**Benefits Claims Decision Support (BCDS) System**

System Design Document



**January 2016**

Version 0.1

Department of Veterans Affairs

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 11/09/2015 | 0.1 | Document Creation | Anthony Cannon / Pro-Sphere |
| 12/15/2015 | 0.2 | Architecture updates | Vasu Rayapati |

.

Artifact Rationale

The System Design Document (SDD) is a dual-use document that provides the conceptual design as well as the as-built design. This document will be updated as the product is built, to reflect the as-built product.

When to Complete Each Section of the SDD

| Section | Completed On or Before PMAS Phase | Rationale |
| --- | --- | --- |
| 1 – Introduction | MS 0 Review; updated thereafter | Conceptual design should inform evaluation of investments |
| 2 – Background | MS 0 Review; updated thereafter | Conceptual design should inform evaluation of investments |
| 3 – Conceptual Design | MS 0 Review; updated thereafter | Conceptual design should inform evaluation of investments |
| 4 – Data Phase | MS 0 Review; updated thereafter | Conceptual design should inform evaluation of investments |
| 5 – Logical Design | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |
| 6 – Process Phase | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |
| 7 – Development Phase | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |
| 8 – Physical Phase | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |
| Attachments | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |

A product’s system design should be defined conceptually prior to the allocation of personnel and resources that occur at project initiation. This gives the enterprise an opportunity to evaluate IT investments before project teams are stood up and funding is allocated. Sections 1- 4 which discuss the high level design should be completed prior to MS 0. All sections should be completed and updated before MS 1. Projects will need to address all SDD approval constraints prior to the MS 2 review. In addition, the SDD should reflect the as-built product going into the MS 2 review.

Instructions

| Activity | New Capability (1) | Feature Enhancement (2) |
| --- | --- | --- |
| **Field Deployment (A)** | Yes | Yes |
| **Cloud/Web Deployment (B)** | Yes | Yes |
| **Mobile Application (C)** | Yes | Yes |

Table of Contents

[1. Introduction 6](#_Toc437944484)

[1.1. Scope 6](#_Toc437944485)

[1.2. Stakeholders 6](#_Toc437944486)

[1.3. User Profiles 6](#_Toc437944487)

[2. Background 6](#_Toc437944488)

[2.1. Overview of the System 6](#_Toc437944489)

[2.2. Overview of the Business Process 6](#_Toc437944490)

[2.3. Overview of the Significant Requirements 6](#_Toc437944491)

[3. Conceptual Design 7](#_Toc437944492)

[3.1. Conceptual Application Design 7](#_Toc437944493)

[3.1.1. Application Context 8](#_Toc437944494)

[3.1.2. High-Level Application Design 10](#_Toc437944495)

[3.1.3. User Interface 11](#_Toc437944496)

[3.1.4. Business Services 11](#_Toc437944497)

[3.1.5. Technical Services 12](#_Toc437944498)

[3.1.6. Data Services 12](#_Toc437944499)

[3.1.7. External Interfaces 12](#_Toc437944500)

[3.1.8. Application Locations 13](#_Toc437944501)

[3.2. Conceptual Data Design 13](#_Toc437944502)

[3.2.1. Project Conceptual Data Model 13](#_Toc437944503)

[3.2.2. Database Information 14](#_Toc437944504)

[3.2.3. User Interface Data Mapping 14](#_Toc437944505)

[3.2.3.1. Application Screen Interface 14](#_Toc437944506)

[3.2.3.1.1. *<Insert name of screen>* 14](#_Toc437944507)

[3.2.3.2. Application Report Interface 15](#_Toc437944508)

[3.2.3.2.1. *<Insert name of report>* 15](#_Toc437944509)

[3.2.3.3. Unmapped Data Element 16](#_Toc437944510)

[3.3. Conceptual Infrastructure Design 16](#_Toc437944511)

[3.3.1. System Criticality and High Availability 17](#_Toc437944512)

[3.3.2. Application servers 17](#_Toc437944513)

[3.3.3. Special Technology 18](#_Toc437944514)

[3.3.4. Technology Locations 18](#_Toc437944515)

[3.3.5. Conceptual Infrastructure Diagram 19](#_Toc437944516)

[3.3.5.1. Conceptual Production String Diagram 19](#_Toc437944517)

[4. Data Phase 19](#_Toc437944518)

[4.1. Data View 19](#_Toc437944519)

[4.2. Data Dictionary 19](#_Toc437944520)

[5. Logical Phase 20](#_Toc437944521)

[5.1. Interface Architecture 20](#_Toc437944522)

[5.2. Human-Machine Interface 21](#_Toc437944523)

[5.3. Interface Design Rules 21](#_Toc437944524)

[5.4. Inputs 21](#_Toc437944525)

[5.5. Outputs 21](#_Toc437944526)

[5.6. Navigation Hierarchy 21](#_Toc437944527)

[5.6.1. Screen [x.1] 21](#_Toc437944528)

