VA Enterprise Design Patterns:

Data-as-a-Service (DaaS)

Office of Technology Strategies (TS)
Architecture, Strategy, and Design (ASD)
Office of Information and Technology (OI&T)

Version 1.0

Date Issued: October 08, 2014

THIS PAGE INTENTIONALLY LEFT BLANK FOR PRINTING PURPOSES

APPROVAL COORDINATION

TIMOTHY L MCGRAIL 111224

Digitally signed by TIMOTHY L MCGRAIL 111224
DN: dc=gov, dc=va, o=internal, ou=people, 0.9.2342.19200300.100.1.1=tim.mcgrail@va.gov, cn=TIMOTHY L MCGRAIL 111224
Date: 2014.10.14 09:29:00 -04'00'

Date:

Tim McGrail
Deputy Director (Acting)
ASD Technology Strategies

PAUL A. TIBBITS 116858

Digitally signed by PAUL A. TIBBITS 116858
DN: dc=gov, dc=va, o=internal, ou=people, 0.9.2342.19200300.100.1.1=paul.tibbits@va.gov, cn=PAUL A. TIBBITS 116858
Reason: I agree to specified portions of this document.
Date: 2014.10.28 17:38:46 -04'00'

Date:

Paul A. Tibbits, M.D.
DCIO Architecture, Strategy, and Design

REVISION HISTORY

Version	Date	Organization	Notes
0.40	04/18/14	ASD TS	Initial Draft
0.60	06/20/14	ASD TS	Initial Draft for review/feedback from internal and external stakeholders
0.75	08/25/14	ASD TS	2nd Draft for review/feedback from internal and external stakeholders
0.85	09/12/14	ASD TS	Updated draft addressing concerns from internal and external stakeholders, including a revised structure to capture key concepts and attributes associated with DaaS.
0.88	09/12/14	ASD TS	Additional updates made based upon vendor feedback submitted after distribution of v0.85. Provides clarification content and terminology updates and grammatical fixes.
0.95	09/25/14	ASD TS	Final draft version for review/comment by internal VA stakeholders. Updates include changes to language around “centralized/decentralized” data, explanation of use of approved VistA Evolution SOA design pattern until a planned incremental update is made, change to VistA Evolution design pattern to highlight DaaS components, and update to Data Security sections addressing confusion around VA 6500 Handbook encryption issues raised during public forum.
0.98	09/29/14	ASD TS	Updated draft document to include finalized use case documents
1.0	10/08/14	ASD TS	Updated final document based upon last round of internal stakeholder comments as well as TS leadership review.

REVISION HISTORY APPROVALS

Version	Date	Approver	Role
0.40	04/18/14	Jacqueline Meadows-Stokes	ASD TS SOA DaaS Design Pattern Lead
0.60	06/20/14	Jacqueline Meadows-Stokes	ASD TS SOA DaaS Design Pattern Lead
0.75	08/25/14	Jacqueline Meadows-Stokes	ASD TS SOA DaaS Design Pattern Lead
0.85	09/12/14	Jacqueline Meadows-Stokes	ASD TS SOA DaaS Design Pattern Lead
0.88	09/12/14	Jacqueline Meadows-Stokes	ASD TS SOA DaaS Design Pattern Lead
0.95	09/25/14	Jacqueline Meadows-Stokes	ASD TS SOA DaaS Design Pattern Lead
0.98	09/29/14	Jacqueline Meadows-Stokes	ASD TS SOA DaaS Design Pattern Lead
1.0	10/08/14	Jacqueline Meadows-Stokes	ASD TS SOA DaaS Design Pattern Lead

TABLE OF CONTENTS

1. INTRODUCTION

1.1. Background

Data-as-a-Service (DaaS) represents a capability that enables applications to obtain seamless access to enterprise data stores in a standardized way, while shielding them from the complexity of their implementations. It is based on the concept that data can be provided on demand to users through web services regardless of the organizational separation of service providers and consumers.

Within VA, IT programs have experienced problems accomplishing enterprise-wide data sharing due to the development of self-contained applications that access application-specific data stores. In many cases, these data stores required proprietary and/or custom software to access and display data, which often constrains users to proprietary standards. Solutions in these cases typically included a software bundle comprised of a data store and the application(s) needed to access the data. Implementation and sustainment of these solutions left programs in a state of vendor lock-in in order to maintain their applications. Additionally, many programs developed applications tightly coupled to specific data stores, resulting in difficult troubleshooting and increasing maintenance challenges as the data stores changed over time. The diagram below provides a notional services "layer cake" representation of the "as-is" state of the VA application development environment. It is meant to specifically highlight the data layer within VA's IT infrastructure, indicating the existence of data stovepipes and silos (red rectangles).

