

Perceptive Reach Integrated Reach Database System (IRDS)

System Design Document



Department of Veterans Affairs

September 2016

Version 5.0

Revision History

Date	Version	Description	Author
9/13/2016	5.0	Management Review	Matthew Robinson
8/16/2016	4.7	Updated title page and page headers to reflect the correct month. Made minor updates throughout the document. Updated screenshots in section “6.4. Navigation Hierarchy”	Kaitlin Reskovac
5/25/2016	4.6	Management Review	Matthew Robinson
5/10/2016	4.5	Updated title page and page headers to reflect the correct month. Added user story PR-970 to section “3.5.1. Overview of Significant Functional Requirements”	Kaitlin Reskovac
4/5/2016	4.4	Updated section “4.2.1. Project Conceptual Data Model”	Kaitlin Reskovac
4/4/2016	4.3	Updated screen shot in section “5.1.1.1. Client Side” Added content to section “6.2.1. User Roles” Updated title page and page headers to reflect the correct month. Updated User Roles Matrix in section “5.1.1.4.2. User Access Authentication and Authorization” Updated screenshots in sections: “6.4.1. Login”, “6.4.2. Surveillance View”, “6.4.3. Facility View”, “6.4.4. Individual View”	Kaitlin Reskovac
4/2/2016	4.2	Added information about removing deceased patients from the IRDS database in section “4.2.1. Project Conceptual Data Model” Added Data Entry Widget user stories to the table in section “3.5.1. Overview of Significant Functional Requirements”	Kaitlin Reskovac
3/31/2016	4.1	Added CCS user story (PR-3861) to section “3.5.1. Overview of Significant Functional Requirements”	Kaitlin Reskovac
3/28/2016	4.0	Updated title page, headers, and footers to reflect the current month	Kaitlin Reskovac
2/16/2016	3.9	Updated title page, headers, and footers to reflect the correct month	Kaitlin Reskovac

Date	Version	Description	Author
1/27/2016	3.8	Updated section “4.3.2. Special Technology”	Kaitlin Reskovic
1/20/2016	3.7	Updated headers and footers to reflect correct month and year Updated screen shots in sections “6.4.1. Login”, “6.4.2. Surveillance View”, “6.4.3. Facility View”, and “6.4.4. Individual View” Added content to section “5.1.1.3. Drag and Drop Widgets”	Kaitlin Reskovic
12/10/15	3.6	Option Year Update & Review	Kaitlin Reskovic
9/14/2015	3.5	Final Review	Monica Mohler/Paul Bradley
9/14/2015	3.4.1	Network architecture image update and minor content modifications	Matthew Robinson
9/4/2015	3.4	Contract Compliance and Quality Assurance Review	Jesse Rogers / Kaitlin Reskovic / Matthew Robinson
9/2/2015	3.3	Technical Updates	Robert Snelling
8/14/2015	3.2	Final Review	Monica Mohler/Paul Bradley
8/11/2015	3.1	Peer Review	Matthew Robinson
8/4/2015	3.0	August Review	Robert Snelling
7/15/2015	2.9	Final Review	Monica Mohler/Paul Bradley
7/15/2015	2.8	Peer Review	Radina Ivanova
7/15/2015	2.7	July Review	Robert Snelling
6/12/2015	2.6	Final Review	Monica Mohler
6/12/2015	2.5	Peer Review	Radina Ivanova/Matt Robinson
6/10/2015	2.4	June Update	Robert Snelling
5/14/2015	2.3	Final Review	Monica Mohler/Paul Bradley
5/13/2015	2.2	Peer Review	Radina Ivanova
5/12/2015	2.1	May Update	Robert Snelling
4/13/2015	2.0	Final Review	Monica Mohler/Paul Bradley
4/7/2015	1.9	Peer Review	Radina Ivanova

Date	Version	Description	Author
4/6/2015	1.8	April Update	Robert Snelling
3/5/2015	1.7	Final Review	Monica Mohler/Paul Bradley
3/4/2015	1.6	Peer Review	Radina Ivanova
3/2/2015	1.5	March Update	Robert Snelling
2/13/2015	1.4	Final Review	Monica Mohler/Paul Bradley
2/5/2015	1.3	Peer Review	Radina Ivanova
2/4/2015	1.2	Added additional content to reflect dashboard technical architecture	Andal FeQuiere
1/14/2015	1.1	Updated to reflect functional changes for the dashboard and the technical architecture.	Robert Snelling, Matthew Robinson
12/11/2014	1.0	Final Review	Monica Mohler/Paul Bradley
12/10/2014	.04	Peer Review	Radina Ivanova
11/26/2014	.03	Added additional content on application context and on continuous integration.	Robert Snelling, Andal FeQuiere, Paul Bradley
11/13/2014	.02	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	.01	Added Database, Analytic, Dashboard, Architecture, Networking, Business Benefits.	Bill Balshem, Kristopher Hoover, Robert Snelling, Paul Bradley
10/07/2014	.00	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 IRDS solution is developed. This document is based on the Project Management Accountability System (PMAS) ProPath Template.

Table of Contents

1. About this document	6
2. Introduction	6
2.1. Purpose of the SDD	6
2.2. Identification	6
2.3. Scope	7
2.4. User Characteristics	7
3. Background	8
3.1. Overview of the System	8
3.2. Overview of the Business Process	10
3.3. Business Benefits	13
3.4. Assumptions and Constraints	15
3.5. Overview of the Significant Requirements	16
4. Conceptual Design	23
4.1. Conceptual Application Design	23
4.2. Conceptual Data Design	26
4.3. Conceptual Infrastructure Design	29
4.4. System Architecture	30
4.5. Hardware Architecture	30
4.6. Software Architecture	31
4.7. Continuous Integration / Continuous Delivery	31
4.8. Network Architecture	31
4.9. Service Oriented Architecture / ESS	32
4.10. Enterprise Architecture	32
5. Detailed Design	32
5.1. Software Detailed Design	33
5.2. Specific Requirements	47
5.3. Continuous Integration / Continuous Delivery	47
5.4. Data Design	48
5.5. Service Oriented Architecture / ESS Detailed Design	52
5.6. Data Analytics Platform	53
5.7. Risk Model	54
5.8. External System Interface Design	54
6. Human-Machine Interface	54
6.1. Interface Design Rules	54
6.2. Inputs	54

6.3. Outputs	56
6.4. Navigation Hierarchy	56
7. Security and Privacy.....	61
7.1. Security.....	61
7.2. Privacy	62
8. Relationship to Other Documents and Plans.....	62
9. Acronyms and Abbreviations	62

1. About this document

The System Design Document (SDD) describes the design activities, Agile development methodology, top-level system architecture, hardware, software, communications, and interface components of the Perceptive Reach system and applications. This document will be prepared and delivered 30 days after contract award and will be updated monthly thereafter.

2. Introduction

The Department of Veterans Affairs (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 Veterans Health Administration's (VHA) 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.

2.1. 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 establishes hardware, software, communication, and interface components. For the purpose of this document, "system" will be used to reference the overall IRDS solution (Dashboard, Database, Direct Message, and Data Analytics) and "application" will be used to reference the portions of IRDS that will interface with an end user (Dashboard and Direct Message).

2.2. Identification

This SDD applies to the Perceptive Reach Integrated Reach Database System (IRDS) as developed under the purview of the VA Center for Innovation (VACI).

2.3. Scope

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

- Reach Database – a Structured Query Language (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.

2.4. User Characteristics

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

- a) 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.
- b) 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.
- c) 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.
- d) 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 generate reports. The report generation process shall be automated.
- e) Sustainment - The application will provide the capability for users to edit and add to the IRDS Risk Stratification Model, permit creation to new models and mapping to interfaces.

3. Background

3.1. 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.

As shown in [Figure 1](#), IRDS will be an integrated system comprised of the following:

- **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 1: IRDS System Overview

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.

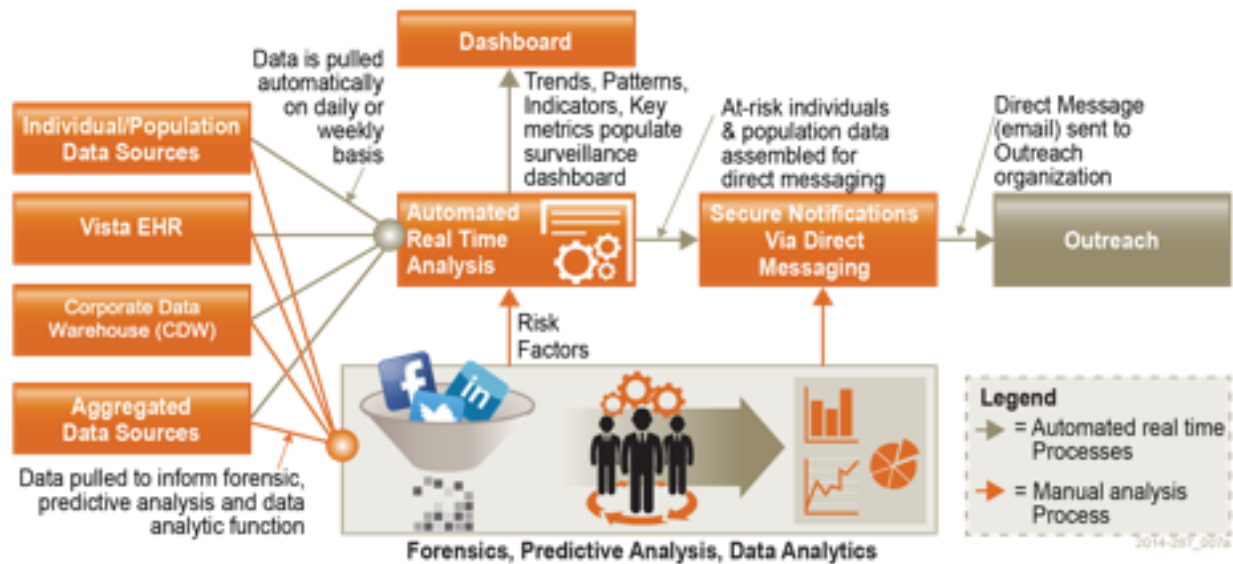


Figure 2: Conceptual System Design

This effort includes data analysis and predictive modelling, done in collaboration with VA stakeholders and clinical subject matter experts (SME), while at the same time leveraging research data from sources such as the Department of Defense (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.

3.1.1. 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)

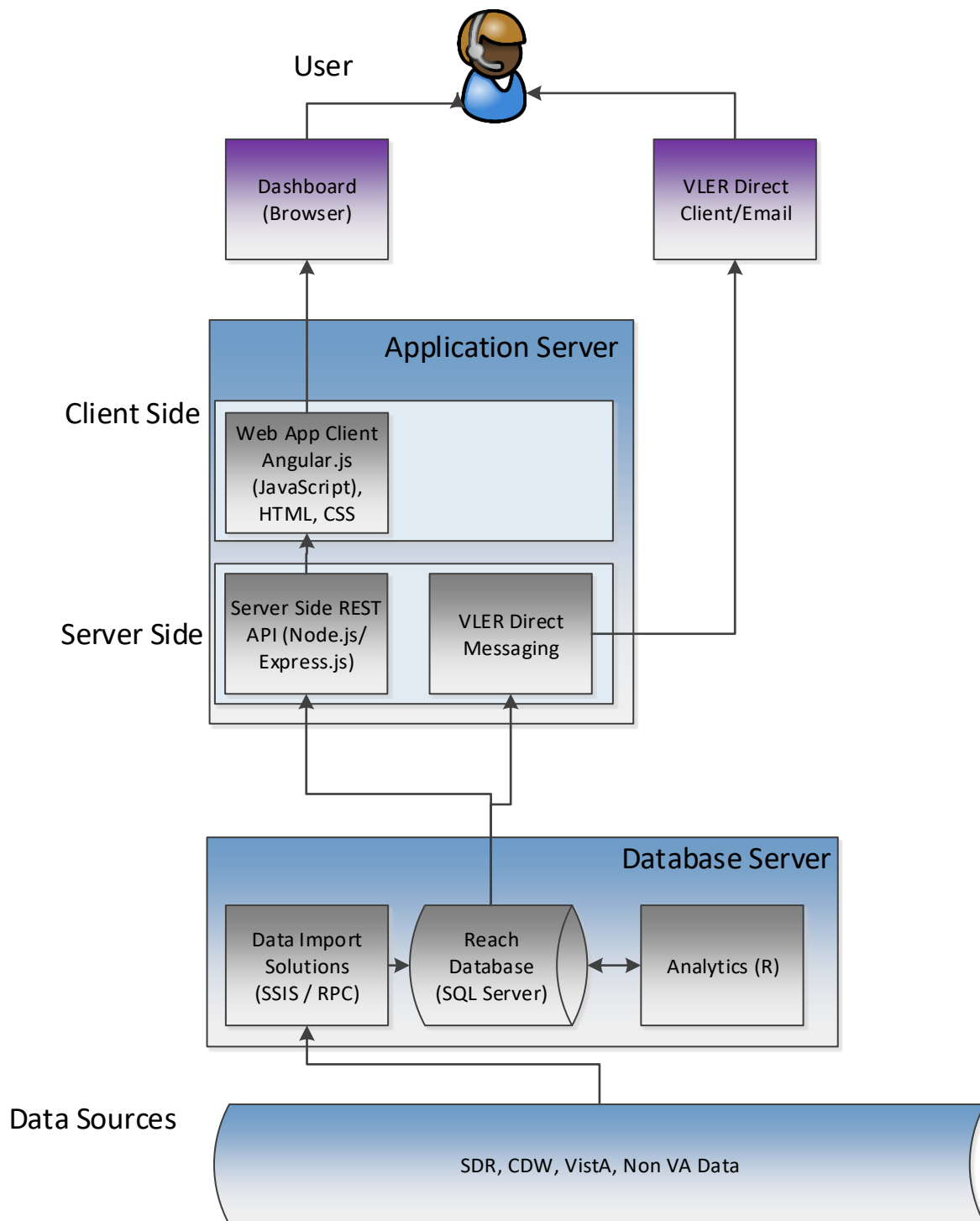
The SDR represents a significant step towards the deployment of a comprehensive suicide surveillance system as it enables a systematic collection of data on completed suicides, attempted suicides, and suicide ideations. The analytical value contained within the rich data sets of the SDR is largely untapped 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.

3.2. Overview of the Business Process

[Section 2.4](#) 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 is 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 Virtual Lifetime Electronic Record (VLER) architecture.
5. A system user will access the Dashboard via a compatible web browser. The Dashboard is a web application that provides a visual presentation in the web browser of data that has been accessed via a REST application programming interface (API), which queries the Reach database for both specific and aggregate data regarding high risk Veterans at their management level (region, state, Veterans Integrated Service Networks (VISN), Veterans Affairs Medical Center (VAMC)). The query results are passed to the client browser and populated in the web page.

