAP with Attachments, SOAP with Message Transmission Optimization Mechanism (MTOM), or Java Message Service (JMS) shall be used for exchanging binary data.
- Web services implemented with SOAP/JMS and XML/JMS shall use CoE-approved implementation of JMS.
- JMS message types shall be TextMessage for XML data or ByteMessage for binary data. To prevent undesirable language coupling, serialized objects should not be passed using the ObjectMessage type.
- If XML payloads are to be exchanged, SOAP messaging shall be used whenever possible to maximize interoperability.
- The XPath and XQuery standards shall be used for querying XML documents.

5-3. Message Criticality and Patient Safety Frameworks

Both VA and DoD provide Patient Safety Frameworks which address the criticality of the message and the handling of problems of the messages with respect to Patient Safety. These frameworks should be adhered to in order to ensure that appropriate prioritization of messages occurs according to patient safety priority

Service Level Agreements (SLA) and Service Documentation shall provide indicators for Message Criticality. Message Criticality implies the urgency of the message to be transported. Patient health related messages will be given priority over administrative messages. The Message Criticality levels are as follows:

- **Urgent** – The message concerns patient health and must be delivered as quickly as possible. The amount of time the message is persisted and the number of retries will be specified in the SLA.
- **Default** – The message may or may not involve patient health and can be delivered using any value within the range of response times defined in the SLA. Message persistence and the number of retries are the default values for the ESB.

While rules-based routing may be applied towards message criticality, at the outset criticality will be addressed using service throttling.

5-4. Service Taxonomy

SOA Service Taxonomy is used for categorization and lookup of Services. The taxonomy allows a narrowing of the search for services associated with a particular set of characteristics. The iEHR SOA has chosen to evolve the taxonomy as needed starting with a simple categorization system. In order to describe the initial taxonomy, we first must define Service and Candidate Services.

- **Service** – A service is a logical representation of a repeatable activity that has a specified outcome. It is self-contained and is a "black box" to its consumers.
- **Candidate Services** – A potential service in the Inception Phase. It has a description and possibly an indication of inputs and outputs.

Having defined several key terms, the taxonomy is here defined. The iEHR SOA Services are categorized into:

- **Utility Services** – Provide generic processing logic that is not classified as business logic. Utility logic is a "cross-cutting" logic because it is ideally agnostic (sufficiently generic, no knowledge of particular parent task) and reusable and therefore multi-purpose in nature. They are positioned at the bottom part of RA6 in the RA.
- **Entity Services** – Encapsulate a specific business entity (such as an invoice, patient record or timesheet). Entity-centric services are useful for creating highly reusable and business process-agnostic services that are composed by an orchestration layer or by a service layer consisting of task-centric business services (or both). They are positioned at the middle part of RA6 in the RA.
- **Task Services** – Encapsulate business logic specific to a task or business process. This type of service generally is required when business process logic is not centralized as part of an orchestration layer. Task-centric business services are reused within the context of the business process they model. They are positioned at the upper part of RA6 or in RA5 in the RA, depending on the implementation and the available SOA/BPM packages.

More explicitly the taxonomy of service can be represented by a matrix describing the extent to which the types of services are or are not domain agnostic. This is shown in Figure 25.

	Business Logic	Utility Logic	Agnostic Logic	Non-Agnostic Logic
Task Services Sub-layer	✓			✓
Entity Services Sub-layer	✓		✓	
Utility Services Sub-layer		✓	✓	

Figure 25 - Taxonomy Matrix

This taxonomy has been applied to the Common Services Catalog and will be used until the CoE decides a need for additional taxonomy.

5-4.1 Taxonomy Attributes

Every taxonomical node SHALL have the following attributes:

Table 3 – Taxonomy Attributes

ID	A sequence of characters capable of uniquely identifying the node within the taxonomy.
Parent ID	The ID of the parent node to which the node is directly subordinate. This is included to support information about the taxonomy structure.
Name	A designation of the node by a linguistic expression intended to be used by humans.
Allowed	An indicator that the node is allowed to be assigned to or used to describe a web service and its attributes ("Yes") or that the node exists strictly to group and categorize other nodes within the taxonomy itself and cannot be assigned to or used to describe a web service and its attributes ("No")
Description	A representation of the node by a descriptive statement.

5-4.2 Documenting Taxonomies

This section presents requirements for documenting taxonomy.

- All taxonomies shall be portrayed through a common set of metadata as described below.
- The taxonomy metadata shall include a namespace.
- If a namespace already exists, that namespace shall be applied.
- If a namespace does not exist, the namespace shall be created and registered.
- The metadata shall include a taxonomy title.
- The metadata shall include the taxonomy identifier, which is identical to the taxonomy's namespace.
- The metadata shall include the name of the creator of the taxonomy.
- The metadata shall include a brief description of the taxonomy.
- The metadata should include a brief description of the purpose of the taxonomy, and where and how it can be applied.
- If the taxonomy was derived in whole or in part from an existing resource, the metadata shall include the name of the resource.
- If the taxonomy was derived in whole or in part from an existing resource, the metadata shall include the network location (URL) of the resource.
- The metadata MAY include any special circumstances when use of the taxonomy is not recommended.
- The metadata shall include a list of values (nodes) described using attributes shown in section 1-8 8.1 above.
- If there are few values, then the metadata shall include the entire list.
- If the list is extensive, then the metadata shall include the name and network address (URL) of an external resource that contains the entire list.
- The metadata should include a brief overview of the structure of the taxonomy, either as a plain language description or as a diagram of the hierarchical relationships among the taxonomy's nodes.
- The metadata should include at least one example of the use of taxonomical nodes, either in a registry or as a part of the description.
- The metadata should include clear instructions on where and how to obtain the taxonomy, together with a network location for every external resource listed in the taxonomy's metadata.

- The metadata should include a statement regarding Intellectual property rights (including copyright) affecting use of the taxonomy, validation routines or validation Application Programming Interface (API), including licensing requirements, etc.
- The metadata should include a condition of use, such as any requirement for registration, payment, or a legal agreement before access to any of the resources.
- 0 provides a template for requesting new taxonomies.

DRAFT

PART 6 REGISTRY AND REPOSITORY GUIDELINES

The Production Services Registry/Repository (SRR) Policies govern both the use of the SRR and the management of its contents. The SRR incorporates registry and repository functionality to provide human-readable and machine-readable metadata and artifacts enabling service discovery and consumption. Governance is necessary to ensure the integrity, consistency and completeness of this service meta-information.

The exposure of service meta-information in the SRR facilitates its design-time discovery and access by authorized consumers. The following policy statements apply for SRR governance:

- The SRR shall be the design-time system of record for all consumable services offered to service consumers and other qualified parties.
- Categorization schemes shall be established for service meta-information contained in the SRR that enables its discovery.
- Programs shall not publish their service contract through any mechanism other than the SRR, except for testing purposes.
- The CoE shall be the administrator of the production SRR. A development Registry/Repository will be made available for implementers.
- All SRR users shall be approved by the CoE before an account is created.
- There shall be one well known URL for SRR access.
- Only authorized service providers shall publish service meta-information to the production SRR.
- Service meta-information shall be approved by the CoE before its publication in the SRR.
- Recommendations and change requests regarding SRR structure and architecture shall be collated by the SRR administrator and elevated to the CoE for consideration and implementation.
- Disputes among service providers and consumers shall be elevated to the CoE, which shall resolve the dispute and delegate accordingly.
- Service meta-information shall be entered into the SRR at the appropriate lifecycle milestones in accordance with the service registration process.
- Discovery of a Service and its associated meta-information in the SRR shall be limited to authorized SRR users.
- The Service Provider, upon approval from the CoE, has the right to restrict access to the Service and associated meta-information in the SRR.
- Consumers shall provide suitable identification and contact information (both machine and human-readable) to the SRR administrator for each registered service they are consuming.
- All services shall have a point of contact. The point of contact manages the details of the service, such as, but not limited to, the inability to use the service, data format, access requests, service registration and instances of QoS failure.
- Custody or ownership transfer of any operational service shall be negotiated and coordinated by the CoE.
- Authorized SRR users shall be able to access meta-information for previous versions of a service.

6-1. Working with the Repository

6-1.1 Definition and Disambiguation

- A Services Repository manages services from a business point of view.

- A Services Registry manages services from a technical point of view.

Initially it was suggested that registries hold metadata and repositories hold data. However, since documents like XML can be considered data as well as metadata, we see more vendors offering integrated registry/repository solutions.

Several distinctions may be helpful in clarifying the boundaries. One useful distinction is between design-time and run-time. Both registries and repositories have design-time and run-time features.

- Design-time metadata are mostly focused on description and discovery. Design-time data typically reflect artifacts such as code, WSDLs, and XSDs.
- Run-time metadata is focused on delivering contract and policy information. Run-time repositories typically store messages and provide query, audit, logging and a variety of archiving capabilities.

Table 4 – Registry / Repository - Design-time / Run-time

	Design-time	Run-time
Registry	<ul style="list-style-type: none"> • Discovery • Description 	<ul style="list-style-type: none"> • Contracts • Policies • Versioning
Repository	<ul style="list-style-type: none"> • Code versions • Documentation 	<ul style="list-style-type: none"> • Message storage that can be queried • Event Logging • Auditing

6-1.2 Design-Time Repository Guidelines

- XML inline comments should be used so that future designers can follow the original design considerations and thought process.
- The “Garden of Eden” XML building pattern should be used for consistency and unified appearance.
- Due to the dispersed nature of the applications and location of users, all of XML Schemata and I/O documentation should specify UTF-8 as their encoding, and all XML date/time types should require either a time-zone or that date/time should be assumed to be in UTC.
- The Repository should contain requirement(s) artifacts describing all needs. These should be linked to all other artifacts to ensure traceability.
- A canonical data model in XML schemata describing all data that are visible to (concern) the Business should be created. In-line comments are strongly suggested for all items. They will be used by the WSDLs later.
- Appropriate namespaces for the data to correspond to the business needs and the overall architectural guidelines should be produced (CoE approved, Architecture supervised.).
- All relevant standard taxonomies (drug prescriptions, interactions, country codes, other health data etc.) should be added into the repository so that no taxonomies should be referenced outside the Repository.
- An additional namespace for faults and errors should be established.
- The business operations on the data should be described. Thus, WSDLs will describe the services needed. (CoE approved, Architecture supervised.)
- Any additional elements needed for the complete description of the services should be linked to new or existing XML schemata (data).

- Any necessary XSL or other Data Transformation Rule should be added to the Repository. Transformation rules should likewise be added to the Registry.
- Any BPEL (BPMN, XPDL, etc.) constructs. (XML entries/artifacts.) should be added to the Repository.
- All supporting documentation – IPO documents, governance stipulations, UML diagrams describing the services' interactions and structure should be added.
- Existing or created SLAs should be entered. These should be linked to the appropriate requirement(s). Do not enter default (empty, "place-holding") SLAs as they may confuse designers (they may be perceived as "completed.")
- Create and enter end-points for the WSDLs/Services. Incorporate the fact that an ESB alters the end-points' path. Enter the ESB configuration too.
- The accuracy of the end-points via a mock-up procedure such as SOAP-UI or CA LISA simulation should be tested.

6-1.3 Design Time Repository Artifacts

6-1.3.1 Non-Versioned Artifacts

- **SLAs** – Even though an SLA is unenforceable at design time, designers need to be aware of it. Versions of SLAs at design time are irrelevant, as only the latest will (possibly) move into production, if ever. In addition, developers cannot target multiple versions (and they will never be expected to.)
- **WS-Policy Documents** – These are not relevant for this phase. The document's latest version only is needed here. If promoted, that WS-Policy document will be enforced later, at run-time.) Just like SLAs, there is no sense versioning these at design time, since only the latest will migrate to run-time.
- **Permalink (for the Registry)** – An unchanging link to the repository page for the artifact being viewed.

6-1.3.2 Versioned Artifacts

- **BUILD Scripts** – For applications. Link to their build scripts here (i.e. MAVEN2/3, Ant, and make/nmake).
- **WSDL**
- **XML Schema(ta)**
- **XSL Transforms/Stylesheets**
- **WS-Policy Documents** – That are relevant at this phase (such as the ones that will be enforced during testing).
- **MIME TYPE (even WSDLs have mime types)**
- **Target Namespace (wherever relevant)** – For WSDL, XSD, XSLT, etc.

6-1.4 Run-Time Repository Guidelines

- Apart from the Service Lifecycle states, artifacts can exist in one of four states.
 - **Staged** – All artifacts promoted to this state are visible to admin only.

- **Live** – All artifacts in this state are discoverable.
- **Maintenance** – All artifacts in this state are discoverable but disabled for maintenance reasons. During discovery all is discoverable, but the invocation end-point (UDDI access point,) is blank.
- **Decommissioned** – All artifacts in this state are discoverable but disabled due to scheduled end-of-life (newer version, etc.). During discovery all are discoverable, except the invocation end-point (UDDI access point) and the UDDI overview URL are both blank.
- The administrator responsible for state transitions should be recording the changes and have his/her contact information associated with the actions.
- Invocation end-points should be reviewed and, if needed, updated from the Design-time values.
- Change management should be narrow and focused (Since the final/desirable state is “LIVE”.) Everything should abide the aforementioned states transition.
- An SLA indicating Repository availability should be implemented.
- When an artifact is “LIVE” its SLA(s) should be connected to the monitoring tool. During “MAINTENANCE” monitoring should not be disabled, but the frequency of alerts may be relaxed. (If disabled, monitoring would inaccurately report the up-time/down-time ratios.)

6-1.5 Run-Time Repository Artifacts

6-1.5.1 Non-Versioned Artifacts

- **Repository Administrator Contact** – Who is to be contacted if the associated entry is wrong or there are versions missing? There might be multiple Repository Administrators – which one has responsibility for this particular artifact or group of artifacts?
- **Permalink (for the Registry)** – An unchanging link to the repository page for the artifact being viewed. Needs no versioning because it will not change, it is frozen to that one artifact.