[5.6.2. Screen [x.2] 21](#_Toc437944529)

[5.6.3. Screen [x.3] 21](#_Toc437944530)

[6. Process Phase 22](#_Toc437944531)

[6.1. Service Oriented Architecture / ESS 22](#_Toc437944532)

[7. Development Phase 22](#_Toc437944533)

[7.1. Software Architecture 22](#_Toc437944534)

[7.1.1. Presentation – User interface 23](#_Toc437944535)

[7.1.2. Service Contract and Adaptors 23](#_Toc437944536)

[7.1.3. Business services and entities 23](#_Toc437944537)

[7.1.3.1. Role-based access and user provisioning 23](#_Toc437944538)

[7.1.3.2. Reporting 23](#_Toc437944539)

[7.1.3.3. Modeling Engine 24](#_Toc437944540)

[7.1.4. Logging and Auditing 24](#_Toc437944541)

[7.1.5. Exception Handling 24](#_Toc437944542)

[8. Physical Phase 24](#_Toc437944543)

[8.1. Hardware Architecture 24](#_Toc437944544)

[8.2. Network Architecture 25](#_Toc437944545)

[9. Attachment A – Approval Signatures 26](#_Toc437944546)

[A. Additional Information 27](#_Toc437944547)

[A.1. Identification of Technology and Standards 27](#_Toc437944548)

[A.2. Constraining Policies, Directives and Procedures 27](#_Toc437944549)

[A.3. Requirements Traceability Matrix 27](#_Toc437944550)

[A.4. Packaging and Installation 27](#_Toc437944551)

[A.5. Design Metrics 27](#_Toc437944552)

# Introduction

This document contains an overview of the architecture components which combine to form the BCDS and VBA system. This document describes the system design, and the strategy for realizing the architecture vision.

## Scope

Insert a link to the Business Requirement Document here:

## Stakeholders

## User Profiles

Describe the intended user base of the proposed system.

Describe the attributes of the user community (and their proficiency with software systems) and the technical community (and their familiarity with support and maintenance).

# Background

## Overview of the System

The Veterans Benefits Administration (VBA) is responsible for administering the issuance of over $40 Billion in Disability Compensation Benefits to deserving Veterans and their survivors annually. VBA adjudicates over a million claims for such benefits, including claims for new benefits, as well as claims to increase the amount of benefits based on increased level of disability or other relevant changes in the Veteran’s health or welfare status.

The global war on terror (GWOT) and other U.S. military engagements over the past two decades, medical advancements that have significantly enhanced survivability from battlefield injuries, previously antiquated VBA adjudication processes and related infrastructure, all collectively operated to strain VBA’s ability to promptly adjudicate claims for disability compensation benefits. As late as 2012, the volume of claims in pending status of 125 days exceeded 600,000. The Obama administration established a departmental goal of eliminating this “claims backlog” by the end of 2015. The Department of Veterans Affairs (VA) is poised to accomplish this goal after having invested considerably in transforming its adjudication processes through the implementation of digitally supporting technical infrastructure, and instituting mandatory over-time for personnel responsible for adjudicating the claims. Sustaining such performance levels can be achieved only by exploring innovative strategies that provide greater assurance of prompt adjudication of claims for these Veteran benefits.

## Overview of the Business Process

Provide an overview of the business processes that this application will support. Provide a link to the RSD.

## Overview of the Significant Requirements

Provide a link to the RSD and the BRD. Provide a link to a Rational report which can provide requirements for the following:

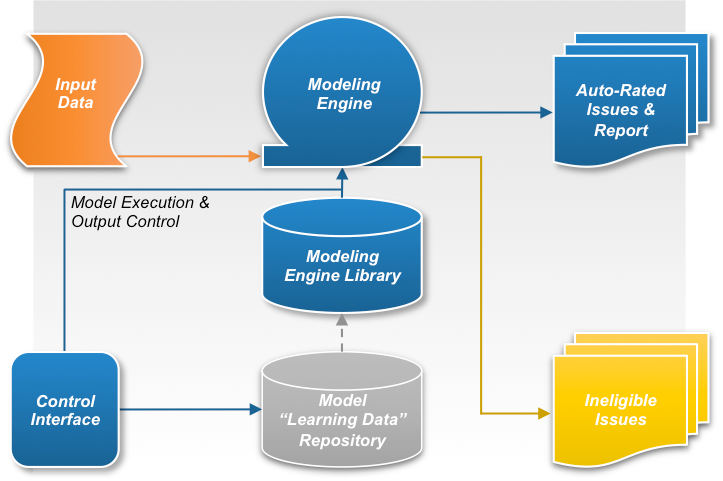
* Overview of significant functional requirements
* Overview of the functional workload/performance requirements
* Overview of operational requirements
* Overview of pivotal technical requirements
* Overview of the security and privacy requirements
* Overview of system criticality and high availability requirements
* Overview of single sign on requirements
* Overview of use of enterprise portals
* Overview of special device requirements

# Conceptual Design

This section of the SDD provides details about the following topics:

* Conceptual Application Design
* Conceptual Data Design
* Conceptual Infrastructure Design

## Conceptual Application Design

Team ProSphere’s proposed solution architecture is depicted **Figure 1**. As this figure suggests, the BCDS System will be composed of four primary components:

***Predictive Model “Model Formulation Data” Repository:*** This component stores statistically relevant enterprise data supporting current and future predictive models. These data will be used to formulate, validate, and refine current and future predictive models;

***Predictive Model Library:*** This component stores the established predictive models for use by the modeling engine to automatically determine ratings for issues contained within relevant claimant data sets.