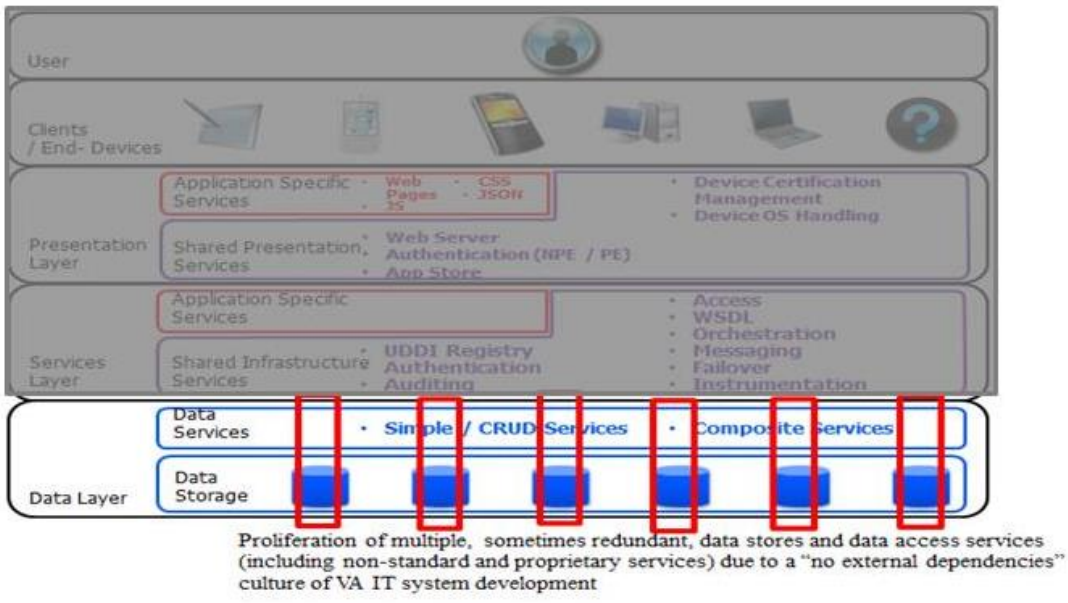

Figure 1 – Notional Representation of VA "As-Is" State Involving Enterprise Data-as-a-Service (DaaS)

VA is planning and executing the evolution of its IT architecture from a set of stove-piped systems to an integrated, modern service oriented architecture (SOA) environment. This evolution will involve design

approaches that support the modernization of existing applications as well as future implementations of new applications that share enterprise services and data using the VA SOA infrastructure to access enterprise data. DaaS will be realized through SOA-based data services in conjunction with additional Enterprise Shared Services (ESS) and data management tools (MDM, ETL, etc.), that will enable data quality to be maintained at a standardized enterprise level, cleansing and enriching data and making it available to different systems, applications, and users on demand. DaaS will aid in simplifying and accelerating application development, eliminating data silos by enabling enterprise-wide data sharing and allowing VA to address challenges with respect to linking various types of customer data and views through a shared enterprise Virtual Data Access Layer (e.g., Health, Benefits, Corporate, and Memorials).

1.2. Purpose

The purpose of this document is to provide strategic direction for the VA to establish a capability for standardized access for multiple applications to enterprise data stores. In the target VA IT environment, enterprise storage, retrieval, and exchange of data will be achieved through the reuse of ESS provided by the VA's SOA infrastructure, which includes the Enterprise Messaging Infrastructure (eMI). These services will include standardized interfaces for enterprise data access, including Create, Read, Update, Delete (CRUD) operations on enterprise data stores. Additionally, these services will include other key attributes such as data aggregation and data validation, as explained in Section 3.

1.3. Scope

The following sections of this first increment DaaS Design Pattern document will present high-level constraints for enabling enterprise data access within VA. It describes key DaaS capability attributes, including the use of ESS, for supporting virtual data access that will be available for all applications. The document addresses:

- DaaS key concepts and context (Section 2)
- DaaS "to-be" architecture attributes (Section 3)
- Descriptions of VA-specific DaaS use cases (Appendices B)
- Applicable technical references, standards, and policies (Appendix D)

The following content is beyond the scope of this document, but may be referenced in appropriate locations to guide further technical planning and coordination with appropriate stakeholders:

- Implementation details for application integration with ESS
- Infrastructure and hardware design specifications
- Details for specific database technologies
- List of authoritative data sources and standards

1.4. Intended Audience

The document will be used by all programs to guide them towards the use of ESS for standardized, enterprise-wide access to enterprise data. This will help programs meet data sharing requirements utilizing enterprise data stores while:

- Developing new VA applications internally
- Modifying existing production systems
- Acquiring and integrating COTS (including open-source) applications

1.5. Document Development and Maintenance

This document was developed collaboratively with internal stakeholders from across the Department and included participation from VA's Office of Information and Technology (OI&T), Product Development (PD), Office of Information Security (OIS), Architecture, Strategy and Design (ASD), and Service Delivery and Engineering (SDE). In addition, the development effort included engagements with industry experts to review, provide input, and comment on the proposed pattern. This document contains a revision history and revision approval logs to track all changes. Updates will be coordinated with the Government lead for this document, who will also facilitate stakeholder coordination and subsequent re-approval depending on the significance of the change.

2. DESIGN PATTERN KEY CONCEPTS

The rise of SOA has rendered the actual platform on which data resides less relevant. SOA assumes distributed resources and use of those resources without knowing the details of their implementations. As such, it sets the stage for more generic services that do not need to expose those implementation details. This applies equally to data services.

DaaS provides enterprise-level shared services for standardized access to enterprise data stores that are available across multiple applications. This section provides high-level overviews of key concepts associated with DaaS that are applicable to solving the recurring problems within the current state of the VA IT environment, as described in Section 1 (Background). These concepts provide the context for the DaaS "to-be" architectural attributes that are described in Section 3, which will guide the establishment of design constraints that will be applied to solution architectures developed by all programs in the VA.

2.1. Data Access

DaaS will enable applications to use enterprise-wide data access services to interact with enterprise data stores. These are intended to be simple, yet coarse-grained services that provide the ability to perform CRUD operations on a single data store, as well as federated services that leverage multiple CRUD operations performed across multiple data stores. Overall, these services abstract the logic required to access underlying data stores via a common data access layer, making application development, configuration and maintenance easier to sustain. Additionally, these services will support data interchange functionality (sometimes referred to as "service agents") when an application must access data provided by an external service. It is important to understand that well-designed user interfaces (whether on a PC or mobile device) should never physically interact directly with databases.

DaaS will be accessible through Enterprise Shared Services (ESS) that will enable seamless data access in accordance with the "to-be" attributes described in Section 3. These services will align to the Information Services category in the ESS layered architecture construct, as derived from the Open Group SOA Reference Architecture. Moreover, they will be generally applicable across business domains, compared to a more domain-specific business process service. The use of these services will:

- Decouple physical and logical locations and avoid unnecessary data replication
- Abstract physical data structures and syntax into views accessible through a common data access layer
- Federate disparate data into useful composites
- Support data integration across both SOA and non-SOA applications

Access to these services will be subject to appropriate access control requirements and restrictions for security, privacy, records management, and retention. This topic will be explored further in a future enterprise design pattern document that addresses enterprise-level authorization and auditing.

