**VA Informatics and Computing Infrastructure**

(VINCI) Enhancements (E)

Data Access Request Tracker (DART)

System Design Document

Milestone 1



May 2016

Department of Veterans Affairs

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 5/19/2016 | 1.12 | Updated section 3.2.3.2 with current list of Reports | PII |
| 4/27/2016 | 1.11 | Added Navigation Diagram (8.4) and Alt Text | PII  PII |
| 3/22/2016 | 1.10 | Updated section 2.3 with tables | PII |
| 2/17/2016 | 1.9 | Updated sections that are NA (VISTA specific), updated software detailed design and removed references to service oriented architecture plus updated for 508 compliance | PII PII  PII |
| 1/14/2016 | 1.8 | Removed future scope per AERB, made URL updates, included hardware sizing, removed VistA specific elements, and major formatting | PII PII |
| 12/03/2015 | 1.7 | TRM updates and updates to include block diagram | PII PII |
| 11/16/2015 | 1.61 | Added Milestone 1 to cover, edited per suggestions of the pre AERB review. | PII |
| 11/10/2015 | 1.6 | Update SQL Server version, clarified scope inclusions. Updates to sections 1-5 | PII and PII PII |
| 10/02/2015 | 1.5 | Updated scope inclusions | PII PII |
| 6/30/2015 | 1.4 | Updated SDD to new template | PII |
| 6/25/2015 | 1.3 | Updated based on responses on ETA checklist | PII |
| 06/12/2015 | 1.2 | Final draft for VA | PII |
| 06/05/2015 | 1.1 | 1st draft for VA review | PII |
| 05/19/2015 | 1.0 | Initial draft created | PII |

Table of Contents

[1. Introduction 1](#_Toc446406367)

[1.1. Scope Inclusions 2](#_Toc446406368)

[1.2. User Profiles 2](#_Toc446406369)

[2. Background 3](#_Toc446406370)

[2.1. Overview of the System 3](#_Toc446406371)

[2.2. Overview of the Business Process 4](#_Toc446406372)

[2.3. Overview of the Significant Requirements 5](#_Toc446406373)

[2.3.1. Overview of Significant Functional Requirements 5](#_Toc446406374)

[2.3.2. Capacity 5](#_Toc446406375)

[2.3.3. Availability 6](#_Toc446406376)

[3. Conceptual Design 7](#_Toc446406377)

[3.1. Conceptual Application Design 7](#_Toc446406378)

[3.1.1. Application Context 9](#_Toc446406379)

[3.1.2. Application Locations 10](#_Toc446406380)

[3.2. Conceptual Data Design 10](#_Toc446406381)

[3.2.1. Project Conceptual Data Model 10](#_Toc446406382)

[3.2.2. Database Information 10](#_Toc446406383)

[3.2.3. User Interface Data Mapping 11](#_Toc446406384)

[3.2.3.1. Application Screen Interface 11](#_Toc446406385)

[3.2.3.1.1. Dashboard Page for Researchers 11](#_Toc446406386)

[3.2.3.1.2. New Request 12](#_Toc446406387)

[3.2.3.1.3. Activity Information Tab 12](#_Toc446406388)

[3.2.3.1.4. Participants Tab 13](#_Toc446406389)

[3.2.3.1.5. Data Sources Tab 13](#_Toc446406390)

[3.2.3.1.6. Documents Tab 14](#_Toc446406391)

[3.2.3.1.7. Submit Tab 14](#_Toc446406392)

[3.2.3.1.8. Administration 15](#_Toc446406393)

[3.2.3.2. Application Report Interface 15](#_Toc446406394)

[3.2.3.3. Unmapped Data Element 15](#_Toc446406395)

[3.3. Conceptual Infrastructure Design 15](#_Toc446406396)

[3.3.1. System Criticality and High Availability 15](#_Toc446406397)

[3.3.2. Special Technology 16](#_Toc446406398)