***Modeling Rules Engine:*** This component provides the platform and modeling logic for ingesting claimant data sets, determining whether the predictive models can be applied, applying the predictive models, and outputting either auto-rated issues and the related report for the claimant data set, or the ineligible issues for the claimant data set;

***Control Interface:*** This component allows authorized users to control how the data ingest and auto-adjudication system operates, provides means to control the format and content of the output, and provides the mechanism to control the predictive model library and refinement of the underlying data.

Team ProSphere’s development approach for the BCDS is depicted in **Figure 2**. We provide a detailed description of the activities conducted in each iterative development cycle in the sections that follow

### Application Context

VBA is seeking to determine whether this analytical methodology can serve as the foundation for developing a solution capable of accurately deriving disability rating determinations with little or no human intervention.

Three specific project objectives are listed below.

1. To develop a BCDS solution that fulfills the summary functional requirements defined in the original solicitation and awarded task order
2. To employ the BCDS solution in a pilot to evaluate the feasibility of deploying such a solution, on an industrial scale commensurate with VBA annual operations
3. To empirically characterize the operational impact such a solution would deliver if implemented by VBA

Provide a diagram showing the context within which the application exists. The diagram should include:

* One object for the system that is the subject of this design,
* One object for each system or external service with which this system interfaces,
* One object for each Program Office system or subsystem with which this system interacts, and
* One for each data store that this system shares with other systems.

Sample Application Context Diagram

Sample Application Context Diagram

Sample Application Context Diagram

Figure 1: Sample Application Context Diagram

Table 5 describes the information in the Application Context Diagram in four sections. Note that the system for which this design applies is represented by a single object (typically in the center of the diagram). Therefore, it is not referred to in Table 5 below.

Table 5 (Grouping): Application Context Description

Object

| ID | Name | Description | Interface Name | Interface System |
| --- | --- | --- | --- | --- |
| < ID from diagram> | <Enter name of external system, organization, or agency> | <High level discussion of the purpose of the information interchange> | <Name of each of the Interfaces to this object> | <Systems with which this system interfaces> |

Interfaces External to OI&T

| ID | Name | Related Object | Input Messages | Output Messages | External Party |
| --- | --- | --- | --- | --- | --- |
| < ID from diagram> | <Interface name from the object rows above> | <Object from the list above that is the source of this interface> | <For each input message, enter a business description of the data being input> | <For each output message, enter a business description of the data being output> | <Name of external party> |

Interfaces Internal to OI&T

| ID | Name | Related Object | Input Messages | Output Messages | External Party |
| --- | --- | --- | --- | --- | --- |
| < ID from diagram> | <Interface name from the object rows above> | <Object from the list above that is the source of this interface> | <For each input message, enter a business description of the data being input> | <For each output message, enter a business description of the data being output> | <Name of external party> |

Externally Shared Data Stores

| ID | Name | Data Stored | Owner | Access |
| --- | --- | --- | --- | --- |
| < ID from diagram> | <Name of the data store> | <Description of the data being stored> | <This System / Name of OIT or external organization> | <Enter the Create, Read, Update, or Delete (CRUD) operations that this system does on this data store> |

### High-Level Application Design

The application architecture diagram shows how the various components of the BCDSS Solution are integrated with each other to build an integrated application.

The BCDS is multi-tiered, modular, and rules-engine based application. It is based on the principles of Service Oriented Architecture (SOA), and employs open standards for interfaces making it configurable to meet VBA requirements.



Service components are built for interoperability. Services adhere to the Web Services Interoperability (WS-I) standard and are based on the Service Contract Specification. Services within this layer are classified into the following categories.

### User Interface

The user interface Layer provides graphical user interfaces to human and system users to access services through various access channels.

The presentation layer for BCDS web-based application use services provided by the business services to facilitate the interaction of the Modeling engine, reporting engine and portal applications. The BCDS presentation architecture is based on the event driven application architecture deployed within the tomcat container.

### Business Services

Business Services encapsulate the application and business logic that supports BCDS business functions. Services within this layer are invoked in various methods. Trusted application components and services in other architecture layers may invoke services directly via open Java EE-compliant application interfaces (APIs). This method of invocation is often preferred for performance reasons when both the consumer and provider are known and within trusted zones; are both contained within the BCDS system boundary; or exist within the same Java container.

