Perceptive Reach

Integrated Reach Database System

(IRDS)

System Design Document



Department of Veterans Affairs

November 2014

Version 0.2

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 11/13/2014 | 0.2 | Added additional conceptual design details for the database, analytic platform, risk model, dashboard, business benefits, and user stories. | Bill Balshem, Kristopher Hoover, Robert Snelling, Paul Bradley |
| 10/24/2014 | 0.1 | Added Database, Analytic, Dashboard, Architecture, Networking, Business Benefits. | Bill Balshem, Kristopher Hoover, Robert Snelling, Paul Bradley |
| 10/07/2014 | 0.0 | First Draft | Paul Bradley |
|  |  |  |  |
|  |  |  |  |

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. Per the Project Management Accountability System (PMAS) Guide, the SDD as a conceptual design is required prior to the Milestone 1 Review. The as-built design for each delivery must be incorporated prior to the Milestone 2 Review

Table of Contents

[1. About this document 1](#_Toc403739890)

[2. Introduction 1](#_Toc403739891)

[2.1. Purpose of the SDD 1](#_Toc403739892)

[2.2. Identification 1](#_Toc403739893)

[2.3. Scope 1](#_Toc403739894)

[2.4. User Characteristics 2](#_Toc403739895)

[2.5. Relationship to Other Documents and Plans 2](#_Toc403739896)

[2.6. Acronyms and Abbreviations 3](#_Toc403739897)

[3. Background 4](#_Toc403739898)

[3.1. Overview of the System 4](#_Toc403739899)

[3.1.1. SDR Database Background 6](#_Toc403739900)

[3.2. Overview of the Business Process 6](#_Toc403739901)

[3.3. Business Benefits 10](#_Toc403739902)

[3.4. Assumptions and Constraints 11](#_Toc403739903)

[3.4.1. Design Assumptions 11](#_Toc403739904)

[3.4.2. Design Constraints 12](#_Toc403739905)

[3.4.3. Design Trade-offs 12](#_Toc403739906)

[3.5. Overview of the Significant Requirements 12](#_Toc403739907)

[3.5.1. Overview of Significant Functional Requirements 12](#_Toc403739908)

[4. Conceptual Design 14](#_Toc403739909)

[4.1. Conceptual Application Design 14](#_Toc403739910)

[4.1.1. Application Context 14](#_Toc403739911)

[4.1.2. High-Level Application Design 17](#_Toc403739912)

[4.2. Conceptual Data Design 18](#_Toc403739913)

[4.2.1. Project Conceptual Data Model 18](#_Toc403739914)

[4.2.2. Database Information 20](#_Toc403739915)

[4.2.3. User Interface Data Mapping 20](#_Toc403739916)

[4.2.3.1. Application Screen Interface 20](#_Toc403739917)

[4.2.3.1.1. <Insert name of screen> 20](#_Toc403739918)

[4.2.3.2. Application Report Interface 21](#_Toc403739919)

[4.2.3.2.1. <Insert name of report> 21](#_Toc403739920)

[4.2.3.3. Unmapped Data Element 22](#_Toc403739921)

[4.2.4. Data Import Design 22](#_Toc403739922)

[4.2.4.1. SQL Server Integration Services (SSIS) 22](#_Toc403739923)

[4.2.4.2. Remote Procedure Calls (RPC) 22](#_Toc403739924)

[4.3. Conceptual Infrastructure Design 23](#_Toc403739925)

[4.3.1. System Criticality and High Availability 23](#_Toc403739926)

[4.3.2. Special Technology 24](#_Toc403739927)

[4.3.3. Technology Locations 24](#_Toc403739928)

[4.3.4. Conceptual Infrastructure Diagram 25](#_Toc403739929)

[4.3.4.1. Location of Environments and External Interfaces 25](#_Toc403739930)

[4.3.4.2. Conceptual Production String Diagram 25](#_Toc403739931)

[4.4. System Architecture 25](#_Toc403739932)

[4.5. Hardware Architecture 25](#_Toc403739933)

[4.6. Software Architecture 26](#_Toc403739934)

[4.7. Network Architecture 27](#_Toc403739935)

[4.8. Service Oriented Architecture / ESS 28](#_Toc403739936)

[4.9. Enterprise Architecture 29](#_Toc403739937)

[5. Data Analytics Platform 29](#_Toc403739938)

[6. Risk Model 30](#_Toc403739939)

[7. Data Design 33](#_Toc403739940)

[7.1. Non-DBMS Files 34](#_Toc403739941)

[7.2. Data View 34](#_Toc403739942)

[8. Detailed Design 34](#_Toc403739943)

[8.1. Hardware Detailed Design 34](#_Toc403739944)

[8.2. Software Detailed Design 35](#_Toc403739945)

[8.2.1. Conceptual Design 35](#_Toc403739946)

[8.2.1.1. Product Perspective 35](#_Toc403739947)

[8.2.1.1.1. User Interfaces 35](#_Toc403739948)

[8.2.1.1.2. Hardware Interfaces 35](#_Toc403739949)

[8.2.1.1.3. Software Interfaces 35](#_Toc403739950)

[8.2.1.1.4. Communications Interfaces 36](#_Toc403739951)

[8.2.1.1.5. Memory Constraints 36](#_Toc403739952)

[8.2.1.1.6. Special Operations 36](#_Toc403739953)

[8.2.1.2. Product Features 36](#_Toc403739954)

[8.2.1.3. User Characteristics 36](#_Toc403739955)

[8.2.1.4. Dependencies and Constraints 36](#_Toc403739956)

[8.2.2. Specific Requirements 37](#_Toc403739957)

[8.2.2.1. Database Repository 37](#_Toc403739958)

[8.2.2.2. System Features 38](#_Toc403739959)

[8.3. Network Detailed Design 38](#_Toc403739960)

[8.4. Service Oriented Architecture / ESS Detailed Design 38](#_Toc403739961)

[8.4.1. Service Description for <Consumed Service Name> 39](#_Toc403739962)

[8.4.2. Service Design for <Provided Service Name> 39](#_Toc403739963)

[8.4.2.1. Introduction 39](#_Toc403739964)

[8.4.2.1.1. Purpose and Scope of Service 39](#_Toc403739965)

[8.4.2.1.2. Links to Other Documents 39](#_Toc403739966)

[8.4.2.2. Service Details 39](#_Toc403739967)

[8.4.2.2.1. Service Identification 39](#_Toc403739968)

[8.4.2.2.2. Service Versions 40](#_Toc403739969)

[8.4.2.2.3. Summary of Design and Platform Details 41](#_Toc403739970)

[8.4.2.2.3.1. SOA Pattern(s) Implemented 41](#_Toc403739971)

[8.4.2.2.3.2. COTS Platform vendor names and versions for hosting platform 41](#_Toc403739972)

[8.4.2.3. Dependencies 41](#_Toc403739973)

[8.4.2.4. Service Design Details 41](#_Toc403739974)

[8.4.2.4.1. Interface Technical Specs 41](#_Toc403739975)

[8.4.2.4.1.1. Service Invocation Type 41](#_Toc403739976)

[8.4.2.4.1.2. Service Interface Type 41](#_Toc403739977)

[8.4.2.4.1.3. Service Name 41](#_Toc403739978)

[8.4.2.4.1.4. Interface 41](#_Toc403739979)

[8.4.2.4.1.5. End Points 42](#_Toc403739980)

[8.4.2.4.1.6. Operations or Methods 42](#_Toc403739981)

[8.4.2.4.1.7. Message Schemas 42](#_Toc403739982)

[8.4.2.4.2. Information Model 42](#_Toc403739983)

[8.4.2.4.2.1. Class Diagram and Description of Entities Involved 42](#_Toc403739984)

[8.4.2.4.2.2. Mappings from ELDM to Standards Based Schemas 43](#_Toc403739985)

[8.4.2.4.3. Behavior Model (AKA Use Case Realization) 43](#_Toc403739986)

[8.4.2.4.3.1. Use Cases (Use Case Model) 43](#_Toc403739987)

[8.4.2.4.3.2. Interaction Diagrams 43](#_Toc403739988)

[8.4.2.5. Gap Analysis 43](#_Toc403739989)

[8.4.2.5.1. Variances from Enterprise Target Architecture 44](#_Toc403739990)

[8.4.2.5.2. Variances from SLDs 44](#_Toc403739991)

[8.4.2.5.3. Variances from Standards and Policies 44](#_Toc403739992)

[8.4.2.5.4. Justification for Exceptions and Mitigation 44](#_Toc403739993)

[9. External System Interface Design 44](#_Toc403739994)

[9.1. Interface Architecture 44](#_Toc403739995)

[9.2. Interface Detailed Design 45](#_Toc403739996)

[10. Human-Machine Interface 46](#_Toc403739997)

[10.1. Interface Design Rules 46](#_Toc403739998)

[10.2. Inputs 46](#_Toc403739999)

[10.3. Outputs 46](#_Toc403740000)

[10.4. Navigation Hierarchy 47](#_Toc403740001)

[10.4.1. Screen [x.1] 47](#_Toc403740002)

[10.4.2. Screen [x.2] 47](#_Toc403740003)

[10.4.3. Screen [x.3] 47](#_Toc403740004)

[11. Security and Privacy 47](#_Toc403740005)

[11.1. Security 47](#_Toc403740006)

[11.2. Privacy 48](#_Toc403740007)

[Attachment A – Approval Signatures 49](#_Toc403740008)

[A. Additional Information 50](#_Toc403740009)

[A.1. RTM 50](#_Toc403740010)

[A.2. Packaging and Installation 50](#_Toc403740011)

[A.3. Design Metrics 50](#_Toc403740012)

[A.4. Acronym List and Glossary 50](#_Toc403740013)

[A.5. Required Technical Documents 50](#_Toc403740014)

[A.6. Attach Documents 50](#_Toc403740015)

# About this document

This document is a “work in progress”. The document will be populated with content as the design evolves with each sprint. The early versions of the document will focus on the conceptual design, key business requirements, and the primary user stories. The detailed design will follow. The sections in the document that contain blue instructional text are incomplete. The instruction text will be left in place to provide guidance to authors and readers alike regarding the expectations of the forthcoming content.