2.2. Database Management Systems (DBMS)

There are a multitude of available database management systems (DBMS) that represent the enterprise's data stores and there exist additional variations of these database types based upon the data needs of a given consumer. For the development of this document, specific database types were characterized for enterprise DaaS delivery due to the data specific needs of VA. These database types are referred to as NoSQL, Relational, and Hybrid throughout the remainder of this document and are what drove the development of this design pattern guidance. Use case information regarding the application of these database types is provided in Appendix B.

2.2.1. Relational Databases

Relational database management systems (RDBMS) are a very mature technology, allowing storage of relational (structured) data in flat two-dimensional tables in a row and column paradigm. The first use case in Appendix B provides context on how an application may call an RDBMS through the eMI. Once data is landed within the RDBMS, it is most commonly accessed via the Structured Query Language (SQL) either directly via JDBC or ODBC connectivity, within a part of application logic or a stored procedure. Within the VA IT infrastructure, RDBMS leverage Structured Query Language (SQL) and are supported by tools in the Technical Reference Model (TRM) that enable the use of Data Access Objects (DAO) and Object-relational Mapping (ORM) to provide abstractions to physical database access. Currently, most of the VA's data stores are based on RDBMS technology, including the Corporate Data Warehouse (CDW), which uses SQL and Extract, Transform, and Load (ETL) batch operations to support business intelligence (BI) applications. Programs are required by the ETA Compliance Criteria to evaluate their data modeling needs early in the system development lifecycle (SDLC), and determine whether application data will be persisted in relational schemas or persisted (e.g., form-based data) in a database that allows for more flexible schemas, such as a NoSQL database.

2.2.2. NoSQL Databases

NoSQL, or "Not Only SQL," databases provide a mechanism for storage and retrieval of unstructured data that doesn't easily lend itself to being efficiently stored and retrieved in the traditional row and columns paradigm that RDBMS employ. NoSQL databases have come about as a result for a need to find a new way to store and manage semi-structured and unstructured data assets as the world continues to generate more data from unlimited data generating sources. The second use case in Appendix B provides context on how an application may call a NoSQL through the eMI. Motivations for this approach include simplicity of design, horizontal scaling and finer control over availability. The data structures utilized in NoSQL data stores (e.g. key-value, graph, or document) enables flexible data storage capability, not requiring a rigid schema definition for defined tables, and capacity to adapt to incoming data without needing extensive tuning to maintain performance as data volume continues to grow. As with all technologies, NoSQL data stores excel in some places where RDBMS falls short and vice versa, which is why a complete migration to NoSQL is not a wise approach. Currently the VA is pursuing an enterprise NoSQL data store that may be used for persistence of unstructured data, which may include use cases such as Patient Generated Data (PGD) for mobile applications. This document will be revised to reflect the status of enterprise NoSQL database availability in the VA.

2.2.3. Hybrid (NoSQL and Relational)

A hybrid database call represent a scenario involving the simultaneous processing of data requests to both a RDBMS and NoSQL database. The third use case in Appendix B provides context on how an application may call an aggregate of both RDBMS and NoSQL databases through the eMI. Hybrid scenarios leverage Not-Only SQL database technologies that are designed to meet the increasing volume, velocity, and variety of data that organizations are storing, processing, and analyzing. This category may also include other categories of data stores that are non-relational by nature, such as flat file repositories and even VistA, which includes implementations on non-relational database platforms such as GT.M.

DaaS implementation as an ESS enables access to a diverse range of available data across these three varying database types within the enterprise environment, introducing the concept of and need for authoritative data sources, as explained in the following subsection.

2.3. Authoritative Data Sources

The term authoritative data source refers to a recognized or official data source, or functional combination of multiple data sources, providing reliable and accurate data for subsequent use by consumers. The concept of authoritative data sources aligns with the ongoing effort to streamline data quality of all data sources in the VA and achieve the goal of establishing a trusted set of enterprise-wide data sources. The authoritativeness of data must be well established, documented, and maintained prior to a commitment to specific set of data or authoritative data source solution(s). Data availability across an enterprise environment presents the need for authoritative data sources that are subject to established policy and standards which ensure data quality and consistency. While the concept of

authoritative data sources is introduced in this design pattern document, specific guidance associated with their identification and governance is out of scope. For additional details on authoritative data sources and VA information management policy requirements, see the draft *VA Directive 6518: Enterprise Information Management*.

3. DESIGN PATTERN "TO-BE" ARCHITECTURE

3.1. Alignment to VistA Evolution SOA Design Pattern

This section describes the "to-be," vendor-agnostic attributes of DaaS with respect to the approved SOA design pattern for the VA. Appendix B provides example use cases of how applications may use these services to make calls to an RDBMS, a NoSQL database, or a Hybrid aggregation that involves data from both an RDBMS and a NoSQL databases. There is an array of currently available data services being used within VA today [e.g., Data Access Service (DAS), Veteran Relationship Management (VRM) Customer Relationship Management (CRM), etc.]. Future increments of this document will provide greater details on them, including when and how they are used.

The target state for the DaaS capability supports service discovery, mediated interactions, and decoupled transactions through the VA SOA infrastructure and ESS. DaaS is currently a subset of the overarching FY14 SOA design pattern for VistA Evolution that was approved by the Deputy CIO, Architecture, Strategy, and Design (ASD) on 8 July 2014. The VistA Evolution design pattern was the initial enterprise IT strategic guidance document created within the VA's Technology Strategies Office. It has been the driver for the development of additional Enterprise SOA design pattern documents addressing aspects of: Authentication, Authorization and Audit, IT Service Management, and Web Technologies Data Sharing, to date. This DaaS design pattern is one of the many existing or planned SOA design patterns and establishes the framework for the use of ESS and the common SOA infrastructure, including the eMI, across all applications, including those that shared data with VistA.