**Figure 3: Dashboard Surveillance**

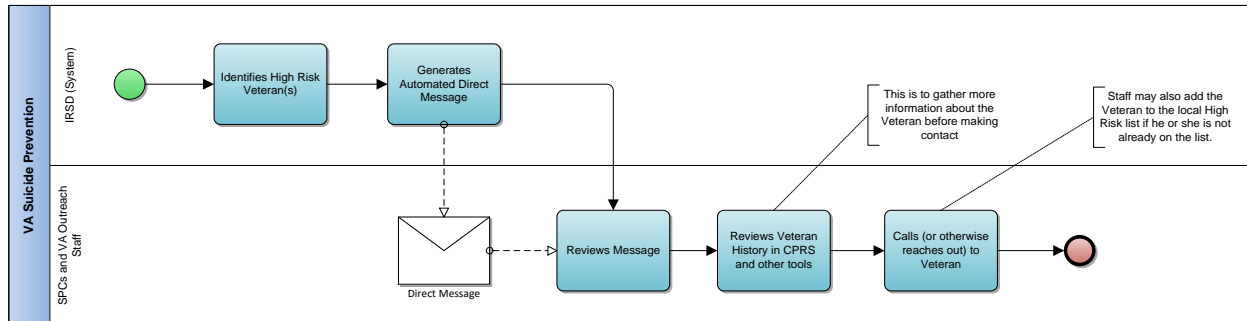


Figure 4: Direct Message, New Veteran Identified

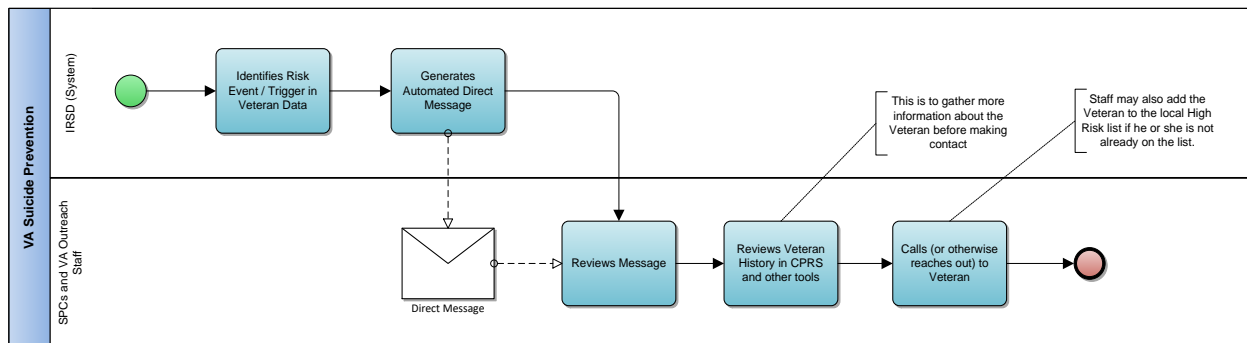


Figure 5: Direct Message, Veteran Data Update

Table 1: 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.

Business Process ID	Business Process Name	Type	Owner	Description
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.

3.3. 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 6: 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 7: System Benefits

3.4. Assumptions and Constraints

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

This SDD will be regularly updated and identified with version numbers to describe the expanded system design.

Unless otherwise stated, software will be open source and compliant with One-VA's Technical Reference Model (TRM).

3.4.1. Design Assumptions

The overall design and development process is based on iterative development and rapid prototyping incorporating key stakeholder input allowing for design factoring and enhancement.

3.4.2. Design Constraints

- System designers have attempted to utilize open source tools wherever possible, including the 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 One-VA Technical Reference Model (TRM) List of Technology or have requested a waiver for any tools not listed on the TRM List of Technology.

3.4.3. Design Trade-offs

It is the system's responsibility to provide a feature-rich and responsive Dashboard user interface (UI). The Reach Database provides the consolidated data services to drive the UI. It is expected that the number of users will be relatively low but the responsiveness of the Dashboard to query and display near real-time data is essential. Thus, the IRDS Dashboard is architected to scale by ensuring that performance does not degrade as the number of users increases and as additional data becomes available. This architectural focus can be demonstrated in the following design trade-offs:

- Maximize the responsiveness of the UI by utilizing client side controllers, and only making network calls to retrieve data. This is achieved through the use of a Single Page Application
- Favor an asynchronous API interface to maximize performance. If an API call is synchronous, it means that code execution will block (or wait) for the API call to return before continuing causing possible performance issues. However, asynchronous calls do not block (or wait) for the API call to return from the server allowing execution to continue in the application. In light of this approach, the Dashboard will cache data when sensible in order to continue performance gains. This approach is further explained in Section 5.2.5 (RESTful Data Services).
- Favor the usage of micro-services, rather than traditional web server clusters, to maximize discrete scalability options and to ensure that services remain loosely coupled.

- The IRDS database is composed on data aggregated and refined through an analytics platform and does not include all available data from external data sources utilized by the analytics platform.

3.5. Overview of the Significant Requirements

3.5.1. Overview of Significant Functional Requirements

The table below includes an overview of the major user requirements associated with the proposed solution. A full listing of the project's Product Backlog is maintained in Jira at <https://opensourceehr.atlassian.net/secure/Dashboard.jspa>. In addition, a more detailed 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 2: Functional Requirements

Jira Key	Epic	User Story
PR-158	As an Outreach Provider, I want to be sent secure notifications via a Direct Message of at-risk Veterans and populations so I can provide outreach services to these groups.	
PR-346		As an Outreach Provider, I want to view a Direct Message that highlights a Veteran at high risk for suicide so I can provide outreach services to them.
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-2306		As an SPC, I want automated Direct Messages to be sent to every site an at-risk Veteran has received care at within the past two years so I will immediately know if a Veteran has received care in multiple locations.

PR-160	As a member of VA leadership, VA Center of Excellence for Suicide Prevention staff, VA Mental Health leaders, and VA Suicide Prevention Coordinators (hereby indicated as a "Dashboard User") 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-1537		As a Dashboard User, I want to view a Veteran's recent medical diagnoses from within the Dashboard so I can view the Veteran's recent issues at a glance and make outreach and care decisions.
PR-1571		As a Dashboard User, I want widgets to be enabled to communicate with each other across the Dashboard. (e.g. Veteran Roster to Supporting Widgets like Appointments or Medications).
PR-1789		As a system administrator, I want the IRDS to generate reports/logs when new data is imported into the system.
PR-1267		As a Dashboard user, I want to see a Veteran's contact information so that I can contact them to provide outreach and intervention services.
PR-1268		As a Dashboard user, I want to see a Veteran's emergency contact information so that I can contact them to provide outreach and intervention services.
PR-1380		As a Dashboard user, I want the ability to update a Veteran's outreach status so that I can track what Veterans have not yet been contacted, which Veterans are in outreach / intervention services, which Veterans have refused service, etc.
PR-1405		As a Dashboard User, I want to view a Veteran's medications from within the Dashboard so I can view the Veteran's prescriptions at a glance and make outreach and care decisions.

PR-1407		As a Dashboard User, I want to view a Veteran's medical appointments from within the Dashboard so I can view the Veteran's appointments at a glance and make outreach and care decisions.
PR-356		As a Dashboard User, 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 a Dashboard User, 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-882		As a Dashboard User, I want to view a graphic in a widget that allows me to see basic information related to a VA facility.
PR-881		As a Dashboard User, I want to see a line graph that shows the change in how many Veterans are at the top .1%,-or 5% of the risk stratification model over time.
PR-877		As a Dashboard User, I want to see a widget that allows me to view a menu showing how many Veterans are within a certain risk stratification.
PR-876		As a Dashboard User, I want to see a widget that allows me to view a "roster" (or list) of Veterans that have been identified by the application as high risk.
PR-996		As a Dashboard User, I want to be presented "Clinical Decision Support" information related to a Veteran's specific information.
PR-994		As a Dashboard User, I want to view individual Veteran information that is relevant to suicide outreach, intervention, and care, so I can make clinical care decisions for treatment of the Veteran.
PR-349	As a Dashboard User, I want to log in to the Perceptive Reach application.	
PR-1676		As a Dashboard user, I want to enter my credentials (username and password) to log in to the Dashboard.
PR-1887		As a Dashboard user, I want to be presented with a "landing page" where I can log in to the application.

PR-1788		As an IRDS Dashboard user I expect the system will automatically terminate my session after a specified period of inactivity (ex. 10 minutes) such that an unauthorized user could not use my system if I walk away from my computer.
PR-1677		As a Dashboard user, I want my log in to be rejected if my credentials are not accurate or valid.
PR-1678		As a Dashboard user, I want to log out of the Dashboard to end my session.
PR-1679		As a system administrator, I want to grant users default access to the Dashboard (Individual Veteran and Facility Level Views).
PR-1680		As a system administrator, I want to grant users supervisory access to the Dashboard (State, Region, VISN, National, plus all other views).
PR-1681		As a system administrator, I want to change a user's access level if appropriate.
PR-1705		As a Dashboard user, I want to be logged in to the Dashboard via VA Single Sign On automatically.
PR-1569		As a Dashboard User, I want to be able to save my Dashboard settings / customizations when I log out of the application and see them again when I have logged back in.
PR-351		As a Dashboard User, I want to see data from my "home" facility when I log in.
PR-161	As a Researcher, I want to access research tools and data in the application so I can perform research-related tasks and projects.	
PR-724		As a Researcher, I want to access the data in the underlying database using BIRT.
PR-725		As a Researcher, I want to access the data in the underlying database using R.
PR-726		As a Researcher, I want to access the data in the underlying database using MySQL.
PR-162	As a Reporter, 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-889		As a Reporter, I want to generate reports from the data in the underlying database using BIRT.

PR-890		As a Reporter, I want to generate reports from the data in the underlying database using R.
PR-891		As a Reporter, I want to generate reports from the data in the underlying database using MySQL.
PR-892		As a Reporter, I want to access a set of "canned" reports that are automatically generated on a regular basis.
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-1787	The IRDS system shall have error detection and handling mechanisms.	
PR-1870		As a System Administrator, I want the application to detect and notify me of network errors so I can troubleshoot and resolve the errors in an expeditious manner.
PR-1871		As a System Administrator, I want the application to detect and notify me of CDW interface errors so I can troubleshoot and resolve the errors in an expeditious manner.
PR-1872		As a System Administrator, I want the application to detect and notify me of remote system errors so I can troubleshoot and resolve the errors in an expeditious manner.
PR-1873		As a System Administrator, I want the application to detect and notify me of data errors (rx data corruption or erroneous) so I can troubleshoot and resolve the errors in an expeditious manner.
PR-353	As a Dashboard User, I want to access Dashboard "views" that summarize data for a specific facility service area, state, VISN, Region, or nationally per corresponding view (Facility, VISN, National).	
PR-1684		As a supervisor, I want to see a summary of how many Veterans are in the top .1% and 5% risk stratifications nationally.

PR-1685		As a supervisor, I want to see a summary of how many Veterans in the top risk stratifications are associated with each VA Medical Center, nationally.
PR-1686		As a supervisor, I want to see a summary of how many Veterans are in each age group in the top .1% and 5% risk stratifications nationally.
PR-1687		As a supervisor, I want to see a summary of how many Veterans in the top .1% and 5% risk served in the various US Military Branches.
PR-1688		As a supervisor, I want to see a summary of how many Veterans are every outreach and intervention status in the top .1% and 5% risk stratifications nationally.
PR-1689		As a supervisor, I want to see a summary of how many Veterans in the top .1% and 5% risk have been to a VA Mental Health clinic in the past 12 months.
PR-1690		As a supervisor, I want to see a summary of how many Veterans in the top .1% and 5% risk stratifications have specific diagnoses, nationally.
PR-1692		As a supervisor, I want to see a summary of how many Veterans in the top .1% and 5% have been flagged as high risk for suicide in SPAN.
PR-3733		As a supervisor, I want to see trends of reportable incidents by viewing a historical summary of the total monthly reportable incidents that occur at a specific facility over the course of the previous 17 months.
PR-3861		As a VISN level (or other higher level user), I want the ability to view data from multiple specific facilities so I can make program decisions across those facilities.
PR-3437	As a clinical care team member, I want to see a widget that allows me to enter and save new patient information in the IRDS Dashboard	
PR-3438		As a Clinical Care Team Member, I want the Data Entry Widget to update based on the Veteran I have selected in the Veteran Roster
PR-3439		As a Clinical Care Team Member, I want to enter a date for when a High Risk Flag was initiated, if / when one has been initiated for the patient.
PR-3440		As a Clinical Care Team Member, I want to be able to copy and paste information from the High Risk Flag read-only text box to the top free-text entry area.

PR-3442		As a Clinical Care Team Member, I want the read-only field to populate with updated SPAN High Risk Flag information on a monthly basis.
PR-3443		As a Clinical Care Team Member, I want to view a history of the High Risk Flag free text entries by selecting the left or right arrows.
PR-3444		As a Clinical Care Team Member, I want to view a history of High Risk Flags by selecting the left or right arrows.
PR-3445		As a Clinical Care Team Member, I want to enter the patient's principal MH provider, if known.
PR-3446		As a Clinical Care Team Member, I want to be able to copy and paste the name of a candidate principal MH provider from the bottom text box to the top free-text entry area.
PR-3448		As a Clinical Care Team Member, I want the read-only field to populate with updated candidate MH provider names on a monthly basis.
PR-3449		As a Clinical Care Team Member, I want to view a history of the principal MH providers free text entries by selecting the left or right arrows.
PR-3450		As a Clinical Care Team Member, I want to view a history of Principal MH Providers by selecting the left or right arrows.
PR-3451		As a Clinical Care Team Member, I want to enter the date that a Safety Plan was initiated or completed.
PR-3452		As a Clinical Care Team Member, I want to be able to copy and paste information from the Safety Plan read-only text box to the top free-text entry area.
PR-3454		As a Clinical Care Team Member, I want the read-only field to populate with updated Safety Plan information on a monthly basis.
PR-3455		As a Clinical Care Team Member, I want to view a history of Safety Plan free text entries by selecting the left or right arrows.
PR-3456		As a Clinical Care Team Member, I want to view a history of Safety Plans by selecting the left or right arrows.
PR-3457		As a Clinical Care Team Member, I want to enter a free-text comment / note when desired.
PR-3458		As a Clinical Care Team Member, I want to view a history of saved comments by selecting the left or right arrows.

PR-3459		As a Clinical Care Team Member, I want to hit the “Save” button to save my changes.
PR-3460		As a Clinical Care Team Member, I want to see a pop-up message if I attempt to select a different Veteran before saving my changes.
PR-3520		As a user, I want to hover over the text field (free text and system of record fields) to see the date of when the field was updated
PR-3522		As a user, I want to have a 128 character limit in the free text fields.
PR-970		As a Dashboard User, I want to export the data in a dashboard widget to an external file.

4. Conceptual Design

4.1. Conceptual Application Design

The application design centers on the Reach Database as depicted in diagram below. Data flows in from VA sources, initially CDW, and is processed through a Risk Assessment which categorizes each Veteran at a risk level. Based on requirements, the IRDS Dashboard and Direct Message notification process will interact with veteran data to provide support through visualizations and notifications to VA SPCs.

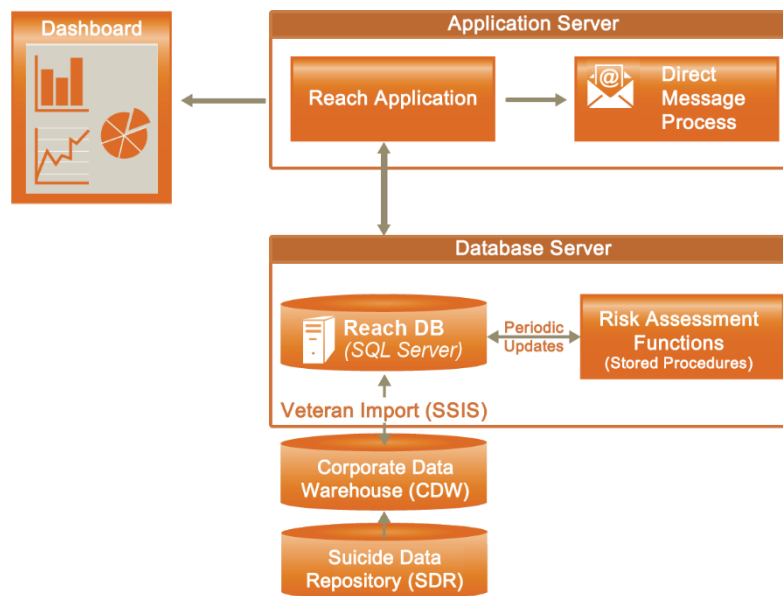


Figure 8: Application Design

4.1.1. Application Context

While traditional interventions must remain in place, this proposed approach introduces an upstream intervention. The IRDS solution is a novel and complimentary approach to more traditional forms of suicide prevention already in place at VA.