Rules Services within solution provide the encapsulation and isolation of business rules logic for implementing policy and complex rules. Business Rules are authored in a consistent, technology-neutral and human-readable format and translated into machine-executable format by the BCDS rule authoring tool.

### Technical Services

Technical services include common architecture services (such as batch management, auditing, and exception handling) and those services used by applications to access the core set of functionality provided by other software that are plugged into the BCDS framework; including business rules management and reporting functionalities.

### Data Services

Data Services provide the application access to shared data. The Data Services components are written in Java and use open standards and specifications. The Data Services are coded to the Java Persistence API (JPA), a data management framework, and employ object to relational mapping techniques to access data within the underlying Database Management System (DBMS).

### External Interfaces

The solution will be used to connect, mediate and manage messaging between heterogeneous services, legacy systems, and other integration instances across the enterprise.

Table 6: Objects in the High Level Application Design

Objects / Components to be Built or Modified

| ID | Name | Description | Service or Legacy Code | External Interface Name | External Interface ID | Internal Interface Name | Internal Interface ID | SDP Sections 1&2 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| < ID from diagram> | <Name of high level service or internal subsystems> | <Business level discussion of the function or role of the service or subsystem> | <Service / modification to legacy system> | <Name of each of the external interfaces to this object> | <ID of each of the external interfaces to this object> | <Name of each of the internal interfaces to this object> | <ID of each of the internal interfaces to this object> | [Approved / Submitted / Being Developed] |

Internal Data Stores

| ID | Name | Data Stored | Steward | Access |
| --- | --- | --- | --- | --- |
| < ID from diagram> | <Name of the data store> | <Description of the data being stored> | <Name of the system/subsystem /service that is the steward for the data> | <Which CRUD operations does this system do on this data store> |

### Application Locations

Use Table 7 to specify the locations at which the application components will be hosted.

Consideration should be given to adopt cloud technologies as potential solutions. Leveraging cloud technologies is part of a larger effort by the Office of Management and Budget (OMB) to reform Federal IT Management. Considerations such as regional deployments etc. should be documented in this section.

Table 7: Application Locations

| Application Component | Description | Location at Which Component is Run | Type |
| --- | --- | --- | --- |
| <Component name> | <Description> | <Facility name> | <Presentation Logic/Business Logic/Data Logic/Interface Code> |