6-1.5.2 Versioned Artifacts

- **Service Author(s)**
- **Service Publisher** – May not be the same as above, e.g. DoD may author a service published by IPO.
- **SLA(s)**
- **Authorized Users/Consumers**
- **Authorized “VIEWERS ONLY”**
- **Security Policies and Assertions**
- **Java Archives (JAR Files), Broker Archives (BAR Files), other libraries or executables must be versioned**
- **ESB Configurations** – This way, the ESB can configure itself from a “single source of truth,” and a ready backup can be found here.
- **BPEL/BPMN** – Any executable business process language should have its own artifact type(s) just like WSDL and XML Schema, etc.
- **XSL Transformations or other Transformation Rules** – Whether used in application or ESB, all data transformations should be versioned and catalogued

- **Target Namespace (wherever relevant)**
- **Deployment vs. Invocation Link** – Between target namespace of deployment and proxy endpoint(s) if a Web Service, Web Service Operation, XSD, XSLT, etc. (Any proxied artifact needs linking back to the non-proxy hard endpoint. This is sometimes represented as a dependency from the ESB Proxy to the actual WSDL, or Service, or both).
- **Links to all dependent items in the Registry/Repository**

DRAFT

PART 7 ARCHITECTURAL PATTERNS

In general, the SOA provides support for the Service Implementation Patterns described in Hoppe et al. [1]. These patterns are preferred as they are industry tested and can be constructed using the SOA Suite. Additional patterns may be considered but must past review from the CoE.

7-1. Supported Web Service Standards

The Web Service Standards of Part 6 shall be preferred above other standards for web service development. Accepted standards include:

- HTTP 1.1
- JMS 2.0

DRAFT

7-2. Interface Management Standards

For iEHR, SOAP Services developed will be exposed using WSDLs during design-time. A WSDL provided by a Service Provider is stored in a registry in accordance with the registry data model to be defined by the CoE. Service consumers can then search the registry and discover the service they are seeking.

DRAFT

7-3. Enterprise Integration Software Application Patterns

Enterprise integration patterns are accepted solutions to recurring problems within a given context. Application Integration patterns are applicable at the application software layer. These patterns provide guidance to system designers and architects. The following sub-sections describe different integration patterns: accessing services through REST and SOAP protocols, real-time access, message transformation, chain of responsibility, interface versus implementation binding, transaction management and compensation pass, a message spraying pattern for notifications, and Universal Description Discovery and Integration (UDDI) service registration and lookup. Some of these patterns are discussed in depth with process diagrams in iEHR Enterprise Technical Architecture- Enterprise Application Integration document. Others may be found in Hoppe [1].

DRAFT

7-4. SOA Service Patterns

Web services are commonly used to implement SOA strategy. In the web service implementation there will be loose coupling between server code and the client code. There are two common types of web service implementations: SOAP based and REST. The SOAP based web service would use XML as the data format, while the REST web service would use would accept or send XML or JSON among other formats.

DRAFT

7-5. Real-time Access

There are two patterns for real time access; synchronous and asynchronous. In a synchronous implementation of a Web Service, the client connection remains open from the time the request is submitted to the server. It is based on a blocking algorithm where the service consumer process is blocked until it receives the response back from the service provider. Asynchronous real time access is conceptually the opposite of synchronous real time access, in that it is based upon a non-blocking algorithm.

DRAFT

7-6. Integration Governance

Software integration establishes communication of information between two or more components in a reliable, high-performance manner. A software integration solution may be exposed as a web service. The purpose of iEHR SOA Integration Governance is to provide normative guidance in the development of integrations solutions within the iEHR program. Integration Governance guidance is intended to define a set of standardized repeatable integration solution patterns to guide integration solution design and development. In addition, Integration Governance provides a standardized integration solution vocabulary to enable accurate documentation and communication of integration solutions.

The iEHR ESB is a high-performance messaging engine. Messaging is the iEHR paradigm for integrating heterogeneous systems. In order to develop repeatable messaging integration solutions using a common messaging integration vocabulary, iEHR integration standards are required. Numerous integration pattern libraries and vocabularies exist. As opposed to reinventing patterns and vocabulary, the iEHR program seeks to adopt and align with widely accepted standards. In particular, patterns from the Hoppe et al are used. Enterprise Integration Patterns (EIP) found in the text are available as Visio icons and are distributed under a Creative Commons attribution license located at: <http://www.eaipatterns.com/index.html>. EIP is adopted as the integration pattern library and integration pattern vocabulary for the iEHR program.

The SOA work stream of the Architecture Branch of the IPO Technical division is developing Integration Governance which consists of two processes: Integration Pattern Governance and Integration Component Certification Governance. Following are the task descriptions to develop these two processes.

7-6.1 Integration Pattern Governance Process

The Integration Pattern Governance PROCESS develops and enforces supporting architecture to the iEHR Architecture. The iEHR Architecture currently provides high level, enterprise technical architecture guidance. In order that project level solutions conform to iEHR Architecture in a consistent and reusable manner, Integration Governance is required. Integration Governance provides low-level, coding level guidance to software developers in order to develop solutions that comply with the iEHR Architecture. Integration Governance provides a limited, standardized, repeatable set of executable level design patterns which are applicable to multiple projects across the iEHR enterprise. Integration Governance can be considered a type of Solution Architecture, design standard and coding standard that supports and implements the overall iEHR Architecture. Integration Governance includes the discovery and specification of common Message Exchange Patterns (MEP), Service Design patterns and Data Integration Patterns. Integration Governance is related to and provides input into other types of SOA design governance and the SOA CoE. Integration Governance is documented and communicated through a standard pattern template. Code samples and proof-of-concepts are produced as resources permit.

7-6.2 Integration Component Certification Process:

The Integration Component Certification process defines and implements criteria for the evaluation of risks, capabilities and approaches for integration of significant components to the ESB. Components, such as large COTS packages are frequently procured for their business value. The business value of a component must be balanced with the Total Cost of Ownership (TCO) of the component. A factor of TCO is the components capability for integration. Such criteria include:

- Does the component provide a SOA interface?
- Does the component provide an API?
- At what level of business function detail is the component interface?

- Does the component expose a canonical information model?
- Does the component meet Information Assurance standards?
- Does the component meet performance criteria?

Attributes such as those mentioned are considered in the Certification of a Component.

DRAFT

7-7. Message Exchange Patterns

WebSphere Message Broker messaging solutions SHALL be derived from the Messaging Integration Governance,

Message Exchange Patterns (MEPs) are implemented primarily by messaging system developers. MEPs provide guidance through the accepted messaging solutions to recurring problems within a given context. MEPs are abstract enough to apply to most messaging technologies, but specific enough to provide hands-on guidance to designers and architects. Patterns also provide a vocabulary for developers to efficiently describe their solution. Using these patterns help integration architects and developers design and implement integration solutions more rapidly and reliably. The iEHR SOA Suite platform that includes IBM WebSphere Message Broker and Layer 7 products among others already implements some of these patterns.

There are approximately 60 patterns identified so far and they are categorized into different areas such as integration styles, channel patterns, message construction patterns, routing patterns, transformation patterns, end point patterns and system management patterns. A summary of these enterprise integration patterns and their designs/architectures can be found at the following link

<http://www.enterpriseintegrationpatterns.com/toc.html>

For convenience, a summary of EIPs are included. A full description of the EIPs can be located at the EIP web site referenced above. EIPs are divided into Solution Integration Patterns that are likely to be constructed by integration developers within iEHR and Component Patterns that are likely to be used within a Solution Pattern. The EIPs provide a common vocabulary to define and describe integration scenarios across the iEHR program. This common integration vocabulary serves to clarify and make explicit communication of integration architecture solutions.

7-8. Solution Integration Patterns

- **Content-Based Router** – Examines the message content and routes the message onto a different channel based on data contained in the message.
- **Pipes and Filters** – An architectural style which divides a larger processing task into a sequence of smaller, independent processing steps (Filters) that are connected by channels (Pipes).
- **Message Router** – A type of Filter which consumes a Message from one Message Channel and republishes it to a different Message Channel depending on a set of conditions.
- **Message Translator** – A type of filter which serves as serves to translate data formats between applications or other filters.
- **Point-to-Point Channel** – A type of Message Channel which ensures that only one receiver will receive a particular message.
- **Publish-Subscribe Channel** – A type of Message Channel which delivers a copy of an event to those users who indicated an interest in receiving the event.
- **Datatype Channel** – A type of Message Channel where all data is of the same type. The sender selects the appropriate channel for each type of data. The receiver knows the type provided by a particular channel and thereby knows what processing is required.
- **Invalid Message Channel** – A special channel for messages that could not be processed by their receivers.
- **Dead Letter Channel** – A special channel for messages that the messaging system determines that it cannot or should not deliver.
- **Request-Reply** – A design pattern in which a request-message is sent on one channel. A response is returned on a different channel.
- **Process Manager** – A central processing component that maintains the state of the sequence and determine the next processing step based on intermediate results.
- **Splitter** – Break out the composite message into a series of individual messages.
- **Message Filter** – A type of Message Router that based on a set of criteria, blocks non-conforming messages from entering the channel and routes conforming messages to the channel.
- **Recipient List** – Determine the list of desired recipients, and then forward the message to all channels associated with the recipients in the list.
- **Messaging Gateway** – A class than wraps messaging-specific method calls and exposes domain-specific methods to the application.
- **Messaging Mapper** – Mapping logic between that maps data between the messaging infrastructure and the domain objects.
- **Dynamic Router** – A type of Message Content-Based Router which reads rules from a control channel to determine where to route different types of messages.
- **Aggregator** – A type of tasteful filter which collects and stores individual messages until a complete set of related messages has been received. The Aggregator then publishes a single message distilled from the individual messages.
- **Re-sequencer** – A type of stateful filter which collects and re-order messages so that they can be published to the output channel in a specified order.
- **Composed Message Processor** – The Composed Message Processor splits the message up, routes the sub-messages to the appropriate destinations and re-aggregates the responses back into a single message.

- **Scatter-Gather** – The Scatter-Gather broadcasts a message to multiple recipients and re-aggregates the responses back into a single message.
- **Envelope Wrapper** – Wraps application data inside an envelope that is compliant with the messaging infrastructure then unwraps the message when it arrives at the destination.
- **Content Enricher** – Read a message, use message data to access an external data source, retrieve external data to add missing information to the message.
- **Content Filter** – Remove unimportant data items from a message leaving only important items.
- **Normalizer** – Routes a message type through a custom Message Translator to convert the message into a common format.
- **Polling Consumer** – AKA Synchronous Receiver, the application explicitly makes a call when it wants to receive a message.

DRAFT

7-9. Component Integration Patterns

- **Messaging Bridge** – A connection between messaging systems which replicates messages between systems.
- **Message Bus** – Connecting middleware between applications that enables various applications to work together using messaging.
- **Document Message** – A message which contains a data structure.
- **Event Message** – A message which contains notification of an event.
- **Return Address** – A Request-Reply design pattern, the identification of the return channel.
- **Correlation Identifier** – A unique identifier within a reply message that indicates the request message the reply is for.
- **Message Sequence** – When a reply is divided into a number of reply messages, the Message Sequence is the order of the reply messages.
- **Message Expiration** – A time limit indicating the period of time that the message is viable.
- **Command Message** – A message which invokes a procedure in another application.
- **Transactional Client** – Make the client's session with the messaging system transactional so that the client can specify transaction boundaries.
- **Claim Check** – Store message data in a persistent store and generate a message lookup key called the Claim Check. The application passes the Claim Check to subsequent components for a message look-up.
- **Routing Slip** – The Routing Slip is attached to each message, specifying the sequence of processing steps.
- **Channel Adapter** – Exposes a non-messaging application to the Messaging System enabling the application to send and receive messages and the Messaging System to execute the Application API.
- **Canonical Data Model** – A data model that is independent from any specific application. Require each application to produce and consume messages in this common format.
- **Messaging System** – IT infrastructure providing the capability to transfer packets of data: frequently, immediately, reliably, and asynchronously, using customizable formats.
- **Message Channel** – A Messaging System component providing the capability to write information to the channel and the other one reads that information from the channel.
- **Message** – A data record that the messaging system can transmit through a message channel.
- **Message Endpoint** – A messaging system client used by applications to send and receive messages.
- **Guaranteed Delivery** – A Message System capability and Quality of Service (QoS) that persists messages such that they are not lost even if the messaging system crashes.
- **Format Indicator** – The format indicator enables the sender to tell the receiver the format of the message. This way, a receiver expecting several possible formats knows which one a message is using and therefore how to interpret the message's contents.

7-10. Implementation Guidance

7-10.1 Message Broker Implementation Patterns

The SOA Suite's WebSphere Message Broker is used for routing and mediation between Service endpoints. Message Broker's internal workflow and development environment provides a drag and drop interface to build different routing and mediation patterns. These patterns can be stored in a Message Broker library of patterns. The use of pre-existing patterns in the library is encouraged prior to developing custom patterns. Custom patterns will be reviewed by the CoE for use by specific projects and included within the Message Broker Patterns Library as appropriate.

7-10.2 Interoperability Guidelines

All SOAP based implementations of a Service must comply with WS-I Standards. In particular, Services must comply with WS Interoperability Basic Profile, and WS Interoperability Basic Security Profile [2, 3] Criteria.

RESTful Services will use messages as described in the sections on Messages (Part 5).

7-11. Business Rules Governance

For the initial release of iEHR Business Rules will only support Rules-based routing of messages. Other uses of the Business Rules Engine will be considered in subsequent releases of the iEHR.

DRAFT

7-12. References

1. Hoppe, G. Woolf, B. Enterprise Integration Patterns. Addison-Wesley Professional, 2003.
2. Basic Profile 1.2, Web Service Interoperability (WS-I) Organization, 2010.
3. Basic Security Profile 1.1, Web Service Interoperability (WS-I) Organization, 2010

DRAFT

PART 8 NAMING CONVENTIONS

All custom built services will use the same naming conventions. This provides a consistent way to identify services, operations, and namespaces. Naming will conform to the following standards:

8-1. Service Naming

As with all Conventions, Service Naming may have exceptions, based on the “consumability” argument above. Services have to be consumable, thus understandable. They will be understood better if they are descriptive in a natural way.