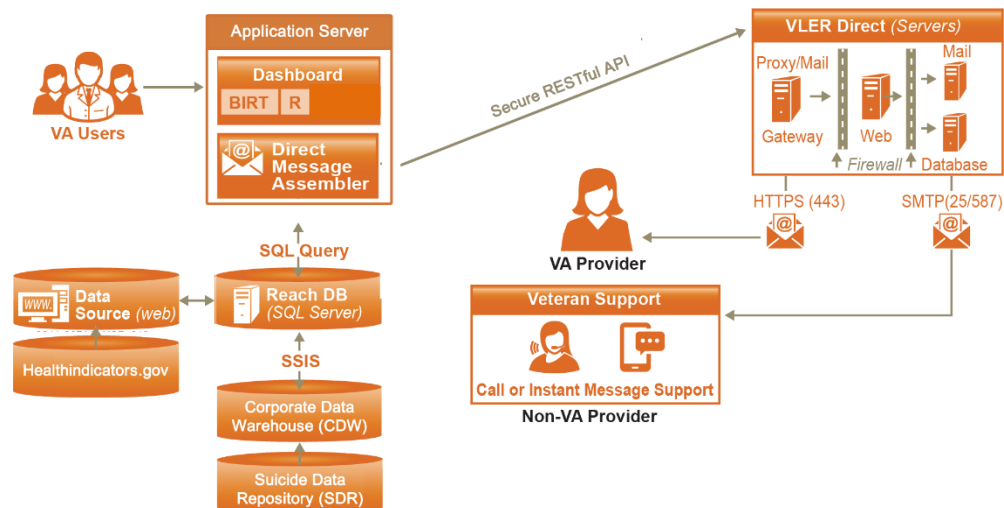


Figure 9: System Diagram

Table 3: (Grouping) Application Context Description Object

Name	Description	Interface Name	Interface System
Dashboard	The IRDS dashboard will be a clinical support and an operations style surveillance tool providing near real-time views of Veteran information, regional and temporal data, trends, events, and key performance metrics.	SQL Query	Reach DB, VA Users
Direct Message Assembler	To facilitate intervention through outreach programs, the IRDS system will create and transmit notification messages via Direct Messaging to VA designated and authorized intervention service providers.	SQL Query, HTTPS	Reach DB, VLER Direct
Data Analytics Platform	A combination of business intelligence (BI) tools and data analytics packages will satisfy three key objectives: data integrity, flexibility, and simplicity. Examples: BIRT, R, & SQL.	SQL Query	Reach DB, VA Users
Reach DB	The Reach Database will be developed to leverage the SDR and other data sources to create a robust data collection against which analytics can be performed.	SSIS, SQL Query	Dashboard, SDR, VLER Direct, External Sources

4.1.2. High-Level Application Design

The IRDS will include data visualization tools. Data visualization tells the story of the analysis results using charts, tables, and other graphics and is the step that makes the body of data analytics work accessible to a broad range of stakeholders. For the IRDS, the most important data to be displayed and visualized is data related to providing outreach, intervention, and clinical care services to Veterans at an increased risk for suicide. This focus will help the program achieve its strategic goals and objectives of proactively providing services to at risk Veterans and in turn helping decrease instances of Veteran suicide.

The surveillance Dashboard will be accessed through a VA approved web browser. The Dashboard will be configurable, providing different user groups distinct views that meet their business needs. As shown in [Figure 10](#), 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 REST API 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.

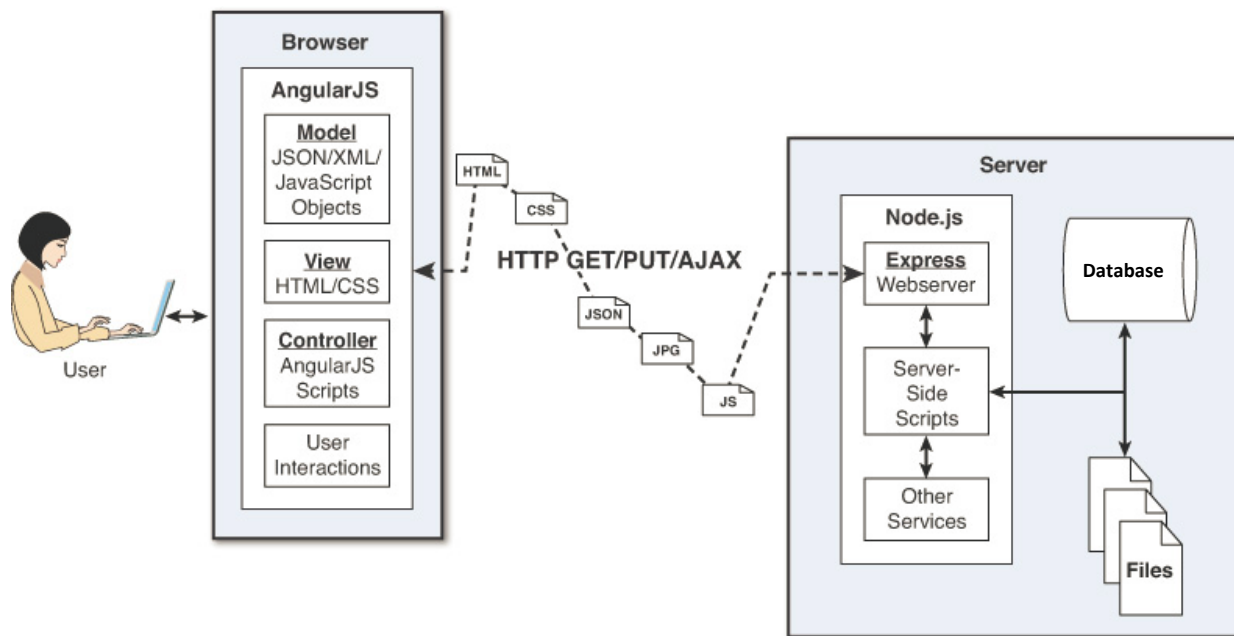


Figure 10: Conceptual Dashboard Application Design

4.2. Conceptual Data Design

4.2.1. Project Conceptual Data Model

The Reach Database contains the following tables:

Data Tables

- Patient** - The master list of Veterans that will be scored for suicide risk in the IRDS system. This table contains:
 - Reach ID - Unique IRDS system ID (all child tables to the Patient table will be linked by Reach ID)
 - VA Identifiers - Patient ICN
 - Demographics - Name, SSN, DOB
 - Current Risk Score and Risk Level
 - Date First Identified as High Risk
- RiskFactors** - The Risk Factors table will store risk factors for each Veteran, which will be used to calculate that Veteran's risk score. One record will contain all of the risk factor values for a Veteran. There will be a 1 to 1 relationship between the Patient table and the RiskFactors table.
- ScoreHistory** - Each time risk scores are re-run for the Veterans the old scores will be moved to a ScoreHistory table, so each Veteran score can be mapped over time. The table will have a one to many relationship with the Patient table.
- Veteran details tables** - Any data that will be displayed in the IRDS Dashboard for high risk Veterans will be stored in one to many child tables to the Patient tables. Ex: There will be an emergency contact table in the Reach database. Any time a Veteran becomes identified as high risk, the emergency contact data for that Veteran will be added to an Emergency Contact table.

Veterans that are identified as deceased in CDW will be filtered from being imported into the IRDS patient population.

Reference Tables

Examples are lists to be used for reporting and normalizing of the data such as a list VAMCs and a list of ICD Codes that contain a diagnosis description associated with each code.

VeteranStation

Each Veteran will be tied to one or more location in the IRDS system. This way, aggregate details about Veterans will be able to be rolled up to Facility, VISN and state levels in the Dashboard. All records in the CDW data have the specific VAMC where a Veteran: had a visit, was prescribed medication, was diagnosed with a condition, etc. This value is stored in the Sta3N field in CDW tables. A list of VAMC's (Sta3N) for where each Veteran has had some kind of activity in the past 2 years will be stored in the Reach database and accessible to the Dashboard when doing data aggregation at a specific level for Veterans.

System Tables

Tables used by the Dashboard such as User roles and Preferences (see Data Access).

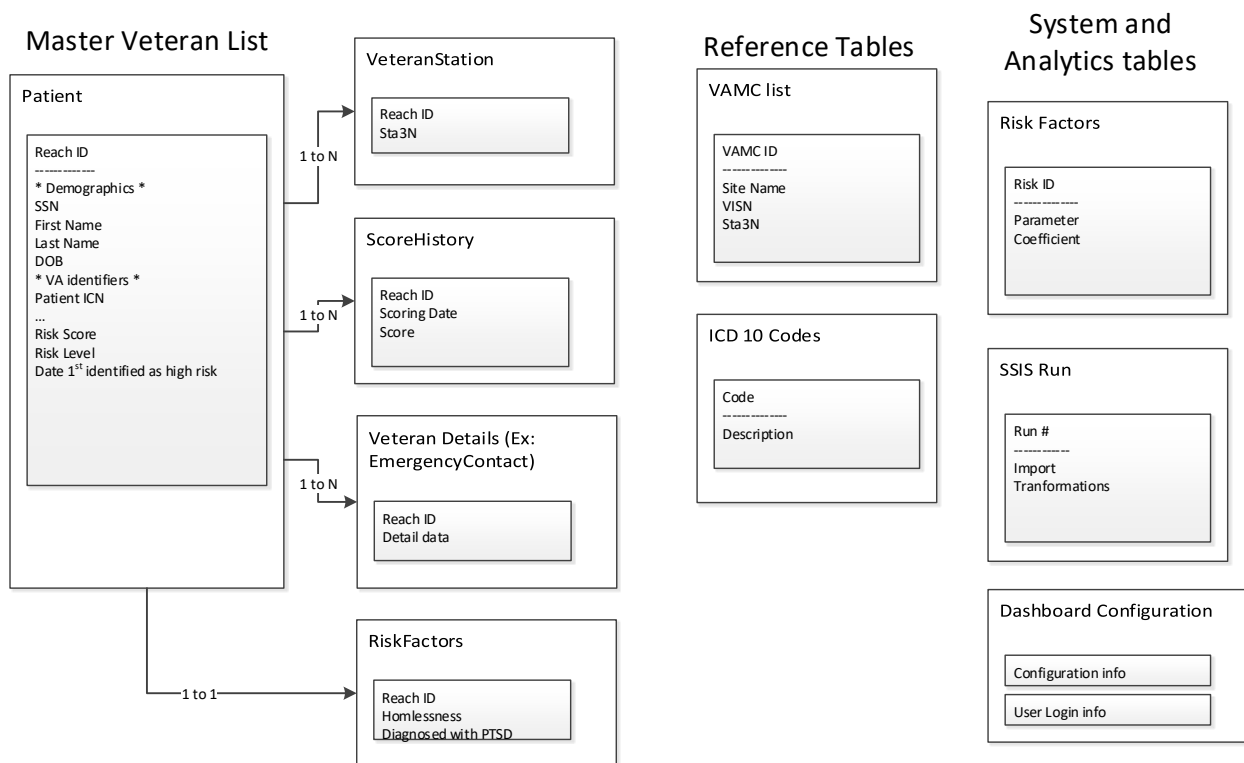


Figure 11: IRDS Reach Database Logical Data Model

4.2.2. Database Information

Table 4: Database Inventory

Database Name	Description	Type	Steward
Reach	SQL Server database(s) that will: <ol style="list-style-type: none">1. Import data from CDW, SDR and other internal/external sources2. Store Analytics output to be used by IRDS Dashboard and messaging	Create	TBD
SDR	See section 5.5, IRDS Data Sources table	Interface	
CDW	See section 5.5, IRDS Data Sources table	Interface	

4.2.3. Data Import Design

4.2.3.1. 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.

4.2.3.2. Remote Procedure Calls (RPC)

(Future Capability) The VA uses the Veterans Health Information Systems and Technology Architecture (VistA) system, for managing Veterans health data. Data will be imported into the IRDS system directly from VistA using RPC calls. VistA data is stored against a MUMPS back end, which uses text-based files for data storage. For each set of VistA data imported 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 an SSIS package

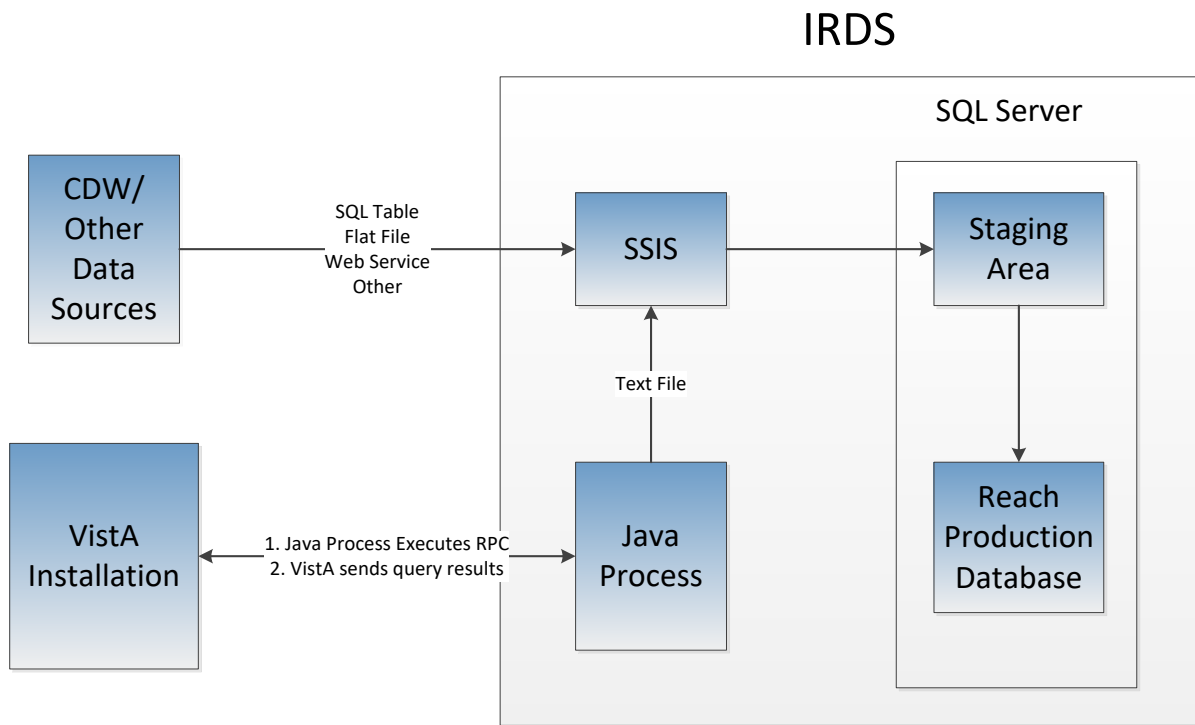


Figure 12: IRDS Data Import Process Flow

4.3. Conceptual Infrastructure Design

The architecture of this system supports cloud computing and the principles of OneVA EA. The IRDS system is intended to be implemented on a virtual or single cloud resource.

4.3.1. System Criticality and High Availability

The IRDS is not a high availability system. System redundancy is not a requirement within this innovation program at this time. The availability of this system is expected to be 365/24 with the exceptions of the times when the underlying infrastructures are not available due to maintenance.

4.3.2. Special Technology

As the IRDS Innovation is granted approval or waivers for any special technologies listed below, the table will be updated.

Table 5: Special Technology Requirements

Special Technology	Description	Location	TRM Status
Business Intelligence Reporting Tools (BIRT)	An open source tool used for reporting.	BIRT exist on the application server desktop.	Yes

Special Technology	Description	Location	TRM Status
R	An open source analytics and reporting tool.	A shortcut for R exist on the application server desktop.	Yes

4.4. 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. The following diagram (also shown in [Section 4.1](#)) provides an overview of the System Architecture.