# Introduction

VA is seeking to expand suicide prevention to include upstream approaches, designed to reduce initiation or escalation of a risk factor. Upstream suicide interventions target individuals or groups who exhibit biological, psychological, or social risk factors that are more prominent among high-risk groups than among the larger population. Understanding the unique needs of our nation’s Veterans and the military culture as it relates to stigma and mental health is important for early intervention. The goal of the Integrated Reach Database System (IRDS) innovation is to promote the general health of the Veteran population and effectively intervene in issues before they escalate in crisis.

The IRDS solution innovates the current process of risk data collection, analysis, and use in effective intervention strategy. The solution will harness the power of large and diverse data stores to aggregate, analyze and identify risk onset as well as reveal previously unidentified at-risk individuals and populations as a holistic and integrated approach.

The IRDS innovation will serve to bolster the three major components of VHA’s Strategic Plan for Suicide Prevention: surveillance, risk and protective factors, and prevention interventions. The IRDS innovation will target antecedent events specific to Veteran populations prior to the onset of risk to mitigate the development of risk.

## Purpose of the SDD

The purpose of this document is to describe in sufficient detail how the proposed system will be constructed. The SDD translates requirement specifications into a document from which the developers can create the actual system. It identifies the top-level system architecture, and identifies hardware, software, communication, and interface components.

## Identification

TBD: 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.).

## Scope

The IRDS SDD describes the architecture, functional components, and interfaces of the IRDS including the:

* Reach Database – a SQL database storing data used for analytic input
* Data Analytics Platform – an integrated collection of tools
* Risk Model – the predictive model(s) used to identify high-risk Veterans
* Dashboard – a multi-view information portal displaying results from the analytic platform and risk model
* Direct Messaging – a secure messaging solution to notify outreach and intervention resources/clinicians about high-risk Veterans

This document should be read in conjunction with the IRDS Interface Design Specification.

## User Characteristics

There are five user interaction scenarios envisioned, upstream at-risk notification, surveillance, research, reporting, and system sustainment, including:

1. Upstream At-Risk Notification – The primary users in this usage model are the VA outreach and intervention teams. The IRDS shall provide secure notification via a Direct Message of at-risk populations and at-risk individuals to these teams. The application will consolidate various data sources, risk factors, and statistical models to identify at risk individuals and populations.
2. Surveillance – The primary users in this model shall include VA leadership, VA Center of Excellence for Suicide Prevention staff, VA Mental Health leaders, and VA Suicide Prevention Coordinators. The surveillance dashboard will be available through a standard web browser that will be updated in near real-time (minimum weekly) with results produced from the continuous monitoring and processing of linked data sources.
3. Research – The users in this usage model are researchers and statisticians looking to leverage the tools and data available through Reach data analytics platform. The solution will provide a framework for these users to utilize the interfaces provided by the assembled tools to perform required research functions.
4. Reporting – This model shall include both direct and indirect users. The direct users are the individuals required to assemble reports. The indirect users are the consumers or target audience of the reports. The direct users will utilize the interfaces provided by the assembled tools to assemble reports. The report generation process shall be automated.
5. Sustainment - The Contractor shall provide the capability for users to edit and add to the IRDS Risk Stratification Model, permit creation to new models and mapping to interfaces.

## Relationship to Other Documents and Plans

The following IRDS documents may be referenced in tandem with the information recorded here:

* Project Management Plan (PMP)
* IRDS Interface Design Specification
* IRDS Requirements Specification Document (RSD)
* IRDS User Research Report
* IRDS Requirements Traceability Matrix (RTM)

## Acronyms and Abbreviations

Table : Acronyms and Abbreviations

| Acronym | Term |
| --- | --- |
| Army STARRS | Army Study to Assess Risk and Resilience in Service members |
| BIRLS | Beneficiary Identification Records Locator System |
| CD | Compact Disk |
| CDC | Center for Disease Control |
| DoD | Department of Defense |
| ETL | Extract, Transform, Load |
| GB | Gigabyte |
| ICD | International Classification of Diseases |
| IM/IT | Information Management/Information Technology |
| IRDS | Integrated Reach Database System |
| IT | Information and Technology |
| NDI | National Death Index |
| OIT | Office of Information and Technology |
| OMHS | Office of Mental Health Services |
| PMP | Project Management Plan |
| RSD | Requirements Specification Document |
| RTM | Requirements Traceability Matrix |
| SAS | Statistical Analysis System |
| SDCD | State Death Certificate Data |
| SDR | Suicide Data Repository |
| SFTP | Secure File Transfer Protocol |
| SMITREC | Serious Mental Illness Treatment Resource and Evaluation Center |
| SPAN | Suicide Prevention Applications Network |
| SQL | Structured Query Language |
| SSIS | SQL Server Integration Services |
| SSN | Social Security Number |
| TB | Terabyte |
| UI | User Interface |
| VA | Department of Veterans Affairs |
| VCL | Veterans Crisis Line |
| VHA | Veterans Health Administration |
| VSSC | VHA Support Service Center |

# Background

## Overview of the System

The IRDS development and field pilot combines technology, outreach and clinical support to realize a clinically based data-driven early intervention and treatment solution aimed at suicide prevention. The application will include capability for analyzing multiple and integrated data sets with cutting-edge data analytic techniques and visualizations to identify at-risk individuals and populations and provide proactive and secure notifications of these results to Veteran support services.

A central component of the IRDS project is VA’s Suicide Data Repository (SDR.). The SDR is a centralized SQL Server database consolidating multiple sources of data containing suicide and mortality data of Veterans. The IRDS project proposes to expand the capabilities of the SDR to include new interfaces to clinical data sources, integrated data analytics/predictive modelling capabilities, a surveillance dashboard, and secure messaging – thus enhancing SDR capabilities to include upstream suicide intervention.

As shown in **Figure 1**, IRDS will be an integrated system that builds upon the existing SDR with the following capabilities:

* **Reach Database.** A SQL database used to aggregate new data sources and relevant SDR data.
* **Data Analytics Platform and Dashboard.** An integrated collection of analytics and visualization tools, including a surveillance dashboard aimed at identifying at-risk individuals and populations.
* **Direct Messaging.** A method to construct and transmit a secure message to authorized outreach and intervention service providers.
* **Outreach and Intervention.** A pilot workflow that includes the process by which outreach and intervention resources are notified and act upon the data provided.



Figure : Conceptual Program Design

The underpinning technology and data analytics platform will provide methods by which at-risk populations and individuals can be identified. Specifically, we propose a programmable and configurable solution that can be tailored and enhanced over time as more data sources become available and as clinical research identifies new risk factors. As depicted in the bottom-center of  **Figure** **2** a significant component in this effort is the identification of Veteran-specific risk factors, a precursor to the design of an automated reporting model.

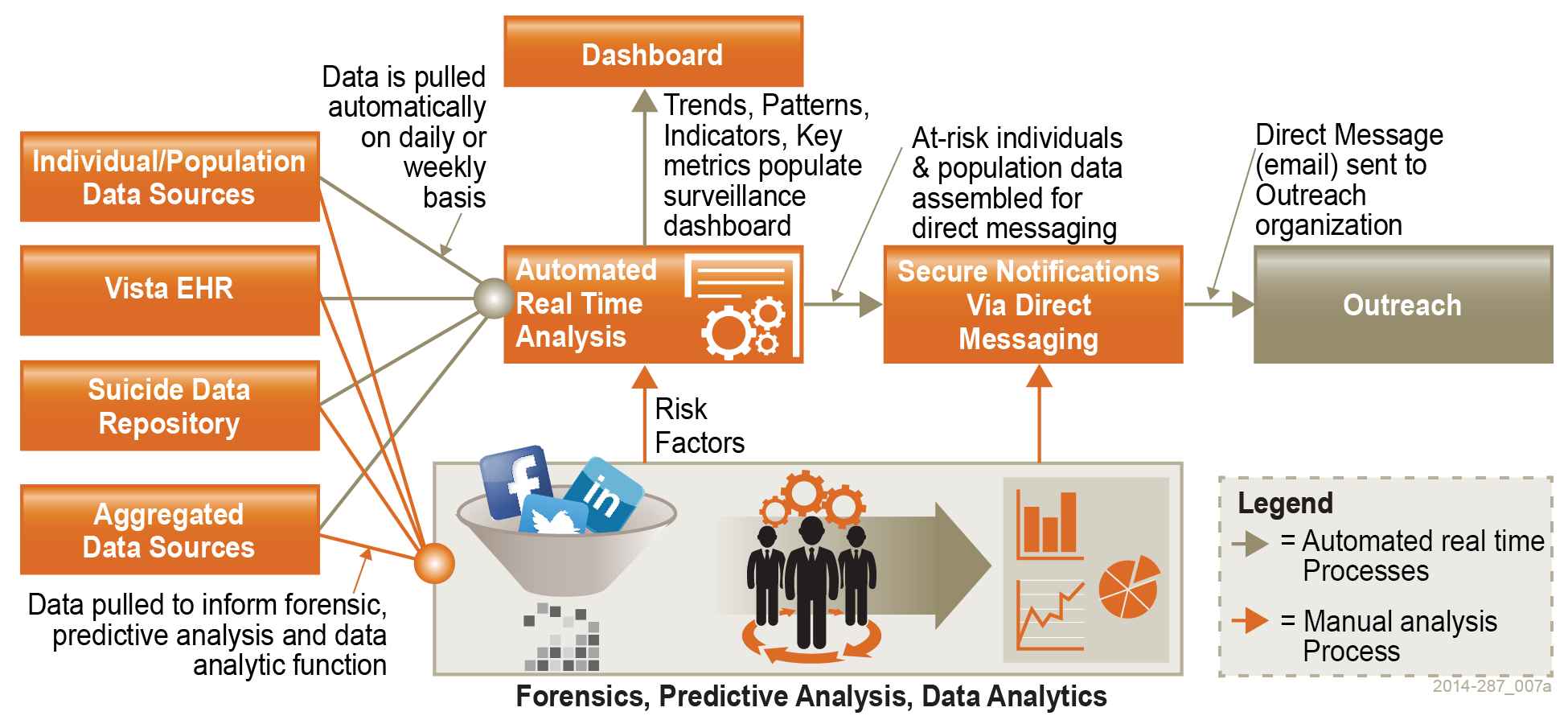


Figure : Conceptual System Design

This effort includes data analysis and predictive modelling, done in collaboration with VA stakeholders and clinical subject matter experts, while at the same time leveraging research data from sources such as DoD (e.g. Army Study to Assess Risk and Resilience in Service members [Army STARRS] and the Military Suicide Research Consortium) and VA (e.g. Center of Excellence for Suicide Prevention). Harnessing the automated reporting model, information will be presented in a customizable national surveillance dashboard and transmitted to authorized officials and Veteran support services organizations via Direct Messaging.