Service candidates with high cross-application reuse potential should always be stripped of any naming characteristics that hint at the business processes for which they were originally built. For example, instead of naming an operation getVAPatientRecordID, simply reduce it to getPatientID.

Entity and Utility services should be nouns. E.g. Patient (Entity), LabResults (Entity), SignOn (Utility), Document (Utility). Entities and utilities are the reusable “offerings” of the business (Entity) or technical (Utility) environments. By being strictly nouns, they are better understood and reused.

Task services should be nouns, or sometimes gerunds or verbs: They describe a “process,” thus they refer to actions that are associated with the service. AmendPaymentAward (from VBA), Auditing (gerund), PatientEncounter, ResultsAnalysis. Tasks, describing processes or process steps, are better understood if they use phrases with clear business meaning.

8-2. Namespaces

In general, a namespace uniquely identifies a set of names so that there is no ambiguity when objects with different origins and the same names are mixed together.

A Web Services Definition Language (WSDL) definition consists of a collection of elements with different origins. Each definition often involves a number of different namespaces. The following common namespaces are used to represent specification-based elements:

- <http://schemas.xmlsoap.org/wsd/>
- <http://schemas.xmlsoap.org/wsd/soap/>
- <http://www.w3.org/2001/XMLSchema/>
- <http://schemas.xmlsoap.org/wsd/http/>
- <http://schemas.xmlsoap.org/wsd/mime/>
- <http://schemas.xmlsoap.org/soap/envelope/>

When assembling a WSDL from iEHR modules, additional namespaces will be introduced, especially when importing XSD schema definitions. When defining the project’s own elements, the Provider will establish more namespaces to represent application-specific parts of the WSDL documents. (It is not uncommon for larger WSDL documents to contain up to ten different namespaces and their associated qualifiers.)

The WS-I Basic Profile requires the use of the target Namespace attribute to assign a namespace to the WSDL as a whole. If the XSD schema is embedded within the WSDL definition, then the WS-I Basic Profile requires that it also be assigned a target Namespace value that can be the same value used by the WSDL target Namespace.

Part 21 provides a template for requesting namespaces.

PART 9 CODING CONVENTIONS

Custom Services will use language specific coding conventions. Names, styles, and idioms should leverage the guidance provided by the language creator. For example, Java development should leverage the standards provided by Oracle. Similarly, .NET development should follow standards preferred by Microsoft.

DRAFT

PART 10 LOGGING, AUDITING AND ERROR HANDLING

10-1. Logging

In a distributed n-tier implementation such as SOA, in the absence of a consolidated logging and error-handling strategy, conducting data analytics and root cause analysis can be challenging. If Service Providers and Service Consumers do not use a consistent set of methods for logs, errors, and audits, traces will not produce a complete picture of events. More significantly, core security and regulatory policies may be violated. This section discusses recommended patterns for logging, auditing, and error-handling within the iEHR SOA. Each of the SOA Suite components produce a number of types of error messages and codes, and service developers are expected to take full advantage of these utilities to display or pass these error messages to the next application layer appropriately. Proper handling of these error messages is also important for help desk support and problem resolution. References to error codes and messages are provided in SOI Governance Volume 3.

- **Logging** – Fundamentally, the purpose of a logging system is to support analysis. This analysis can be associated with monitoring, maintaining SLAs, or troubleshooting. For a large distributed system as is the case for the iEHR implementation, the log management system must conform to a stringent set of requirements.
- **General Consideration for Logging** – In order to reduce overhead, services will send messages to a log/audit server (going forward this will be referred to as the log server). If the log management system is to be useful then it must satisfy the following criteria:
- **Reliability** – The logging system should be resilient to network interruptions, system overloads, and emergency shutdowns.
- **Availability** – The logging system should be accessible to authorized users and systems. In particular, for high availability purposes, the logging system should use clustered logging.
- **Scalability** – The logging system should be able to address increasing numbers of requests.
- **Event Sequence Management** – The logging system should be able to sequence incoming events based on timestamp or event sequence ID. Geographically distributed logging should account for time and location differences. Additionally, the log management system should have a process to mark and reconcile event collisions should they occur.
- **Query Support** – The log management system should support reads of the logs even as they are being updated.
- **Consideration Regarding Protocols** – Log messaging can use a variety of protocols. Since the SOA will be running largely on a Linux environment, the role of Syslog may be considered. Historically, Syslog has used User Datagram Protocol (UDP). However, while UDP has low overhead, it does not support reliable delivery. A TCP/IP based protocol is preferred, for several reasons:
 - **Delivery Acknowledgements** – In the absence of queue-based delivery mechanisms, TCP/IP provides a level of delivery reliability.
 - **Packet Sequencing** – TCP/IP will place packets in the proper order independent of the order in which they were received.
 - **SSL/TLS support** – SSL/TLS are supported by TCP/IP. IHE Audit Trail and Node Authentication (ATNA) requires the use of TLS.

The iEHR RA expects applications to use TCP/IP based logging as part of a reliable delivery mechanism. OSI Layer 7 protocols that support reliable delivery are acceptable. Assured delivery mechanisms that support Store and Forward, such as Message Queues, are optimal.

- **Queue Based Logging** – The SOA Suite provides logging functionality. The various components of the SOA Suite produce their own logs. Given the number of interacting components and the

presence of multiple requests and responses, log management can go beyond merely monitoring the logs of each component and manually doing reconciliation across components. An asynchronous log management system is therefore desirable. The log management system processes logs from the various components and creates a consolidated log record. One such approach to log management is described as follows.

- In order to maintain low-overhead and minimize locks on the file system(s), it is not unusual to use queue-based logging. Queue-based logging implies that all log messages are sent to a message queue. The queues forward to a log manager that writes the information to the appropriate data stores. Figure 26 below shows a basic pattern for logging that uses message queues to sequence log messages and assure delivery.

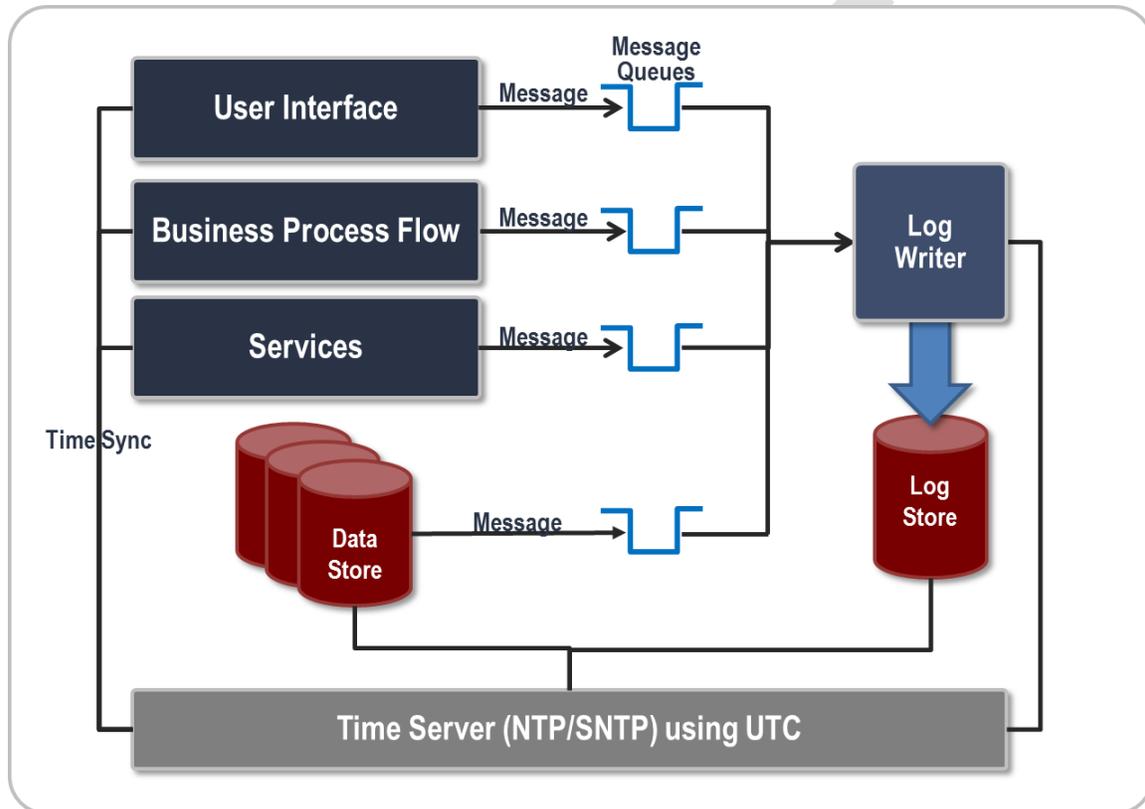


Figure 26 – Default Logging Pattern

- A more general log management system using clustering and distribution is shown in Figure 27 below.

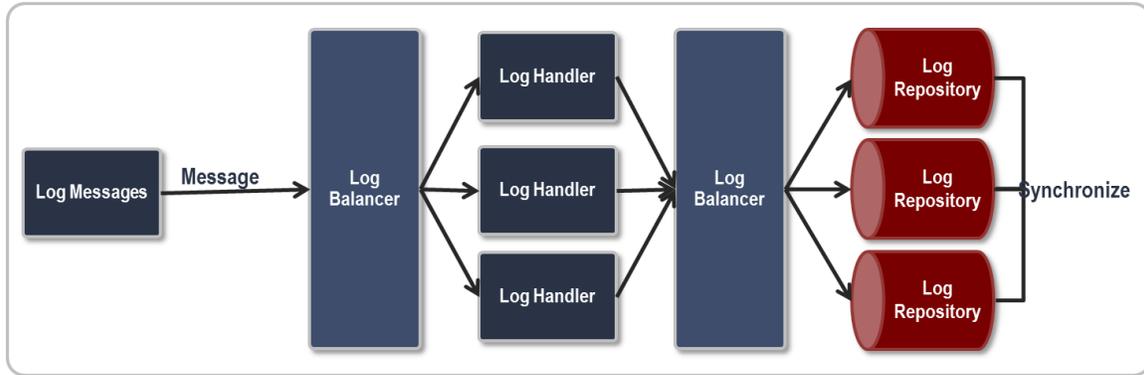


Figure 27 – Clustered Log Handling

- As before, the pattern of Figure 27 requires time synchronization against an authoritative time server.
- **Log Message Format** – In order for Log Messages to be parsed, a formal log structure must be in place. The structure provided here is based on the NwHIN/ATNA auditing criteria [1, 2]. The full NwHIN format is discussed in the Audit Section below. The log levels correspond to SL4J definitions [3]. The Message must have the following core components.
 1. **Message ID** – A global unique identifier (GUID) for the Message
 2. **Message Source** – The server IP address, the service name and the method generating the message
 3. **Consumer Identifier** – The requestor sending the request
 4. **Provider Identifier** – The service executing the request
 5. **Sent Message Timestamp** – The UTC timestamp of the message.
 6. **Received Message Timestamp** – The UTC timestamp of the message
 7. **Action/Event Status** – *TBD*
 8. **Success** – The action executed and returned the result to the Consumer without error.
 9. **Failure** – The action executed and returned to a failure message to the Consumer with an error.
 10. **Partial** – The action involves asynchronous messaging and Success or Failure must be determined by the response to the message.
 11. **Message Text** – A human readable message. Optionally, a before and after Snapshot can be maintained.
 12. **Log Level** – One of DEBUG, INFO, WARN, ERROR
- **Log Message Schema** – The default log message format will be as follow the Message Hierarchy described below in Figure 28.

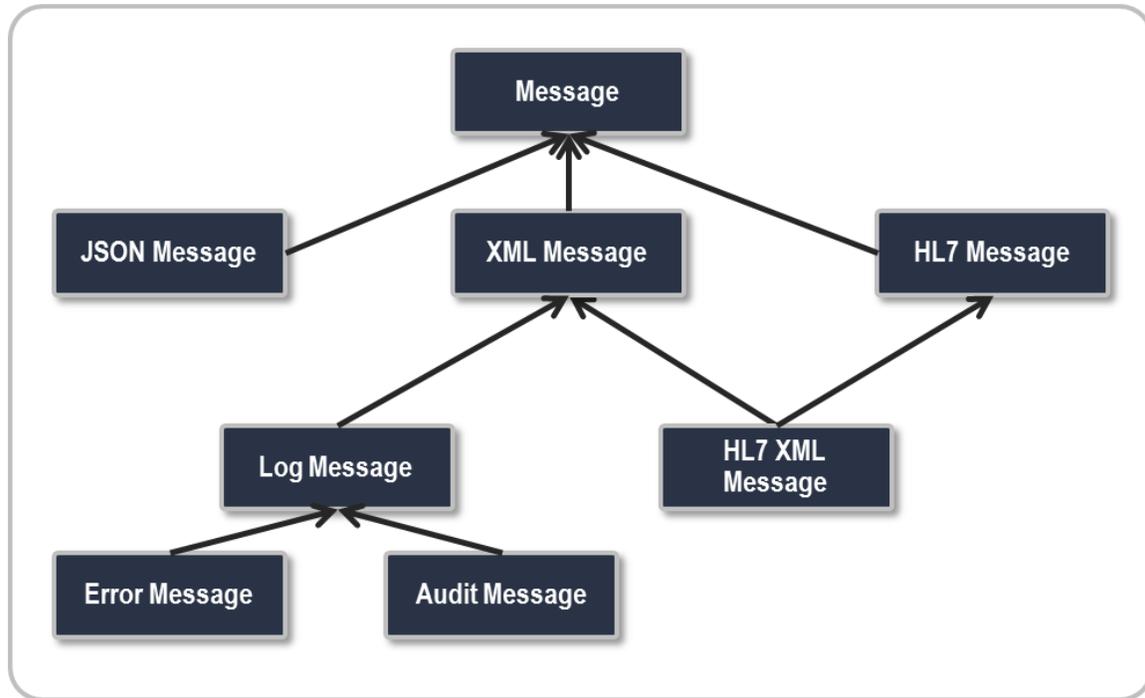


Figure 28 – Message Hierarchy

10-2. Auditing Records

In order to conform to Healthcare Reference Models, and regulatory constraints, Audit Records are a necessary part of any EHR. In particular, Audit Records are necessary for purposes of security and Health Insurance Portability and Accountability Act (HIPAA) compliance and ATNA. The ATNA Standard and DICOM extensions will be used as it satisfies both requirements. The ATNA audit criteria and the associated RFC 3881 have been extended by the DICOM committee. The iEHR SOA will leverage the DICOM extended version. This specification will not be described here. The reader is directed to the ATNA and DICOM specifications [2, 4].