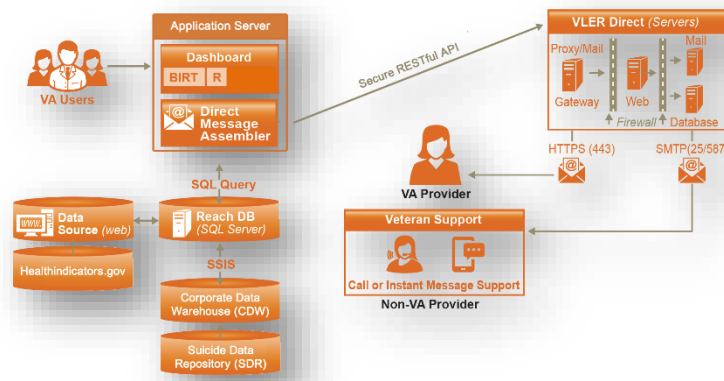


Figure 13: System Architecture

4.5. Hardware Architecture

The solution will be deployed within the VA's enterprise environment.

Table 6: Initial Requirements (Cloud Based – Prototype Server)

Type	OS	Processor	Memory	Storage	Users	Applications
Cloud	MS Windows Server 2012 64-bit	Intel Xeon CPU E5-2670, 2.6GHZ	16 GB	60 GB	16	MS SQL Server 2012 Enterprise Edition

Table 7: Planned Requirements (Cloud Based – Development/Test Server)

Type	OS	Processor	Memory	Storage	Users	Applications
Open	MS Windows Server 2012 64-bit	Intel Xeon CPU E5-2670, 2.6GHZ or better	64 GB	560 GB (all drives)	16	MS SQL Server 2012 Enterprise Edition

For further details on the Hardware Architecture, please refer to [Section 5 Hardware Architecture](#).

4.6. 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. For details on the Software Architecture, please reference [Section 5.2 Software Detailed Design](#).

4.7. Continuous Integration / Continuous Delivery

The IRDS solution will be maintained through a Continuous Integration / Continuous Delivery process (CI). This will isolate changes as they are added to the larger code base and are immediately tested and reported on. The goal of CI is to provide rapid feedback so that if a defect is introduced into the code base, it can be identified and corrected as soon as possible.

4.8. Network Architecture

The following illustration shows the 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.

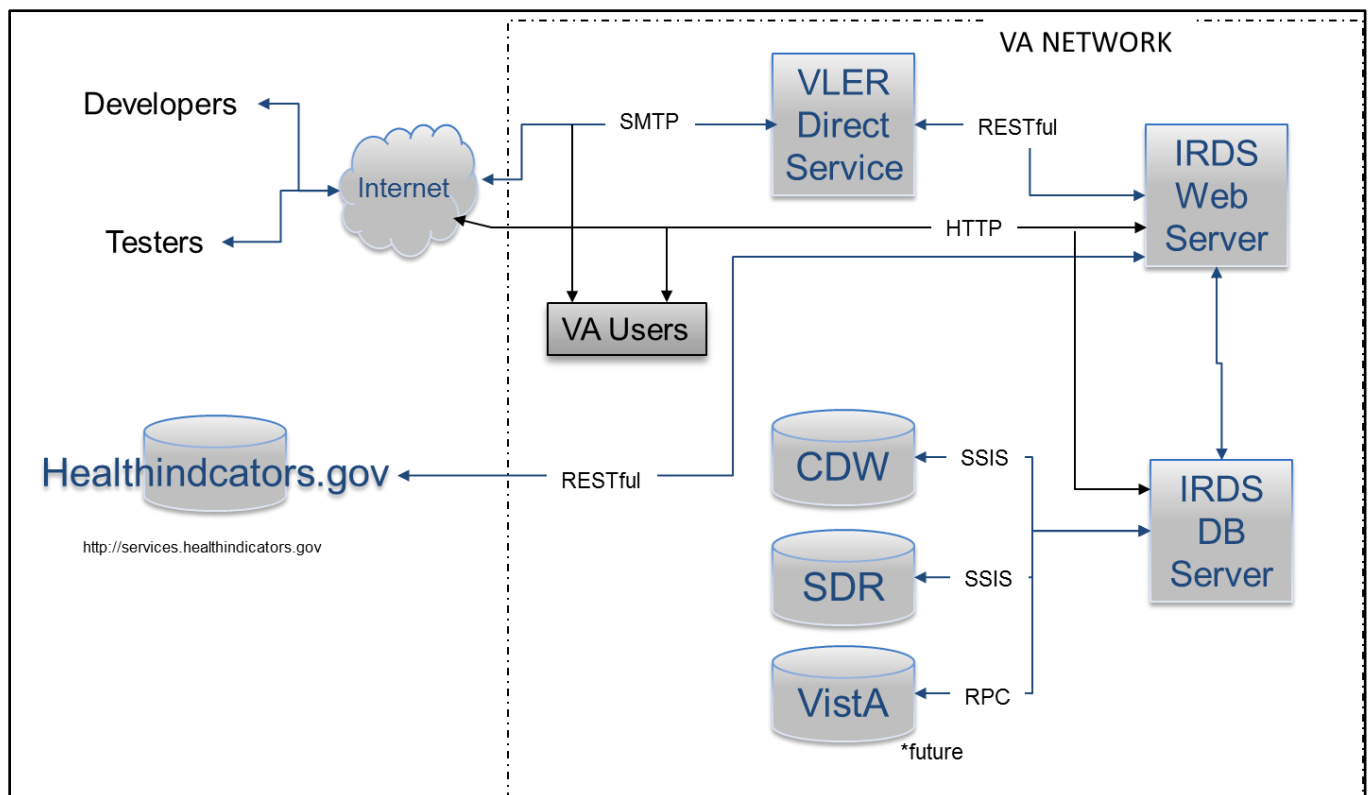


Figure 14: Network Architecture

4.9. 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. However, it is expected for IRDS to consume the VLER Direct ESS for Direct Message.

4.10. 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 portability, modularity, and scalability consistent with VA directives.

5. Detailed Design

The following section details each aspect of the IRDS solution. This will be expanded to include each component of the solution as each are defined via the requirements process

5.1. Software Detailed Design

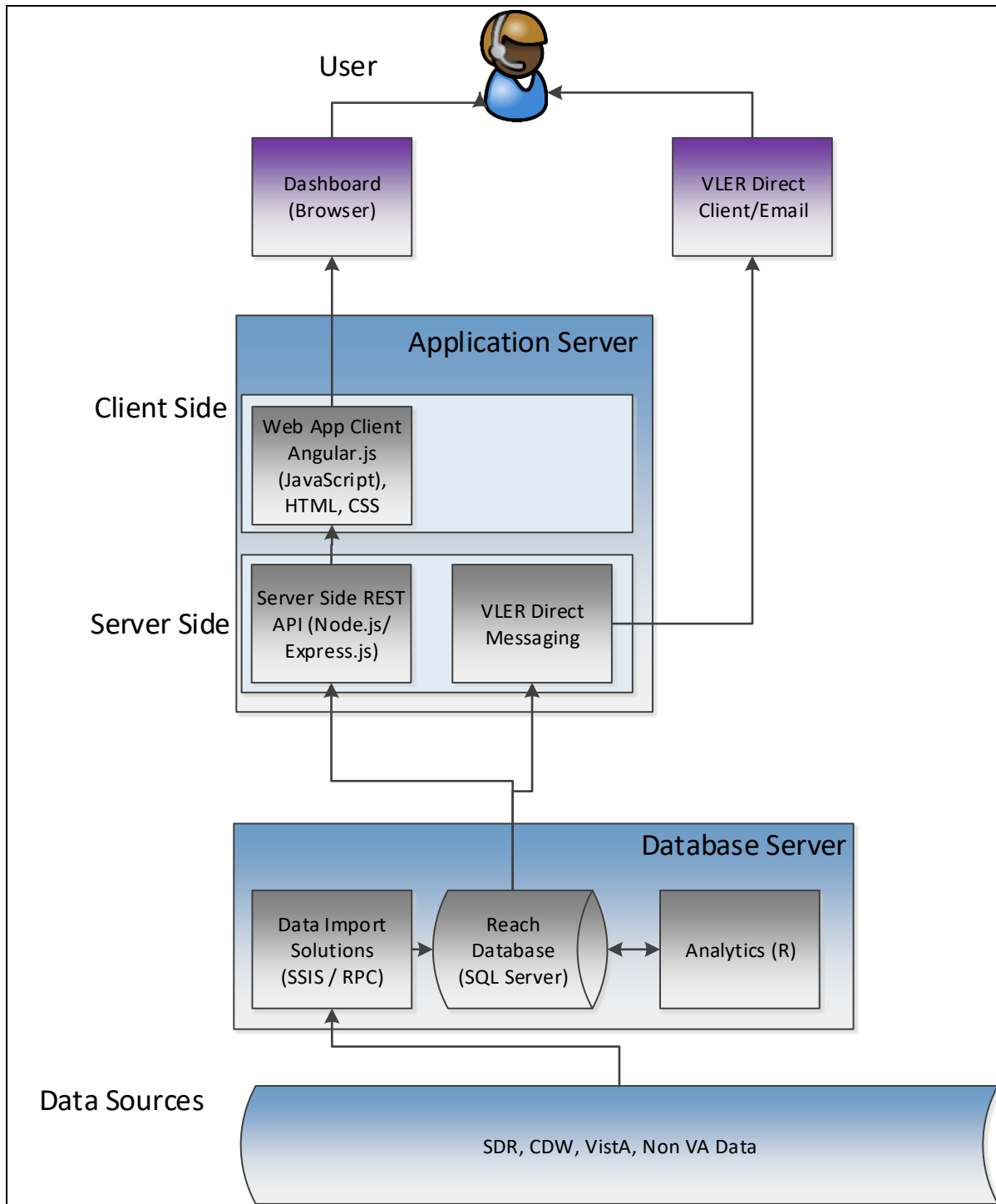


Figure 15: Architecture Overview

5.1.1. Dashboard

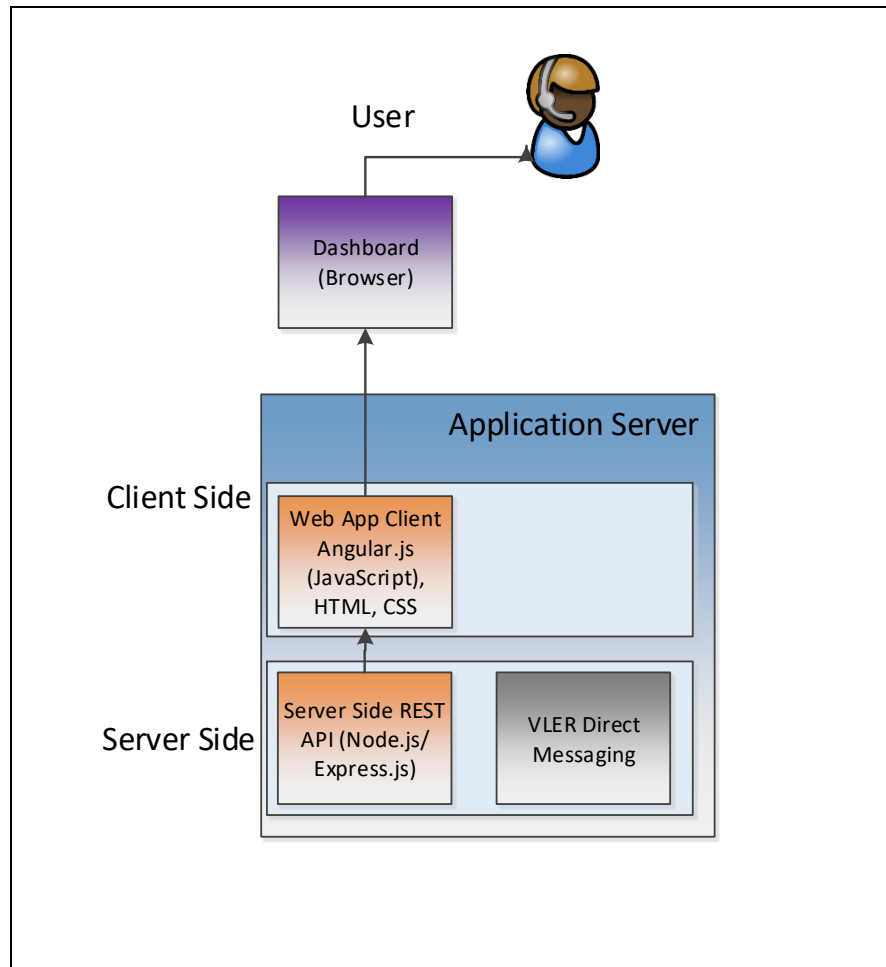


Figure 16: Dashboard

The role of the Dashboard is to provide visualization tools that display the status of metrics and key performance indicators (KPIs) for IRDS. The Dashboard will consolidate and arrange preselected metrics and analytic results within a browser based web application. The Dashboard is designed as a Single Page Application (SPA) which loads as one HTML page that redraws its UI without round trips to server. The client is a browser that runs JavaScript code to request the data and render updates to the page. The server is an API oriented architecture exposed through HTTP for items such as authorization, data access, business logic via CRUD (create, read, update and delete). Node.js provides 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)

5.1.1.1. Client Side

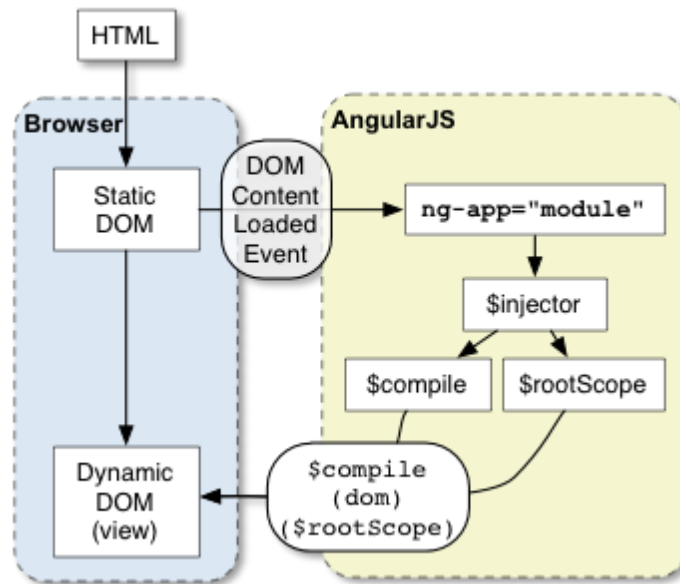


Figure 17: AngularJS Architecture

The Dashboard client is a dynamic SPA built primarily with AngularJS providing the framework for client-side model–view–controller (MVC) architecture. The library works by first reading the HTML page, which has embedded into it additional custom tag attributes. Those attributes are interpreted as directives telling Angular to bind input or output parts of the page to a model that is represented by standard JavaScript variables. AngularJS provides the inner workings of the application with data-binding, basic templating directives, form validation, routing, deep-linking, reusable components, dependency injection.

While AngularJS is the core technology of the client application, the Dashboard is using an open sourced AngularJS generic Dashboard framework which provides a uniform look and feel to the user interface. Using a system of containers, the Dashboard provides common blocks of functionality called widgets that allow for modularization of functionality.