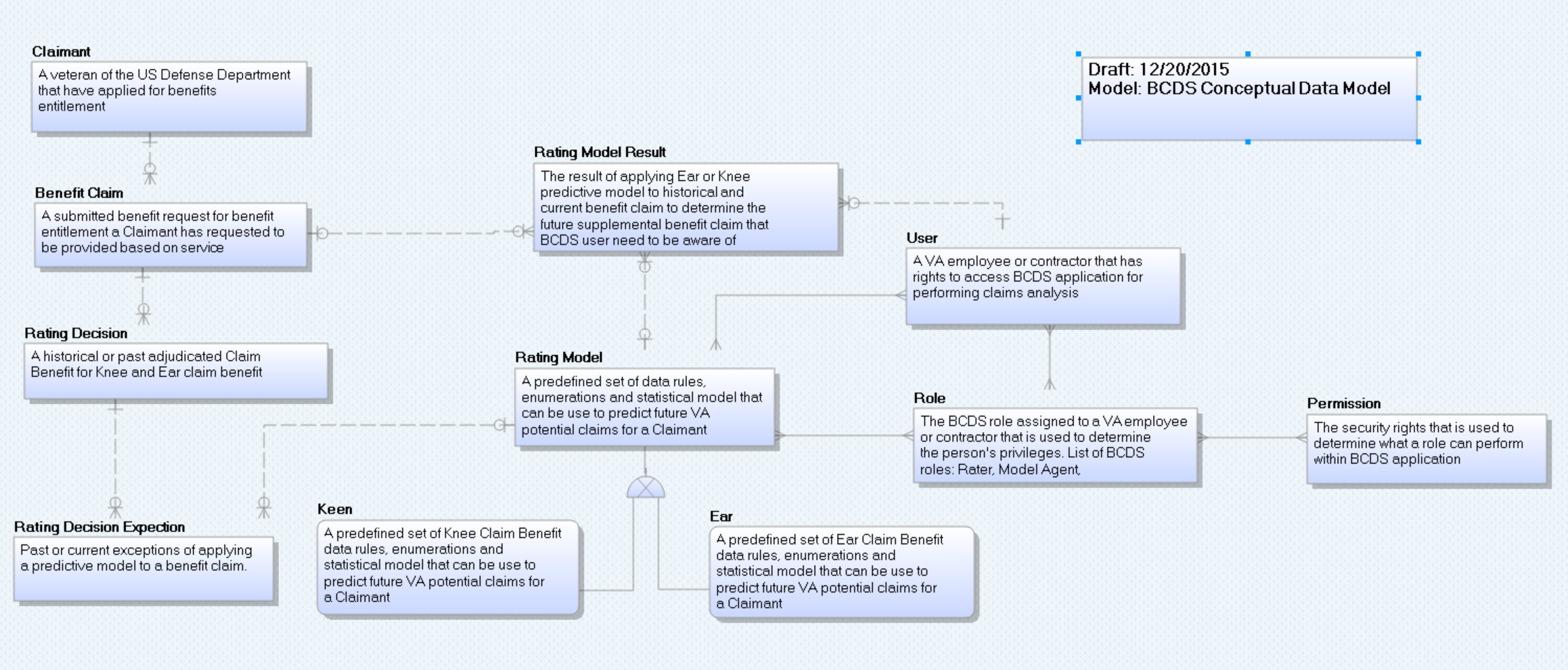
Table 8: Application Users

| Application Component | Location | User |
| --- | --- | --- |
| <Component name> | <Facility name> | <Role> |

## Conceptual Data Design

### BCDS Conceptual Data Model

Figure 3: BCDS Conceptual Data Model



BCDS CDM Data Dictionary

| **Entity Name** | **Object Type** | **Definition** |
| --- | --- | --- |
| User | Entity | A VA employee or contractor that has rights to access BCDS application for performing claims analysis |
| Rating Decision | Entity | A historical or past adjudicated Claim Benefit for Knee and Ear claim benefit |
| Ear | Entity | A predefined set of Ear Claim Benefit data rules, enumerations and statistical model that can be use to predict future VA potential claims for a Claimant |
| Keen | Entity | A predefined set of Knee Claim Benefit data rules, enumerations and statistical model that can be use to predict future VA potential claims for a Claimant |
| Rating Model | Entity | A predefined set of data rules, enumerations and statistical model that can be use to predict future VA potential claims for a Claimant |
| Benefit Claim | Entity | A submitted benefit request for benefit entitlement a Claimant has requested to be provided based on service |
| Claimant | Entity | A veteran of the US Defense Department that have applied for benefits entitlement |
| Rating Decision Expection | Entity | Past or current exceptions of applying a predictive model to a benefit claim. |
| Role | Entity | The BCDS role assigned to a VA employee or contractor that is used to determine the person's privileges. List of BCDS roles: Rater, Model Agent, Administrator, Modeling Analyst |
| Rating Model Result | Entity | The result of applying Ear or Knee predictive model to historical and current benefit claim to determine the future supplemental benefit claim that BCDS user need to be aware of |
| Permission | Entity | The security rights that is used to determine what a role can perform within BCDS application |

### Database Information

Table 9: Database Inventory

| Database Name | Description | Type | Steward |
| --- | --- | --- | --- |
| Model Repository | A data repository responsible for persisting Ear & Knee predictive models and the associated data for operation of BCDS Application | N/A | TBD |
| Reporting Repository | A data repository responsible for persisting the results of a the predictive model and ratings determination for reporting and analysis purpose | N/A | TBD |
| Staging Repository | A data repository responsible for staging of data and preparation of data and predictive models before processing | N/A | TBD |

### User Interface Data Mapping

ffBCDS Application user interface (UI) in the application presentation layer will be available for end user to enter data data into the system. Direct end user access to the data repositories of BCDS will not be available.

#### Application Screen Interface

Create a new subsection for each screen of the Graphical User Interface (GUI) that users will have access to, in order to enter or update information in the database.)

##### *<Insert name of screen>*

Figure 4: <screen name> Screen represents the screen that <describes what the screen accomplishes>; Table 10 describes it. Paste a screenshot below and complete the table to describe the screen.

Figure 4: *<screen name>* Screen

Table 10: *<screen name>* Screen Description

| Graphical User Interface (GUI) Field | Table (Database Table that field connects to) | Field (Field in Table that the GUI field connects to) | Comments |
| --- | --- | --- | --- |
| <Name> | <xxx> | <PATIENT\_ NAME> | <Add any comments or descriptive information that would be relevant to the tester> |
| <SSN> | <xxx> | <SSN> |  |
| Date of Birth (Age) | <yyyy> | DATE\_OF\_BIRTH DATE\_OF\_DEATH (if deceased) |  |

#### Application Report Interface

This section describes and defines the reports that will be available in the user interface, if applicable.

##### *<Insert name of report>*

<Create a new subsection for each report> Figure 6 represent <name> screen and Table 16 describes it…

Figure 5 represents the <report name>; Table 11 describes it. Paste a screenshot of the report below and complete the table to describe the report.

Figure 5: *< Report name>* Report

Table 11: *<Report name>* Description

| Report Column | Data Source *<Table Name. Fieldname>* |
| --- | --- |
| Patient | <xxx.PATIENT\_NAME> |
| SSN | <xxx.SSN> |
| DoB | <yyyy.DATE\_OF\_BIRTH> |

#### Unmapped Data Element

In this section describe any database element that was not mapped to a screen and the reason the data element(s) was not mapped. This section may be skipped if there is no User Interface involved in the project, such a building a service offering etc.