To maintain consistency across design pattern documents, the following figure represents an intermediate revision of the current, approved SOA design pattern. It highlights the specific components within that pattern pertaining to DaaS, which are shaded in orange and contained within the larger Data-as-a-Service rectangle. Updates to the SOA design pattern are currently underway to move from a depiction of the SOA environment for VistA Evolution, to a more accurate representation of the full Enterprise SOA target vision for the VA IT infrastructure. The updated design pattern will show the "to-be" vision for a common, enterprise SOA service layer that is available via the eMI and provides access to the enterprise DaaS layer. Upon approval of that design pattern by the Deputy CIO, ASD, this DaaS design pattern document will be updated accordingly.

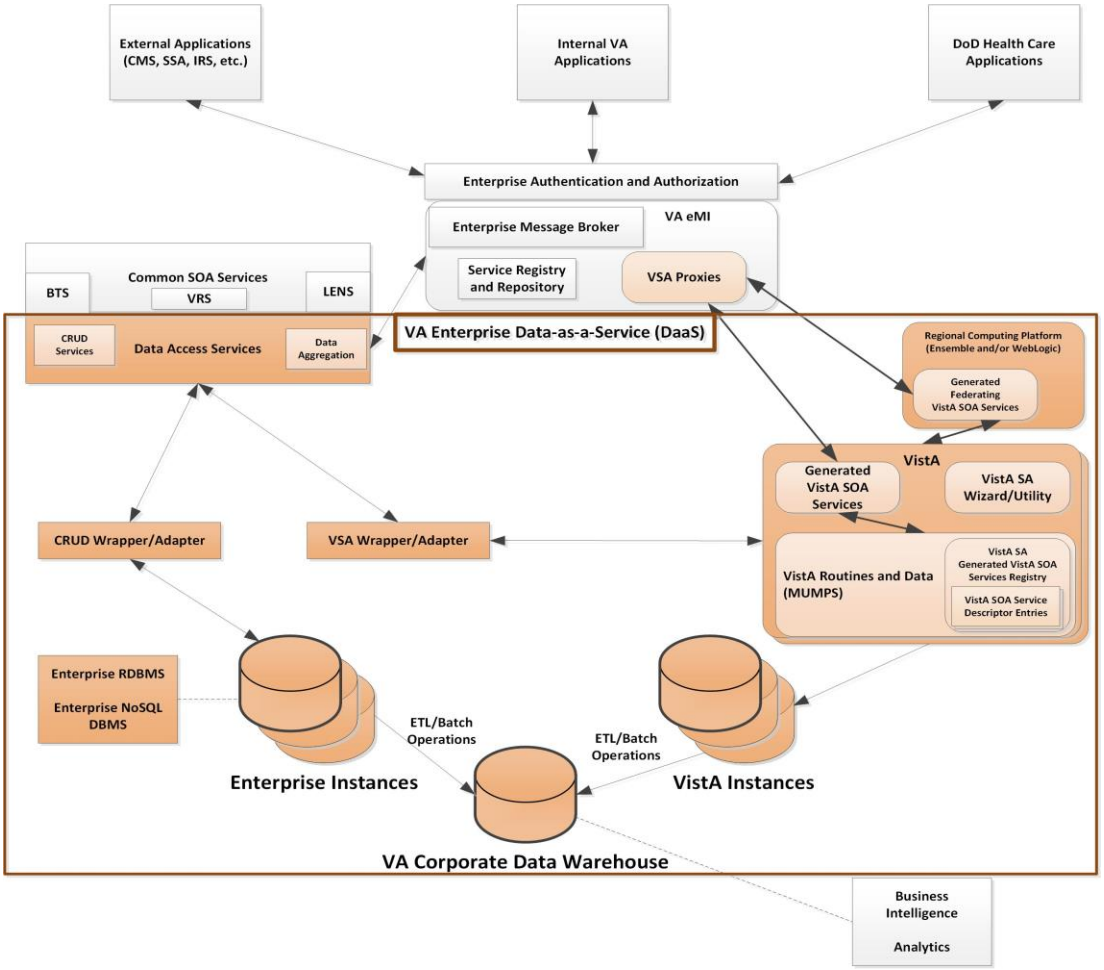

Figure 2 – VA Enterprise DaaS Context (Orange) Based on Approved FY14 VistA Evolution SOA Design Pattern

DaaS encapsulates portions of the existing approved SOA design pattern and represents an enterprise-wide "data layer" that includes enterprise CRUD services for data access as well as the shared enterprise data stores themselves (e.g., VistA, Health Data Repository, and the CDW). Applications will be developed such that they will leverage ESS for security (via Identity and Access Management (IAM) Access Services) as well as common SOA capabilities via Enterprise Messaging Infrastructure (eMI), such as service registry or messaging.

Currently, the SOA design pattern permits development and deployment of CRUD services to provide virtual data access through service adapters or wrappers, thereby abstracting physical database implementation logic. These services will be defined to be web services that provide the ability to perform CRUD operations on a single data store. Additionally, DaaS will allow for the development of a composite set of CRUD services which leverage multiple CRUD operations performed across multiple data stores. These composite CRUD services may be used to support aggregation or manipulation of data that leverage a diverse set of data elements.

The design pattern for SOA in the VA will be revised in FY15 to ensure that the SOA capabilities framework is consistent across each business domain. The revision will include a more comprehensive set of enterprise-wide capabilities available within the VA, including the runtime environment and service containers for ESS. This DaaS design pattern will be revised to ensure proper alignment to the updated framework with regard to enterprise CRUD services and enterprise data stores. The updated design pattern will also provide additional information on officially designated ESS that are promoted and deployed to meet the capability needs for DaaS in the VA.

3.2. Data-as-a-Service (DaaS) Attributes

The following DaaS attributes have been identified as an initial set of key components involved in ensuring an effective enterprise data-as-a-service capability. The content associated with each of these attributes combines an understanding of the "as-is" VA IT infrastructure, knowledge of required internal VA and external government policy, and the application of industry best practices to create the enterprise-level constraints that will guide the realization of the "to-be" vision for VA enterprise Data-as-a-Service. This descriptive language and constraining guidance was developed through collaboration with both internal and external government and industry stakeholders.