### SDR Database Background

As previously mentioned, the SDR is a centralized SQL Server database, which currently consolidates multiple sources of data containing suicide and mortality data of Veterans. The SDR will be one of the data sources for the IRDS innovation. Data from the sources below are periodically imported into the SDR. Sources include:

* National Death Index (NDI) based on DoD and VA NDI search criteria
* State Death Certificate Data (SDCD)
* Suicide Prevention Applications Network (SPAN)
* Veterans Crisis Line (VCL)

The SDR represents a significant step towards the deployment of a comprehensive suicide surveillance system as it enables a systematic collection of data on completed suicides, attempted suicides, and suicide ideations. The analytical value contained within the rich data sets of the SDR is largely untapped but primed for extraction via digital tagging, discovery, and analysis. The SDR provides a means for VA to quantify and monitor the scope of the suicide problem by analyzing SDR data using analytics to identify characteristics associated with higher or lower risk of suicide, and track changes in the suicide rate over time to evaluate the effectiveness of suicide prevention interventions.

Importantly, the SDR provides a view into the unique characteristics of Veterans pertaining to mortality, suicide ideations, suicide attempts, crisis line communications, and mental healthcare. The data contained within the SDR combined with additional real time clinical data and other data sources, such as public records, presents an opportunity to more accurately identify at-risk Veterans using data unique to Veterans.

## Overview of the Business Process

[Section 1.4](#_User_Characteristics) of this document outlines the major use cases associated with the IRDS application. Of these use cases, the ones that have the most significant business process component are Surveillance and Upstream At-Risk Notification. The business process diagrams below represents a high level overview of how the IRDS application will integrate within these use cases, however it should be noted that the details of each business process may vary on a case by case and facility by facility basis. As more business processes are identified during the development process, this document will be updated.

1. Data sources are imported into the IRDS system via SQL Server Integration Services (SSIS) import solutions. Each data source will have its own SSIS package (.dtsx). In the case of importing data from a VistA installation, one or more RPC calls will be executed from within an SSIS import. The import solution will transform the data and load into the appropriate tables in the Reach database.
2. An R program is run periodically (every year or so) on the production server to update the Risk model. The results of the run are stored in a table in the Reach database.
3. On a regular basis (daily, weekly) a SQL Server process runs that does surveillance against a list of veterans tracked in the Reach database tables against the risk model. The process identifies who are at high risk of attempting suicide.
4. The results of the surveillance run are emailed to the appropriate contacts for those veterans via a secure message that leverages the VA VLER architecture.
5. A system user opens up their dashboard via a compatible web browser and a client side java component connects to a server side java component, which queries the Reach database for both specific and aggregate data regarding high risk veterans at their management level (region, state, VISN, VAMC). The query results are passed to the client browser and populated in the web page.



Figure : Dashboard Surveillance



Figure : Direct Message, New Veteran Identified



Figure : Direct Message, Veteran Data Update

Table : Business Processes

| Business Process ID | Business Process Name | Type | Owner | Description |
| --- | --- | --- | --- | --- |
| 1 | Dashboard Surveillance | Modernized | VA Mental Health Staff | High-level description of how VA staff identifies high-risk Veterans and adds them to the local “High Risk List,” the primary tool for tracking and monitoring high risk Veterans within a facility’s service area. Staff will also use the dashboard for ad hoc research and lookups not associated with a defined business process. |
| 2 | Direct Message, New Veteran Identified | New | VA Mental Health Staff | Description of automated message generation and delivery when the application has identified a Veteran not previously displayed in the system. |
| 3 | Direct Message, Veteran Data Update | New | VA Mental Health Staff | Description of automated message generation and delivery when the application identifies a Veteran who has been previously identified, but has a significant negative event or trigger in the database which may induce an increased risk for suicidal behavior. |

## Business Benefits

Surveillance, identification of risk and protective factors, and interventions are three components of a holistic suicide prevention program. IRDS aims to implement and test this approach using data, technology and clinical expertise to establish a systematic approach, automating the data collection, data analysis/predictive modelling, identification of risk factors and Veterans at risk, notification, reporting, and continuous monitoring processes.

The IRDS concept will address two major elements of suicide intervention; information and time.

* Can information on Veterans both under and outside VHA care be aggregated to produce a useful prediction of suicide risk?
* Can effective interventions be developed and deployed in time to avoid problems from escalating into crises?

We contend that the proposed model will realize a real-time surveillance and intervention solution that will answer both questions in the affirmative.

|  |
| --- |
|  |
| **Source:** CRS analysis of major components of U.S. Department of Health and Human Services (HHS) Office of the Surgeon General and National Action Alliance for Suicide Prevention, 2012 National Strategy for Suicide Prevention: Goals and Objectives for Action, Washington DC: HHS, September 2012 |

Figure : Surveillance Process Model

|  |
| --- |
| Impact |
| * The IRDS innovation will dramatically change the manner in which VA plans, funds, manages, and assesses suicide intervention and prevention programs. * IRDS will provide a cost effective framework from which health data can be studied, hypotheses tested, and where proven analytic methods can be automated, including the automation of report and notification messages. * As risk factors change over time, and as new risk factors are identified, new analytical models and new data sources can be incorporated into IRDS and the method of identification, notification, and intervention can be re-applied. * Through near real time analysis, and a surveillance dashboard, IRDS will allow VA to respond to regional and temporal events and trends with more agility and precision, while also providing a means to monitor and measure the results from specific initiatives. * Likewise, the precision provided in the system will reduce program expenditures as more focused funding can be applied, and ineffective programs can be assessed and terminated. |
| Benefits |
| * The most significant benefit of the IRDS innovation will be the reduction in attempted and completed suicides through early identification, effective intervention, and early treatment. * Due to the early identification and treatment for a broad range of clinical, socioeconomic, and environmental conditions, outreach and intervention programs leveraging IRDS will promote wellness and are therefore likely to decrease the probability of more serious health conditions in the future. * The IRDS solution is aimed at identifying individuals and populations with characteristics that may, if left untreated, increase the probability of future suicidal crises. That is, we are looking for early warning signs, and the application of preventative care that will reduce suffering, suicides and treatment expenses. * The organization and visualization of near real time information will simplify VA business processes, minimizing or eliminating the costs of producing static reports, and eliminating the costs of actions taken on stale data. * Program funding and resources can be optimized and tailored to specific regional needs, preventing waste. * The IRDS innovation will highlight the need for policies and governance surrounding the use of public and non-public data to manage both population and individual health outcomes. |
| Scalability |
| * The IRDS system will provide a framework of data aggregation, data analysis/predictive modelling, reporting, notifications, and visualizations. The integrated system and each of the individual components will be defined and architected using standards and design paradigms that enable interoperability and scalability. * This framework provides a platform on which new analytic techniques, tools, and theories can be tested and studied, eliminating the need to construct new research platforms from scratch – thus saving time and money on future programs. * This framework and reference design will be the basis from which an enterprise level solution can be designed, implemented, and deployed within VA, consistent with OneVA Enterprise Architecture principles and requirements. |

Figure : System Benefits

## Assumptions and Constraints

This section describes the assumptions, and constraints that impacted the design of the system.

The details of the system design are subject to change as requirements are being gathered in parallel with development.

### Design Assumptions

Identify any specific assumptions that were made which influenced the design of this system.

### Design Constraints

* System designers have attempted to utilize open source tools wherever possible. This design of the user interface / front end presentation layer of the system, testing tools, and statistical / analytics tools.
* System designers used VA tools approved for use in the VA Technical Reference Model (TRM) or have requested a waiver for any tools not included in the TRM

### Design Trade-offs

Discuss the trade-offs involved with the design chosen and the reasons for your choices.

Example 1: an increase in security controls will likely entail a decrease in ease-of-use

Example 2: an increase in the flexibility of a system will entail a decrease in the simplicity of that system

For this reason, the designer must decide to put a higher value on some attributes over others. Some areas to consider include:

* Flexibility
* Interoperability
* Performance
* Reliability and robustness
* Usability (including 508 compliance)

## Overview of the Significant Requirements

The material in this section is not to replace either the existing functional or technical requirements documents, nor serve as the basis for the Requirements Traceability Matrix, but only to inform non-project personnel reading this document of the basis for the design.

### Overview of Significant Functional Requirements

The table below includes an overview of the major user requirements, or “Epics,” associated with the proposed solution. A full listing of the project’s Product Backlog is maintained in Jira[[1]](#footnote-2). In addition, a snapshot of the project’s major functional requirements will be included in the project’s RTM. The PR tags in the table below, for example, “PR-158,” derive from the tracking system in Jira.

Table : Functional Requirements

|  |  |
| --- | --- |
| ID (JIRA Ticket Number) | **Description** |
| PR-158 | As an Outreach Provider, I want to be sent secure notification via a Direct Message of at-risk Veterans and populations so I can provide outreach services to these groups. |
| PR-160 | As a member of VA leadership, VA Center of Excellence for Suicide Prevention staff, VA Mental Health leaders, or VA Suicide Prevention Coordinator, I want to view a surveillance dashboard with results produced from the continuous monitoring and processing of linked data sources so I can monitor and understand Suicide Outreach outcomes. |
| PR-161 | As a Researcher I want to access the access the tools and data in the application so I can perform research-related tasks and projects. |
| PR-162 | As a Researcher, I want to generate reports using the data and automated tools in the application so I can use reports as management and communication tools. |
| PR-163 | As a User, I want to edit, add to, and create new IRDS Risk Stratification Models and mapping to interfaces so the application can be updated over time. |
| PR-346 | As an Outreach Provider, I want to view a Direct Message that highlights Veterans at high risk for suicide so I can provide outreach services to them. |
| PR-349 | As an SPC, I want to log in to the Perceptive Reach application. |
| PR-351 | As an SPC, I want to see data from my "home" facility when I log in. |
| PR-352 | As an SPC, I want to view newly identified at-risk Veterans during a specific time frame. |
| PR-353 | As an SPC, I want to view information about groups of at-risk Veterans filtered by geographic area. |
| PR-354 | As an SPC, I want to view information about groups of at-risk Veterans filtered by other data attributes (see details in Jira ticket). |
| PR-355 | As an SPC, I want to view information related to the change in suicide rates over time. |
| PR-356 | As an SPC, I want to pick and choose which screen elements I see on the dashboard, so I can first see only the data that is important to me. |
| PR-357 | As an SPC, I want to move screen elements I see on the dashboard so I can customize the look of the dashboard to suit my preferences. |
| PR-505 | As an Outreach Provider, I want to view a Direct Message when a Veteran experiences a high risk trigger or event, so I can provide outreach services to them. |
| PR-521 | As a Perceptive Reach user, I want to see a sortable / filterable list of high risk veterans. |
| PR-522 | As a Perceptive Reach user I want to click a high risk Veteran list so I can see more detailed information about the Veteran I selected. |

# 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

This section provides the conceptual design of the application that is being produced by this project. [There should be a “To-Be” and a “This-Increment” view of the design, in addition to an “As-Is” view if an existing system. The “To-Be” view should include the future application context, and application high level design. The “This-Increment” view should have application context and high level design.]