## Conceptual Infrastructure Design

The Infrastructure Architecture design of the system defines the way in which the system architecture components are logically related and deployed to support the runtime execution of the BCDS solution. The BCDS application environment is deployed in two configurations: High Availability and Non-High Availability. The Production environment and prod-like environments are built as a High-Availability environment; while the majority of the non-production environments (i.e. Development, Testing) do not require the same availability and are built with less resilience and fail-over capability.

The below High Level Non-High Availability diagram describes the various connections and transport mechanisms used within the BCDS applications.



### System Criticality and High Availability

The infrastructure architecture includes the appropriate deployment and configuration of firewalls to define network segments. We place web servers in a Demilitarized Zone (DMZ) so that end-users can easily access these resources to conduct their business. The DMZ is protected from the Internet by a set of edge firewalls. Assets storing data are placed deeper in the network on an internal segment that is protected from the DMZ from an additional set of firewalls. Highly critical assets may be placed even deeper in the network in a highly secure network segment that is separated by yet more firewall.

Our solution employs secure data transmission protocols, including the secure sockets layer (SSL) protocol and public key authentication, signing and encryption.

The following figure describes the production high availability environment. This design is also used for select non-production environments that require the same configuration as production for testing purposes.



In the event of a system failure, or when a technology component or network becomes unavailable, the solution employs continuity of interface operations through high-availability and redundant databases and devices. The HA solution provides automated forwarding and manual options to reconfigure devices and quickly switch from backup to primary, or passive to active. These features offer continuity of operations for our interfaces in the event of a local outage, loss of component availability, or system failure

### Application servers

The BCDS system is to deploy applications on tomcat server. Using tomcat clustering capabilities these applications can be spread across multiple JVM process instances. Tomcat supports clusters of server processes, with transparent load balancing and failover support among the processes. This gives the solution inherent high-availability support. Moreover, cluster instances need not be deployed in the same machines or even the same physical data-center. Tomcat applications typically provide linear scalability; moreover, it is very simple to add additional machines or Tomcat server processes – even within a running production system.

.

### Special Technology

If any special technology was identified in Section 2.5.9 as part of this system, describe the device and the type of location at which it will be installed. This information may be provided using Table 12.

Table 12: Special Technology Requirements

| Special Technology | Description | Notional Location | TRM Status |
| --- | --- | --- | --- |
| <Name> | <Business language description> | <At what type of location will this technology be deployed?> | <Is this technology in the TRM?  (Yes / No)> |

### Technology Locations

This section describes the various technology components that will be used. If known, provide the name of the datacenter at which the technology will be installed. If not, specify as Site A, Site B etc. Provide this information in Table 13.

Table 13: Technology Location Details

| Technology Component  Production 1 | Location | Usage |
| --- | --- | --- |
| Workstations |  |  |
| Special Hardware |  |  |
| Interface Processors |  |  |
| Legacy Mainframe |  |  |
| Legacy Application Server |  |  |
| Legacy Databases |  |  |
| Other |  |  |

| Technology Component  Production 2 | Location | Usage |
| --- | --- | --- |
| <copy from Prod 1 set, or enter new ones as appropriate> |  |  |

| Technology Component  Certification | Location | Usage |
| --- | --- | --- |
|  |  |  |

| Technology Component  Education | Location | Usage |
| --- | --- | --- |
|  |  |  |

| Technology Component  Test | Location | Usage |
| --- | --- | --- |
|  |  |  |

| Technology Component  Development | Location | Usage |
| --- | --- | --- |
|  |  |  |

### Conceptual Infrastructure Diagram

The Infrastructure design focuses on the orchestration and virtualization of computational, network, and storage resources. The Infrastructure Layer describes the overall capabilities provided by the Infrastructure and the products supporting those services.

#### Conceptual Production String Diagram

Create a diagram to show the configuration of a single production string.

Additional components, such as the mainframe, other Web servers, or other major components should be included if they are expected to be required.

Figure 7: Conceptual Production String Diagram

# Data Phase

This section outlines the design of the database management system (DBMS) and non-DBMS files associated with the system. For networks, detail the distribution of data and identify any changes to the logical data model that may occur due to software or hardware requirements.

Note: Provide a data dictionary appendix showing data element name, type, length, source, validation rules, maintenance, data stores, outputs, aliases, and description.

## Data View

A "Data View" should be included in the Architectural Representation whenever persistent data objects are included in the system (they are typically present in most software systems). The data view describes the logical data model of the system and includes an Entity Relationship Diagram (ERD). For a description of Entity Relationship diagramming please refer to the whitepaper <<http://www-106.ibm.com/developerworks/rational/library/content/03July/2500/2785/2785_uml.pdf>>

## Data Dictionary

# Logical Phase

The BCDS architecture employs a layered and modularized approach to implementing the BCDS system architecture. The proposed approach provides a clean separation between the presentation layer, and application services layer, which allows VBA to maximize flexibility and reduce the costs associated with maintaining the BCDS system infrastructure. The functionality provided by the application is accessed via a set of shared business services. The SOA architecture connects, mediates and manages messaging between heterogeneous services, legacy systems, and other external interfaces across the enterprise.