For HIPAA compliance requirements the reader is directed to the HIPAA requirements [5]. A summary of HIPAA guidelines is as follows.

- **General Controls** – Record who did what to which object, when, and on which system.
 - Record which events each system is capable of logging.
- **Events to capture**
 - Machine startup and shutdown; startup and shutdown of audit function.
 - Successful/unsuccessful login and logout of users; denial of service events.
 - Add, modify, and delete actions on all data/files/objects; plus read/view actions on data classified as restricted.
 - Use of all privileged accounts and utilities.
 - Changes to user accounts or privileges (creation, modification, deletion).
 - Automatic logout of a user after exceeding a locally defined time of inactivity or excessive login attempts.
 - Switching to another user's access or privileges after logging in.

- Software or hardware modification.
- All access to security files, attributes, or parameters; any action to circumvent security controls including access to anti-virus software.
- **Operation events to capture**
 - Login attempts with failed identification or authentication, also known as failed login attempts.
 - Changes of the time or date of the system clock.
 - Emergency mode operation.
 - Detection of a virus.
 - Detectable hardware and software errors; log failure and restart events.
 - Changes to log files (creation, deletion, and configuration).
- **Communication events to capture**
 - Network link failures.
 - Device connection failure due to device identification or authentication failure (also known as a failed connection attempt).
 - Network and device connections dropped.
 - Data integrity verification failure for information transmitted over a network.
 - Message authentication failure for information transmitted over a network.
 - Overrides of network abnormality alarms and alerts.
 - IP addresses of successful and unsuccessful connections.
 - Changes to network security configuration (e.g., firewalls).
- **Content of audit trails**
 - Date, time, type, and any applicable error condition of event.
 - The ID of the user who caused the event.
 - The application that created the audit event.
 - The application(s) responsible for executing the event.
 - The component or workstation that initiated the event, and where the event happened.
 - Description of the event, which may include before and after images.
- **Monitoring**
 - Follow up on suspicious events such as intrusion attempts, authorized accesses at unusual times, and unusual changes to infrastructure devices.
 - Identify, investigate, report, and respond to inappropriate activity.
 - Ensure that audit requirements and activities do not unduly disrupt critical business processes.
 - Identify the individuals performing event analyses. Each shall be independent from those setting audit trail rules. Ensure they are available and that they record who, what, when, where, and why sensitive information is released. Rules-of-evidence integrity must be maintained.
 - Document all event capturing and analysis procedures, requirements, and responsibilities, including when to involve forensics specialists. Develop a process to ensure that users comply with access control procedures, including strong password creation and protections.

- Audit all user activity where risk levels warrant.
- Employ event analysis support tools and/or e-intelligent methods of correlating log data to detect suspicious activity and reduce volume.
- **Maintenance and storage of audit trails**
 - Audit trails must be managed only by authorized staff.
 - Audit trail retention varies according to legal requirements and business needs. PHI and audit trails must be archived for six years. Other Federal laws and regulations may stipulate other retention periods; always use the most stringent guideline when the data is covered by more than one policy, law, or regulation.
 - Audit trail records management retention and disposal rules must be documented.
- **Audit Record Architecture Pattern**
 - The overall audit design pattern is the same as that in Figure 28, where the Log Repository is replaced by an ATNA Audit Repository as shown in Figure 29.

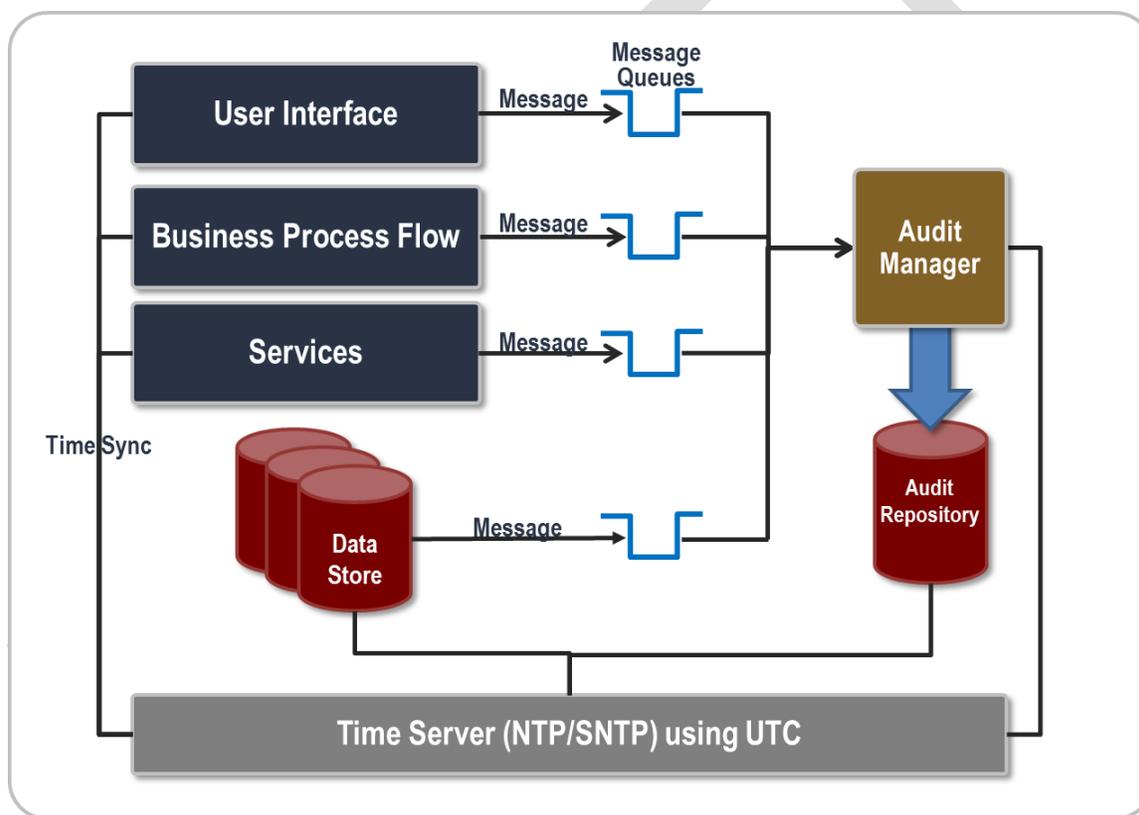


Figure 29 – ATNA Audit Pattern

- **Exception and Error Handling** – As mentioned earlier, a strategy for Exception and Error Handling in a distributed services environment is fundamental to successful operations of the architecture. This section will provide a distinction between exceptions and errors and will proceed to describe design patterns for exceptions and errors.
- **Exceptions vs. Errors** – The iEHR SOA makes the following distinction between Exceptions and Errors based on Java Exceptions and Errors [6].
- **Exceptions** – Addresses application level conditions. In general, a service should use exception handling to either address the exception locally, or else re-throw the exception to the next handler in the exception chain. If the messaging is using SOAP, the Exception should be wrapped in a

SOAP fault. Regardless of the message format, the exception contains the information described previously in the section on Logging. Exceptions that affect the user should propagate to advise the Consumer of the appropriate action.

- **Errors** – Addresses system and infrastructure conditions that are outside of the Providers or Consumers control. Errors should be handled by a chain of error-handlers. Errors should be logged and the overall error-handling strategy between Consumer and Provider should allow the failure to occur gracefully with appropriate messages sent to the Consumer.
- **Privacy** – Logs, Exceptions, Errors, and Audits shall not contain Personal Identity Information (PII) or Personal Health Information (PHI). For traceability and troubleshooting purposes, only de-identified data should be used.
- **Exception/Error Message Format** – The Error Message Format is also an extension of the Log
 - Message Format and contains:
 - **Type** – Exception or Error
 - **Message** – A human readable message describing the Error or Exception.
 - **Code** – A Code indicating additional descriptive information available in documentation.
 - **Code Type** – The protocol associated with Exception or Error (e.g. HTTP/S, HL7, MLLP, JMS, etc.).
 - **Number** – A numerical reference to the code for look up purposes.
 - **Recommended Action** – Optionally, provide a suggestion as to how to recover from the error/exception.
 - **Stack Trace** – Optionally, the Error/Exception Message can contain a stack trace associated with the error.
- **Analysis Activities and Tools** – The iEHR RA does not prescribe a particular analysis strategy for logs and audits. It is the responsibility of the development and maintenance organization to select and implement analysis techniques such as Reporting Tools or Log Analyzers.

10-3. Error Codes

As of this writing, the SOA ESB team has not defined explicit error codes. However, error code ranges have been defined. These ranges are as follows:

- **001-999 – Message** – An error has occurred in processing a message.
- **1000-1999 – Security** – An error has occurred in implementing security constraints.
- **2000-2999 – Data** – An error has occurred with data read/write.
- **3000-3999 – Message Broker** – An error has occurred in Message Broker's ability to execute a flow.
- **4000-4999 – Rules Engine** – An error has occurred in the processing of rules by the Rules Engine.
- **5000-5999 – Consumer Interface** – An error has occurred in the User Interface or System-to-System Interface.
- **6000-6999 – BPM** – The Business Process Management Engine has encountered an error.
- **7000-7999 – Service** – A Service specific error has occurred.
- **8000-8999 – Service Component** – A Component comprising a Service has encountered an error.
- **9000-9999 – Packaged Application and Legacy** – A COTS/GOTS or other legacy access has resulted in an error.

10-4. References

1. Audit Log Query Service, Nationwide Health Information Network, v1.3.1. 2009.
2. Audit Trail and Node Authentication, IHE, http://wiki.ihe.net/index.php?title=Audit_Trail_and_Node_Authentication
3. Simple Logging Façade For Java (SLFJ). <http://www.slf4j.org/>
4. Digital Imaging and Communications in Medicine (DICOM). Supplement 95: Audit Trail Messages, 2004.
5. An Introductory Resource Guide for Implementing the Health Insurance Portability and Accountability Act (HIPAA) Security Rule. NIST Special Publication 800-66 Revision 1. October 2008.
6. Java Exceptions. <http://docs.oracle.com/javase/tutorial/essential/exceptions/>

DRAFT

PART 11 VERSIONING

A Service is comprised of multiple components. The executable code itself, the service descriptions, the Service Level Agreement (SLA) description, and any supporting documentation must therefore be versioned.

The service, including its related artifacts and code, will be controlled by change management and will be properly versioned. The policy statements in Table 5 refer to identification of service implementation versions, and specifically to the published version identifier for those implementation versions. There is no assertion or assumption that other related artifacts are versioned in lockstep with the service implementation.

Multiple versions of the service will co-exist to allow consumers the ability to migrate to newer versions independent of the deployment of the new version.

A key factor in the Versioning Policy Statements is the concept of “backward compatibility”. The service contract represents an agreement between service provider and service consumers. A backward compatible change is any change to the service implementation after which existing.

Table 5 - Versioning Approach

Policy Specification/Standard/Guideline	Web Service Versioning	
	Notes	Required/Recommended/Prohibited
The versioning identifier scheme will use lower case “v” followed by a sequential version number as an element of the namespace	“v1”, “v2”, etc.	Recommended
Any non-backwards compatible change to the service contract requires a new version identifier.		Required
The version identifier included in the endpoint URL must be unique across versions of the service that are NOT backwards compatible.	The version identifier included in the endpoint URL may be reused across versions of the service that are backwards compatible.	Required
A version identifier must be included in the WSDL namespace.	<definitions targetNamespace=”http ... /v2” >	Required

11-1. Versioning Policies

Services and supporting artifacts (Service Assemblies) will use a versioning comprised of three types of Updates

- Service Updates which require a change to the Technical Service Contract, and also require changes by consumers (i.e., not backward-compatible) shall be categorized as Major updates.
- Service Updates which require a change to the Technical Service Contract, but do not require changes by any known consumers (i.e., backward-compatible) shall be categorized as Minor updates.
- Service Updates which do not require a change to the Technical Service Contract (i.e. bug fixes, or Quality of Service enhancements) shall be categorized as Revision updates.

The versioning shall follow the convention of “v<major version>,<minor version>,<revision version>.” Initial production releases of Service Assemblies will be labeled. “v1.0.0.”

Example values are:

- V1.0.0 – The initial production version of a service
- V1.0.1 – A revision update has been implemented
- V2.1.3 – A Major Update, a Minor Update, and a 3 Revision Updates have been implemented.

DRAFT

PART 12 SECURITY

As described in documents such as the Electronic Health Record System Functional Model (EHR-S FM) and the Health Information Technology Standard Panel (HITSP) specifications, security should minimally address the following key principles described in Table 6 below.

12-1. Security Considerations

Table 6 – Security Principles for iEHR

Principle	Description
Authentication	Authentication requires that the identities of participants in an exchange be verified.
Authorization	Authorization requires that an exchange of information has been approved (e.g., an individual has the rights to review a patient record)
Confidentiality	Confidentiality demands that unauthorized individuals are not allowed to read messages that are transmitted.
Integrity	Integrity requires that exchanged messages have not been altered.
Non-Repudiation	Non-repudiation requires that an interaction is not deniable after the fact.
Availability	Availability in a security context implies that a system is hardened against attacks that render the system unable to serve its function.
Auditing	Due to the sensitive nature of health care information, auditing of users, connections, and activities is required, for instance Nationwide Health Information Network (NwHIN) and Audit Trail and Node Authentication (ATNA)- related auditing [1,2]

MHS and VA currently have security solutions in place that address these requirements. However, as mentioned earlier, MHS and VA use multiple solutions for different applications that are not necessarily congruent. These include the use of multiple instances of Active Directory (AD) and multiple AD technologies (e.g., Snareworks for AHLTA) as well as the Identity Authentication Service (iAS) with its own directory services.