### Application Context

The following figure represents the context in which the application will exist.

Please 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.

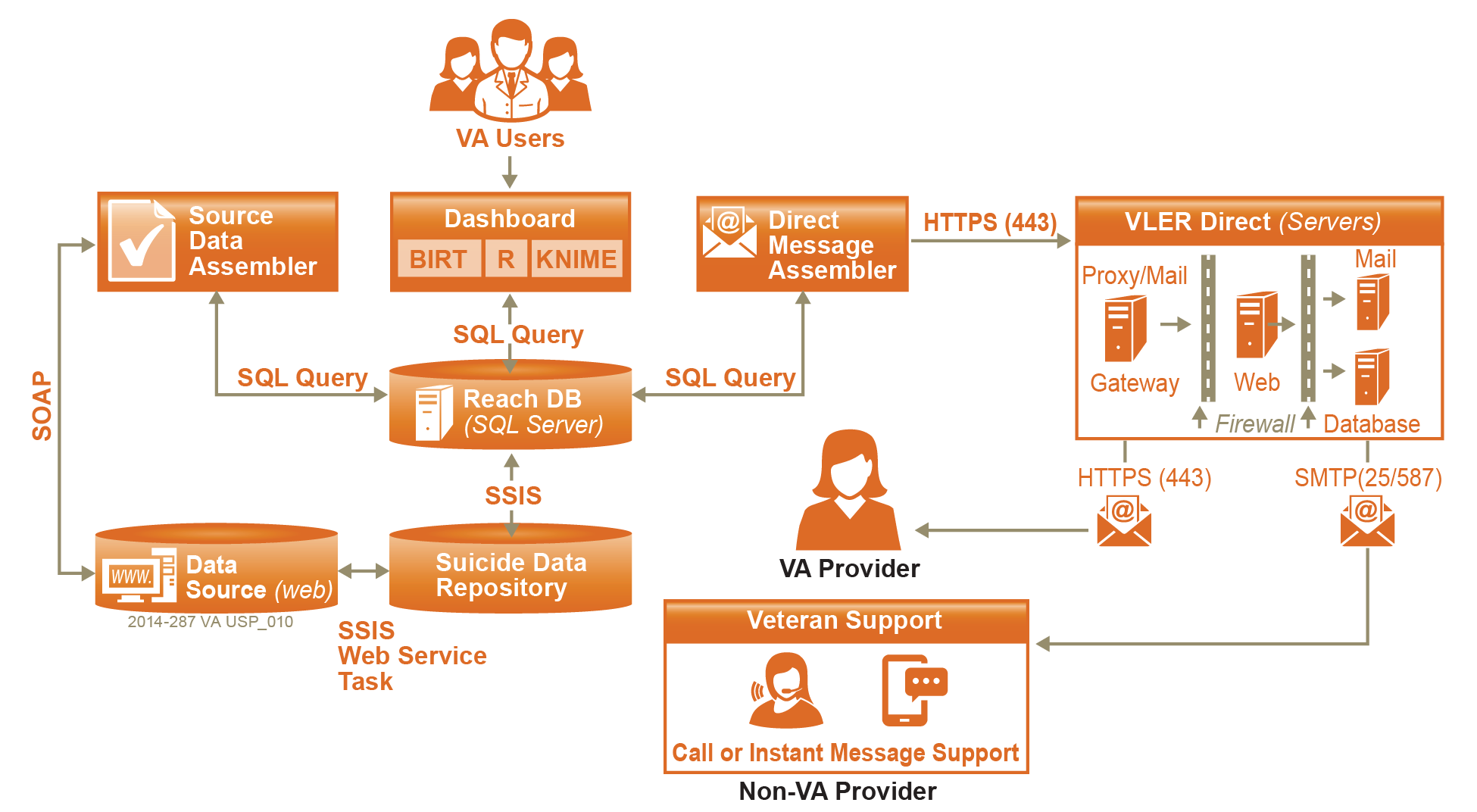


Figure : System Diagram

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 : (Grouping) Application Context Description Object

| ID | Name | Description | Interface Name | Interface System |
| --- | --- | --- | --- | --- |
| < ID from diagram> | Improved Risk Model | Risk model to predict risk of suicide for veterans | <Name of each of the Interfaces to this object> | <Systems with which this system interfaces> |

Interfaces

| 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> |

Table : IRDS Data Sources

| **ID** | **Name** | **Description** | **Internal to VA** |
| --- | --- | --- | --- |
| 1 | VA Suicide Data Repository (SDR) | Periodically imports data from 4 VA data sources into SQL format:  1. National Death Index (NDI) Mortality Search Results  2. State Death Certificate Data  3. Suicide Prevention Applications Network (SPAN)  4. Veterans Crisis Line | X |
| 2 | Corporate Data Warehouse (CDW) | Warehouses multiple VBA and VHA data sources in SQL Format. | X |
| 3 | Veterans Health Information Systems and Technology Architecture VistA | Electronic health record system for VA patients. | X |
| 4 | LexisNexis | Provider of multiple data sources such as legal, risk management, corporate, government, law enforcement, accounting, and academic. (POTENTIAL DATA SOURCE) |  |
| 5 | Public Access to Court Electronic Records (PACER) | Electronic public access service that allows users to obtain case and docket information online from federal appellate, district, and bankruptcy courts. (POTENTIAL DATA SOURCE) |  |

### High-Level Application Design

The High-Level Application Design identifies the major components of the application and the relationships of the major application components to each other and to the surrounding applications. The major components of the application are at the subsystem or top-level service area. Many different graphical formats are acceptable for the High-Level Application Design Diagram. Lower-level services will be defined and documented in the Logical Application Design section.

**Error! Reference source not found.** illustrates a High-Level Application Design in the form of a dataflow diagram. This diagram differs from the diagram in Figure 2 in that the single object representing this system in Figure 2 is decomposed into its major components. Use

The IRDS System will involve visualization tools for the IRDS System. Data visualization tells the story of the analysis results using charts, tables, and other graphics. Data visualization is the step that makes the body of data analytics work accessible to a broad range of stakeholders.

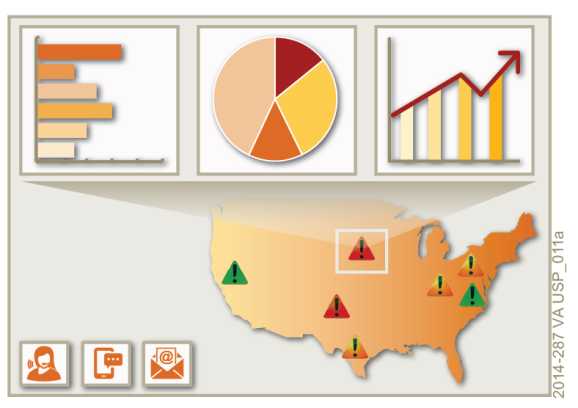


Figure : Dashboard Mockup

The surveillance dashboard will be accessed through a standard web browser. The dashboard will be configurable, providing different user groups distinct views that meet their business needs. As shown in the figure the dashboard will be designed to support interactive viewing and formatting, and created using the open source tools. The design of the dashboard will involve the input of stakeholders and end users. The application will leverage a Node.js runtime environment which will manage the server-side functions (http server, database connections, APIs, and webservice interfaces). The visualization components will be browser based and built using Angular.js, an open source web application framework. This will provide for flexibility and ease of coding.



Figure : System Communication

## Conceptual Data Design

### Project Conceptual Data Model

The Reach database will store source imported data on 2 levels:

* Veteran Demographics
* Veteran Case Data

For each veteran, a master record will be created for that individual. A record for that be created in the 'Veterans' table and a unique ID will be created for that individual within the reach system 'ReachID'. The table will also contain basic demographic for that individual, such as Name, SSN, DOB, Gender, etc... The table also contains a 'Score' field, which the surveillance piece of this system will populate to for that vet to determine if he or she is at high risk. If another record is imported into the system for that individual from another source, the currently existing Veteran record that exists for the vet will be updated with any additional demographic data the additional source contains.

All other data elements imported for veterans will be stored in case level tables. There will be a one to many relationship from the Veteran table to any case level table. For instance, any data that may be available from the SDR on previous suicide attempts for that individual will stored in a suicide attempt table. For each suicide attempt that is documented for a veteran, a 'SuicideAttempt' record will be created, including a ReachID column that will link the suicide Attempt record back to the veteran.

The Risk Factors determined by the IRDS Risk Model will captured in the RiskFactors table. This table will be updated any time the Risk model is updated, by running the methodology encapsulated in an R program that exists on the IRDS server.



Figure : IRDS Data Model

### Database Information

Table : Database Inventory

| Database Name | Description | Type | Steward |
| --- | --- | --- | --- |
| <Reach | SQL Server database(s) that will   1. Import data from SDR and other internal/external sources 2. Store Analytics output to be used by IRDS dashboard and messaging | Create | ? |
| SDR | See section 3.1.1, IRDS Data Sources table | Interface |  |

### User Interface Data Mapping

This section describes and defines the format and information that will be available for users of the product to be able to enter data into the database or to retrieve information from the database, if applicable.