The BCDS Web application makes HTTP/HTTPS calls to modeling engine, reporting engine and services. Business logic components executes SQLs to read/write data from database. All business logic components communicates with local/API calls. External partners will send data loads through FTP/SFTP or email, if we need to load more than once. BCDS system uploads data to third party systems/locations using common protocol(s).

## Interface Architecture

Describe the interface(s) between the system being designed and other systems. Include the interface architecture(s) being implemented, such as wide area networks, gateways, etc. Provide diagrams showing the communications path(s) between this system and other systems.

## Human-Machine Interface

Describe the human-machine interface (i.e., GUI) relative to the user. Additional information may be added if the suggested headings are inadequate.

## Interface Design Rules

Identify conventions and standards for designing the GUI.

## Inputs

Identify the input media used by the user (i.e., operator) for providing information to the system, such as data entry screens, optical character readers, bar scanners, etc.

Identify the messages associated with operator inputs, including the following:

* Form(s) if the input data is keyed or scanned for data entry
* Access restrictions
* Security considerations.

## Outputs

Describe the system output design relative to the user. System outputs include reports, data display screens, query results, etc.

Identify the following, if appropriate:

* Access restrictions or security considerations
* Description of the purpose of the output
* Report requirements, including frequency of periodic reports
* Screen contents. (Provide a graphic representation of each layout. Define all data elements associated with the layout).

## Navigation Hierarchy

Provide a diagram of the navigation hierarchy that shows how a user moves through the GUI.

### Screen [x.1]

Provide the layout of all input data screens or GUIs. Provide a graphic representation of each GUI, for example, a low-resolution screenshot. Define all data elements associated with each screen or GUI, or reference the data dictionary. Label each data input screen and/or GUI.

### Screen [x.2]

Provide a graphic representation of each GUI, for example, a low-resolution screenshot. Define all data elements associated with each screen or GUI, or reference the data dictionary.

### Screen [x.3]

Provide a graphic representation of each GUI, for example, a low-resolution screenshot. Define all data elements associated with each screen or GUI, or reference the data dictionary.

# Process Phase

## Service Oriented Architecture / ESS

The system architecture utilizes the Service Oriented Architecture (SOA) as a foundation for the BCDS technical solution. Service oriented architecture is a technology approach and not a specific hardware or software component. Instead, SOA represents an architectural model in which functionality is decomposed into small, distinct units (services), which can be distributed over a network and can be combined together and reused to create business applications.

# Development Phase

## Software Architecture

The BCDS web application is a lightweight, browser based application that do not require installation of other software on the user's desktop. The web application is browser-independent and is designed to operate across the most commonly used web browsers, including Fire Fox, Apple Safari and Internet Explorer, with only minimal hardware related requirements. In addition to access via the most commonly used desktop browsers, users can access the BCDS web application via mobile hand-held devices.



### Presentation – User interface

The BCDS web applications incorporate usability features for individuals with disabilities and provide the BCDS system with ADA compliance. Additionally the application supports multiple modalities for consumers when interacting with the BCDS system, including support for browsers that work on mobile platforms, including Safari on iPhone and iPad; Chrome on Android phones & tablets and IE on Windows phones and tablets.

### Service Contract and Adaptors

The most basic interaction in SOA is between service providers and consumers. The two parties must come to an agreement or contract that specifies the details of the service the provider is performing. The contract loosely couples the relationship between provider and consumer. The service adaptors efficiently convert data or application requests into native language of the data without requiring custom code.

### Business services and entities

#### Role-based access and user provisioning

The BCDS application’s global navigation menu allows the user to navigate to all functional areas of the application. Users are provided with tab-based navigation that allows the user to enter the rating details in the order.

The application is designed to meet the needs of the various user types such as Model Agent, Rater, Administrator and Modeling Analyst.

The application allows the user to edit field data if they have the appropriate user access levels. Fields that a user does not have access rights for are not viewable by the user. The user only sees those fields they can access based upon role based security functionality provided by the BCDS system. Security is added to the application to prevent users from editing certain data, based on data type and user type.

Access roles, rules, permissions, and groups are configurable in the BCDS administration module. The solution has granular capabilities to define roles, permissions, groups, and rules. These capabilities are restricted to selected administrators only.

#### Reporting

The BCDS reporting engine is a server-based reporting dashboard that offers useful features including charts, matrices, custom layouts, ad hoc reports, custom report items and a variety of presentation formats.

Types of reports include:

* Required standard reports
* Reporting histories for trending and monitoring of operational performance.

#### Modeling Engine

The BCDS modeling engine is a real-time, accurate rating determinations are supported by a modern, flexible Business Rules Engine where non-technical users can participate in the rules creation and maintenance process and that can be provided as a shared service for use by other programs.