[3.3.3. Technology Locations 16](#_Toc446406399)

[3.3.4. Conceptual Infrastructure Diagram 17](#_Toc446406400)

[3.3.4.1. Location of Environments and External Interfaces 18](#_Toc446406401)

[3.3.4.2. Conceptual Production String Diagram 18](#_Toc446406402)

[4. System Architecture 18](#_Toc446406403)

[4.1. Hardware Architecture 18](#_Toc446406404)

[4.1.1. DART Application Servers 19](#_Toc446406405)

[4.2. Software Architecture 19](#_Toc446406406)

[4.2.1. Process View 19](#_Toc446406407)

[4.2.2. Deployment View 20](#_Toc446406408)

[4.2.3. Implementation View 20](#_Toc446406409)

[4.2.4. Modules 21](#_Toc446406410)

[4.3. Network Architecture 23](#_Toc446406411)

[4.4. Service Oriented Architecture / ESS 23](#_Toc446406412)

[4.5. Enterprise Architecture 23](#_Toc446406413)

[5. Data Design 27](#_Toc446406414)

[5.1. DBMS Files 28](#_Toc446406415)

[5.2. Non-DBMS Files 28](#_Toc446406416)

[5.3. Data View 29](#_Toc446406417)

[6. Detailed Design 29](#_Toc446406418)

[6.1. Hardware Detailed Design 29](#_Toc446406419)

[6.2. Software Detailed Design 29](#_Toc446406420)

[6.2.1. Conceptual Design 29](#_Toc446406421)

[6.2.1.1. Product Perspective 29](#_Toc446406422)

[6.2.1.2. User Interfaces 30](#_Toc446406423)

[6.2.1.3. Hardware Interfaces 30](#_Toc446406424)

[6.2.1.4. Software Interfaces 30](#_Toc446406425)

[6.2.1.5. Communications Interfaces 30](#_Toc446406426)

[6.2.1.6. Memory Constraints 30](#_Toc446406427)

[6.2.1.7. Special Operations 30](#_Toc446406428)

[6.2.1.8. Product Features 30](#_Toc446406429)

[6.2.1.9. User Characteristics 31](#_Toc446406430)

[6.2.1.10. Dependencies and Constraints 31](#_Toc446406431)

[6.2.2. Specific Requirements 31](#_Toc446406432)

[6.2.2.1. Database Repository 31](#_Toc446406433)

[6.2.2.2. System Features 31](#_Toc446406434)

[6.2.2.2.1. Design Element Tables 31](#_Toc446406435)

[6.2.2.2.2. Routines (Entry Points) 32](#_Toc446406436)

[6.2.2.2.3. Templates 32](#_Toc446406437)

[6.2.2.2.4. Bulletins 32](#_Toc446406438)

[6.2.2.2.5. Data Entries Affected by the Design 32](#_Toc446406439)

[6.2.2.2.6. Unique Record(s) 32](#_Toc446406440)

[6.2.2.2.7. File or Global Size Changes 32](#_Toc446406441)

[6.2.2.2.8. Mail Groups 32](#_Toc446406442)

[6.2.2.2.9. Security Keys 32](#_Toc446406443)

[6.2.2.2.10. Options 32](#_Toc446406444)

[6.2.2.2.11. Protocols 32](#_Toc446406445)

[6.2.2.2.12. Remote Procedure Call (RPC) 32](#_Toc446406446)

[6.2.2.2.13. Constants Defined in Interface 32](#_Toc446406447)

[6.2.2.2.14. Variables Defined in Interface 32](#_Toc446406448)

[6.2.2.2.15. Types Defined in Interface 33](#_Toc446406449)

[6.2.2.2.16. GUI 33](#_Toc446406450)

[6.2.2.2.17. GUI Classes 33](#_Toc446406451)

[6.2.2.2.18. Current Form 33](#_Toc446406452)

[6.2.2.2.19. Modified Form 33](#_Toc446406453)