3.2.1. Data Aggregation

Data aggregation allows for the use of one or more data sources to construct a usable business entity and account for combinations of structured data with semi-structured or unstructured data. To provide seamless virtual data access to consumer applications, DaaS will provide functions such as data aggregation, data de-duplication/rationalization, and data synchronization through the use of enterprise SOA infrastructure including the Enterprise Messaging Infrastructure (eMI). These functions will support the composition of responses from multiple data stores and provide a semantically harmonized, aggregate response to the end user. Appendix B describes example use cases for data aggregation (referred to as "Hybrid") via both a relational (RDBMS) and NoSQL database.

As of FY14, the VA is currently developing the VistA Exchange capability as an enterprise-wide data aggregation service for the healthcare domain. VistA Exchange will provide 'native federation' of all appropriate longitudinal health record data. This means data from DoD systems, all VA sites, and eHealth Exchange partners will be aggregated, indexed and normalized (to the extent possible for specific data sources) for use by point of care applications. Future deployments of ESS will enable data aggregation capabilities for domains supporting the benefits, memorials, and corporate domains to satisfy business needs internal and external to the VA.

3.2.2. Create, Read, Update, Delete (CRUD) Operations

Create, Read, Update, and Delete (CRUD) operations refer to all of the major functions that are implemented in database applications, and they are the four basic functions of persistent storage. CRUD functionality can be implemented with an object database, an XML database, flat files, or custom file formats, for example.

In the "to-be" state for the VA infrastructure there will be shared data that will allow a standardized set of enterprise-wide CRUD operations for each of the Health, Memorials and Benefits domains. Currently, the following CRUD constructs will be permissible through DaaS:

- CRUD Interchange Operations: Requires the use of the ESB provided by the eMI to ensure data validation, audit logging, and access control due to the need for calling data access services to a database that is not owned by the application. Programs will need to assess application performance for data calls passing through the message broker, as opposed to retrieving data through direct calls to data access services.
- Direct CRUD Operations: Provides a "fast track" for a direct call to data access services without the use of the eMI. This scenario depends on business needs where routing messages through the eMI may hinder performance. An example involves making a direct call to data stores to obtain vital signs information in the operating room of the VA Medical Center (VAMC). This scenario may apply to new applications that may be able to make a direct call for CRUD operations on a specific database, such as the Patient Generated Data (PGD) database. Future versions of this design pattern will provide greater detail addressing the potential need for scenario specific implementations of local access control privileges that allow bypassing of eMI.

3.2.3. Scalability and Elasticity

Scalability is the ability to add resources to a solution as the demand for that solution or the specific resources used by the solution increases. This can be done through added hardware resources or through using additional tuning capabilities inherent in the technology of which the system is comprised (i.e., utilizing compression capability in a database to increase input/output (IO); adding more CPU and RAM to a server running software components of your solution). Within the VA IT infrastructure, there will be the ability to scale out based upon demand, in order to accommodate increased data transaction volume without impacting performance.

Elasticity describes the degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic or manual manner. DaaS will provide the elastic capability to add or remove data nodes on the fly as needed without impacting production. Achieving this notion of elasticity is essential in ensuring consistent performance at the data layer.

3.2.4. Capacity Management

Capacity describes the availability of space/resources at a given time within an IT infrastructure. The need for appropriate resource capacity planning arises from the fact that functionality and demand for resources within an enterprise IT system is never constant. These changes over time increase the demand for resources, which can cause wait times for data delivery and query processing as performance is greatly impacted when a system does not have the capacity to satisfy the demands of the enterprise. Understanding of the necessary workload to be supported by an implemented solution will ensure acceptable performance and system reliability as its resources won't be stretched too thin to

handle the workload demand. Increases in capacity will be implemented efficiently to reduce the need for system downtime and maintaining consistent performance.

VA will need to implement a capacity management process to project what will be needed at initial system implementation. Monitoring is of critical importance as a prerequisite for projections and planning and VA will need to ensure a monitoring system is in place that tracks throughput in order to be proactive in identifying when more capacity will need to be added. This process will need to provide the ability to increase resources as needed by distinguishing between the nature of given workloads (transaction heavy vs. analytics heavy) in order to apply the appropriate configuration for maximum throughput. It is also important to note, that while capacity planning is important at the data layer, there are several tiers in the overall system architecture where capacity planning will need to take place as well. An example of this is the network bandwidth of the ESB and Shared Services server resources, that will facilitate access to the data layer as requests come in. If these two components aren't sized accurately, the underlying data sources can perform at lightning speed but if the delivery mechanism to the end user is always saturated, a query that takes one second to process at the data layer, may take minutes for the results to get back to the end user due to increased latency added in the end-to-end process.

3.2.5. Data Security

Data security is always of paramount importance within an IT environment because it ensures that available data is protected from external and internal threats. While other layers of the VA SOA provide a number of security mechanism to ensure data access requests are coming from a known, authenticated source, this cannot be assumed as true. DaaS will use DBMS capabilities to assure data is protected from a variety of potential attacks, such as SQL injection, hacks into the database, and unauthorized user (internal and external) access, viewing, and manipulation of data. Security measures will also be in place that not only guard against malicious attacks, but also protect against accidental damage to data integrity. For example, the DBMS will need to enforce referential integrity and protect against accidental deletion or modifications of data – actions which, while not malicious in intent, can still cripple a database. Enabling separation of duties with respect to all authorized users (particularly internal users) is another example of a need associated with ensuring data security. This approach safeguards against any single user having 100% autonomy to do as they please with respect to the administration of data sources.

In the "to-be" VA enterprise data environment, implemented data services will provide full end-to-end FIPS 140-2 encryption compliant modules in support of encrypted data, both in transit and at rest. Data-at-rest encryption, for both static (e.g., archive) and inconstant (e.g., reference tables) data, will comply with VA 6500 Handbook, NIST and DISA STIG security controls and policy. At the level of patient generated data (PGD) encryption (i.e., PII/PHI), data-at-rest and in transit will also comply with HIPAA controls in addition to these security policies. These requirements will be met during the development and execution of solution level architectures and implementation guidance in adherence with all

applicable security and Enterprise Information Management policy and controls. More detailed, security-specific guidance can be found in the current Authentication, Authorization and Audit (AA&A) Design Pattern document(s).