TBD – The PwC Analytics team is currently working with the VA to determine:

* Which VA sources will be imported into the Reach database
* Which, if any external sources will be imported into the Reach database
* What data elements from SDR will be imported into the Reach database

#### 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 :*<screen name>* Screen

Table : *<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 : *< Report name>* Report

Table : *<Report name>* Description

| Report Column | Data Source *<TableName. 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.

### Data Import Design

#### SQL Server Integration Services (SSIS)

SSIS will be the primary tool for importing external data sources into the IRDS Reach database

For a specific data import, an SSIS package will be developed to

1. Make a connection to the source (SQL table, text file, other)
2. Import the data into a staging area
3. Make the appropriate data transformations (cleaning, standardization)
4. Load the transformed data into the appropriate Reach data store tables

The execution of SSIS packages (.dtsx files) can be automated by scheduling them as a Windows process via SQL Server Agent.

#### Remote Procedure Calls (RPC)

Data will be imported into the IRDS system directly from the Vista using RPC calls. VistA data is stored against a MUMPS back end, which uses text based files for data storage. For each source of VistA data imported from into IRDS:

1. Either a custom RPC will be written (in M) or a currently existing one will be leveraged
2. An automated java process will execute the RPC and return the query results in text format
3. Those results will be stored in a flat file on the IRDS server to be imported into the reach database via a SSIS package (using the steps listed in the section above)



Figure : IRDS Data Import Process Flow

## Conceptual Infrastructure Design

The Conceptual Infrastructure Design should describe any unique technology that will be used, which are either part of this system, or will attach to this system.

All information should be provided to the extent that it is known. Because the system is at a preliminary design stage, it is expected that the information provided may need to be changed during later design stages or increments.

The Conceptual Infrastructure Design is a high-level overview of the infrastructure that will be used to support the application. Primary emphasis is on the environments that will be required and the locations at which they will be installed. The Conceptual Infrastructure Design becomes more detailed at later stages as more information is collected regarding the system, and the infrastructure requirements (i.e., capacity requirements) are better known.

### System Criticality and High Availability

Describe the approach that will be taken to meeting the system criticality and high availability requirements identified in Section 2.5.6, including the extent to which geographically distributed, high availability designs are planned. Describe the approach that is taken towards high availability as well as any workload distribution scheme that is planned to support the high availability implementation (e.g., restricting updates to a single node).

If the system is not mission critical and high availability is not required, then describe the approach that will be taken to provide the requisite level of availability and disaster recovery.

### 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 : 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 : 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

#### Location of Environments and External Interfaces

Create a diagram to show the environments that will be supported. As illustrated in TBD, the diagram should show the following:

* Local networks to which they will be attached (Production, Test, or Development)
* Locations at which they will be installed
* External connections (each external interface should be shown in terms of where it enters the network).

#### Conceptual Production String Diagram

Create a diagram to show the configuration of a single production string to the extent that it is known. It is likely that this diagram will be highly notional and may show such items as enterprise service bus, application servers, and database servers.

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

Figure : Conceptual Production String Diagram

## System Architecture

The system developed under the Perceptive Reach (IRDS) project will be designed to run on a cloud-based environment consistent with the principles of OneVA EA.

## Hardware Architecture

The solution will be deployed within the VA’s enterprise environment. Initial Requirements (Cloud Based – Prototype Server):

* MS Windows Server 2012 64-bit
* Intel Xeon CPU E5-2670, 2.6GHZ
* 16GB RAM
* 60GB HDD
* MS SQL Server 2012 Enterprise Edition
* Users: 16 (8 simultaneous logins, minimum)

Planned Requirements (Cloud Based – Development/Test Server):

* MS Windows Server 2012 64-bit
* Intel Xeon E5-2600 Family (2670 or 2690), 2.6GHZ or better
* 32GB RAM
* 500GB HDD
* MS SQL Server 2012 Enterprise Edition
* Users: 16 (8 simultaneous logins, minimum)

Future Requirements:

* TBD

For further details on the Hardware Architecture, please refer to section 5.1 Hardware Detailed Design.

## Software Architecture

The IRDS innovation will develop and demonstrate a new SQL database that aggregates both VA and non-VA data sources to be used to facilitate identification of at-risk individuals and populations, an integrated data analytics solution that includes open source data analysis and visualization tools, and an open standard based secure messaging solution to inform authorized individuals of analysis results.

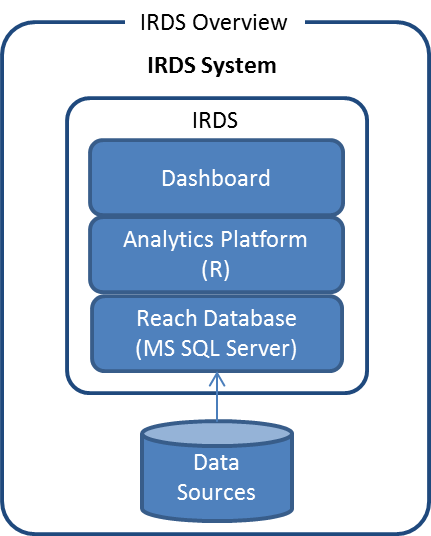


Figure : Architecture Overview

**Reach Database:**

* Platform: SQL Server 2012
* Data Importing and ETL Tools: SQL Server Integration Services, FMQL or RPC (for interfacing directly with VISTA)

**Analytics Platform:**

The enhanced risk model will be coded in the open source statistical language R. Once an enhanced model is finalized, the model coefficients will either stay in R or be transitioned into SQL code. If the model remains in R, this may make running the model daily potentially a more difficult process due to lengthy data processing times. The alternative to this possible issue might be to store the data processing and model coefficients in SQL. If this approach is used, R would still be used at a specified frequency to refresh the parameter estimates of the risk model.

* R 3.1.2 (a language and environment for statistical computing and graphics)
* RStudio (a powerful and productive user interface for R)

**Dashboard:**

The role of the dashboard is to provide visualization tools that display the current status of metrics and key performance indicators (KPIs) for IRDS. The dashboard will consolidate and arrange preselected metrics and analytic results on within a browser based web application. Node.js will provide the base architecture for the web server providing the primary services of the dashboard through HTML, JavaScript, and CSS to drive the behavior of the application.

* Node.js (JavaScript runtime engine)
* Express.js (Node.js module used to handle routing of HTTP calls)
* HTML, JavaScript, and CSS
* Angular.js (JavaScript framework for extending HTML)
* Bootstrap (powerful front-end framework for faster and easier web development)

For further details on the Software Architecture, please refer to section 5.2 Software Detailed Design.

## Network Architecture

The following illustration shows the network notional network configuration leveraged by the IRDS system. The system will utilize VA LAN and WAN networking resources to transfer data from various data repositories to the IRDS database, and to support direct message emails sent to VAMC resources and to external resources such as partners at Rutgers UBHC.

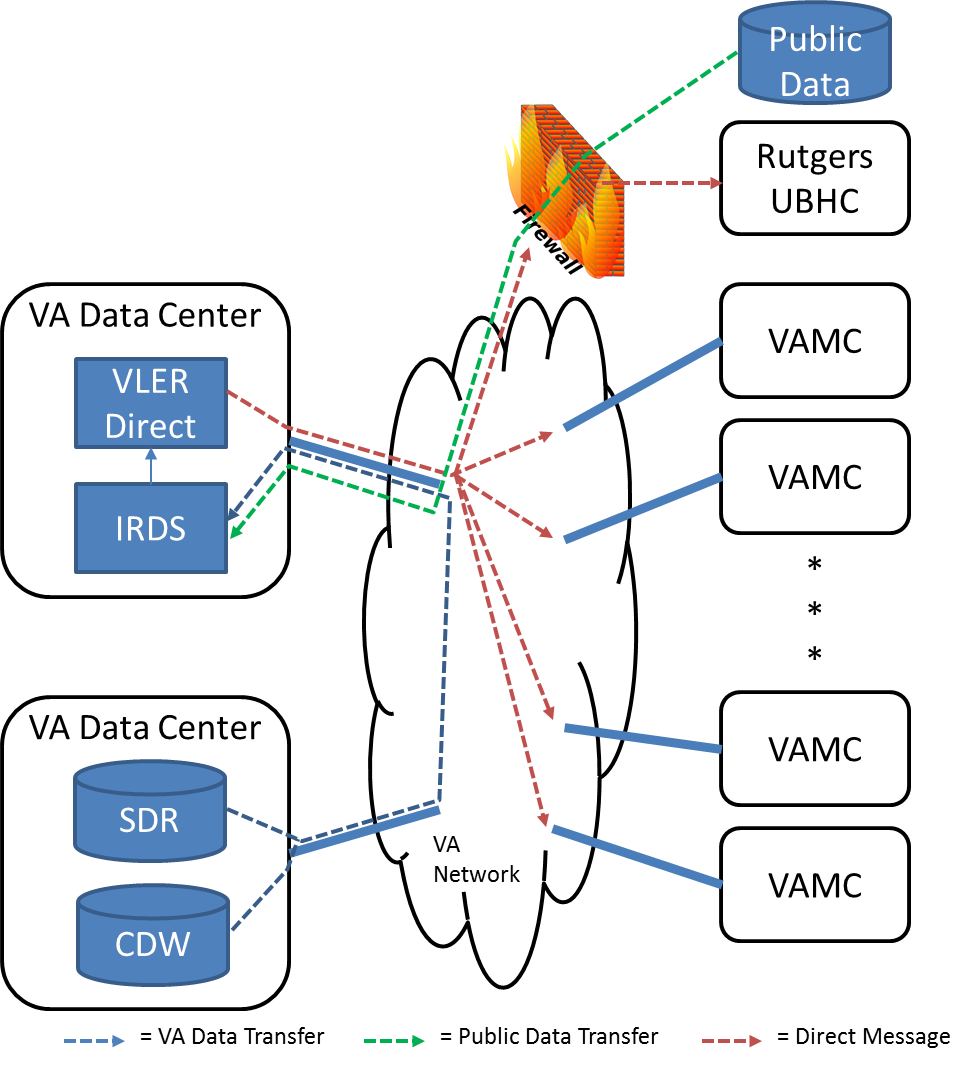


Figure : Notional Network Architecture

## Service Oriented Architecture / ESS

While the IRDS may be extended to provide Enterprise Shared Services in the future the innovation objectives do not include the delivery of web services functionality. The IRDS provides a dashboard application accessed through a web browser, and secure messaging delivered through email client or web portal. The Data Analytics Platform will be access through direct access to the server on which the IRDS resides. IRDS will consume the VLER Direct ESS to deliver Direct Messaging.