The MHS OCIO Policy 11-001 mandates the use of the MHS Joint Active Directory (JAD) for all applications excluding those mentioned as exceptions [3]. JAD is a centralized solution to the pre-existing 11 AD instances that were being used. JAD receives updates from each Service (i.e., Army, Navy, and Air Force) and synchronizes with the VA Active Directory Instances. JAD associates individuals with security groups and provides authentication and authorization information based on those groups. As of this writing, JAD among other Identity Management solutions are under consideration for iEHR.

12-1.1 Additional Access Control Topics

There are two additional security issues that need to be considered as part of Governance of the SOA.

The first is the use of Security Assertions Markup Language (SAML) assertions. DoD security reference architectures stress a migration away from protocols such as direct use of Lightweight Directory Access Protocol (LDAP) and towards SAML [4]. This can be handled by ensuring that AD and other Directory Service instances are hidden behind SAML services.

The second is the use of Attribute Based Access Control (ABAC). Currently, the MHS and VA systems primarily use Role Based Access Control (RBAC) with some attributes being included in certain security tokens. The DoD Privilege Management Roadmap discusses moving towards ABAC using Identity and Access Management (IdAM) suites [4]. Consequently, the iEHR must have plans in place for ABAC to support DoD defined attributes as well as MHS and VA specific attribute extensions.

Taken together, SAML, ABAC, and IdAM should be accounted for. The Layer 7 Gateway will act as a point of implementation for RBAC and ABAC. Using Layer 7, SAML and Extended Access Control Markup Language (XACML) can be supported. Layer 7 supports multiple access control mechanisms and

can act as a Security Token Service (STS) when needed. In conjunction with the Gateway, additional IdAM controls may be required. In the presence of a substantial commitment to AD at DoD and VA, the iEHR will likely consider IdAM solutions that consume AD.

DRAFT

12-2. Security Policies

- Programs shall implement security consistent with:
 - NIST Special Publication 800-95 Guide to Secure Web Services [NIST800-95].
 - Extensible Access Control Markup Language (XACML)
 - Security Assertion Markup Language (SAML v 2.0)
 - IETF RFC 2459: PKIX Profile
 - IETF RFC 3280: X.509 PKI certificate and certificate revocation list (CRL) profile, X.509v3 certificates and CRL v1
 - W3C XML Signature
 - Secure Sockets Layer v3 (SSLv3)
 - IETF RFC 2246 Transport Layer Security (TLSv1.1 or higher)
 - SOAP based Services shall use:
 - OASIS WS-Security 1.1
 - WS-I Basic Security Profile version 1.1
 - RESTful Services shall comply with the following:
 - The same security mechanisms as used in DoD and VA web applications will apply to RESTful Services.
 - All RESTful Services will use Security Frameworks accepted by the CoE. No custom made Security Frameworks will be supported.
 - Hash-based message authentication (HMAC) will be used whenever possible.
 - IETF RFC 2560: Online Certificate status protocol (OCSP) – certificate validation
 - IETF RFC 2251: Lightweight Directory Access protocol (LDAP) v3

12-3. References

1. Audit Log Query Service, Nationwide Health Information Network, v1.3.1. 2009.
2. Audit Trail and Node Authentication, IHE, [http://wiki.ihe.net/index.php?title=Audit Trail and Node Authentication](http://wiki.ihe.net/index.php?title=Audit_Trail_and_Node_Authentication)
3. MHS-OCIO Policy 11-001. Military Health Systems 2010.
4. Department of Defense Privilege Management Roadmap by The Office of the Assistant Secretary of Defense for Networks and Information Integration / DoD Chief Information Officer; 6 January 2010.

DRAFT

PART 13 EXTERNAL GOVERNANCE DEPENDENCIES

13-1. Business Governance

Business Governance corresponds to the policies and processes that define how the organization is run. It is the activities and policies required by the executive level and their implications at the project level. In the absence of well-defined business governance, Business Processes will be ad hoc and not amenable to structured implementations in terms of the SOA. Consequently, the ROI of the SOE is adversely affected.

DRAFT

13-2. Information Governance

Information Governance corresponds to the processes and policies for managing data and its associated context. The absence of well-defined syntactic and semantic data, along with authoritative data sources implies that even with a modular, agile SOA, the information propagated will not contribute to an improved ROI.

DRAFT

13-3. Infrastructure Governance

Infrastructure Governance corresponds to the processes and policies that manage the software and hardware that enable the SOA. A low maturity level for Infrastructure Governance compromises the maturity level of the SOA. The SOA may provide rich functionality, but will not be effective if issues associated with networks and hardware does not allow that functionality to be optimally leveraged.

DRAFT

PART 14 IEHR SOA COMPLIANCE CRITERIA

IEHR SOA Compliance Criteria establishes minimum compliance criteria of services and service oriented architecture in order to assist iEHR investment decision-makers, program managers and system developers in ensuring alignment of iEHR programs, projects, initiatives or investments

The iEHR SOA Compliance model is adopted from level 4 of The Open Group Service Integration Maturity Model, Version 2 Technical Standard (OSIMM) [1]. Level 4 indicates the organization has matured to the point of supporting and enabling the development of systems in the form of services. OSIMM identifies 7 dimensions or system views through which to analyze how level 4 is implemented by the organization. A relevant excerpt of OSIMM is provided in Part 23. In addition to OSIMM criteria, specific compliance criteria within each dimension are adopted from VA and DoD artifacts [2,3]. The iEHR SOA Compliance model is mapped to PMAS milestones as listed in Part 23.

In order for a compliance statement to be considered satisfied, the compliance statement must be addressed by one or more of the PMAS artifacts, from any of the PMAS milestones associated with the statement. This section provides a convenient and authoritative “checklist” of the iEHR SOA compliance criteria. This section details WHAT the criteria is. HOW to meet iEHR SOA Compliance criteria is defined first elsewhere in this volume, then secondly in industry best practices.

Service meta-data description criteria are specified at a granular level of detail. Within the SOA Suite, WSRR registry, the service meta-data capture the state and outcomes of service development life-cycle project reviews and gates. Compliance scripts are written within WSRR which query project review outcomes to assist in governance compliance determinations.

This section should serve not only as a sort of compliance checklist, but also as an indicator as to what should be addressed in PMAS documentation.

14-1. Business Dimension: Componentized Business Provides and Consumes Services

Table 7 – Business Dimension

PMAS Milestone	Compliance Statement
0,1	The service development project has established and documented its iEHR business justification thorough established means such as BJP, CARD, etc.
0,1	The business justification includes support of health care services for veterans, soldiers and their families.

14-2. Organization and Governance Dimension: Emerging SOE Governance

Table 8 – Organization and governance Dimension

PMAS Milestone	Compliance Statement
1	The service complies with SOE CoE Governance Policies. Areas of non-compliance and waiver are identified
1	A plans is developed to bring non-compliant aspects of service into compliance, or seek necessary waivers from the iEHR SOE CoE
1	The service identified for consumption or development is selected from the iEHR SOA Service Catalog
1	A proposed service addition or change to the iEHR SOA Service Catalog is approved through the iEHR SOA CoE Governance process.
1,2,3	A Service Owner is identified for service development

DRAFT

14-3. Method Dimension: Service-Oriented Modeling

Table 9 – Method Dimension

PMAS Milestone	Compliance Statement
0,1,3	The service complies with business architecture standards
1,3	The service development process design model(s) adopt a service-oriented approach
1,3	The service development process identifies applicable Service Lifecycle Management (SLM) Phase(s).
1,3	The service development process meets entry and exit criteria for each SLM phase
1,3	The service development process specifies the standard iEHR testing framework
1,3	The service development process designs unit tests for all service operations as well as service component functions and public methods
1,3	The service development process automates unit tests.
1,3	The service development process and service execution leverages tools of the iEHR SOI
1,3	Services are registered within the WSRR
1,3	Service development adopts CoE Guidance
1,3	Services implement SOA methods, practices, principles and patterns

14-4. Application Dimension: Service Design

Table 10 – Application Dimension

PMAS Milestone	Compliance Statement
1,3	Services use standardized service contracts
1,3	Services are loosely coupled
1,3	Service apply appropriate level of abstraction
1,3	Services are reusable
1,3	Services are stateless
1,3	Services are discoverable
1,3	Services implement SOA CoE design patterns
1,3	Service leverage the ESB for application integration
1,3	Services implement SOA CoE common exception handling
1,3	Services scale horizontally across additional commodity processors without code changes
1,3	The SOA performance requirements are identified
1,3	The SOA performance requirements are met
1,3	The Service SLA requirements are identified
1,3	The Service SLA requirements are met

14-5. Architecture Dimension: Emerging SOA

Table 11 – Architecture Dimension

PMAS Milestone	Compliance Statement
1,3	Services are layered with each service operating primarily within a designated layer.
1,3	Services implement separation between presentation, business logic and data access layers.
1,3	The communication between the layers happens via loosely coupled interface components
1,3	The service application logic access and manage data via a data access layer or a data access service instead of directly accessing the data.
1,3	Data service implement the data adapter pattern separating the service interface from the data adapter which handles communication with the data source.
1,3	Services appropriately implement data virtualization
1,3	Services implement iEHR security and trust model
1,3	Services implement iEHR standard logging
1,3	Services appropriately implement mediation patterns for information exchange, translation and transformation

14-6. Information Dimension: Information as a Service

Table 12 – Information Dimension

PMAS Milestone	Compliance Statement
1,3	Data is not permanently stored on end user devices?
1,3	Transient data such as cookies are purged at the end of the user session?
1,3	Permanent storage uses enterprise class data store software and hardware?
1,3	Service comply with SOE CoE metadata guidance
1,3	Use of iEHR SOA Data Services
1,3	Applicable iEHR data standards and information models are employed (e.g. RLUS, CIIF, VDR, VPR, CIMI, CLIM, CTS, EHR-S FIM, FHIM, HDD, RIM, etc.)
1,3	data sources are authoritative
1,3	A Common Information Model is adopted
1,3	The Common Information Model meta-data registry is leveraged
1,3	Procedures and permissions are established to ensure local copies of authoritative data are kept in sync
1,3	Data access is logged
1,3	Service are identified that handle Personally Identifiable Information (PII) or Personal Health Information (PHI)
1,3	Means are implemented to handle and protect PII and PHI consistent with law and iEHR policy

14-7. Infrastructure and Management Dimension: Project-based SOA Environment

Table 13 – Infrastructure and Management Dimension

PMAS Milestone	Compliance Statement
1,3	Infrastructure meets federal Cloud First policy and implements appropriate cloud deliver model of Infrastructure as a Service (IaaS) or Platform as a Service (PaaS) or Software as a Service (SaaS)
1,3	The chosen cloud deliver model meets FedRAMP and NIST standards
1,3	The user Identity Management and Authorization requirements are identified
1,3	Virtualization is employed to abstract services from the hardware layer
1,3	Service load testing is implemented to determine infrastructure capacity requirements
1,3	Services are monitored for performance, availability and SLA compliance
1,3	Disaster recovery standards are defined and implemented
1,3	Backup and restore standards are defined and implemented
1,3	Information Assurance standard are defined and implemented
1,3	Services utilize designated secure message and access paths.
1,3	Service implementation technologies comply with the iEHR Technical Reference model

DRAFT

14-8. References

1. The Open Group Service Integration Maturity Model. Open Group. 2009.
2. Department of Veterans Affairs, One-VA Enterprise Architecture, Enterprise Technical Architecture.
3. Wisnosky, et al, DoD Business Mission Area Service-Oriented Architecture to Support Business Transformation, Software Engineering Technology, October 2008.

DRAFT

PART 15 SLM ENTRY AND EXIT CRITERIA

Each SLM phase has entry and exit criteria. The following table describes the criteria for the phases.

Table 14 – SLM Entry and Exit Criteria

SLM Phase	Description	Entry Criteria	Exit Criteria
Inception	Initial set of iEHR SOA Software Services; or a newly proposed Service	Any iEHR organization may propose a Service by providing a description and justification. SOE CoE Acceptance Proposal is complete	Service Owner Assigned. Service Acquisition Approach and Developer Approved. SOE CoE Governance Approval. Service Documentation Template completed
Design	Service Developer has begun design work	Service Development Plan submitted	Service Definition Complete. Performance goals/thresholds established. Service Documentation Template (Updates). Service Design Document. Test Plan for Services
Construction	Coding under way	Funding, Software Development and Project Management Teams in place	Service Specification Complete. Development Service Registry Information posted. Code Quality Verified
Test	Perform Service Development and Integration Testing	Test Plan(s) developed. Test Environment established and approved	All Test Phases Complete. QoS/SLA established. Service Quality Verified
Deployment	Installing and Validating a Service in its production runtime environment	Build and Deployment Plan developed. Production and Operational Support Documentation available. Service Configuration documented and controlled	Service Installed at one or more production nodes. Runtime management tools configured. Service Registry updated w/operational data.SAT complete; Production and CoE approvals in place
Operation	Activation, Runtime Monitoring and Support, SLA Enforcement and Version Management	Service activated in production. QoS/SLA/Performance Monitoring in place. Versioning/Change Mgmt in effect	Service End-of-Life justification approved. Replacement Service(s) approved
Depracation	Alert Service Consumers and manage Transition Period prior to Service Termination	Service Consumer impact evaluation initiated	Service Retirement/Transition Plan developed
Retirement	Service removed from production environment and alternate service(s) activated where applicable	Service Consumers ready to employ new/alternate service(s). Alternate service(s) in Operation Phase	Service termination date approved and synchronized, as needed, to availability of production-ready alternate service(s) with adequate capacity and performance

PART 16 SLM CHECKLISTS

This Section contains checklists for each of the Service Lifecycle Phases. These checklists should be used in order to verify that all entry and exit criteria have been met for each phase.