3.2.6. Troubleshooting

DaaS will be capable of identifying errors associated with data source access, including timeouts and CRUD operations. Depending upon the gravity of the error, the DaaS may pass some exceptions through the eMI for concurrency violations, unresponsive queries, deadlock etc. This is another example of a specific scenario where direct CRUD may be necessary. Future DaaS design pattern increments will need to address these scenarios in order to drive solution architectures that will leverage services that are compliant with the ESS Exception Handling Guideline.

3.2.7. Data Validation

In accordance with the Enterprise Target Application Architecture (ETAA) SOA Layer Implementation Guide Section 4.1, DaaS will also provide an effective data validation technique, which will validate all input data from other layers or third party components prior to its inclusion in the physical data store and before being made available for query. To ensure an effective enterprise data validation capability, VA will be required to establish and implement a Master Data Management (MDM) Plan and associated MDM tools.

3.3. Enterprise DaaS Constraining Principles and Strategic Guidance

The following table summarizes the constraining principles associated with DaaS concepts and architectural attributes. These principles will be used to drive implementation guidance to programs that will integrate with enterprise data stores via DaaS.

Table 1 – DaaS Constraining Principles and Strategic Guidance

#	DaaS Concept/Attribute	Principles & Strategic Guidance
1	Data Access	Enterprise data will be accessible through non-proprietary protocols and communication methods.
2	Data Access	Authoritative data sources will conform to the OneVA EA Enterprise Logical Data Model (ELDM), which is designed to ensure a consistent, accurate and reliable view of the data.
3	Data Security	Upon the creation of records or collection of records from joins, federation, or aggregation, the security and access levels will raise to the level of the data element with the highest security classification.
4	Data Security	Implemented data services and enterprise data stores will provide full end-to-end FIPS 140-2 encryption compliant modules in support of encrypted data, both in transit and at rest.
5	Data Security	DaaS security, including database security and exception handling will comply with VA 6500 Handbook, NIST and DISA STIG security policies.

#	DaaS Concept/Attribute	Principles & Strategic Guidance
6	Data Security	At the level of patient generated data (PGD) encryption (i.e., PII/PHI), data at rest and in transit will comply with HIPAA controls in addition to VA 6500 Handbook, NIST, and DISA STIG security policies.
7	Data Aggregation	DaaS will provide functions such as data aggregation, data de-duplication/rationalization, and data synchronization through the use of enterprise SOA infrastructure including the Enterprise Messaging Infrastructure (eMI).
8	Scalability	DaaS will provide a high level of availability for service consumers and facilitate outwardly scalability.

Appendix A. USE CASES

The following three(3) use case scenarios involve user/application data access via relational, NoSQL, and Hybrid database services and were initially developed for increment one of the DaaS Design Pattern effort. They were created to aid in identifying best practices and lessons learned, both in industry and VA implementations, as well as help to define the scope of the overall design pattern document.

User/Application Access to Data through Relational Database Service

Purpose

This Use Case is meant to provide a generalized example describing the basic flow of events for a data request interaction between a consumer/application, enterprise data services, and a relational database residing within the VA IT infrastructure. It describes interactions in a scenario where a user of a non-health application is attempting to identify the location of the cemetery and burial site where a deceased veteran relative is buried. This request involves an application initiating a data request that will interact with the Burial Operation Support System (BOSS). This request is executed via enterprise Data-as-a-Service (DaaS), which is available via the enterprise Messaging Interface (eMI) and the "To-Be" Enterprise Virtual Data Access Layer.

Assumptions

Assumptions for this use case include:

- Authentication and Authorization processes have been successfully completed
- The consumer is authorized to request memorials data from database via enterprise capability

Basic Flow of Events

Data Access:

1. Non-health application requests memorials data from Relational Database Service by querying for specific burial site records for a particular veteran (Burial Operation Support System (BOSS))
2. Data call goes through enterprise Messaging Infrastructure (eMI) provided by common services layer with appropriate message header (SQL call and stored procedures)
3. eMI routes request through Enterprise Virtual Data Access Layer (DaaS) based upon specific burial record data requested
4. Enterprise Virtual Data Access Layer receives request from eMI
    a. Using simple CRUD read operation/Composite data manipulation, determines specific data access service and data source (BOSS) to which request message should be sent and routes request
5. BOSS Data Access Service and Database receives data request
    a. Uses SQL call and stored procedures to look up requested data
    b. Database returns veteran burial record information in a data message to Enterprise Virtual Data Access Layer

6. Enterprise Virtual Data Access Layer converts data message to format for consumption by consuming non-health application
7. Enterprise Virtual Data Access Layer relays message to consuming non-health application
8. Non-health application uses received burial records to report location of cemetery and coordinates of grave site
9. Requested data is successfully received by User for location of cemetery and grave site

Use Case Context Diagrams

The following figure shows, at a high level, the flow of events associated with the proposed "To-Be" Use Case for non-health application access to burial data (BOSS) via eMI and the Enterprise Virtual Data Access Layer.