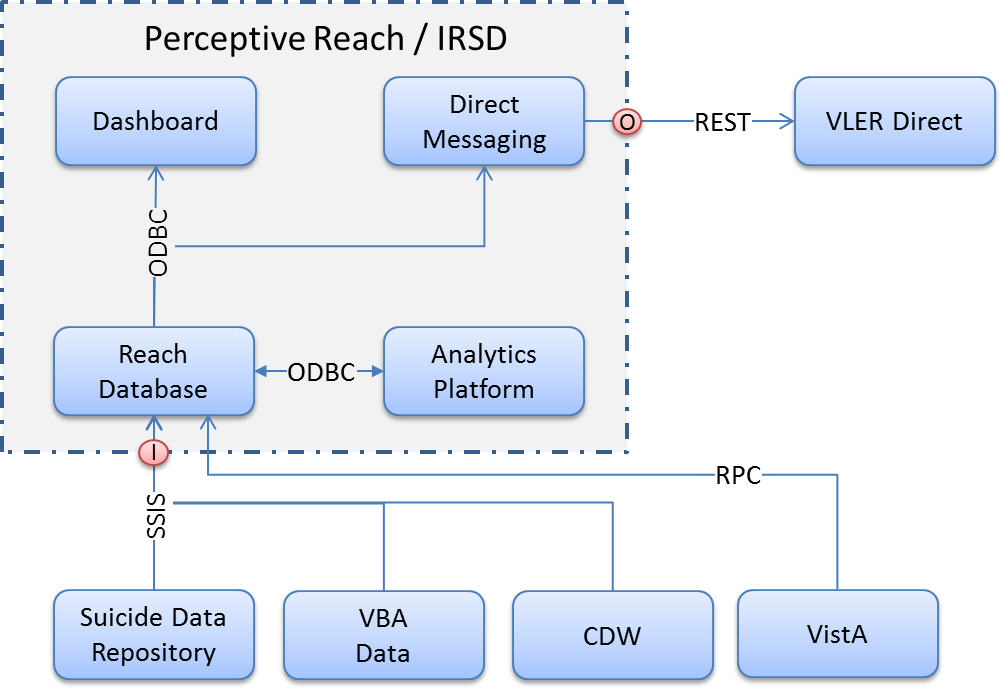


Figure : Architecture Overview

## Enterprise Architecture

IRDS conforms to the principles of OneVA Enterprise Architecture, utilizing technologies approved on the VA Technical Reference Model (TRM), open and standardized interfaces. Through standard design patterns and the use of virtualization and cloud technology the IRDS architecture supports portabability, modularity, and scalability consistent with VA directives.

# Data Analytics Platform

To assess the functionality of the current Risk Model and its potential modifications/improvements, the Perceptive Reach team requires that the appropriate analytical tool is made available in the Perceptive Reach environment. There are a number of criteria that an analytical tool must meet for it to be appropriate for use:

* Open source tool
* Can run on unix/linux
* Can connect to various databases
* Ability to calculate statistical tests: t-test, chi sq test, etc and basic modeling: linear, logistic, etc
* Can execute advanced analytical/statistical models
* Can import, merge, manipulate, and export data
* Can visualize data
* Is approved for use in the VA environment
* Experienced users/programmers of the tool are members of the Perceptive Reach team

A number of tools were assessed to determine whether they satisfy each of the abovementioned criteria: SAS, SPSS, Stata, R, Python, Weka, Gephi, Eclipse, Business Intelligence and Reporting Tools (BIRT), Jasper Reports, Konstanz Information Miner (KNIME). The only tool that, met all of the criteria was R.

R is the leading open source statistical analysis tool used across many disciplines. The strength of R lies in its large and devoted academic and industry user base. This has allowed for the language/software to contain a wide variety of in-built capabilities and remain a cutting edge tool. Since the model improvements for the Risk Model have not yet been identified, the tool that will be used for the analysis must have a wide range of capabilities, which R has. Because of this, it is unlikely that R will have substantial applicability limitations when assessing the model and considering/implementing model improvements.

# Risk Model

The main goal of the modeling effort is to continue to utilize and improve the existing suicide completion risk model developed by VA so that (1) near-time data on veterans can be used to predict the risk of suicide completion for a specific veteran and (2) the VA Suicide Prevention Coordinators and outreach staff are notified of veterans with elevated risk levels for suicide and can take appropriate preventative actions. The current risk model uses a logistic regression framework and approximately 380 inputs from VHA and NDI data. Using this model as a starting point, potential improvements will be considered and implemented if they are shown to improve the out-of-sample predictive power of the existing model. Potential improvements may include but will not be limited to: new data sources, alternative input variable definitions, testing of variable interactions, and alternative model structure.

New data sources may be utilized and variables constructed and tested to determine if they can be used as enhancements to the current risk model. Alternative methods to logistic regression may be considered to determine if such can improve the predictive power of the current risk model.

The strength and robustness of a predictive model is dependent on the data inputs that are used to develop the model. Thus, identifying a robust and reliable list of potential data inputs is essential. By reviewing academic literature and holding discussions with clinicians and other subject matter specialists on both risk and protective factors for suicide ideation and completion, a number of variable categories have arisen as critical inputs into the risk model: demographics, clinical, and socioeconomic. It is expected that the list of specific data inputs will increase as academic literature review and discussions with subject matter specialists continue.

After identifying the data points potentially useful for model development, data sources will need to be identified that capture the information on relevant variables. The current risk model uses data from VHA and NDI. It is expected that some of the additional data inputs will be derived from the same VHA and NDI data sources; however, additional data inputs will be considered from new data sources. One of the data sources that may be useful for model development is VBA. From the initial discussions with subject matter specialists on the VBA data, there are multiple datasets at VBA that contain information on veterans including: military service, financials, medical, and demographics.

Model input data will be housed within a SQL Server database behind a VA firewall. To be able to access the data, R statistical software will connect directly to the database to access the data for analysis.

The ultimate output from the enhanced risk model will be a veteran-specific risk score based on the predicted likelihood of suicide risk. The risk score may be a numeric score bound between two limits, an unbounded numeric score, an ordinal categorical score (for example: High, Medium, Low), or an unordered categorical variable (for example: PTSD, Divorce, Family Death, etc). The final decision on risk score methodology will be made after the risk model is finalized and dashboard/notifications end users are consulted.

The outputs from the risk model will be stored in the SQL Server instance that also houses the raw data inputs into the model. Since the exact output of the statistical model is not yet defined, there is no definition as to the exact storage protocol of model outputs, only that they will be stored in the SQL Server. The logic behind storing the model outputs in the SQL Server is to allow the dashboard and messaging applications to pull these data from a single source rather than multiple sources.



Figure : Risk Model Data Sources

**Suicide Completion Risk Model Inputs**

It is expected that at the minimum, the suicide completion risk model will utilized VHA and SDR/NDI data. However, if feasible, these data will be supplemented by additional information from VBA data sources, and potentially third party data sources (e.g., local unemployment rates, country-level general suicide incidence from CDC, DoD data):

VHA Data – The VHA data contains veteran health information including but not limited to inpatient and outpatient care administered, diagnoses, and prescriptions for veterans who have used VHA services. The VHA data will be stored in the Reach Database behind the VA firewall. These data will be used both for developing the suicide completion risk model as well as for iterative, near-time development of suicide likelihood scores for veterans.

SDR/NDI Data – The SDR and/or NDI data are used to identified veterans who have committed suicide; thus, these data are crucial for constructing the dependent variable in the models. These data will be imported into R and used to train and test the risk model.

VBA Data – The VBA data contains, but is not limited to: benefit, military, financial, pension, and disability information on veterans who are using VBA services. Similar to the VHA data, the VBA data will be stored in the Reach Database behind the VA firewall. If feasible, these data will be used both for developing the suicide completion risk model as well as for iterative, near-time development of suicide likelihood scores for veterans.

Third Party Data – There are third party data sources are being considered to obtain information on veterans that is not contained within VHA or VBA data. Some of these sources are: county unemployment rates, county suicide rates, LexisNexis, court records, and social media data.



Figure : Data Sources

**Suicide Completion Risk Model Outputs**

Predicted Suicide Completion Risk – The developed model in R will produce, for each veteran in the training and testing cohort, the predicted probability of completing suicide. The predicted values will be computed based on fitted model coefficients. It is not necessary for these predicted probabilities to be imported into the SQL server (see section below on Model Coefficients).

Model Coefficients – The suicide completion risk model coefficients (for all included variables) derived in R will be imported into the SQL server. Using these coefficients, for every record (veteran) with an update to variable values, the coefficients will be applied to veteran data to re-calculate the risk of suicide completion. This updating will occur on the SQL server (and outside of R).

**Components**

R – The risk model will be trained and tested using R. When training the model, R will be importing data from tables in the Reach Database. The tables will contain VHA, VBA, SDR/NDI data and third party data.

SQL Server – The data used for the risk model in the Reach Database will be stored in a SQL Server instance. R will pull data from SQL Server in order to train and test the risk model. The suicide completion risk will be calculated/updated directly on the server using veteran data and model coefficients developed in R.

**Dependencies within Components**

R to SQL Server – In order to be able to develop a risk model and save model coefficients, a connection between R and the SQL Server must exist to allow for the transferring of data between the two components. R has a packaged called RODBC which allows for this functionality. Utilizing the RODBC package, R and SQL Server will be able to transfer data between the two components allowing for modeling development and suicide completion risk scoring calculations.

# Data Design

The Reach Database component of IRDS will use SQL implemented with Microsoft SQL Server Enterprise Edition 2012.The logical and physical data models are not yet defined.

At this time assumptions are:

1. SDR Data will be pulled directly from the SDR server/databases via a SQL connection and querying of the data
2. VBA data will pulled from SQL tables located at Corporate Data Warehouse (CDW) via a SQL connection and querying of the data
3. VISTA data will be accessed and imported through either currently exiting and/or custom RPC calls, or FMQL.
4. It is possible that some data sources may be interfaced by other means such Web Service calls or file formats such as FLAT files.