SLM Inception Phase Checklist

Table 15 – SLM Inception Phase Checklist

Project Name:	<<Enter project name>>	Module/File Name:	<<Enter the module Name>>			
Reviewer	<<Enter reviewer name>>	Date	<<Enter the date of review>>			
Verified By:						
#	Item/Description	Evidence/Comments	Status			Date Verified
			Y	N	N/A	
Identify - Involves identifying Service candidates that are aligned with business needs and Consumer requirements						
1.	Is the business requirement for the Service creation documented?					
2.	Has the iEHR IPO enterprise Service repository been searched to find out if a Service already exists that implements similar business functionality?					
3.	Have the 'SOA Service identification guidelines' been followed in identification of the candidate Services?					
4.	Has the Service Documentation Template been updated? (The Service Identification, Operations and NFR tabs should be completely filled in.)					
CoE Initial Review - This activity involves review and approval of the Service Documentation by the CIO MB						
5.	Has the CoE reviewed the Service Documentation Template?					
6.	Have all the CoE comments/suggestions been incorporated in the Service Documentation Template?					
7.	Did the CoE approve of this Inception phase?					
TWG Initial Review - This review involves validating the need for creating the Service and identifying other potential Consumers of the Service						
1.	Has the TWG reviewed the Service Documentation Template?					
8.	Have all the TWG comments/suggestions been incorporated in the Service Documentation Template?					
9.	Did the TWG approve of this Inception phase?					
Plan - Involves initiation and planning for the design and development of the Service offering						
1.	Verify completion of the Service documentation template and its approval by TWG and CoE.					
10.	Have all the necessary resources been acquired to develop and implement the Service?					
11.	Has proper project Management been established to monitor the progress of the Service development and					

	implementation?					
12.	Has Lessons Learned been documented for this phase? Check for evidence on completion of Lessons Learned documentation.					
Exit Criteria – Exit Criteria for Inception Phase						
1.	Has the following Inception Phase outputs been completed, approved and checked into document repository? <ul style="list-style-type: none"> Service Documentation Template 					

SLM Design Phase Checklist

Table 16 – SLM Design Phase Checklist

Project Name:	<<Enter project name>>	Module/File Name:	<<Enter the module Name>>			
Reviewer	<<Enter reviewer name>>	Date	<<Enter the date of review>>			
#	Item/Description	Evidence /Comments	Status			Date Verified
			Y	N	N/A	
Design - Involves design of the Service in accordance with the functional and non-functional requirements						
1.	Has the Service Design document been created?					
13.	Have the EA Service Design principles been referenced and adhered to?					
TWG Design Review - This review involves validating the need for creating the Service and identifying other potential Consumers of the Service						
1.	Has the TWG reviewed the Service Design document?					
2.	Have all the TWG comments/suggestions been incorporated?					
14.	Did the TWG approve of this Design phase?					
15.	Has Lessons Learned been documented for this phase? Check for evidence on completion of Lessons Learned documentation.					
Exit Criteria – Exit Criteria for Design Phase						
1.	Has the following Design Phase outputs been completed, approved and checked into document repository? <ul style="list-style-type: none"> Service Documentation Template (Updates) Service Design Document Test Plan for Services 					
2.	Is the Service information available in the iEHR IPO enterprise Service repository?					

SLM Construction Phase Checklist

Table 17 – SLM Construction Phase Checklist

Project Name:	<<Enter project name>>	Module/File Name:	<<Enter the module Name>>			
Reviewer	<<Enter reviewer name>>	Date	<<Enter the date of review>>			
#	Item/Description	Evidence/Comments	Status			Date Verified
			Y	N	N/A	
Build - Involves code development per the design to ensure compliance with the design						
1.	Does the program follow coding standards as defined in the iEHR IPO Quality Assurance and Surveillance Plan (QASP)? Check if the appropriate code standards used for consistency.					
2.	Does the code follow the project coding naming conventions? Check that the coding naming conventions have been followed. Variable naming, indentation, and bracket style should be used.					
3.	Did the program perform code peer reviews?					
4.	Has the code been version controlled and checked into a version control system?					
TWG Build Review - Involves review of the Service implementation documentation						
1.	Has the TWG reviewed all the Service Development?					
2.	Have all the TWG comments/suggestions been addressed?					

SLM Testing Phase Checklist

Table 18 – SLM Testing Phase Checklist

Project Name:	<<Enter project name>>	Module/File Name:	<<Enter the module Name>>			
Reviewer	<<Enter reviewer name>>	Date	<<Enter the date of review>>			
#	Item/Description	Evidence/Comments	Status			Date Verified
			Y	N	N/A	
Build - Involves code development per the design and testing activities to ensure compliance with the design						
1.	Has the Test Plan been created?					
2.	Have unit test cases and test scripts that encompass functional and non-functional requirements been created?					
3.	Has unit test been conducted on the code and the results documented? Check if the developer has unit tested the code before sending it for review. All the limit cases should have					

	been tested.					
4.	Has the code been subjected to the iEHR IPO Software Code Quality Check (SCQC) review? Check if the developer has submitted the code to SCQC for review.					
5.	Has all the SCQC findings addressed by the developer? Check for evidence that all the SCQC findings have been addressed.					
6.	Have integration test cases and test scripts been created that encompass functional and non-functional requirements?					
7.	Has integration test been conducted and the results documented?					
8.	Has the program completed the Development and Integration Test (DIT) processes and procedures prescribed by the iEHR IPO? Check for evidence that the DIT findings have been addressed.					
9.	Has the program completed the System Integration Test (SIT) processes and procedures prescribed by the iEHR IPO? Check for evidence that the SIT findings have been addressed.					
10.	Has the program completed the System Qualification Test (SQT) processes and procedures prescribed by the iEHR IPO? Check for evidence that the SQT findings have been addressed.					
TWG Build Review - Involves review of the Service implementation documentation						
1.	Has the TWG reviewed all the Testing Documentation?					
2.	Have all the TWG comments/suggestions been addressed?					
3.	Did the TWG approve of the Build and Deployment to the Acceptance test environment?					
TWG Test Review - Involves review of the Service implementation documentation						
1.	Has the TWG reviewed all the Test results?					
2.	Have all the TWG comments/suggestions been addressed?					
3.	Did the TWG approve of the Test results?					

SLM Deployment Phase Checklist

Table 19 – SLM Deployment Phase Checklist

Project Name:	<<Enter project name>>	Module/File Name:	<<Enter the module Name>>			
Reviewer	<<Enter reviewer name>>	Date	<<Enter the date of review>>			
#	Item/Description	Evidence/ Comments	Status			Date Verified
			Y	N	N/A	
Deploy - Involves deploying the services for use by the consumers						
1.	Has the operators guide been created?					
2.	Has the Service build and deploy documentation been created?					
TWG Build Review - Involves review of the Service implementation documentation						
1.	Did the TWG approve of the Build and Deployment to the Acceptance test environment?					
Deploy - Involves deploying the Service in the runtime environment						
1.	Has the iEHR Operations Team provided its approval of the build and deployment documentation?					
2.	Has the code been deployed to test environment?					
3.	Have sample tests been performed to ensure the deployed code works as desired?					
4.	Has the code been checked into the Configuration Management Repository and available for deployment team?					
16.	Has the Service registry been updated with the Service access URL?					
Validate - Involves verification and validation that the Service works properly before activating for consumption by the Service Consumers in the production environment						
1.	Has the program and the Service Consumers completed the System Acceptance Test (SAT) processes and procedures prescribed by the iEHR IPO? Check for evidence that the SAT findings have been addressed.					
2.	Has the program obtained approval for Enterprise wide deployment? Check for evidence of approval Enterprise wide deployment.					
TWG Test Review - Involves review of the Service implementation documentation						
1.	Have all the TWG comments/suggestions been addressed?					
2.	Did the TWG approve of the Service code deployment to production environment?					

SLM Operational Phase Checklist

Table 20 – SLM Operational Phase Checklist

Project Name:	<<Enter project name>>	Module/File Name:	<<Enter the module Name>>			
Reviewer	<<Enter reviewer name>>	Date	<<Enter the date of review>>			
#	Item/Description	Evidence/Comments	Status			Date Verified
			Y	N	N/A	
Manage - Involves post deployment Management activities including change Management						
1.	Has the Service been deployed to production and activated?					
2.	Have all the Service runtime monitors and alerts been activated? This includes all non-functional requirements and SLAs					
3.	Are all Service runtime statistics been captured and reported to the Service owners and Service Consumers?					
4.	Are all non-compliances been reported to the Service owners?					
5.	Are Service change Management processes implemented and available?					
17.	Has Lessons Learned been documented for this phase? Check for evidence on completion of Lessons Learned documentation.					

SLM Deprecation Phase Checklist

Table 21 – SLM Deprecation Phase Checklist

Project Name:	<<Enter project name>>	Module/File Name:	<<Enter the module Name>>			
Reviewer	<<Enter reviewer name>>	Date	<<Enter the date of review>>			
#	Item/Description	Evidence/Comments	Status			Date Verified
			Y	N	N/A	
TWG Deprecate Review - Involves review and approval of the Service deprecation plan						
1.	Has the Service deprecation plan been created? The Service deprecation plan should at minimum include business reasons for deprecation, date of deprecation and retirement. The plan should ensure minimal disruption for the Consumers and enough time for migration to the new/alternate Service.					
2.	Has the CoE & TWG approved Service deprecation plan?					
Retired - Involves notification to the Service Consumers and decommissioning the Service						
1.	Have the Service Consumers been alerted to stop using the Service?					

2.	At the end of the deprecation period, has the Service been deactivated and removed from the iEHR IPO enterprise Service repository?					
3.	Has the Service been decommissioned?					
4.	Has the Service registry been updated for Service deprecation?					
5.	Has Lessons Learned been documented for this phase? Check for evidence on completion of Lessons Learned documentation.					

SLM Retired Phase Checklist

Table 22 – SLM Retired Phase Checklist

Project Name:	<<Enter project name>>	Module/File Name:	<<Enter the module Name>>			
Reviewer	<<Enter reviewer name>>	Date	<<Enter the date of review>>			
#	Item/Description	Evidence/Comments	Status			Date Verified
			Y	N	N/A	
TWG Deprecate Review - Involves review and approval of the Service deprecation plan						
1.	Has the Service retirement plan been created?					
2.	Has the CoE & TWG approved retirement plan?					
Retired - Involves notification to the Service Consumers and decommissioning the Service						
1.	Has the Service been decommissioned?					
2.	Has the service portfolio been updated?					

PART 17 SLM WAIVER FORM

Table 23 – SLM Waiver Form

Project Name	<<Enter Project Name>>	Service Name	<<Provide Name Of The Service Being Built>>	Date	<<Provide Name Of The Service Being Built>>
Requestor Name	<<Enter Waiver Requestor's Name>>	Requestor Contact Phone	<<Enter Waiver Requestor's A Number>>	Requestor Email ID	<<Enter Waiver Requestor's Email Id>>
Service Sponsor Name	<<Enter Service Sponsor's Name>>	Service Sponsor Contact Phone	<<Enter Service Sponsor's Contact Phone Number>>	Service Sponsor Email ID	<<Enter Service Sponsor's Email Id>>
#	Criteria	Evidence/Comments			
	Describe in detail the reason for requesting the waiver				
	Which phase of SLM is the waiver requested for?				
	Has the Service Sponsor approved the waiver request?				
	Have the project stakeholders been notified about the waiver request?				

Sponsor Review Results:

Table 24 – Sponsor Review Results

Reviewer Name(s)	<<Enter Reviewer Names>>	Review Date	<<Enter Date Of Review>	Review Results	<<Enter Approved/Denied/Conditional Approval>>
Review Comments					

TWG Review Results:

Table 25 – TWG Review Results

Reviewer Name(s)	<<Enter Reviewer Names>>	Review Date	<<Enter Date Of Review>	Review Results	<<Enter Approved/Denied/Conditional Approval >>
Review Comments					

CoE Review Results:

Table 26 – CoE Review Results

Reviewer Name(s)	<<Enter Reviewer Names>>	Review Date	<<Enter Date Of Review>	Review Results	<<Enter Approved/Denied/Conditional Approval >>
Review Comments					

IPO Signatory (if needed)

Date

PART 18 SERVICE DOCUMENTATION TEMPLATE

18-1. Service Documentation - Blank

Service Identification

Please enter Service Identification information below:

Table 27 – Service Identification

Identifier #	
Name	
Description/Purpose	
Business Function(s)	
Categorization	
Version Information	
Precondition for Service Invocation	
Post Conditions after Service Completion	
Network Segments	
Network Domains	<input type="checkbox"/> .mil <input type="checkbox"/> .com <input type="checkbox"/> .gov

- Yes No: Has an ICD been provided to the IPO SOE COE and to Harris?
- Yes No: Has WSDL been provided?
- Yes No: Has XML Schema been provided?
- Yes No: Have Policy Assertions been provided?

Inputs/Outputs

Please enter service inputs below.

Table 28 – Service Inputs

Operation	Input			
	Parameter	Type	Description	Required?

Please enter service outputs below.

Table 29 – Service Outputs

Operation	Output			
	Parameter	Type	Description	Required?

Exception / Error Codes

Please enter error codes below.

Table 30 – Exception / Error Codes

Error Code	Description

Quality of Service (QoS) Requirements

Please enter requirements below.

Table 31 – QoS Requirements

ID	QoS	Requirement	Description
Q1	Service Response-time		In the absence of network or data store problems, the Service Response-time shall be such that the Total Response-time requirement is met. Total Response-time = Service Response-time + Data Store Response-time + Network Response-Time
Q2	Availability		The service will be accessible to Consumers during the date and time intervals defined by the SLA.
Q2a	Availability Classification		A1 Recoverable (Lowest) A2 Cold Standby A3 Hot Standby A4 Fail Safe (Highest)
Q2b	Failure Handling		Methodology or process for failure and recovery
Q3	Throughput		The service will support the number of transactions per unit time (e.g., transactions/minute) as defined in the SLA
Q4	Data Volume		The service shall support the response-time, availability, and throughput given the maximum message size Data Volume = Throughput*Maximum_Message_Size
Q5	Security		The service shall conform to the security constraints of MHS while satisfying other QoS requirements

Access Information

Please enter access information below.

Table 32 – Access Information

WSDL location	
Endpoint URL	

Ports and Protocols

Please enter information below.

Table 33 – Ports and Protocols

Port Numbers	Protocols	Message Types (XML, SOAP, HL7, MLLP, HTTP, JMS, DICOM, TCP/IP, (S)FTP, other)	Message Size per Message Type (Min, Nominal, Max Packet Size)	Transactions per Second per Message Type

Use Case (Optional)

Please enter use case steps below.