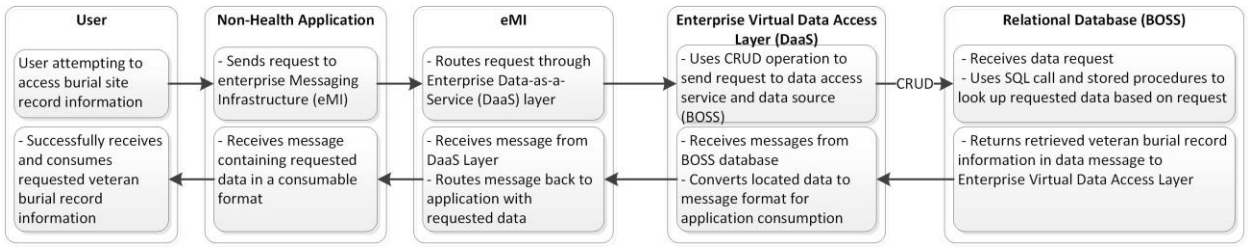

Figure 3: High-Level Context Diagram for Enterprise Relational Data Access (BOSS)

User/Application Access to Data through NoSQL Database Service

Purpose

This Use Case is meant to provide a generalized example describing the basic flow of events for a data request interaction between a consumer/application, enterprise data services, and a NoSQL database residing within the VA IT infrastructure. It describes system interactions in a scenario where a consuming User requests the latest information on a service treatment record (STR) to be retrieved from a VA NoSQL data store. This request involves an internal VA application utilizing the VA Data Access Service (DAS) within the "to-be" enterprise Data-as-a-Service (DaaS) layer, via the enterprise Messaging Infrastructure (eMI), to seamlessly retrieve the necessary STR information.

Assumptions

Assumptions for this use case include:

- Authentication and Authorization processes have been successfully completed
- The consumer is authorized to request memorials data from database via enterprise capability

Basic Flow of Events

Data Access:

1. Consuming User, through internal VA application, requests latest information on a service treatment record (STR) via VA Data Access Service (DAS)
2. Data call is sent to enterprise Messaging Infrastructure (eMI)
3. eMI routes data request to VA DAS via enterprise DaaS layer with appropriate REST message header based upon specific data requested
4. VA DAS receives request from eMI
    a. Using simple CRUD operation/Composite data manipulation, determines specific database to which request message should be sent
5. Enterprise NoSQL database receives data request
    a. Performs index query for requested data based on input from DAS
    b. Database returns result of query to DAS
6. DAS converts located data to message format for consumption by internal VA application
7. DAS relays message back to internal VA application through enterprise DaaS layer and via the eMI
8. Requested data for latest service treatment record is successfully received by User

Use Case Context Diagram

The following figure shows, at a high level, a depiction of the flow of events associated with the proposed "To-Be" Use Case Scenario above.

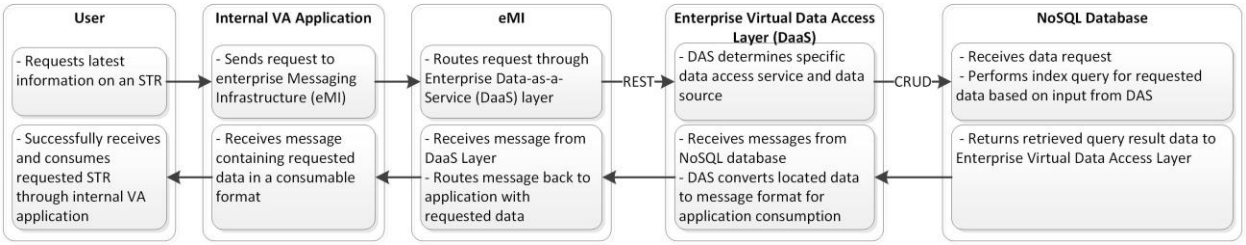

Figure 4: High-Level Context Diagram for Enterprise NoSQL Data Access

User/Application Access to Data through Hybrid Database Service

Purpose

This Use Case is meant to provide a generalized example describing the basic flow of events for a data request interaction between a consumer/application, enterprise data services, and a "hybrid" database service (i.e., an aggregation of data retrieved from both relational and NoSQL data sources) residing within the VA IT infrastructure. It describes interactions in a scenario where a user requests pharmaceutical healthcare data based upon geo-location, PII/PHI, and data range search criteria. This request involves the Joint Legacy Viewer (JLV) user interface (UI) application initiating a health data request and seamlessly receiving an aggregated message response from a "hybrid" database service. Data is retrieved from multiple data sources, both relational and NoSQL, and returned as a single

consumable message using enterprise Data-as-a-Service (DaaS), which is available via the enterprise Messaging Interface (eMI) and the "To-Be" Enterprise Virtual Data Access Layer.

Assumptions

Assumptions for this use case include:

- User Authentication and Authorization processes have been successfully completed to enable JLV access
- User has appropriate role authorizing ability to request healthcare data from databases via enterprise capability based upon PII/PHI
- JLV is authorized to access patient health data from databases via enterprise capability

Basic Flow of Events

Data Access:

1. Joint Legacy Viewer (JLV) application requests pharmaceutical health data to determine number of prescriptions filled by given patient within a specific location and date range based upon available PII/PHI
2. Data call goes through enterprise Messaging Infrastructure (eMI) provided by common services layer with appropriate message headers
    a. NoSQL – Index (Prescription details)
    b. Relational - SQL call and stored procedures (PII/PHI, geo-location, date range)
3. eMI routes requests through Enterprise Virtual Data Access Layer (DaaS) based upon specific data requested
4. Enterprise Virtual Data Access Layer receives requests from eMI
    a. Using enterprise CRUD operation (federated) data manipulation, determines specific data access service and data sources to which request messages should be sent and routes message appropriately
5. Data-as-a-Service (DaaS) Layer
    a. Enterprise Relational Database
        i. Receives data request
        ii. Uses SQL call and stored procedures to look up requested data
        iii. Returns data to Enterprise Virtual Data Access Layer
    b. Enterprise NoSQL Database
        i. Receives data request
        ii. Performs index query for requested data
        iii. Returns data to Enterprise Virtual Data Access Layer
6. Enterprise Virtual Data Access Layer receives data messages from Relational and NoSQL data stores
7. Enterprise Virtual Data Access Layer logic recognizes differing data types and performs a data aggregation step to consolidate into single message format consumable by JLV
8. Enterprise Virtual Data Access Layer relays message through eMI

9. eMI routes message back to JLV with number of prescriptions filled by the identified patient within a designated geo-location area during a given date range
10. Requested data is successfully retrieved, aggregated, and received by JLV/user

Use Case Context Diagram

The following figure shows, at a high level, the flow of events associated with the proposed "To-Be" Use Case for JLV pharmaceutical data access via eMI and the Enterprise Virtual Data Access Layer.