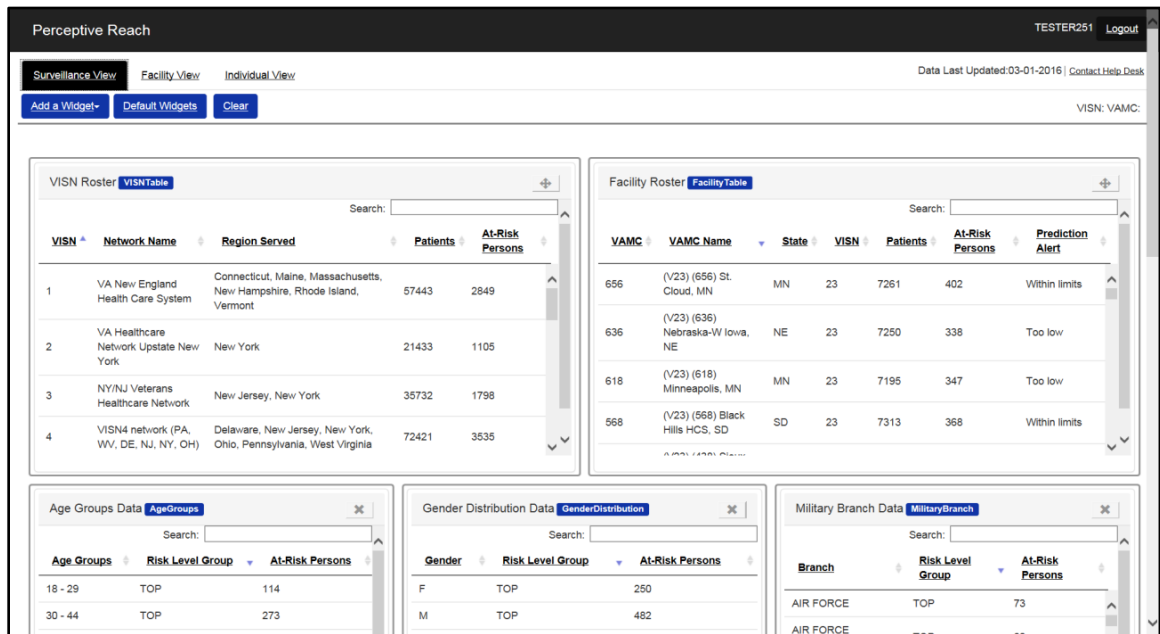


Figure 18: Example Widget Structure

The Dashboard application is able to access data via the RESTful API's provided by the server side of the application. The AngularJS library has the ability to consume API's that provide data in a JSON format. The data is then manipulated via the AngularJS directives for display within the template or for use in logic to control the application.

5.1.1.2. Dashboard Framework

The Dashboard framework is an open sourced AngularJS generic Dashboard framework composed of Angular JS Directives, Templates, Controllers and Services of which enables the construction of dynamic, configurable, and user friendly Dashboard layouts. This approach uses widgets which are small components of the user interface that enables a user to view data, perform a function or access a service. Widgets within the Dashboard support the following features:

- Dynamically adding/removing widgets
- Drag and drop widgets (with Angular-gridster)
- Horizontal and vertical widgets resize (with Angular-gridster)
- Fluid layout (widgets can have percentage-based width, or have width set in any other unit)
- Support real-time data access (REST, etc.)

Displayed below is an architecture diagram of how components tie with one another within this framework.

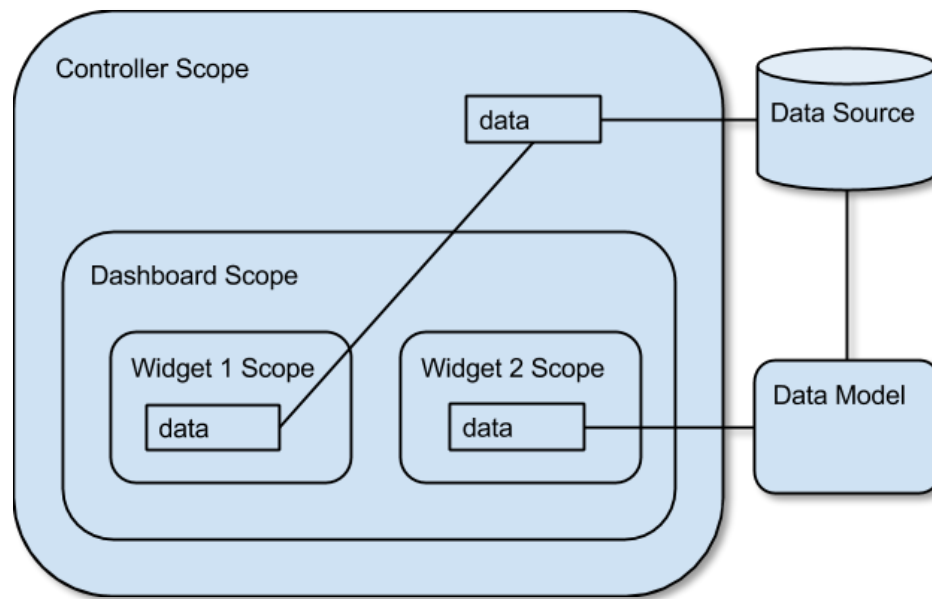


Figure 19: Dashboard Framework Architecture

Initially a Controller Scope for the page is defined to provide a foundation to instantiate a Dashboard layout and related data objects connected to the Reach Database. When a specific Dashboard layout is instantiated a Dashboard Scope is created along with a list of widgets and data objects that correspond to the layout. Widgets inherit the controller scope providing access to variables and properties. A Data Model object is provided to interface between the Reach Database and data objects that are needed for each widget's scope. Each widget's scope will have direct access to a separate instance of the data model. In order to enable widget to widget communication, a shared scope element has been added to the parent data model, *WidgetDataModel*. This shared scope element provides a broadcast and listen service for widgets to respond to. An example of this would be a click event in one widget could update multiple widgets based on the value selected.

When the Dashboard is instantiated, a configurable Dashboard Options Object can be specified for a specific user experience if desired. Examples of the options available in the widget definitions are: default widgets, widget buttons, widget storage, adding widgets, saving Dashboard, etc. Each widget within the Dashboard is able to have static/dynamic size settings.

The Dashboard framework also includes a means to save the state of the user's Dashboard when desired. Example features the Dashboard can automatically save if configured are:

- Instantiated widgets
- Size of widgets (width and height)
- Order that widgets are displayed
- Widget titles

For further detailed information please reference the [Malhar Angular Dashboard Documentation](#).

5.1.1.3. Drag and Drop Widgets

The Dashboard widgets are driven by Angular-gridster, an implementation of gridster-like widgets for Angular JS (<http://manifestwebdesign.github.io/angular-gridster/>). Angular-gridster allows for the building of intuitive draggable widgets which use a columnar approach. Section 508 requires 100% compliance with accessibility mandates that accommodate those with disabilities and other circumstances. Use of Angular-gridster achieves a 508 requirement by allowing users to move widgets solely using the keyboard. Instructions on how to resize and move widgets using the keyboard can be found in the IRDS User Manual.

5.1.1.4. Server Side

The Dashboard server is built on the Node.js platform and using Express for the HTTP server side web framework. Express provides a thin layer of fundamental web application features on top of Node.js' asynchronous event driven framework. Express provides for HTTP access to the RESTful Data Services which use the node-mssql driver ([node-mssql documentation](#)) to access data from MSSQL Reach Database. The Dashboard application will be managed by Internet Information Services (IIS) for Windows® Server.

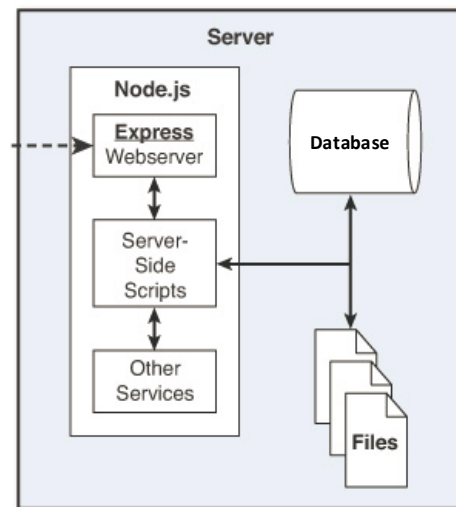


Figure 20: Node.js and Express

5.1.1.4.1. IIS for Node Management

IIS is a flexible, secure and manageable web server for hosting web applications. The iisnode project provides a native IIS 7/8 module that allows hosting of node.js applications in IIS 7/8 and IIS 7/8 Express (WebMatrix). This enables a consistent and reliable approach to process management. Some of the advantages of hosting node.js applications in IIS using the iisnode module are:

- Process management - The iisnode module takes care of lifetime management of node.exe processes making it simple to improve overall reliability. There is no need to implement infrastructure to start, stop, and monitor the processes.

- Scalability on multi-core servers - Since node.exe is a single threaded process, it only scales to one CPU core. The iisnode module allows creation of multiple node.exe processes per application and load balances the HTTP traffic between them, therefore enabling full utilization of a server's CPU capacity without requiring additional infrastructure code from an application developer.
- Auto-update - The iisnode module ensures that whenever the node.js application is updated (i.e. the script file has changed), the node.exe processes are recycled. Ongoing requests are allowed to gracefully finish execution using the old version of the application, while all new requests are dispatched to the new version of the app.
- Integrated debugging - The iisnode module is fully integrated with the node-inspector debugger.
- Access to logs over HTTP - The iisnode module provides access to the output of the node.exe process (e.g. generated by console.log calls) via HTTP.

IIS allows the developer to benefit from a range of IIS features, among them:

- Port sharing (hosting multiple HTTP applications over port 80)
- Security (HTTPS, authentication and authorization)
- URL rewriting
- Compression
- Caching
- Logging

5.1.1.4.2. User Access - Authentication and Authorization

Access to the Perceptive Reach Application requires a user to pass an Authentication and Authorization process. Authentication is the process by which a user is verified against a master directory. Authorization is the process by which the user's defined access level is determined and applied to the application.

With the implementation of the IIS as a part of the Dashboard application solution, the use of Active Directory and User Domain accounts allow for the easy authentication of a user. The process is a two tier process for Perceptive Reach where a user is validated against the Perceptive Reach User Table. If they exist in the Perceptive Reach User Table then their credentials are authenticated via IIS against Active Directory by Domain.

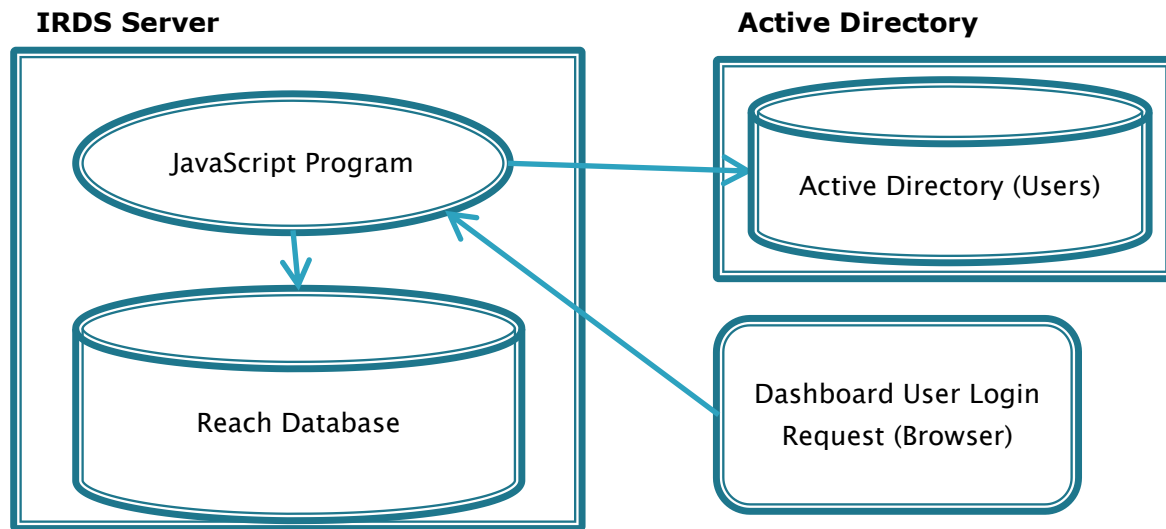


Figure 21: User Access Scheme

Once the user is authenticated successfully, the access level is determined by looking up the user's account for a predetermined user role. A user account most importantly contains User Role and Active Status information to control access and the experience for each user.

A User Profile represents a complete picture of a user's information within the Dashboard. This consists of the following:

- User Account data: Username, First Name, Last Name, etc.
- User Role data: Role Type and Access properties
- User Dashboard data: Widgets, Widget properties (size and location), and Accessed Views

User Role Matrix		<div> Clinical Care Team Members* Clinical Care Supervisor Supervisors - National Level Researchers Reporters System Administrators </div>							
Individual Veteran View Widgets	X	X	X		X	X			
Facility View Widgets	X**	X	X		X	X			
VISN / State / Region / Nation View Widgets		X***	X		X	X			
R / BIRT / MySQL Database Access				X	X	X			
System Administration Tools						X			
<p>*Clinical Care Team is a catchall term for a staff member who provides outreach and intervention services. (e.g. SPC, Social workers, care managers, physicians, psychologists, nurses, etc.)</p> <p>**Clinical Care Team Members will have access to the Facility View for their home Facility only by default</p> <p>***Clinical Care Supervisors (CCS) can be assigned which specific facilities they can see in their facility-level view. This allows VISN level users, for example, the ability to see the sites in their VISN (or a handful of sites), but not see all sites nationally.</p>									

Figure 22: User Access Roles

The Clinical Care Supervisor (CCS) user role has been added to the IRDS system. A user mapped to the CCS role is someone at the VISN or regional level that needs to access data for multiple facilities where they are assigned. This user role was created to give a user access to multiple facilities without giving them access to all national facilities.

5.1.1.4.3. RESTful Data Service

The Dashboard application will implement RESTful Data APIs to be accessed internally from the client code and provided by the server code. These services will provide CRUD (create, read, update and delete) options for data stored in the Perceptive Reach database. All data related to Veterans will be accessed via Read Only APIs, however data related to the actual Dashboard application such as user settings, application settings, and configuration parameters may be accessed via all of the available API actions. These RESTful APIs will be hosted in the server process based on Node.js using Express.js as the HTTP framework.

The REST APIs in Express have three components, the Express route, the API controller, and the node-mssql driver for database access. The route defines the end-point to be accessed by the Dashboard client requests such as “/api/getSampleData” and the internal route to the proper API controller. The API controller provides functionality to acquire and transform the data into JSON for a response back via the route. Within this functionality, node-mssql driver is used to pull data

from the database through the use of request input parameters with defined datatypes. Overall this provides a seamless request and response process that Express's HTTP process manages.

5.1.1.5. Build Process

The Dashboard Application (client and server) build process is managed by Grunt, The JavaScript Task Runner. This enables the build process to be consistent and automates tasks like minification, compilation, unit testing, linting, etc. Grunt and Grunt plugins are installed and managed via npm, the Node.js package manager. A typical Grunt setup will involve adding two files to a project:

- `package.json`: This file is used by npm to store metadata for projects published as npm modules.
- `Gruntfile`: This file is named `Gruntfile.js` and is used to configure or define tasks and load Grunt plugins.

The Dashboard Application is currently using Grunt to do the following:

- Compile all source code
- Configure directory structure (HTML, CSS, JavaScript, Templates, etc)
- Minification of JavaScript to server
- Manage the Express.js HTTP server by environment
- Validate Test Driven Development (TDD) scripts via Karma
- Provide LiveReload for the development environment which enables real-time code changes to be loaded in the server

In addition to these current tasks, Grunt can be configured to run custom tasks. These custom task can be referenced either in the Gruntfile or in an external JavaScript file.

5.1.2. Reach Database

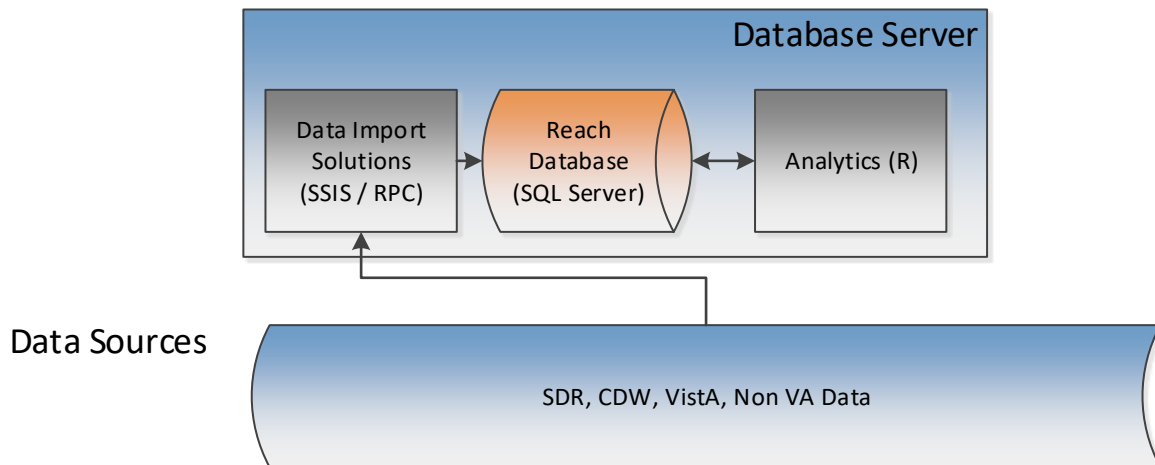


Figure 23: Reach Database

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

For more detailed information on the Reach Database, please see [Section 5.5](#).