Table 34 – Use Case (Optional)

Precondition	Actor	Action Taken	Outcome / Post Condition

Service Invocation Example

Please enter service request and response details below.

Table 35 – Service Invocation

Request	Response

18-2. Service Documentation - Completed

Please enter Service Identification information below:

Table 36 – Service Identification

Identifier #	XX-XXX-XX
Name	Provider Lookup
Description/Purpose	The service provides search capability to find healthcare practitioners and MTFs within MHS
Business Function(s)	For Pharmacy, Lab, Immunization services
Categorization	MHS Global/Core Services/Provider
Version Information	1.0
Precondition for Service Invocation	Valid patient ID, and user authorization
Post Conditions after Service Completion	Returns list of providers for pharmacy, lab, immunization services
Network Segments	DISA Montgomery (MGM) Region, DoD MAAG MESA San Antonio, NIPRNET MNS VPN Cloud
Network Domain	<input checked="" type="checkbox"/> .mil <input type="checkbox"/> .com <input type="checkbox"/> .gov

- Yes No: Has an ICD been provided to the IPO SOE COE and to Harris?
- Yes No: Has WSDL been provided?
- Yes No: Has XML Schema been provided?
- Yes No: Have Policy Assertions been provided?

Inputs/Outputs

Please enter service inputs below.

Table 37 – Service Identification Inputs

Operation	Input			
	Parameter	Type	Description	Required?
Find Doctor	FirstName	Text	First name of the provider	Y
	LastName	Text	Last name of the provider	Y
	PracticeName	Text	Name of provider's practice	N
Find Hospital	ZipCode	Text	Zip code for searching	Y
	Specialization	Drop-down	Type of the hospital/facility to search	Y
	HospitalName	Text	Name of the hospital	N

Please enter service outputs below.

Table 38 – Service Identification Outputs

Operation	Output			
	Parameter	Type	Description	Required?
Find Doctor	FirstName	Text		Y
	LastName	Text		Y
	Specialization	Text		Y

Operation	Output			Required?
	Parameter	Type	Description	
	Address	Text		Y
	Phone	Text		Y
Find Hospital	HospitalName	Text		Y
	Address1	Text		Y
	Address2	Text		Y
	ContactNumber	Text		Y
	ContactEmail	Text		Y

Exception / Error Codes

Please enter error codes below.

Table 39 – Service Inputs

Error Code	Description
_InvalidRequest	Indicates that the request is considered invalid since it does not match the criteria defined by the service specification.
_DisabledFunction	The function is currently disabled.
_InternalError	There was an internal error.

Quality of Service (QoS) Requirements

Please enter requirements below.

Table 40 – QoS Requirements

ID	QoS	Requirement	Description
Q1	Security	WS-Security	
Q2	Authentication	Username/Password	
Q3	Other Compliance	WS-I Basic Profile	
Q4	Fault Reporting	SOAP-Fault	The SOAP-FAULT should contain the appropriate ErrorCode within the <detail> element of the soap-fault
Q5	Response Time	The response should be within 10ms	
Q6	Availability	Service will be available 24 hours a day 7 days a week	
Q6a	Availability Classification	A2 Cold Standby	A1 Recoverable (Lowest) A2 Cold Standby A3 Hot Standby A4 Fail Safe (Highest)
Q6b	Failure Handling	Retry request after configurable timeout	
Q7	Throughput	Max 25 Service requests per second Nominal 12 requests per second	The service will support the number of transactions per unit time (e.g., transactions/minute) as defined in the SLA
Q8	Data Volume	Data Volume = Max 25 requests * 10KB per second; Nominal 12 requests * 10KB per second Max Data Volume = 250KB per	The service shall support the response-time, availability, and throughput given the maximum message size Data Volume =

ID	QoS	Requirement	Description
		second Nominal Data Volume = 120KB per second	Throughput*Maximum_Message_Size

Access Information

Please enter access information below.

Table 41 – Access Information

WSDL location	http://hostname:port/provider/lookupService?wsdl
Endpoint URL	http://hostname:port/provider/lookupService

Ports and Protocols

Please enter information below.

Table 42 – Ports and Protocols

Port Numbers	Protocols	Message Types (XML, SOAP, HL7, MLLP, HTTP, JMS, DICOM, TCP/IP, (S)FTP, other)	Message Size per Message Type (Min, Nominal, Max Packet Size)	Transactions per Second per Message Type
443, 8443	SOAP/HTTPS	HL7 V2 MLLP	10KB Nominal	Max 25 Service requests per second Nominal 12 requests per second
8080, 5001	HTTP, SSH, TCP/IP	HL7 V2 MLLP	10KB Nominal	Max 25 Service requests per second Nominal 12 requests per second

Use Case (Optional)

Please enter use case steps below.

Table 43 – Use Case

Precondition	Actor	Action Taken	Outcome / Post Condition
Valid user logon and authorization to access provider database			
		Input Doctor Name	Receive doctor practice name, address, phone number
		Input Hospital Name	Receive doctor practice name, address, phone number

Service Invocation Example

Please enter service request and response details below.

Table 44 – Service Invocation

Request	Response
<pre> <soapenv:Envelope xmlns:ser="http://server.domain.com/" xmlns:soapenv="http://schemas.xmlsoap.org/ soap/envelope/"> <soapenv:Header> <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http:// docs.oasis-open.org/wss/2004/01/oasis- 200401-wss-wssecurity- secext-1.0.xsd"> <wsu:Timestamp wsu:Id="Timestamp- 17852335" xmlns:wsu="http:// docs.oasis-open.org/wss/2004/01/oasis- 200401-wss-wssecurity- utility-1.0.xsd"> <wsu:Created>2011-06- 30T15:10:12Z</wsu:Created> <wsu:Expires>2011-06- 30T15:15:12Z</wsu:Expires> </wsu:Timestamp> <wsse:UsernameToken wsu:Id="UsernameToken-12478552" xmlns:wsu="http://docs.oasis- open.org/wss/2004/01/oasis-200401-wss- wssecurity-utility-1.0.xsd"> <wsse:Username>wball</wsse:Username> <wsse:Password Type="http://docs.oasis- open.org/wss/ 2004/01/oasis-200401-wss-username-token- profile- 1.0#PasswordText">wball</wsse:Password> <wsse:Nonce>VpJSJ78ArCIR4sWMbmSKJA ==</wsse:Nonce> <wsu:Created>2009-04- 14T15:10:12.562Z</wsu:Created> </wsse:UsernameToken> </wsse:Security> </soapenv:Header> <soapenv:Body> <FindDoctor> <FirstName>John</FirstName> <LastName>Doe</LastName> </FindDoctor> </soapenv:Body> </soapenv:Envelope> </pre>	<pre> <soapenv:Envelope xmlns:ser="http://server.domain.com/" xmlns:soapenv="http://schemas.xmlsoap.org/soap/ envelope/"> <soapenv:Header> <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http:// docs.oasis-open.org/wss/2004/01/oasis-200401- wss-wssecurity- secext-1.0.xsd"> <wsu:Timestamp wsu:Id="Timestamp- 17852335" xmlns:wsu="http:// docs.oasis-open.org/wss/2004/01/oasis-200401- wss-wssecurity- utility-1.0.xsd"> <wsu:Created>2011-06- 30T15:12:12Z</wsu:Created> <wsu:Expires>2011-06- 30T15:12:12Z</wsu:Expires> </wsu:Timestamp> <wsse:UsernameToken wsu:Id="UsernameToken-12478552" xmlns:wsu="http://docs.oasis- open.org/wss/2004/01/oasis-200401-wss- wssecurity-utility-1.0.xsd"> <wsse:Username>wball</wsse:Username> <wsse:Password Type="http://docs.oasis- open.org/wss/ 2004/01/oasis-200401-wss-username-token- profile- 1.0#PasswordText">wball</wsse:Password> <wsse:Nonce>VpJSJ78ArCIR4sWMbmSKJA==</ wsse:Nonce> <wsu:Created>2009-04- 14T15:10:12.562Z</wsu:Created> </wsse:UsernameToken> </wsse:Security> </soapenv:Header> <soapenv:Body> <FindDoctorResponse> <FirstName>John</FirstName> <LastName>Doe</LastName> <Specialization>Pediatrician</Specialization> <Address>141 main street, falls church, va</Address> <Phone>571-222-2222</Phone> </FindDoctorResponse> </soapenv:Body> </pre>

Request	Response
	</soapenv:Envelope>

DRAFT

PART 19 TAXONOMY REGISTRATION TEMPLATE

Table 45 – Taxonomy Registration Template

Taxonomy Identifier	<The Taxonomy Namespace or Identifier - Required>			
Taxonomy Title	<Taxonomy Name – Required>			
Taxonomy Creator	<Name of the Developer of the Taxonomy – Required>			
Taxonomy Description	<Brief accounting of the taxonomy – Required>			
Purpose	<The reason for proposing the taxonomy >			
Source Name	<If the taxonomy was derived from another taxonomy – specify the originating taxonomy – Conditionally Required>			
Source URI	<Location of the Source Taxonomy>			
Overview of Structure	<Plain language or diagrammatic representation of the taxonomy hierarchy>			
Rights and Licensing	<Intellectual or Copyright considerations as well as licensing>			
Terms and Conditions	<Any conditions of use e.g. legal, financial requirements>			
Steward Name	<Point of Contact for management of taxonomy and metadata – Required>			
Steward Organization	<Organization associated taxonomy management and metadata – Required>			
Submitter Name	<Point of Contact for taxonomy submission>			
Submitter Organization	<Organization submitting the taxonomy>			
Submission Date	<Date of submission to the CoE>			
Taxon ID	Taxon Name	Parent Taxon	Assignable	Description
<An Identifier for the Taxon>	<The Name associated with the Taxon>	<The parent node>	<Is this node assignable to a web service or does it exist purely as a grouping of other nodes -- (Yes/No)>	<Brief accounting of taxon>

PART 20 NAMESPACE REGISTRATION TEMPLATE

Table 46 – Namespace Registration Template

Name (URN)	<The namespace as specified in terms of the urn>
Description	<A brief accounting of the namespace>
Steward Name	<Point of Contact responsible for maintaining the namespace>
Steward Organization	<Organization associated with Point of Contact>
Steward Email	<Email address for the Point of Contact>
Submitter Name	<Point of Contact for party requesting namespace inclusion>
Submitter Organization	<Organization associated with Point of Contact>
Submitter Email	<Email address for the Point of Contact>
Parent Namespace	<The parent namespace for this namespace specified in terms of urn>
Related Namespaces	<Any related namespaces specified in terms of URI. The relationship must also be provided>
Submission Date	<The Date the namespace request was submitted to the CoE>
References	<Documents providing details on the namespace>

PART 21 ROLES AND RESPONSIBILITIES

- **SOE CoE** – Primarily a SOA Governance Organization, the Service Oriented Enterprise Center of Excellence (SOE CoE) is responsible for the compliance, waiver and communication policies and processes associated with the iEHR SOA, including the iEHR SOI, SOA and SOE domains. A key SOA artifact maintained by the SOE CoE is the iEHR SOA Software Service Registry. The SOE CoE has both a Business Working Group (BWG) and a Technical Working Group (TWG). In addition to the governance roles, the SOE CoE provides guidance, advice and technical assistance for implementation and application of the iEHR SOA throughout the “iEHR Enterprise”.
- **Service Developer** – Responsible for the development of the wrapper around the Commercial off-the-shelf (COTS) or Government off-the-shelf (GOTS) service or Application Programming Interface (API). If an existing or off-the-shelf service is not available, the Service Developer may develop a new service.
- **Service Owner** – An owner is a role assumed by a participant who is claiming and exercising managerial ownership over one or more services throughout the Service Lifecycle (e.g. service identification, maintenance, etc.), whether the participant executes the Service Lifecycle responsibilities directly or through delegation.
- **SOA Suite Development Team** – Responsible to support the Service Developer. Provide software development kits (SDKs) and integration support for developed services.
- **SOA Suite Operation Team** – This team is responsible for connection of applications/services to the Enterprise Service Bus(ESB) after SOA certification process complete. Provide help desk services, capacity planning and management, system updates and migration.

PART 22 OSIMM

22-1. OSIMM Level 4: Service

Composite applications are built from loosely-coupled services. The way that services may be invoked is based upon open standards and is independent of the underlying application technology. Services run on an IT infrastructure that is supported by the appropriate protocols, security mechanisms, data transformation, and service management capabilities. The services may therefore interoperate across all of the parts of the organization and even across different organizations within the eco-system, and are often managed by assigning responsibilities for managing Service-Level Agreements (SLAs) to segments of the organization. The business functionality has been analyzed in detail and is broken down into services residing within a business architecture that ensures that services will interoperate at the business level. In addition, it is possible to define the services via a specification language – such as WSDL or Service Component Architecture (SCA) – that unambiguously defines the operations performed by the service, permitting the construction of a catalog of services. The combination of IT and service architectures permits the construction of systems based upon these services, operating right across the organizations in the ecosystem. However, at this stage the composition of services and flow of control within a composite application are still defined by developers writing bespoke code, rather than by a declarative flow language. This limits the agility of the development of new business processes as services.

DRAFT

22-2. OSIMM 7 Dimensions

22-2.1 Business

The Business dimension is focused on the business architecture; i.e., the organization's current business practices and policies; how business processes are designed, structured, implemented, and executed. The Business dimension also addresses how the cost of IT capabilities is allocated across the enterprise, and how well the IT capabilities support the flexibility of the business, agility, and SLAs. The Business dimension includes IT strategy. And thus includes the necessary value proposition for moving from one maturity level to a higher level maturity level. A discussion of these value propositions are in Benefits of Moving to Higher Maturity Levels.

22-2.2 Organization & Governance

The Organization & Governance dimension is focused on the structure and design of the organization itself and the necessary measures of organizational effectiveness in the context of an SOA and SOA governance. The Organization aspect is focused on organizational structure, relationships, roles, and the empowerment necessary to adopt a service-oriented strategy. This includes the types and extent of skills, training, and education that are available within the organization. Governance is associated with formal management processes to keep IT activities, service capabilities, and SOA solutions aligned with the needs of the business. Governance guides many aspects of the other maturity dimensions, including how management is structured and costs are allocated.