[6.2.2.2.20. Components on Form 33](#_Toc446406454)

[6.2.2.2.21. Events 33](#_Toc446406455)

[6.2.2.2.22. Methods 33](#_Toc446406456)

[6.2.2.2.23. Special References 33](#_Toc446406457)

[6.2.2.2.24. Class Events 33](#_Toc446406458)

[6.2.2.2.25. Class Methods 33](#_Toc446406459)

[6.2.2.2.26. Class Properties 33](#_Toc446406460)

[6.2.2.2.27. Uses Clause 33](#_Toc446406461)

[6.2.2.2.28. Forms 33](#_Toc446406462)

[6.2.2.2.29. Functions 33](#_Toc446406463)

[6.2.2.2.30. Dialog 33](#_Toc446406464)

[6.2.2.2.31. Help Frame 34](#_Toc446406465)

[6.2.2.2.32. HL7 Application Parameter 34](#_Toc446406466)

[6.2.2.2.33. HL7 Logical Link 34](#_Toc446406467)

[6.2.2.2.34. COTS Interface 34](#_Toc446406468)

[6.3. Network Detailed Design 34](#_Toc446406469)

[6.4. Security and Privacy 34](#_Toc446406470)

[6.4.1. Security 34](#_Toc446406471)

[6.4.2. Privacy 34](#_Toc446406472)

[6.5. Service Oriented Architecture / ESS Detailed Design 35](#_Toc446406473)

[7. External System Interface Design 35](#_Toc446406474)

[7.1. Interface Architecture 35](#_Toc446406475)

[7.2. Interface Detailed Design 35](#_Toc446406476)

[8. Human-Machine Interface 35](#_Toc446406477)

[8.1. Interface Design Rules 35](#_Toc446406478)

[8.1.1. Visibility of System Status 35](#_Toc446406479)

[8.1.2. Match between System and the Real World 36](#_Toc446406480)

[8.1.3. User Control and Freedom 36](#_Toc446406481)

[8.1.4. Consistency and Standards 36](#_Toc446406482)

[8.1.5. Error Prevention 36](#_Toc446406483)

[8.1.6. Recognition Rather Than Recall 36](#_Toc446406484)

[8.1.7. Flexibility and Efficiency of Use 36](#_Toc446406485)

[8.2. Inputs 36](#_Toc446406486)

[8.3. Outputs 36](#_Toc446406487)

[8.4. Navigation Hierarchy 37](#_Toc446406488)

[8.4.1. Screen [x.1] 37](#_Toc446406489)

[8.4.2. Screen [x.2] 37](#_Toc446406490)

[8.4.3. Screen [x.3] 37](#_Toc446406491)

[9. Attachment A – Approval Signatures 38](#_Toc446406492)

[A. Additional Information 39](#_Toc446406493)

[A.1. Identification of Technology and Standards 39](#_Toc446406494)

[A.2. Constraining Policies, Directives and Procedures 39](#_Toc446406495)

[A.3. Requirements Traceability Matrix 39](#_Toc446406496)

[A.4. Packaging and Installation 39](#_Toc446406497)

[A.5. Design Metrics 39](#_Toc446406498)

List of Tables

Table 1: Scope Inclusions 12

Table 2: User Classes 12

Table 3: Business Process 14

Table 4: Objects in the High Level Application Design 15

Table 5: Application Context Description 16

Table 6: Application Locations 17

Table 7: Application Users 17

Table 8: Database Inventory 18

Table 9: Technology Locations 24

List of Figures

Figure 1: Business Process Diagram 14

Figure 2: High-Level Application Design 15

Figure 3: Application Context Diagram 16

Figure 4: UI of Dashboard Page for Researchers 19

Figure 5: UI of "Information" tab for New Request 19

Figure 6: UI of Activity Information Tab 20

Figure 7: UI of “Participants” Tab 20

Figure 8: UI of “Data” tab, used to select the desired Data Sources 21

Figure 9: UI of “Documents” tab’ used to upload and view documents 22

Figure 10: UI of “Submit” tab, used to submit a request for review by the appropriate Admin group(s) 22