Requirements are currently being gathered to determine what data will be imported into the IRDS system and how the data model will be designed to store the data. It has been decided that the are 2 components to the data

1. Tables to store source data that will be imported into the Reach database
2. Tables to store analysis output from R programs which have code to encapsulate a risk model and a surveillance model

Each of the data components above will be contained in their own schema. The source data tables will be stored in the default .dbo schema, and the analysis output tables will be stored in the ‘Analytics’ schema. An additional schema may be added for system tables and temp tables accessed by processes such as SSIS packages.

## Non-DBMS Files

It is assumed that some VHA data will be imported from the VISTA system, which uses A MUMPS data store and will be accessed through either RPC calls and/or FMQL.

## Data View

Requirements are currently being gathered for which data sources and data elements from those sources will be imported into the reach database and what the relationship will be between those elements when they are stored in the database.

# Detailed Design

Requirements are currently being gathered.



Figure : System Overview

## Hardware Detailed Design

(Virtualized Machines / Cloud Based – Specifications):

* MS Windows Server 2012 64-bit
* Intel Xeon E5-2600 Family (2670 or 2690), 2.6GHZ or better
* 32GB RAM
* 500GB HDD
* MS SQL Server 2012 Enterprise Edition
* Users: 16 (8 simultaneous logins, minimum)

## Software Detailed Design

TBD – Requirements are currently being gathered

### Conceptual Design

#### Product Perspective

This subsection of the SDD should put the product into perspective with other related products. If the product is independent and completely self-contained, it should be stated here. If the SDD defines a product that is a component of a larger system, then this subsection should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software.

A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful.

Sections of the Requirements Specification Document (RSD) can be referenced in the subsections, if applicable.

##### User Interfaces

This subsection should specify the logical characteristics of each interface between the software product and its users. This includes those configuration characteristics necessary to accomplish the software requirements (e.g., screens, roll and scroll, GUI interface).

Recommendation: Create a block diagram showing the user interfaces.

##### Hardware Interfaces

This subsection should specify the logical characteristics of each interface between the software product and the hardware components of the system. This includes configuration characteristics (for example, hardware platform or mainframe versus personal computer). It also covers matters such as what devices the system will support, how they will be supported, and protocols. Examples include scanners, pen driven devices, and radio frequency devices.

Recommendation: Create a block diagram showing the hardware interfaces.

##### Software Interfaces

This subsection should specify the use of other required software products (e.g., VA Kernel, VA FileMan, Windows NT); and interfaces with other applications or other systems such as commercial off-the-shelf (COTS) or national databases. Specify the application interfaces (e.g., the linkage between an accounts receivable system and a general ledger system and a COTS software package that will be interfaced using an existing interface). This section should provide the following information for each required software product:

* Name
* Version number
* Discussion of the purpose of the interfacing software as related to this software product
* Definition of the interface in terms of message content and format (e.g., Health Level Seven [HL7], electronic data interchange).

##### Communications Interfaces

This subsection should specify the various interfaces to communications such as local network protocols, e-mail, Transmission Control Protocol (TCP), modems.

Recommendation: Create a block diagram showing the communications interfaces.

##### Memory Constraints

This subsection should specify any applicable characteristics and limits on memory or partition size.

##### Special Operations

This subsection should specify the special operations required by the user such as backup, recovery, and archiving operations.

This section should also include any operations for external devices or COTS systems.

#### Product Features

This subsection should provide a summary of the major features of the software.

For example, an SDD for an accounting program might use this section to address customer account maintenance, customer statement, and invoice preparation without mentioning the vast amount of detail that each of those features requires.

Note: For clarity, remember these items when creating this section of the SDD:

* The features should be organized in a way that makes the list of features understandable to the customer or to anyone else reading the document for the first time.
* Textual or graphical methods can be used to show the different features and their relationships.
* Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables.

#### User Characteristics

This subsection should describe the general characteristics of the intended users of the product, including experience and technical expertise. It should not be used to state specific requirements but rather should provide the reasons why certain specific requirements are specified in the RSD.

#### Dependencies and Constraints

This subsection should provide a description of any other items that will limit the developer’s options. The following list includes items that limit the developer’s options.

* Regulatory policies
* Hardware limitations (for example, signal timing requirements)
* Interfaces to other applications
* Parallel operation
* Audit functions
* Control functions
* Higher-order language requirements
* Reliability requirements
* Criticality of the application
* Safety and security considerations
* Usability (including 508 compliance)

This section of the SDD should contain all the software design to a level of detail sufficient to enable programmers to develop a system that satisfies the requirements defined in the RSD. It should be detailed so as to make it easy for technical staff to find the methods to complete the designed function.

These requirements should, at minimum, include the following items:

* An indication of the associated requirement(s) in the RSD which is being designed
* A description of the functionality being designed
* The design entities (and their attributes) affected
* The algorithm executed (where appropriate) to implement the functionality.

Because the Dependencies and Constraints section is often the largest and most important part of the SDD, the following principles apply:

* Specific design should be cross-referenced to earlier, related documents (e.g., the RSD).
* All design should be uniquely identifiable.
* Items in this section should be identified from a technical level rather than an end user level. (i.e., an option name should be identified rather than the menu text for that option).

### Specific Requirements

#### Database Repository

The Database Repository section in the RSD can be referenced in this section.

If a logical database design is a part of the system, it should be listed here. Logical database design should specify the logical requirements for any information that is to be placed into a database. This may include:

* Types of information used by various functions
* Frequency of use
* Accessing capabilities
* Data entities and their relationships
* Integrity constraints
* Data retention requirements.

Recommendation: Create a block diagram showing the databases and where the data resides.

TBD – see section 5.1

#### System Features

Describe the system features, functional requirements, sub-requirements, etc. which can be organized in an outline format that matches the RSD. Specific formatting and organization of the paragraphs (i.e., section numbering) is left to the discretion of the author and is dependent on the level of detail essential to fully describe the design. Some designs may only require two levels; others may require multiple levels. The information necessary to define the items or to specify modifications to the items affected by the functionality being designed should be provided in the appropriate design element tables. Where feasible, instead of duplicating the RSD, it can be referenced via a link, to avoid unnecessary duplication. The key goal is to provide traceability to requirements.

## Network Detailed Design

Provide enough detailed information about the communication requirements to build and/or procure the communication components for the system. This section should provide sufficient detail to support the procurement of hardware for the system installation. Include the following information in the form of detailed designs (as appropriate):

* Details of servers and clients to be included on each area network
* Specifications for bus timing requirements and bus control
* Format(s) for data being exchanged between components
* Diagrams showing connectivity between components, data flow (if applicable), and distances between components
* LAN topology.

## Service Oriented Architecture / ESS Detailed Design

This section provides details of provided and consumed services as follows:

* Consumed Services: Provide link to Service Description Document for each consumed service.
* Provided Services: Give service design for each provided service.

The information you provide here will be used to upload to the ESS Registry and Repository. At some point in the near future, we do not expect these SOA artifacts such as SLA, Service Description, etc. to be static documents. They will be dynamically generated from the ESS Registry and Repository tool in the form of reports. Any application and service integration design is also documented here.

A list of currently available Enterprise Shared Services is available here: <insert link to ESS list>

### Service Description for <Consumed Service Name>

Provide link to Service Description document for the consumed service. This section will repeat for each consumed service. The Service Description includes Service Interface and Service Level Definition (SLD) to address anticipated capacity requirements.

### Service Design for <Provided Service Name>

This section should describe the detailed service design for each ESS and SOA service needed to obtain an intended result. The Service Design includes Service Interface and Service Level Definition (SLD) to address anticipated capacity requirements.

This section will repeat for each **provided** service.

#### Introduction

##### Purpose and Scope of Service

This service was described at a high level in the charter document. Please refer to it here via a link.

##### Links to Other Documents

Provide links to other documents created for this service so far in the SOA lifecycle. At a minimum, provide links to:

* Service Charter
* Service Roadmap
* Service Description

#### Service Details

##### Service Identification

This section will be written as a table to provide a quick reference to the service's what, where, why and how - cheat sheet.

Table

| Service Attribute | Value |
| --- | --- |
| Name and Alias (if any) | Name of the service and other names for the service, which might be used by someone searching for this service. Please follow ESS naming standards. |
| Overview | Brief textual overview of the service. |
| Version | Version number of the service being described here |
| Latest Status | This field shows the latest status for the above referenced version of this service! The status of a service shows the progress of the service from initiation through development, deployment, and eventual retirement. The status also has a status date associated with the status - and we will be using the latest one here in this document. Valid values include: Inception, Design, Provisioning, Certification / Testing, Operation, Deprecated, Retired, Rejected - Owner has decided not to develop the service. |
| Service Type | Used to define applicable architecture patterns. Examples (from Open Group):  • Interaction  • Process  • Information  • Partner  • Business Application  • Access  • Service Connectivity |
| Architecture Layer | Referred to as class in VA Service template. Used to define applicable architecture patterns and relationships to governing bodies. Examples:  • Solution  • Process  • Information  • Utility  • Underlying |
| Business Domain | Business Vertical or Business Division where this service belongs. |
| Service Domain | The service or technical domain that the service belongs to. Can be used to establish the namespace. |
| Business Organization and Owner | Person who approves this service & any changes. Include email. |
| Technical Organization and Owner | Person responsible for provisioning (specifying, acquiring certifying) this service. Include email. |
| Development Organization and Owner | Person who is responsible for the development processes and activities for this service. Include email. |
| Support Organization and Owner | Person who is responsible for the support of this service while in production. Include email. |
| Target Consumer Organization(s) and Owner(s) | Organizations and/or developers roles that service is intended for. |

##### Service Versions

Table

|  |  |  |
| --- | --- | --- |
| Version Numbers | Current Status of Version | A Brief Description of the change implemented in that version |
| This version | Being Designed |  |
| Example: version 2 | Example: In production. Will be retired with this release. | Example: This release added the ability to look up a person by address.  Provide a link to each version of the service. |
| Example: version 1 | Example: Retired. | Example: This release provided the base minimum functionality to look up a person by name.  Provide a link to each version of the service. |

##### Summary of Design and Platform Details

###### SOA Pattern(s) Implemented

Name of the SOA pattern implemented – for instance, this may be a Pub/Sub model. Just a name and reference to the document or book with the pattern is sufficient for popular patterns or VA's own patterns. If you are using some esoteric pattern, more details will help.