22-2.3 Method

The Method dimension is focused on the methods and processes employed by the organization for its IT and business transformation, and the organization's maturity around the Software Development Lifecycle such as the use of requirements management, estimation techniques, project management, quality assurance processes, design methodologies and techniques, and tools for designing solutions.

22-2.4 Application

The Application dimension is focused on application style, structuring of the application and functional decomposition, re-usability, flexibility, reliability, and extensibility of the applications, understanding, and uniform use of best practices and patterns, whether multiple applications have been created to serve different lines of business with essentially the same functionality, and the availability of enterprise schema and object models.

22-2.5 Architecture

The Architecture dimension is focused on the structure of the architecture which includes topology, integration techniques, enterprise architecture decisions, standards and policies, web services adoption level, experience in SOA implementation, SOA compliance criteria, and typical artifacts produced.

22-2.6 Information

The Information dimension is focused on how information is structured, how information is modeled, the method of access to enterprise data, abstraction of the data access from the functional aspects, data characteristics, data transformation capabilities, service and process definitions, handling of identifiers, security credentials, knowledge management, business information model, and content management.

22-2.7 Infrastructure & Management

The Infrastructure & Management dimension is focused on the organization's infrastructure capability, service management, IT operations, IT management and IT administration, how SLAs are met, how monitoring is performed, and what types of integration platforms are provided.

PART 23 PMAS MILESTONES

Table 47 – PMAS Milestones

#	Milestone	PMAS Milestone Artifacts
0	New Start	Project Charter Business Requirements Document (BRD)
1	Planning	Requirements Specification Document (RSD) Project Management Plan (PMP) Project Schedule Risk Log or Risk Register System Design Document (SDD) Quad Chart Spend Plan (Process Only) Product Evaluation and Decision Analysis (Buy Only) Acquisition Strategy Contract Information Outcome Statement Customer Acceptance Criteria Plan PMAS Readiness Checklist Operational Acceptance Plan (OAP) Confirmation of Release Requirements/Artifacts (ProPath) Submitted Acquisition Package (Virtual Office of Acquisition – VOA) Executive Decision Memorandum (EDM)
2	Provisioning	Contract Award (VOA) Updates to MS1 documents
3	Active	Success Criteria Customer Acceptance Form IPT Charter updates to MS1 documents

DRAFT

PART 24 IEHR SERVICE METADATA

Table 49 contains the metadata associated with a service as defined by VA and the iEHR Program Office. It is meant to be a comprehensive mechanism to capture the attributes of a service.

Each metadata element in Table 49 has the attributes described in Table 48.

Table 48 - Service Metadata Attributes

Column	Column Description
#	A unique identifier number for each service meta-data attribute. Identifier numbers are in dot (.) format to preserve attribute hierarchy.
Attribute	Attribute is the name of the Service Meta-Data property. Attributes are organized into PropertyGroup. Attributes ending in the PropertyGroup suffix are not for the purpose of containing meta-data property values but rather to identify groups of properties. Attribute and sub-attribute relationships are depicted by the dot (.) numbering scheme.
Description	Description is a definition of the Service Meta-Data attribute.
Type	Type is the data type of the attribute.
Life-Cycle Phase	The earliest service life-cycle phase that requires an entry for the attribute. Attributes without an entry in this column are at all times optional. Service Life-Cycle phases are listed below.
Cardinality	Enter multiple if multiple entries for an attribute or attribute group is permitted. No entry indicates a cardinality of single.
Options	Indicates the terms that are permitted as an attribute value.

Table 49 - Service Metadata

#	Attribute	Description	Type	LifeCycle Phase	Cardinality	Options
1.1	ServiceIdentityPropertyGroup	Provides identification of the service	Text			
1.1.1	ServiceIdentity.Name	Descriptive name of the service	Text	Inception		
1.1.2	ServiceIdentity.Description	Description of what the service does	Text	Inception		
1.1.3	ServiceIdentity.Version	The version of the service being described	Text	Deployment		
1.1.4	ServiceIdentity.VersionReplacing	The version of the service replaced by the version described	Text			
1.1.5	ServiceIdentity.Identifier	A unique string, number or identifier for the service	Text	Inception		
1.1.6	ServiceIdentity.URI	A unique URI/URL defining the service	Text	Deployment		
1.1.7	ServiceIdentity.LifeCyclePhase	The lifecycle phase the service is currently in	Text	Deployment		Life-Cycle Phases
1.1.8	ServiceIdentity.ProjectedImageDate	The date the service is forecasted to go into production	date			
1.1.9	ServiceIdentity.ActualDeploymentDate	The date the service actually went into production	date	Deployment		
1.1.10	ServiceIdentity.OperationPropertyGroup	Provides specification of service operation	Text	Design	Multiple	
1.1.10.1	ServiceIdentity.Operation.Name	Name of an operation	Text	Design		
1.1.10.2	ServiceIdentity.Operation.Description	Description of an operation	Text	Design		
1.1.10.1	ServiceIdentity.Operation.InputParameter.Name	Input to the operation, enter as many separate input parameters as required, maps to 1.4.1 ServiceInteraction.StandardRepresentation	Text			
1.1.10.2	ServiceIdentity.Operation.InputParameter.Description	Input to the operation, maps to 1.4.1 ServiceInteraction.StandardRepresentation	Text			
1.1.10.3	ServiceIdentity.Operation.InputParameter.isRequired	Input to the operation, maps to 1.4.1 ServiceInteraction.StandardRepresentation	Boolean			
1.1.10.4	ServiceIdentity.Operation.OutputParameter.Name	Output from the operation, enter as many output parameters as required, maps to 1.4.1 ServiceInteraction.StandardRepresentation	Text	Design		
1.1.10.5	ServiceIdentity.Operation.OutputParameter.Description	Output From the operation, maps to 1.4.1 ServiceInteraction.StandardRepresentation	Text	Design		
1.1.10.6	ServiceIdentity.Operation.OutputParameter.isRequired	Output From the operation, maps to 1.4.1 ServiceInteraction.StandardRepresentation	Boolean			
1.2	ServiceProviderPropertyGroup	Description of the party providing the service	Text			
1.2.1	ServiceProvider.Unit	Service Provider organization	Text	Operation		
1.2.2	ServiceProvider.ContactPerson	Person representing the provider unit	Text			
1.2.3	ServiceProvicer.ContactPerson.email	Email of the contact person	Text	Operation		
1.2.4	ServiceProvicer.ContactPerson.phone	Phone of the contact person	Text			

#	Attribute	Description	Type	LifeCycle Phase	Cardinality	Options
1.2.5	ServiceProvider.ContactPerson.address	physical address of the contact person	Text			
1.3	ServiceFunctionPropertyGroup	Describes the function and operating characteristics of the service	Text			
1.3.1	ServiceFunction.Description	A narrative description of the service for human readers as an aid to understanding	Text			
1.3.2	ServiceFunction.Effect	What happens or results from the service	Text			
1.3.3	ServiceFunction.TechnicalAssumptions	Technical assumptions or physical constraints that must be met to consumer the service.	Text		Multiple	
1.3.4	ServiceFunction.AssociatedPolicies	Policies declared with respect to this service	Text, URI		Multiple	
1.3.5	ServiceFunction.AssociatedMetrics	Metrics and values characterizing operational performance	Text		Multiple	
1.3.6	ServiceFunction.SLANumber	The unique identifier for a SLA contract instance. SLA may be implemented as a collection of policies that are measurable and enforceable at run time. Such implementation requires components to create, maintain, store, find, access and manage policies and contracts.	Text			
1.3.7	ServiceFunction.Source	Source of the service requirements	Text			
1.4	ServiceInteractionPropertyGroup	Sufficient information for service consumer to interact with service	Text			
1.4.1	ServiceInteraction.StandardRepresentation	OWL-S or other standard representation of the interaction description. Used by development tools to read and automatically generate code stubs.	URI			
1.4.2	ServiceInteraction.SupplementalInformation	Supplemental information affecting the interface (e.g. consumers in the US use endpoint x, consumers in Europe use endpoint y)	Text			
1.4.3	ServiceInteraction.InterfaceType	Enables automates searching by interface type such as RESTful, SOAP, JASON,	Text			
1.4.4	ServiceInteraction.ActionModel	Define the actions of the operations	Text			
1.4.5	ServiceInteraction.ProcessModel	Define the sequence dependencies between the service operations	Text			
1.4.6	ServiceInteraction.Vocabulary	Semantics and structure of vocabularies used by this service	XML			
1.4.7	ServiceInteraction.Status	Status of service availability	Text			
1.4.8	ServiceInteraction.Policies	Policy in WS-Policy or other canonical format	Text		Multiple	
1.4.9	ServiceInteraction.Metrics	Identification of available service metrics and means of access. Metrics require components to access,	Text		Multiple	

#	Attribute	Description	Type	LifeCycle Phase	Cardinality	Options
		gather and store metrics				
1.4.10	ServiceInteraction.ServiceConsumed.ServiceIdentity.Identifier	Service this service invokes	Identifier		Multiple	
1.4.11	ServiceInteraction.ServiceConsumedBy.ServiceIdentity.Identifier	Service this service is invoked by	Identifier		Multiple	
1.5	ServiceAccessPropertyGroup	Special conditions required for consuming applications to access the service	Text			
1.5.1	ServiceAccess.SecurityMechanisms	Identifies the security mechanisms required to access the service	Text			
	ServiceAccess.AuthorizationProcess	Criteria and process that a potential consumer needs to go through in order to gain authorization to use the service	Text			
1.5.2	ServiceAccess.Restriction	Restrictions places on users that are allowed access to service.	Text			
1.6	Taxonomy.PropertyGroup	Attributes to connect the service with internal iEHR PM Processes				
1.6.1	Taxonomy.BASegment	Business Architecture Category	Text			
1.6.2	Taxonomy.HL7-EHR-SFM-ID	HL7 EHR Function	Text			HL7 EHR SFM Ids
1.6.3	Taxonomy.HL7Capability	HL 7 Capability Category	Text			HL7 Capabilities
1.6.4	Taxonomy.iEHRCapabillity	iEHR Capability	Text			iEHR capabilities
1.6.5	Taxonomy.ServiceLayer	Service model layer consisting of: application; support; data	Text	Design		
1.6.6	Taxonomy.iBRM-Activity	iEHR Business Reference Model Activity	Text			iEHR iBRM activities
1.6.7	Taxonomy.iEHR-CIPT	Assigned capability IPT	Text			iEHR CIPTs
1.6.8	Taxonomy.Increment	Assigned iEHR increment	Text			iEHR increments
1.6.9	Taxonomy.Release	Assigned release within increment	Text			iEHR increment release
2	SLAPropertyGroup					
2.1	SLA.Number	The unique identifier for a SLA Contract instance	Text			
2.2	SLA.ActionGuarantee	Description of SLA guaranteed service action	Text			
2.3	SLATimePropertyGroup	SLA Performance time commitments	Text			

#	Attribute	Description	Type	LifeCycle Phase	Cardinality	Options
2.3.1	SLATime.Duration	Duration element is a required element in the CommittedTime element which is the duration to complete the service.	Text			
2.3.2	SLATime.Latency	Latency is an optional element for the time delay for starting the service.	Text			
2.3.3	SLATime.Start	Latency is an optional element for the date and time to start the service	Date time			
2.3.4	SLATime.Completion	Date and time for committed completion time	Date time			
2.4	SLAAvailabilityPropertyGroup	Performance availability commitments	Text			
2.4.1	SLAAvailability.From	Date and time for availability starting time	Date time			
2.4.2	SLAAvailability.To	Date and time for availability ending time	Date time			
2.5	SLAThroughPutPropertyGroup	SLA Throughput Commitments	Text			
2.5.1	SLAThroughPut.ThroughPutDuration	This is the duration to complete the service throughput	Numeric			
2.5.2	SLAThroughPut.ThroughPutQuantity	It is the numbers for the throughput, with an attribute of unit of measurement, such as EA, pounds, cubic-feet, etc	Text			
2.5.3	SLAThroughPut.ThroughPutLatency	The time delay for starting the service throughput	Date time			
2.6	SLACostPropertyGroup	SLA Cost commitments	Text			
2.6.1	SLACost.Units	Describe units of service cost	Text			
2.6.2	SLACost.Amount	Describe amount of service cost	Numeric			

PART 25 ADDITIONAL REFERENCES

1. Data Definition Table, Health Level 7 v2.5 Appendix A, 2007.
2. iEHR Integration Governance 01_Draft_WS_0416.docx.
3. MHS-OCIO Policy 11-001. Military Health Systems 2010.
4. Department of Defense Privilege Management Roadmap by The Office of the Assistant Secretary of Defense for Networks and Information Integration / DoD Chief Information Officer; 6 January 2010.
5. <http://www.opengroup.org/soa/source-book/osimmv2/model.htm>
6. Department of Veterans Affairs, One-VA Enterprise Architecture, Enterprise Technical Architecture.
7. Wisnosky, et al, DoD Business Mission Area Service-Oriented Architecture to Support Business Transformation, Software Engineering Technology, October 2008
8. WS-Eventing - <http://www.w3.org/Submission/WS-Eventing/>
9. Enterprise Integration Patterns - <http://www.enterpriseintegrationpatterns.com/toc.html>
10. iEHR Enterprise Technical Architecture- Enterprise Application Integration Volume
11. X. Wang, L. Nayda, R. Dettinger, Infrastructure for a clinical-decision intelligence system, IBM Systems Journal, Vol 46, No. 1, 2007
12. H. Li, M. Xue, Y. Ying, A Web-Based and Integrated Hospital Information System, IEEE Computer Science, 2004
13. U. Abdullah, M. Sawar, A. Ahmed, Comparative Study of Medical Claim Scrubber and A Rule Based System, IEEE 2009
14. A Gomez-Perez, M Fernandez-Lopez, O Corcho, Ontological Engineering, Springer 2004
M. Hepp, P. Leenheer, A Moor, Y. Sure, Ontology Management, Springer, 2008