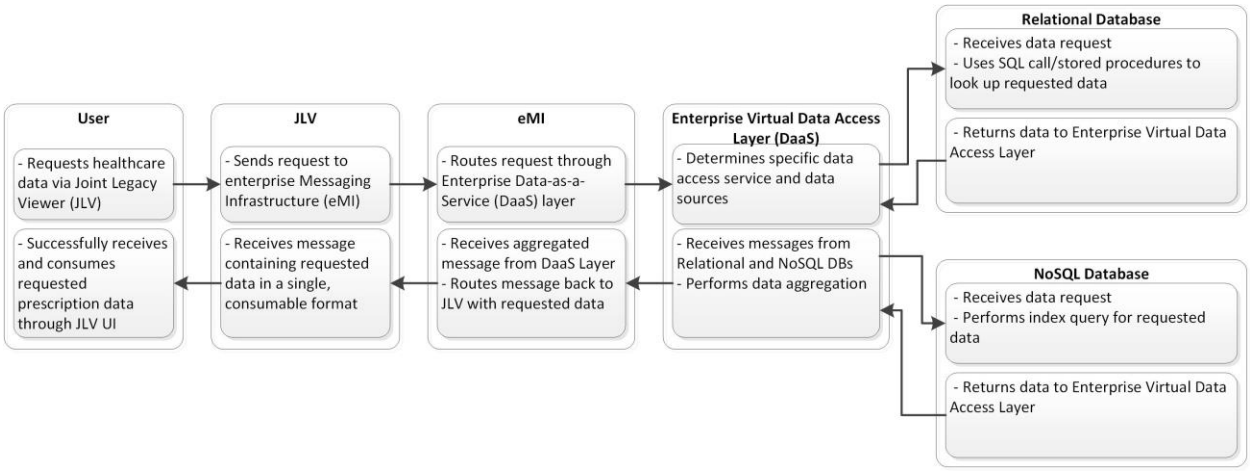

Figure 5: High-Level Context Diagram for Enterprise Hybrid Data Access

Appendix B. VOCABULARY

Acronym	Description
ACID	Atomicity, Consistency, Isolation, Durability: Set of properties that guarantee that database transactions are processed reliably
BI	Business Intelligence
COOP	Continuity of Operations Plan (Disaster Recovery)
DaaS	Data-as-a-Service
Database	Technology used by the VA to provide an organized collection of data for storage and retrieval. Please note that “database” and “data store” are used interchangeably throughout this document.
eCRUD	Enterprise Create, Read, Update, Delete
DAS	VA Data Access Service
eMI	Enterprise Messaging Interface, or the SOA Infrastructure and Enterprise Service Bus capability within the VA IT infrastructure
EIM	Enterprise Information Management
ESS	Enterprise Shared Services
IAM	Identity and Access Management
ISI	Information Sharing Initiative
JSON	JavaScript Object Notation: Text-based open standard designed for human-readable data interchange
MDM	Master Data Management
MVI	Master Veteran Index: VA application that provides Patient Identifier Cross Reference between VA, DOD and external partners
NoSQL	Database solution that provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases
PHI	Protected Health Information
PII	Personally Identifiable Information
RDBMS	Relational Database Management System
SDLC	System Development Lifecycle
SDD	System Design Document
SQL	Structured Query Language: Standard language for relational database management systems (ANSI)
TCO	Total Cost of Ownership
VistA	Veterans Information Systems and Technology Architecture
XML	Extensible Markup Language

Appendix C. APPLICABLE REFERENCES, STANDARDS, AND POLICIES

The DaaS design pattern is aligned to the following pertinent standards, policies and references, which include guidance that is currently incorporated into the OneVA Enterprise Technical Architecture (ETA):

#	Issuing Agency	Policy, Directive, or Procedure	Purpose
1	VA	VA 6500 Handbook	• Defines the overall security framework for VA including data storage, retrieval, and exchange.
2	VA	VistA Evolution SOA Design Pattern – COTS Application and Non-COTS Applications	• Provides references to the use of enterprise CRUD services and persistent data stores as part of the integration with SOA support infrastructure services. These documents are intended to standardize and constrain the solution architecture of all healthcare applications in the VA. • DaaS represents the "data layer" of the VA SOA design pattern
3	VA	SOA Layer Implementation Guide Section 4.1	• Provides implementation guidance for application developers with regard to a virtual data access layer, focusing on the Java EE platform but conceptually is applicable to all platforms using data services.
.4	VA	Enterprise Application Architecture (EAA) Sections 2 and 5	• Section 5 provides technical underpinnings for cross-cutting development concerns for VA applications including data architecture. • Section 2 prescribes the architecture principles for all applications that form the basis of enterprise design patterns. Specifically, Principle 2 regarding the separation of presentation, business, and data logic applies to the to-be vision of virtual data access and aggregation via Data-as-a-Service
5	VA	SOA Technical Framework (SOA-TF) Section 7	• Provides technical underpinnings for data logic concerns for VA applications. • Includes the concepts of data access for direct access to a data store owned by an application, and data interchange for performing CRUD operations on a data store owned by an applications. Data interchange requires the use of the ESB to route messages and perform data validation.
6	VA	ESS Strategy	• Provides the overarching strategy for developing, deploying, and managing ESS throughout the VA
7	VA	OneVA Enterprise Technology Strategic Plan (ETSP)	• The DaaS design pattern will help programs develop applications in alignment with IT Vision Attributes 6, 8, 9, 10, 12 (scale-out of enterprise data stores)
8	VA	DAS System Design Document (SDD)	• Describes how the VA DAS will be constructed and translates identified business needs, business requirements, functional requirements, and non-functional requirements into a document from which developers can create the system
9	VA	DRAFT VA Directive 6518: Enterprise Information Management	• Establishes official policy for the implementation of Authoritative Data Sources in the VA • DaaS provides access to enterprise data stores that will contain authoritative data sources in compliance with this directive