###### COTS Platform vendor names and versions for hosting platform

Example, TIBCO.

#### Dependencies

The Dependency Model identifies other services, systems, databases, etc. that [Service Name] is dependent upon or interacts with to perform its function.

This section should clearly identify all sources and external systems that are accessed by this service to fulfill the service consumers’ request. This section should include diagrams to show as much detail as necessary to inform the developer. Provide a context diagram for the service.

Note: Here our primary audience includes the providers of the service. So this document in general will emphasize system components and sub-systems as much as external interactions.

#### Service Design Details

The next sub-section on Interface Technical Specs **could be** just a copy from the corresponding sub-section in Interface section in the Service Description Document. Here, you could provide more detail necessary for building this service but **the interface spec needs to be consistent between this document and the Service Description Document**. This section contains all information necessary to fully describe an interface published by this service...

##### Interface Technical Specs

The technical specification allows developers of service consumers to locate and discover the service for run time consumption.

###### Service Invocation Type

Such as: SOAP over HTTP, REST.

###### Service Interface Type

Such as: WSDL via Web Service 2.0

###### Service Name

Technical Service Name. Comply with ESS naming standards.

###### Interface

Link to WSDL or other interface document.

###### End Points

Provide if known! Calls that can be made into the service. Can be referenced to the WSDL or can be in a separate table.

###### Operations or Methods

In the table below, the technical names of the operations, inputs and outputs are used. Inputs and outputs, if parameters, must have a data type.

Non-primitive data types must be defined in the Service Information Model section.

This table could be generated automatically from the WSDL content or its equivalent.

Style can take any of these values: Parameters or Document; and One-way or Request-response or Solicit-response or Notification.

Use a separate column for the operation purpose if you wish.

You might use abbreviations in the Faults column and explain the abbreviations used below the table. For example, NF = Not Found, MI = Missing Input.

Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Operation Name | Inputs | Outputs | Transactional Qualities if relevant (Updating?, Atomic?, Can participate in transaction?) | Pre and Post Conditions | Exception (s) |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Provide a link to the Service Information model so that the consumer of your system knows the schema for the input and output parameters.

###### Message Schemas

Provide definitions or links to definitions of the message(s) related to the service operations. These may be dependent on the implementation style and protocol binding of the interface.

##### Information Model

Even though this section looks similar to the corresponding section 3.2 in Service Description, remember that the primary objective here is to facilitate construction and to gain approvals from governing bodies. So you will provide more of a “white box” view of the design here to help your developers code the service.

###### Class Diagram and Description of Entities Involved

Map out all entities involved in the service: input, output, exceptions, entities manipulated in persistent media/DBs, intermediate entities created in memory etc.

###### Mappings from ELDM to Standards Based Schemas

Provide mappings from your native schema to any standards based schemas your service will use to communicate outside. For instance, if you are using HL7 based messages then you will show how data is converted from your native schema to HL7.

##### Behavior Model (AKA Use Case Realization)

The Behavior Model defines the actions and processes supported by the service. Actions and methods represented in the use cases and sequence diagrams shown below are further defined by the operation contracts and the message payloads.

###### Use Cases (Use Case Model)

How does this service fit into the larger use case model of the consumer? You may need multiple models for multiple consumers. Focus is **not** on the internal workings of the new service instead of the calls made from external consumers. Just a summary or the Use Case Diagram may be sufficient. List the alternative and exception flows. Reference the detailed design documents via a URL.

###### Interaction Diagrams

Cut and paste screen shot from RSA or similar tool or provide link to the model. Provide description to help developers build your service. The interaction diagrams should depict external interactions and internal sequences of calls between internal components. The sequence diagram should cut through all layers to show the main, alternate and exception flows.

#### Gap Analysis

Provide a Gap Analysis (Reference) to demonstrate compliance of this service with various standards, policies, guidelines and laws. The Gap Analysis may take the form of a matrix as shown in the sample below. This will help the governance boards expedite your request.

Table

| Design Elements🡪  Policies / SLD elements etc↓ | Design  Element A | Design  Element B | Design  Element C | Comment for non-conformance |
| --- | --- | --- | --- | --- |
| Policy X | Match |  |  |  |
| Policy Y |  | Partial |  |  |
| Policy Z |  |  |  | Commercial encryption server in prod will have to address this policy. |
| Policy A |  |  |  | Compliance with this policy not required until next year. |
| New / Additional Features |  |  | New element minimizes manual intervention |  |

##### Variances from Enterprise Target Architecture

This list of “variances” will become a submission to the ESS dispensation process.

##### Variances from SLDs

This list of “variances” will become a submission to the ESS dispensation process.

##### Variances from Standards and Policies

This list of “variances” will become a submission to the ESS dispensation process.

##### Justification for Exceptions and Mitigation

This section will list out any non-functional and functional requirements that are not being met. The non-conformance may be in violation of elements of SLDs, enterprise architecture (TRM Technology Reference Model), privacy policies or guidelines. For each exception provide:

1. Reasons for non-conformance (cost, time, technology, etc.)
2. Mitigating actions taken to reduce the impact of non-conformance
3. Plan (roadmap) to come back into conformance

This list can grow depending on what the Review bodies may ask for.

# External System Interface Design

This section details interfaces external to system, that are NOT services (ESS/SOA). Typically, these may include, RPCs, Flat Data Files etc.

External systems are systems that are not within the scope of the system under development, regardless of whether the other systems are managed by the vendor or its client.

In this section, describe the interface(s) between the system under development (i.e., the system that is the subject of this SDD) and external systems and/or subsystem(s).

It is best to illustrate these sections with annotated diagrams to clearly identify the various elements of the interfaces.

## 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.

## Interface Detailed Design

Provide sufficient detail about the interface requirements for the development team to format, transmit, and/or receive data across the interface.

Include the following information (as appropriate):

* Data format requirements; if data must be reformatted before it is transmitted or after incoming data is received. Describe the tools and/or methods for the reformat process.
* Specifications for hand-shaking protocols between systems; content and format of hand-shake messages, timing for exchanging these messages, and errors handling.
* Format(s) for reports exchanged between the systems.
* Graphical representation of the connectivity between systems, showing the direction of data flow.
* Query and response descriptions.
* Describe the individual data elements that the interfacing entity(s) will provide, store, send, access, and receive, such as:
* Names/identifiers
  + Data Element Name
  + Data Format/Length
  + Data Type
  + Definition
  + Non-Technical Name
  + Non-Technical Synonyms
  + Specifications
  + Synonyms
* Range or enumeration of possible values (e.g., 0-99)
* Accuracy and precision (number of significant digits)
* Priority, timing, frequency, sequencing, and other constraints
* Security and privacy constraints
* Sources (setting/sending entities) and recipients (using/receiving entities).

Describe the data element assemblies (records, messages, files etc.) that the interfacing entity(s) will provide, store, and send, such as:

* Names/identifiers
  + Technical Name, e.g., data structure name
  + Non-technical Names, e.g. synonyms
* Data elements
* Medium/structure of data elements/assemblies
* Visual characteristics (e.g. layouts, fonts, icons etc.)
* Relationships among assemblies
* Security and privacy constraints
* Sources and recipients.

Describe the communication methods that the interfacing entity(s) will use for the interface, such as:

* Communication links, bands, frequencies, and media
* Message formatting
* Flow control (e.g. sequence numbering)
* Data transfer rate
* Routing
* Transmission services
* Safety
* Security and privacy considerations.

Describe characteristics of the protocols that the interfacing entity(s) will use for the interface, such as:

* Priority/layer of the protocol
* Packeting
* Legality checks, error control
* Recovery procedures
* Synchronization
* Status, identification, and other reporting features.

Where appropriate describe other characteristics, such as physical compatibility of the interfacing entity(s) (dimensions, tolerances, loads, voltages, plug compatibility, etc.)

# 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.

# Security and Privacy

## Security

Describe specific security mechanisms at the application level, as guided by NIST 800-53 revision 3 (or most current version). Also, summarize the security mechanisms to be provided by the VA GSSs. Reference the Security Risk Assessment.

The following information will be provided to address security controls:

A high-level description of the security controls, grouped according to the 18 control families identified in NIST 800-53 revision 3 (or most current version). A description of all 18 control families must be addressed; if a control family is not applicable, then state that control family does not apply and explain why it does not apply.

A description of the specific security controls that will be provided by existing VA infrastructure or VA GSSs.

Describe the planned use by the application of the infrastructure’s centralized security mechanisms and VA GSSs (in particular, the identification and authentication, access control, and audit mechanisms), and infrastructure mechanisms, (e.g., Directory Services) to store user account information. Sufficient detail should be provided to show the feasibility of the integration and/or inter operation of application security mechanisms with infrastructure security mechanisms.

## Privacy

Identify privacy design considerations. Describe specific privacy mechanisms at the application. Describe how the application’s privacy requirements will be met. Reference the System Security Plan (SSP) and Privacy Impact Assessment (PIA).

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 Chair of the governing Integrated Project Team (IPT), Business Sponsor, IT Program Manager, and Project Manager are required to sign.

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

Signed: Date:

< Integrated Project Team (IPT) Chair >

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

Signed: Date:

< Business Sponsor >

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

Signed: Date:

< IT Program Manager >

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

Signed: Date:

< Project Manager >

1. Additional Information

Attach any addition information that supplements the design specification.

* 1. RTM

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.

* 1. Acronym List and Glossary

Identify and define all acronyms and terms that establish meaning within the context of the plan.

Table : Glossary

| Term | Meaning |
| --- | --- |
|  |  |
|  |  |
|  |  |

* 1. Required Technical Documents

The following documents must be submitted for review to support proper approval:

* Conformance Validation Statement (CVS) - Section 508
* For additional information regarding how to obtain proper approval for this project, refer to the following documents:
* IT Infrastructure Standards
* Systems Engineering and Design Review (SEDR) process
* Enterprise Architecture Web page
* One-VA TRM
  1. Attach Documents

Once the SDD is approved, submit the AERB Design Compliance Decision Certificate as an attachment to the completed and approved SDD.

1. https://opensourceehr.atlassian.net/secure/Dashboard.jspa [↑](#footnote-ref-2)