5.1.3. Analytics Platform

The risk model that will be used within the IRDS solution has been developed using the open source statistical programming language R. The finalized model coefficients have been stored to a SQL table that will be referenced in an SSIS package. The model coefficients are applied to veteran data on an ongoing basis using SQL, therefore, exporting data to calculate risk scores in R is not required. However, if the model coefficients are to be updated at a future date, the model will need to be refitted using R.

The tools used to redevelop the risk model for the IRDS solution are:

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

For more detailed information on the Analytics Platform, please see [Section 5.7](#).

5.1.4. Direct Messaging

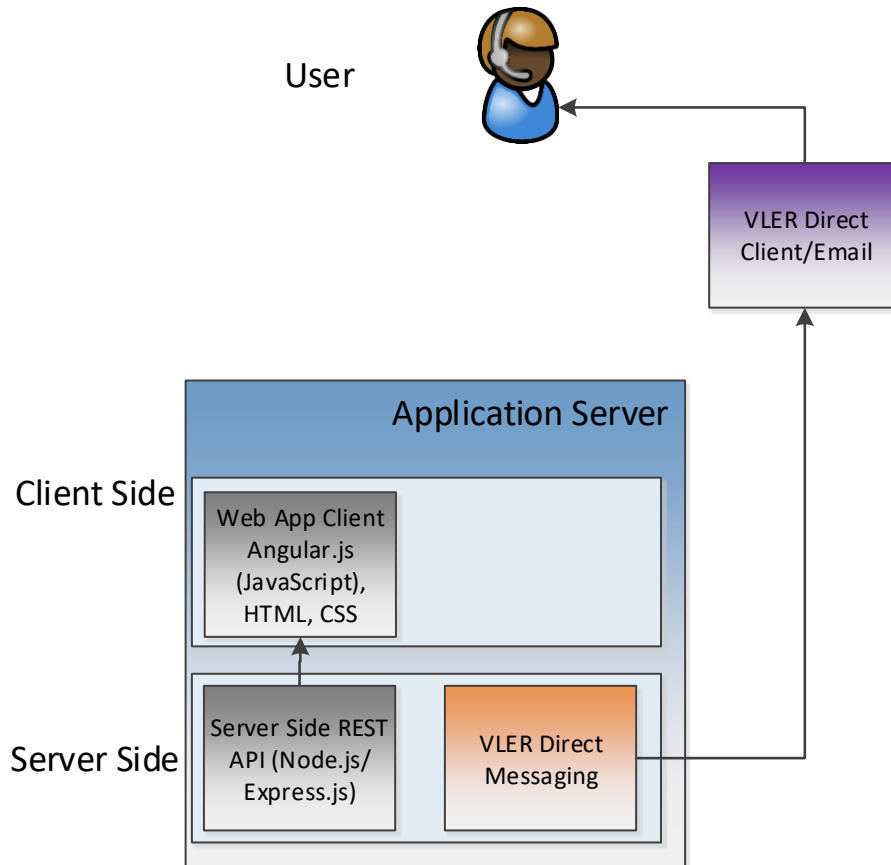


Figure 24: Direct Messaging

The Direct Messaging component will integrate into the existing VLER Direct Messaging architecture. The Direct Messaging will create unique messages based on new and existing data annotated in the IRDS. The application will run on a Node.js application server and communicate to the VLER Direct Messaging over the RESTful web services and hash-based message authentication code (HMAC) for secure authorization, sending, and receiving.

- Node.js (JavaScript runtime engine)
- RESTful web services (HTTP, XML, JSON)
- HMAC (hash-based message authentication code)

The Direct Message process will feature a notify queue table (Direct_NotifyQueue) that is monitored by the Perceptive Reach Notify application, a Node.js application. The queue table will be populated via the Perceptive Reach Database Risk Score routine for newly identified Veterans. The Notify application will run daily (configurable) to monitor the message queue table.

Given the synchronous nature of the messaging process and data, the Notify Application implements Async.js (<https://github.com/caolan/async>). Async.js is a utility module which provides straight-forward, powerful functions for working with asynchronous JavaScript in a synchronous manner. The essence of this process is to provide an approach where each subsequent action in the building of the message waits until the previous step is completed. This is managed with a callback feature that tells the Notify Application when to process the next record. Rather than utilizing typical JavaScript iterators which are asynchronous and fire independently of the next iteration, the Notify Application utilizes the `async.eachSeries` which provide a fire and wait approach (wait for the callback function to notify the iterator to process the next record).

The Notify application will query the queue table for additions, changes, and updates. The `MsgStatus` column will be monitored for statuses of “Send”. When found, a message is created with the Veterans information (Name, Last 4 SSN, Phone, Identified Risk, and Date) and sent to the location and/or resource such as Suicide Prevention Coordinator (SPC). The following rules help to elaborate on some of the actions that will take place in the Notify application:

- Generalized Rules:
 - Notifications are sent out for newly identified Veterans in the Middle and Top (newly = risk level on the previous day is different from today)
 - Notifications are ONLY sent out when a Veteran is increasing in risk level (i.e. Not identified to Middle or Middle to Top)
 - There is a *N day (configurable) window based on the previous message’s risk level to limit notifications within that window of time for same type of message.
 - TOP - 30 day window (should be configurable)
 - MIDDLE - 90 day window (should be configurable)
 - Scenario: Veteran identified as Top 20 days ago and a DM was sent. The Veteran fell back to Middle 15 days ago. Today Veteran identified as Top again, if the window is 30 days for a Top, no message is sent.
 - Notifications will consolidate Veterans on to the same message when multiple Veterans are identified for a location’s recipient(s)
 - A location can have one or more recipients, if more than one, then all recipients will receive the same message
- Specialized Rules:
 - Follow Up Rule:
 - Any Veteran identified in at a risk level that is not acknowledged within *N days (outreach status not updated away from empty) through the Roster Widget in the Dashboard will trigger a re-notification message to the location’s recipient(s). The Risk Level properties will determine follow-up window, if any.
 - TOP - 30 day window (should be configurable)
 - MIDDLE - It is currently not recommended that MIDDLE generate re-notification messages due to the volume of MIDDLE Veterans and the work load that would impose on the VA Staff in addressing the messages alone.

Once sent, the Notify application updates the MsgStatus column if the message is successfully sent. If a message is not sent successfully the error(s) will be recorded by the Notify application and the MsgStatus column will be noted in “error”.

The Notify application reviews the Veteran information to determine which SPC should receive a message. The Veteran’s facility listed in the record is compared to the SPC associated with that facility to determine where to send the message. If no SPC is associated with the facility listed then the direct message is sent to the administrative staff and noted in the error table. If a Veteran record does not include a facility then a message is sent to the administrative staff and noted in the error table.

The Notify application will build messages conforming to the VLER Direct Messaging API protocols and standards. The message will include the following fields: To, From, Subject, Body. The “To” field will contain the destination address of the SPC that is being notified. The “From” field will contain the Perceptive Reach administrative account. The subject line of each DM will say “Perceptive Reach: N Veteran(s) to Review” with “N” indicating the number of Veterans summarized in the body of the DM. The Body of the message will contain a direct link to the Perceptive Reach Dashboard and information to reach at risk Veteran including name, phone number, last 4 of SSN, identified risk, and facility.

The Direct Message application will store messages in the Perceptive Reach Database in the Direct_Message table. The table will contain the message id, date sent, error status, message type, message recipient and number of Veterans included in the message.

The Direct Message application will store errors in the Perceptive Reach Database in the DM_Error table. The table will contain detailed error logs including transmission errors, details related to any failed messages, and debugging info. The DM_Error table will store the message id, message details to be reviewed by the administrator.

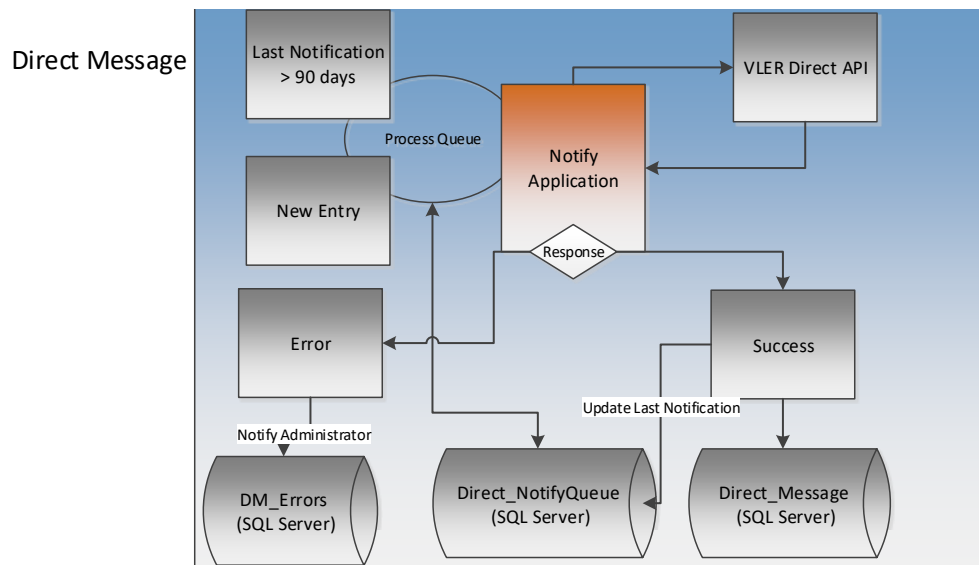


Figure 25: Direct Message

5.2. Specific Requirements

For further details on the specific functional requirements of the system, please refer to the IRDS Requirements Specification Document.

5.3. Continuous Integration / Continuous Delivery

The IRDS solution will be maintained through a Continuous Integration / Continuous Delivery process. This is an automated process, initiated when source code is submitted to the GitHub source code repository. Jenkins detects the submission and initiates a build and test process utilizing build tools such as Gulp and testing tools such as Selenium, Cucumber, and Maven.

As shown in the figure below, the development and testing (Dev-Test) environment and the pre-production environment are very similar. The Dev-Test environment is located in the VA Cloud and will not connect to production systems or utilize any PII/PHI. All testing will be done with a test data set. The Pre-Production environment is located behind the VA firewall and will connect to production systems such as the SDR and CDW databases and utilize PII/PHI for development, testing and operations.

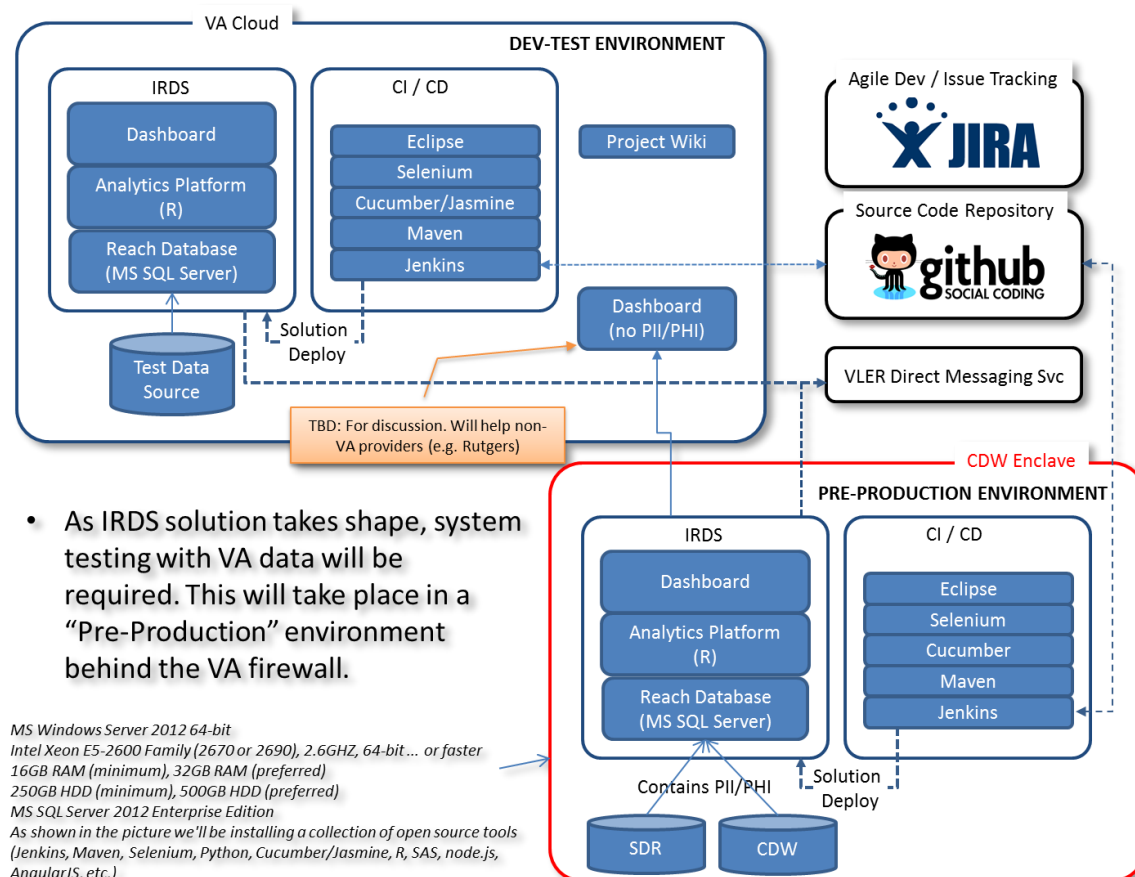


Figure 26: Continuous Integration

5.4. Data Design

- The IRDS system will use SQL Server 2012 as its data platform. All data will be stored in the Reach database.
- Data import solutions will be developed with SQL Server Integration Services (SSIS).
- On a Regular basis (at this time it is assumed weekly):
 - Patient demographic and risk factor data will be pushed from a Corporate Data Warehouse (CDW) server to the IRDS server.
 - Patient demographic data will be updated with MedSAS data. The Reach database contains a table containing patient demographics and their VA Scrambled SSN, that was imported from MedSAS SAS files.
 - Risk Scores will be calculated for each patient. For all patients identified as "At Risk", Dashboard widget data (Appointments, Emergency Contacts, etc.) will be pushed from CDW
 - The code for scoring patients based on the developed risk model, is contained in a stored procedure
 - Additional clinical data, which will be presented in the Dashboard, will be pulled to the IRDS system from the Suicide data Repository (SDR).
- On a nightly basis a stored procedure will run that will identify any necessary VLER messages that need to be sent to IRDS users and set those messages up on the VLER queue.

Table 8: 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 VistA, 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	Healthindicators.gov	Provider of data which is a collaboration of many Agencies and Offices within the Department of Health and Human Services. The HIW is maintained by the CDC's National Center for Health Statistics.	

5.4.1. Physical Data Model

The following database diagram provides an overview of the physical data model.