By centralizing rules within Modeling engine and sharing rules across BCDS system, VBA can eliminate having to maintain business rules in multiple systems.

### Logging and Auditing

The BCDS application includes a robust audit trail, tracking user transactions to identify who initiates each action. It captures the ID of the person who made the change, the time and date of the change, the network location/IP address of the person, the new and previous values, and the outcome, if any, of the event. This data is accessible by authorized users for reporting and Quality Assurance processes.

### Exception Handling

The user is provided with on-screen validations when invalid entries are made. Error messages presented to the user are clear and expressed in nontechnical terms, providing the user with the information needed to resolve errors and complete a successful interaction with the system. One example is that if a required field on a page is not completed, a user is warned that the data is missing and is provided a link to that particular field.

# Physical Phase

## Hardware Architecture

The BCDS project team is going to work with VBA team to select hardware configurations that will me VBA performance and scalability requirements. This includes utilization of pre-configured hardware, such as Solaris/Linux servers, supporting application needs. The hardware configurations should have the capability to significantly scale to meet future needs. Use of virtualization reduces the costs by lowering processor counts and associated processor-count based software licenses.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Hardware Item # | Hardware Item | Environment | Description | Operating System | Count |
| 1 | Load balancers | Production | Apache HTTP server 8GB RAM and 100 GB Storage | Linux | 2 |
| 2 | DR Router | Production | Cisco | N/A | 1 |
| 3 | DR Firewalls | Production | Cisco | N/A | 2 |
| 4 | DR Switch | Production | Cisco | N/A | 1 |
| 5 | Application Servers | Production | Tomcat and Reports server 16 GB RAM and 1TB Storage | Linux/Solaris | 4 |

## Network Architecture

The BCDS application requires a flexible, scalable, reliable and high performance infrastructure. These requirements are our guiding principles for the following network design. There are a number of network services to be considered when designing a network such as Domain Name Service (DNS), Dynamic Host Configuration Protocol(DHCP), IP Addressing, and network operations.



# Attachment A – Approval Signatures

This section is used to document the approval of the System Design Document. The review should be conducted face to face where signatures can be obtained ‘live’ during the review. If unable to conduct a face-to-face meeting then it should be held via LiveMeeting and concurrence captured during the meeting. The Scribe should add /es/name by each position cited. Example provided below.

The Business Sponsor and Project Manager are required to sign.

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

Signed: Date:

< Business Sponsor >

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

Signed: Date:

< Project Manager >

1. Additional Information

Attach any addition information that supplements the design specification.

* 1. Identification of Technology and Standards

Identify the system and software which apply to the SDD, including: identification number(s), title(s), abbreviation(s), version number(s), and release number(s). Identify all standards (e.g., American National Standards Institute [ANSI], International Organization for Standardization [ISO], Institute of Electrical and Electronics Engineers [IEEE], etc.).

* 1. Constraining Policies, Directives and Procedures

Identify any constraints or requirements placed on this document by policies, directives, or procedures.

* 1. Requirements Traceability Matrix

Include an RTM that traces modules and data structures to the software requirements. A reference to the location of the RTM is also acceptable.

* 1. Packaging and Installation

Outline any special considerations for software packaging and installation.

* 1. Design Metrics

Describe all metrics to be used during the design activity.

Template Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| June 2015 | 2.10 | Changed Heading 1 default setting to eliminate page break before | Process Management |
| May 2015 | 2.9 | Edited for Section 508 conformance and remediated with Common Look Office tool | Process Management |
| February 2015 | 2.8 | Incorporates revisions from PMAS Reform Lockdown; namely removing requirements for information that can be obtained from other PMAS authoritative sources. | Andrew Slawter, Office of Technology Strategies |
| September 2014 | 2.7 | Adds Enterprise Shared Services terms and requires AERB Compliance Certificate attachment. | Process Management |
| August 2014 | 2.6 | Signature block update authorized by AERB CR\_018934 | Process Management |
| March 2014 | 2.5 | Section 508 repairs to new version approved by AERB Chair approved | Process Management |
| August 2013 | 2.3 | Replaced the Service Architecture sub-section with new sub-sections for consumed and provided services. Also applied miscellaneous feedback from VA team. | ASD Enterprise Shared Services (ESS) Work Group |
| June 2013 | 1.3 | Upgraded to MS Office 2007-2010 format | Process Management |
| June 2013 | 1.2 | Address inconsistencies in Section 3, Conceptual Design, Correct headings | Process Management |
| March 2013 | 1.1 | Formatted to documentation standards and edited for Section 508 conformance | Process Management |
| January 2013 | 1.0 | Initial Document | PMAS Business Office |

Place latest revisions at top of table.

The Template Revision History pertains only to the format of the template. It does not apply to the content of the document or any changes or updates to the content of the document after distribution.

The Template Revision History can be removed at the discretion of the author of the document.

Remove blank rows.

See TOGAF® 9.1, Part III: ADM Guidelines & Techniques, Gap Analysis on TOGAF website at <http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap27.html>