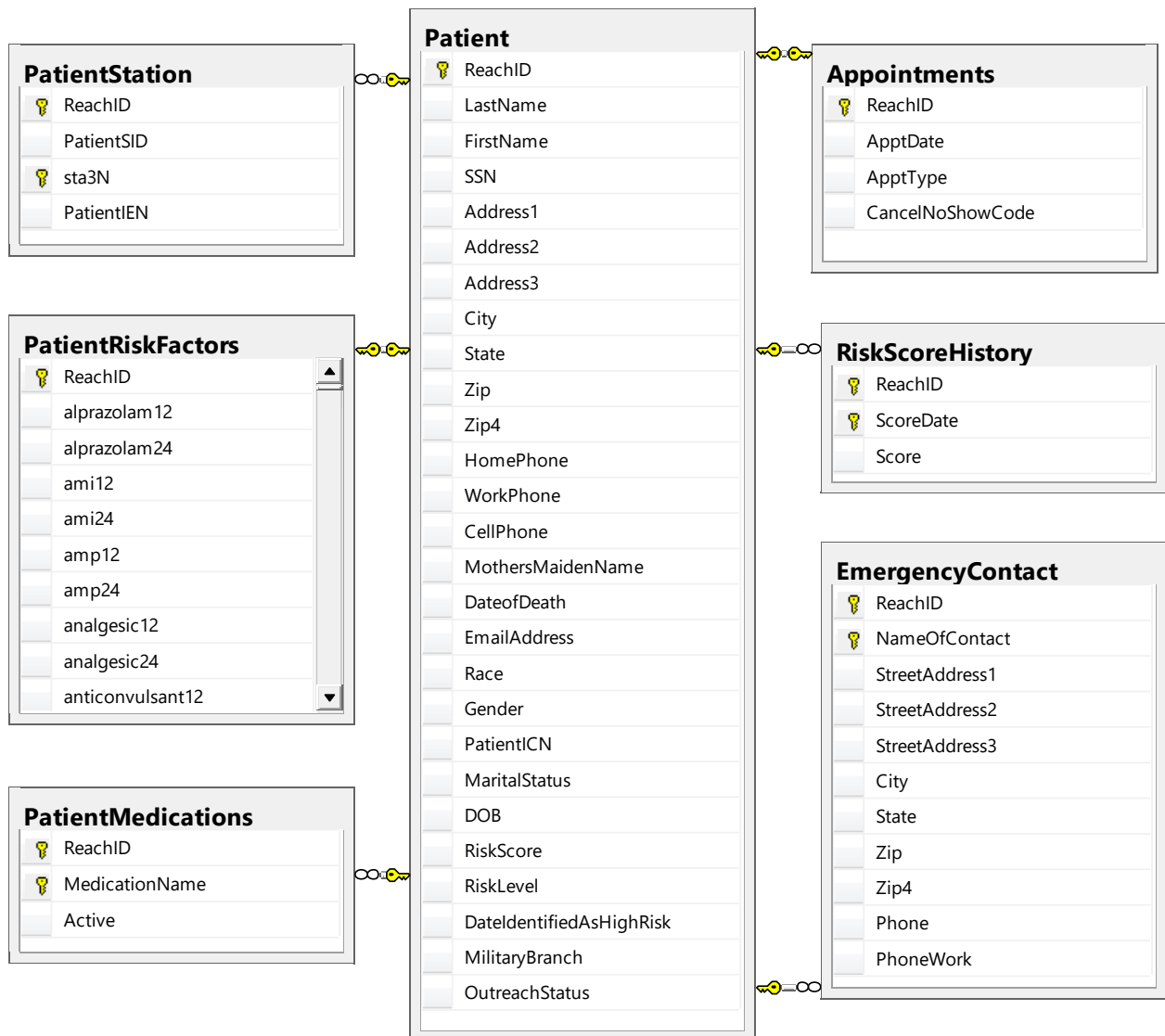


Figure 27: Data Model

5.4.1.1. CDW Project Database Objects

The IRDS project has been given a database (VACI_IRDS) on the server to be used as part of the automated processes for pushing data from the CDW database to Reach database on the IRDS system. All objects are maintained in the ETL schema.

Below is an inventory of database objects on the VACI_IRDS database.

Tables

Work tables for the CDW Import SSIS package:

- Patient

- PatientQueue
- PatientStation
- PatientSubset
- ReachIDs
- RiskFactors
- VHAPatientsRanked
- VHA_1DaysUse

Work tables for the Patient assessment SSIS package:

- AlreadyAtRisk
- NewAtRisk

Views

Views to improve performance:

- vwCPT_vProcedure
- vwCPT_Visit
- vwCSC_Visit
- vwPrescription_MedLog
- vwPrescription_RxOupatFill
- vwDaysUse_Visit
- vwFirstUse
- vwICD9_Visit
- vwLagca_Vdiagnosis
- Attempts_Vdiagnosis
- Attempts_InpatientDiagnosis
- VetInd_Visit

Work tables for the CDW Import SSIS package:

- vw_RaceCodes
- vw_MaritalCodes
- vw_MilitaryBranchCodes
- vw_MilitaryBranch
- vw_MedicationsNewAtRisk
- vw_MedicationsAlreadyAtRisk
- vw_DiagnosisNewAtRisk
- vw_DiagnosisAlreadyAtRisk

Stored Procedures

Stored procedures run during CDW Import:

- GetDaysUse
- Run_ICD9factors

Additional data elements are currently being investigated for importing into the Reach database. Their exact location within the VA data sources is being determined and access/documentation to those sources is in the process of being requested.

5.4.2. Data Dictionary

For details on the Data Dictionary, please reference [Database Design Specification](#).

5.4.3. Veteran De-Duping Process

The Reach data model will contain one master list of all Veterans imported into the IRDS system. As each Veteran is imported into the database through one of the data sources, a record will be created for that Veteran in the Veteran table and a unique ID will be assigned. It is possible that an Individual might be imported into the system through multiple sources. When this occurs, the multiple Veteran records created for the individual will be merged and all case level data will be linked to that one merged Veteran record. To achieve this, a de-duping process will be run after each data import is run.

This de-duping process will be contained in an SSIS package, which does the following:

1. Reviews the Veteran table for possible duplicates
2. Duplicate groups are determined (2 or more records that could be duplicates)
3. For each duplicate group a text file is created containing the pertinent demographic information(Name, SSN, DOB, Gender) for all records in that group
4. A python program processes these files and evaluates all the records in a group via a record de-duping algorithm
5. The python program returns the results to the SSIS process via another set of text files
6. The results from these files are placed in a temporary SQL table and the Veteran table is reconciled accordingly, by merging any records that were determined to be duplicates

Perceptive Reach Server

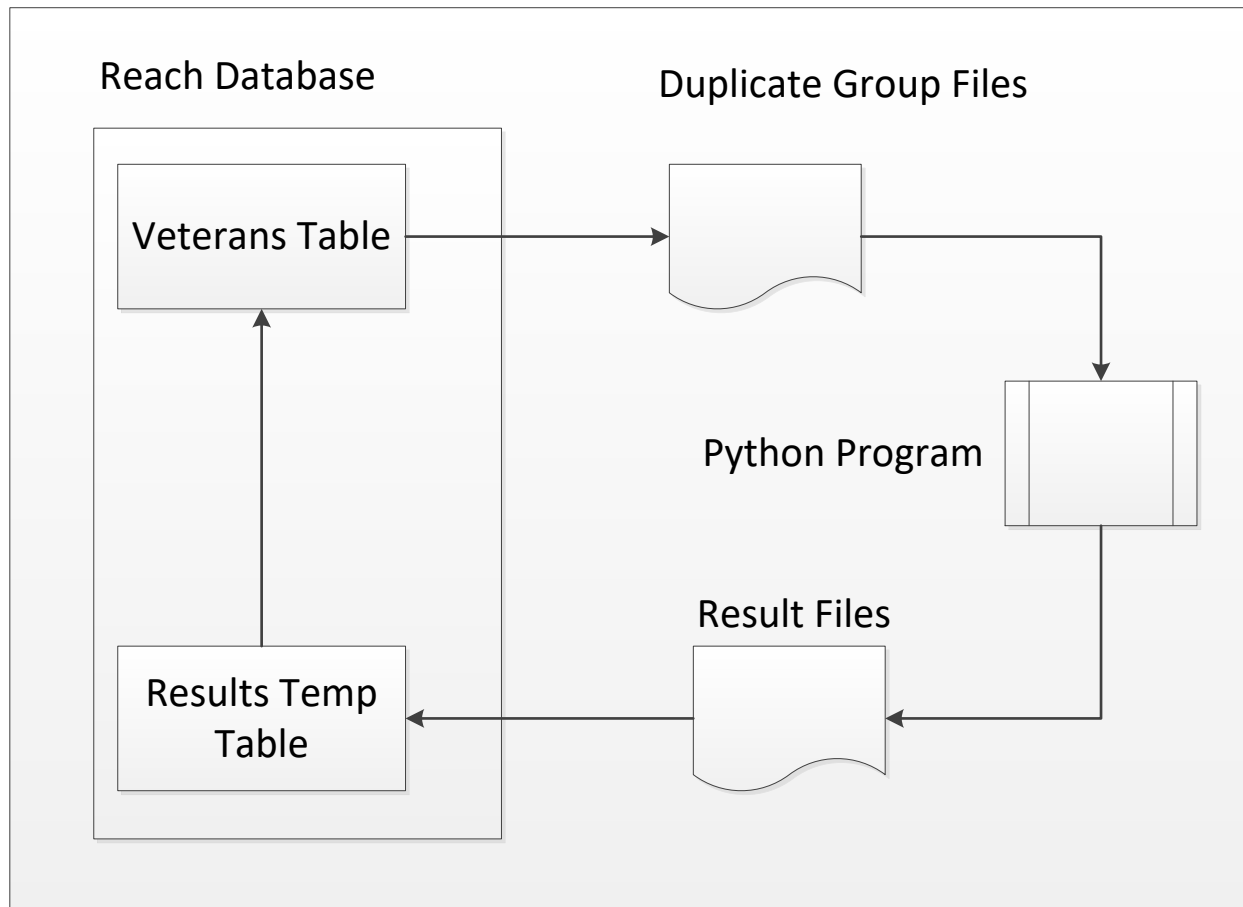


Figure 28: Veteran De-duping Overview

5.4.4. 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.

5.4.5. Data View

The system's "Data View" is included in [Section 5.5.1](#).

5.5. Service Oriented Architecture / ESS Detailed Design

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 accessed through direct access to the server on which the IRDS resides. IRDS will consume the VLER Direct ESS to deliver Direct Messaging. The overall architecture is still being developed for this section as requirements are finalized.

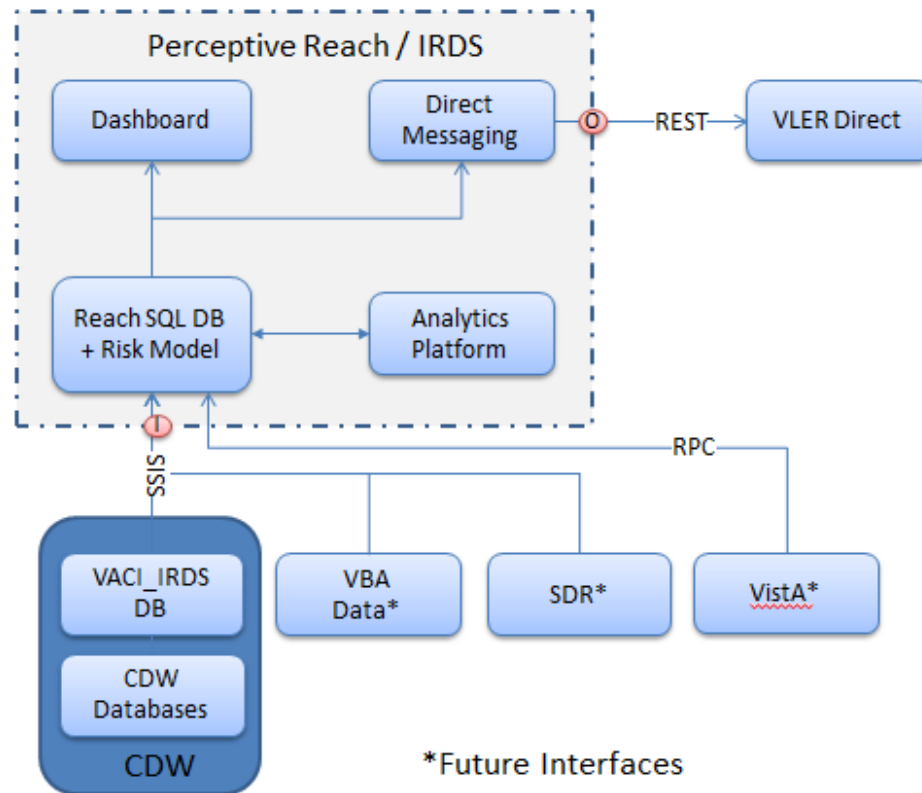


Figure 29: Architecture Overview

5.6. Data Analytics Platform

In order to redevelop the risk model originally created by the SMITREC team within VA; and test coefficient stability and out of sample accuracy of the redeveloped risk model, the appropriate analytical tools must be 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
- Ability to run on Unix/Linux
- Ability to connect to various databases
- Robust statistical modeling capabilities (basic and advanced)
- Data management capabilities
- Visualization capabilities
- Approved for use in the VA environment / One-VA TRM List of Technology

A number of tools were assessed to determine whether they satisfy each of the abovementioned criteria: SAS, SPSS, Stata, R, Python, Weka, Gephi, Eclipse, BIRT Jasper Reports and 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 the R programming language/software to grow and include a wide variety of built-in data analytic and statistical capabilities and remain an innovative tool.

For more information, reference the project's [Data Analytics and Risk Factors](#) specification document.

5.7. Risk Model

The risk model that has been developed to identify VHA users at a heightened risk of suicide within the IRDS solution is based on the previous risk model developed by the SMITREC team. The new risk model for the IRDS solution uses data from both CDW and MedSAS data sources. Specifically; data including prescriptions, diagnosis, procedures, Military Sexual Trauma, emergency department/urgent care visits and VHA usage comes from the CDW source, and demographic information comes from the MedSAS source. In total, there are 381 variables including interaction terms that are used in the logistic regression risk model that is in the IRDS solution. For further details on all aspects of the risk model, please refer to the Perceptive Reach Data Analytics and Risk Factors documentation.

5.8. External System Interface Design

For further details on the System Interface Design, please refer to IRDS Interface Design Specification.

6. Human-Machine Interface

6.1. Interface Design Rules

The user interface shall comply with VA standards, guidelines, Privacy Rules and section 508 requirements.

6.2. Inputs

Users of the Integrated Reach Database system consist of intervention service providers, VA mental health providers, and leadership resources. In addition, approved researchers may also be granted access to the system to analyze data and run reports. For a full description of the user classes, refer to the User Research Report.

Authorized users gain access to the application through login page, which requires the entry of a Lightweight Directory Access Protocol (LDAP) authentication logon user name and password.

6.2.1. User Roles

Table 9: User Roles

User Class	Description
Clinical Care Team Members	User class includes Suicide Prevention Coordinators (SPC), clinical professionals and other Mental Health staff. Some but not all will have limited experience using IT tools. Experience in Social Work and Clinical Psychology is typical.
Clinical Care Supervisors (CCS)	User class includes VA leadership at the region or VISN level that need access to data for multiple facilities but not all national facilities. VA staff in this user class have backgrounds in Nursing, Public Health, Social Work, and other social sciences.
VA leadership, VA Center of Excellence for Suicide Prevention staff, VA Mental Health leaders	User class includes VA staff in leadership positions with backgrounds in Medicine, Clinical Psychology, Nursing, Public Health, Social Work and various social sciences. Expertise with information technology tools varies broadly depending on background and role at VA. User class may also include Outreach and Intervention Team Members described above.
Researchers and Statisticians	User class includes individuals with backgrounds in Statistics, Epidemiology, Public Health, Medicine, Psychology, and various social sciences. Technical expertise for many users will be high, especially in using tools for statistical analysis and related research methods.
Reporting Users	User class includes individuals who generate reports related to Veteran suicide. These users may include individuals described in the user classes above plus additional VA managers, analysts, and administrators who are asked to generate reports. Expertise with various technology tools for many users will likely be high, especially in areas related to reporting tools, databases, and statistics.

Reporting Consumers	This indirect user class includes individuals in the target audience for reports, including internal VA clinical and administrative staff in addition to groups external to VA including other government agencies, US Congress, news media, Veterans Service Organizations, and the general public. Expertise in understanding and interpreting statistical data will vary across target audiences.
Sustainment Staff	User class includes technical VA staff with backgrounds in Information Technology, Computer Science, Statistics, and related fields with the ability to create new statistical models in the application and map the application to new interfaces.

6.3. Outputs

Outputs include data displayed on a web-based Dashboard, Direct Messages regarding identified at-risk Veterans, and data sets resulting from research or reporting activities.

6.4. Navigation Hierarchy

The following details the navigation hierarchy of Perceptive Reach.

Perceptive Reach Flow

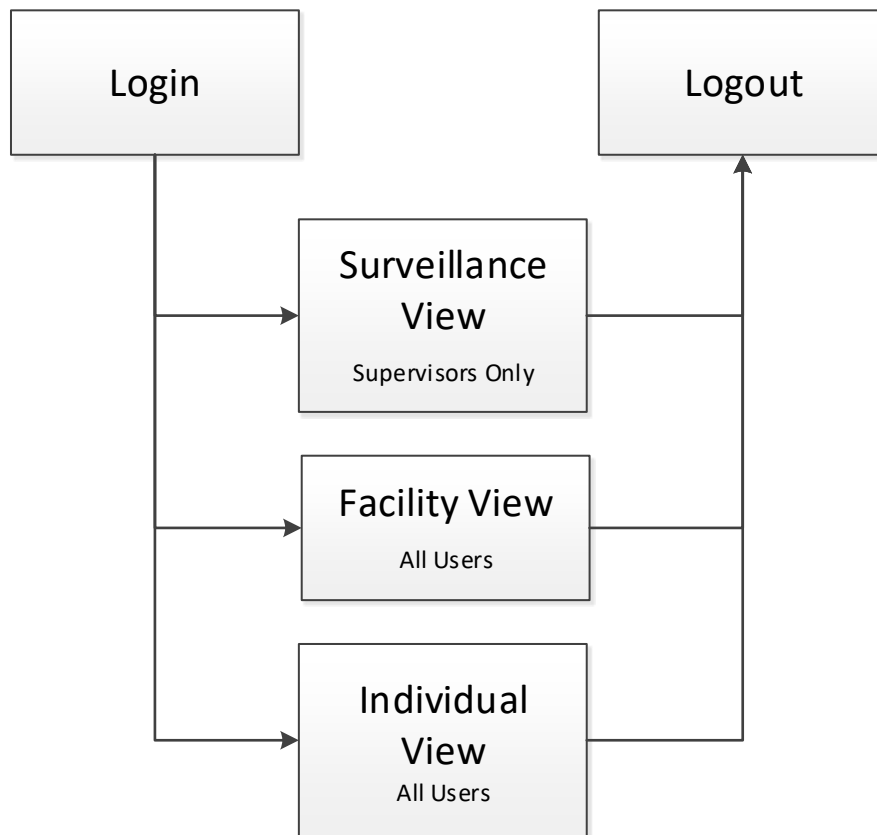


Figure 30: Perceptive Reach Flow

6.4.1. Login

Perceptive Reach

Perceptive Reach Login

This U.S government system is intended to be used by authorized VA network users for viewing and retrieving information only, except as otherwise explicitly authorized. VA information resides on and transmits through computer systems and networks funded by VA. All use is considered to be with an understanding and acceptance that there is no reasonable expectation of privacy for any data or transmissions on Government Intranet or Extranet (non-public) networks or systems. All transactions that occur on this system and all data transmitted through this system are subject to VHA Handbook 1605.02 directive Minimum Necessary Standard for Protected Health Information as users are expected to only access the minimum data necessary to perform their specific job functions. All transactions that occur on this system and all data transmitted through this system are subject to review and action including (but not limited to) monitoring, recording, relieving, copying, auditing, inspecting, investigating, restricting access, blocking, tracking, disclosing to authorized personnel, or any other authorized actions by all authorized VA and law enforcement personnel. All use of this system constitutes understanding and unconditional acceptance of these terms. Unauthorized attempts or acts to either (1) access, upload, change, or delete information on this system, (2) modify this system, (3) deny access to this system, or (4) accrue resources for unauthorized use on this system are strictly prohibited. Such attempts or acts are subject to action that may result in criminal, civil, or administrative penalties. This application may automatically log users out after fifteen minutes of inactivity.

If you have any further questions regarding Perceptive Reach and associated resources, Please contact the Help Desk at VAPerceptiveReachSupport@va.gov

Username

Password

Consent
☐ I accept the terms and conditions for accessing the system described above. Hit the spacebar as needed to check this box.

Login

Figure 31: Perceptive Reach Login

6.4.2. Surveillance View

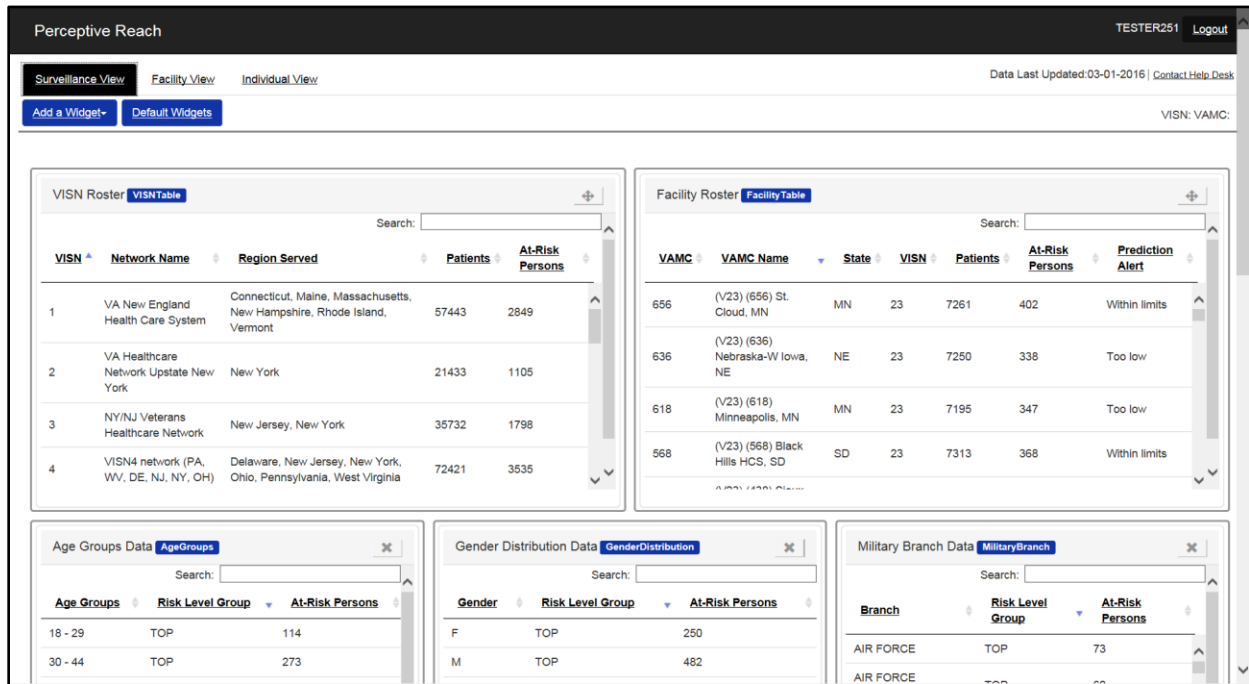


Figure 32: Surveillance View

6.4.3. Facility View

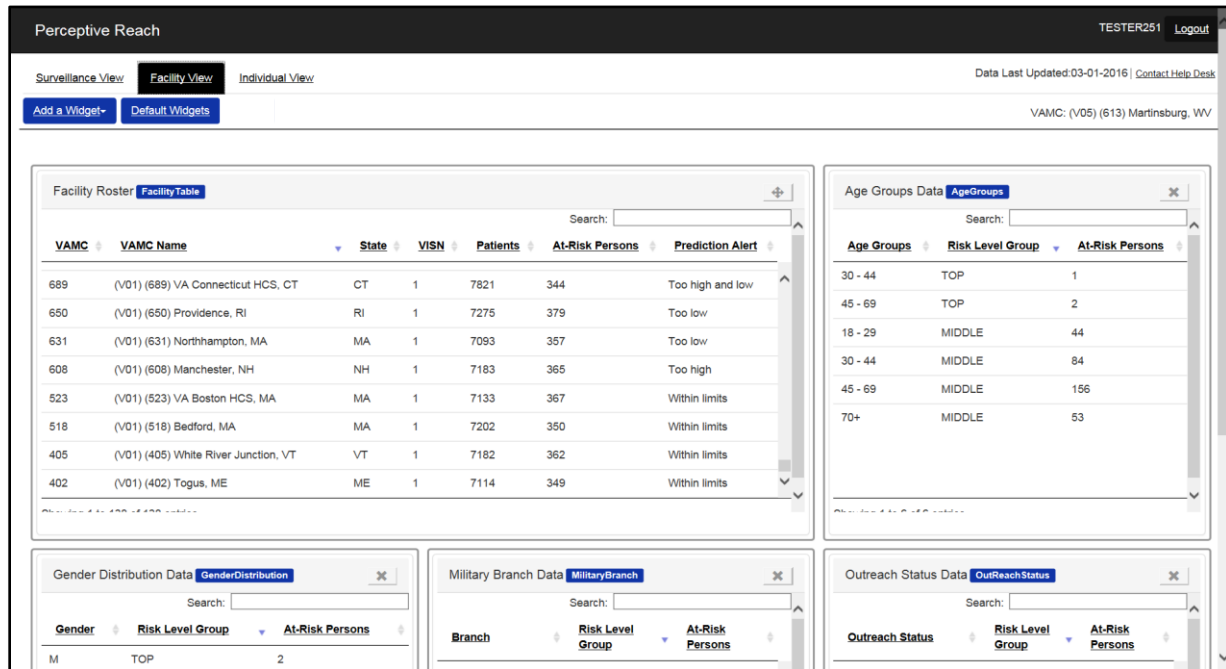


Figure 33: Facility View

6.4.4. Individual View

The screenshot displays the 'Individual View' of the Perceptive Reach system. At the top, there are tabs for 'Surveillance View', 'Facility View', and 'Individual View'. Below these are buttons for 'Add a Widget', 'Default Widgets', and 'Clear'. The main content area is divided into several sections:

- Clinical Decision Support:** Contains information about 'Chronic MIDDLE Risk Features' and 'Action' steps for managing these patients.
- Patient Roster by VAMC:** A table listing patients with columns for Name, SSN, Phone, Date First Identified, Statistical Risk Level, and Outreach Status. The table includes data for several veterans, such as 'Veteran_109258' and 'Veteran_109574'.
- Data Entry:** A section for entering data, including 'High Risk Flag Information' and 'SPAN Records'.
- Patient Contact:** A section for entering contact information for the selected patient.
- Emergency Contact Information:** A section for entering emergency contact details.

Figure 34: Individual View

7. Security and Privacy

The IRDS system meets VA security and privacy standards, and obtained an Authority to Operate in September 2015.

7.1. Security

The IRDS system design will comply with specific security mechanisms at the application level in accordance with the NIST 800-53 Rev 4 Security Controls as described in the IRDS System Security Plan (SSP). A Security Risk Assessment was performed as part of the A&A package. IRDS is categorized as a FISMA “High” system and all 800-53 security controls have been assessed and the information is detailed within the IRDS SSP. Please see the SSP for additional details. These documents are available within the VA’s EO instance of the Governance Risk and Compliance (GRC) tool, RiskVision.

Encryption of data in transit is protected using Federal Information Processing Standard (FIPS) 140-2 compliant encryption algorithms and products to protect patient data. A SQL Server database will be used to store data used within IRDS. All database reads and writes occur within an encrypted connection and the data is encrypted during transport via the SQL Server Always Encrypted feature. Always Encrypted makes encryption transparent to applications. An Always Encrypted-enabled driver installed on the client computer achieves this by automatically encrypting and decrypting sensitive data in the SQL Server client application. The driver

encrypts the data in sensitive columns before passing the data to SQL Server, and automatically rewrites queries so that the semantics to the application are preserved. Similarly, the driver transparently decrypts data, stored in encrypted database columns, contained in query results.

7.2. Privacy

The IRDS system takes a defense in depth approach to protecting PII/PHI data to include the following protection mechanisms (referenced in [Section 7.1](#)):

- IRDS is in compliance with privacy controls per NIST 800-53 as documented in SSP
- Only accessible from VA network
- API protected by a policy enforcement/policy decision point
- VA hosts in AITC are protect by VA boundary protections at the hosting facility and only administrators have access to the machines
- Data encryption in transit using SQL Server Always Encrypt on any network traffic beyond the local enclave

IRDS has a current and signed Privacy Threshold Analysis (PTA) and Privacy Impact Assessment (PIA). The official copies of these documents are stored in the VA's Governance Risk and Compliance (GRC) tool, RiskVision.

8. Relationship to Other Documents and Plans

The following IRDS documents may be referenced in tandem with the information recorded here. The documents can be located at:

https://internal.vacloud.us/wiki/pages/81X0Z7X5/Perceptive_Reach_Deliverables.html

- [Project Management Plan \(PMP\)](#)
- [Interface Design Specification](#)
- [Requirements Specification Document \(RSD\)](#)
- [User Research Report](#)
- [Requirements Traceability Matrix \(RTM\)](#)
- [Database Design Specification](#)
- [Data Analytics Sandbox Specification](#)
- [Dashboard Design Document](#)

9. Acronyms and Abbreviations

Table 10: Acronyms and Abbreviations

Acronym	Term
Army STARRS	Army Study to Assess Risk and Resilience in Service members
API	Application Program Interface

Acronym	Term
CDC	Center for Disease Control
CDW	Corporate Data Warehouse
CI	Continuous Integration
DB	Database
DoD	Department of Defense
EA	Enterprise Architecture
ETL	Extract, Transform, Load
FMQL	FileMan Query Language
GB	Gigabyte
HMAC	Hash-based Message Authentication Code
HTTPS	Hypertext Transfer Protocol Secure
IIS	Internet Information Services for Windows Server
ICD	International Classification of Diseases
IRDS	Integrated Reach Database System
IT	Information and Technology
JDBC	Java Database Connectivity
JSON	JavaScript Object Notation
NDI	National Death Index
ODBC	Open Database Connectivity
PMP	Project Management Plan
REST	RESTful
RPC	Remote Procedure Call
RSD	Requirements Specification Document
RTM	Requirements Traceability Matrix
SAS	Statistical Analysis System
SDCD	State Death Certificate Data
SDR	Suicide Data Repository
SPA	Single Page Application
SPAN	Suicide Prevention Applications Network

Acronym	Term
SPC	Suicide Prevention Coordinator
SQL	Structured Query Language
SSIS	SQL Server Integration Services
SSN	Social Security Number
TBD	To Be Determined
TRM	Technical Reference Manual
UBHC	University Behavioral Health Care
UI	User Interface
VA	Department of Veterans Affairs
VACI	VA Center for Innovations
VAMC	VA Medical Center
VCL	Veterans Crisis Line
VHA	Veterans Health Administration
VISN	Veterans Integrated Service Networks
VistA	Veterans Health Information Systems and Technology Architecture
VLER	Virtual Lifetime Electronic Record
VSO	Veterans Service Organizations
PMAS	Project Management Accountability System
SDD	System Design Document
DBMS	Database Management System
HHS	U.S. Department of Health and Human Services
CRS	Congressional Research Service
PR	Perceptive Reach
KNIME	Konstanz Information Miner
BIRT	Business Intelligence and Reporting Tools
DOB	Date of Birth
ID	Identification
MUMPS	Massachusetts General Hospital Utility Multi-Programming System

Acronym	Term
CPU	Central Processing Unit
LAN	Local Area Network
WAN	Wide Area Network
MVC	Model View Controller
PII	Personally identifiable information
PHI	Protected health information
PACER	Public Access to Court Electronic Records
TBI	Traumatic Brain Injury
VR&E	Vocational Rehabilitation and Employment
VBA	Veterans Benefit Administration
VHA	Veterans Health Administration
PTSD	Post-Traumatic Stress Disorder
US	United States