Figure 11: Conceptual Infrastructure Diagram 25

Figure 12: VINCI DART Infrastructure Diagram 25

Figure 13: Software Architecture 27

Figure 14: DART Architecture (Application Layer) 28

Figure 15: DART Data & Control Flow 29

Figure 16: Conceptual Infrastructure Diagram 30

Figure 17: Data Design Diagram 34

Figure 18: DBMS File Structure 35

Figure 19 : DART Component Diagram 37

# Introduction

The mission of the Department of Veterans Affairs (VA), Office of Information & Technology (OI&T), is to provide benefits and services to veterans of the United States Armed Forces. In meeting these goals, OI&T strives to provide high quality, effective, and efficient Information Technology (IT) services to those responsible for providing care to the veterans at the point-of-care, as well as throughout all the points of the veterans’ health care. The VA depends on Information Management/Information Technology (IM/IT) systems to meet mission goals.

The VA Informatics and Computing Infrastructure (VINCI) Program is intended to fulfill the need of the VA Research community and business intelligence for a secure and powerful analytical environment that provides access to appropriate data. The VINCI Program will provide data, data analysis tools, software, software development tools and support for the VA Researchers and is expanding the VA capability for informatics research and business intelligence.

Data Access Request Tracker (DART) is a software application available in the VINCI Program. It is a custom web application that provides researchers and operations staff with a streamlined workflow to facilitate transparent and timely decision responses. DART is a standardized application used to gather requests access requirements, specify request locations and associated personnel, determine and gather the appropriate required documentation, distribute the documentation to the appropriate reviewers, and provide functionality for reviewers to approve requests.

The VINCI DART project will focus on gathering request access requirements; specify request locations and associated personnel; determine and gather the appropriate required documentation; distribute the documentation to the appropriate reviewers; and provide functionality for reviewers to approve requests.

Specifically, this design covers the following aspects of DART:

* Efficient data management. SQL Server 2014 delivers high mission-critical scale, more efficient IT, and expanded reporting and analytics through self-service business intelligence.
* Usability. DART will provide intuitive web-based interfaces to users with varying levels of access to DART data and system resources, and enforce user security policies.
* Interoperable and portable computing. DART will incorporate the Hibernate Configuration for connecting to the database and performing standard data modifications, and SQL Server 2014 – as a relational model database server and storage space.
* VINCI Business Requirements Document (BRD)

SQL Server 2014 is currently in an evaluation period on the VA TRM site. It is currently being used by VINCI and CDW through the TRM waiver process (see embedded email below).

.

Information provided by reference: BISL SDE Mr. PII .

## Scope Inclusions

The below table describes In-Scope items.

Table : Current Increment Scope Inclusions

| Priority | Description |
| --- | --- |
| **1** | **Preparatory to Research –** Incorporate the Preparatory to Research process into the DART application so that all research related processes are in a single location. |
| **2** | **Reporting –** Provide automated and ad hoc reports as defined by the DART Reporting Business Requirements Document (BRD). |
| **3** | **Add-Remove Review –** Allow the Initial National Data System (NDS) Reviewer the ability to select and deselect reviews as needed. Adding or removing a review will have an effect on the tracking history and status timeline information. Additionally, the reason for the change will need to be captured. |
| **4** | **Data Sources by Individual** – Allow end-user the ability to distinguish which user requires access to data sources that are provided on an individual basis. When a Principal Investigator (PI) submits a request in the DART application, not all Researchers on the project will require access to Compensation and Pension Record Interchange (CAPRI)/VistAWeb or the Mainframe. Currently, the application requires a form for each Researcher associated with the request. Application needs to allow the PI to select which requestors require access to certain data sources, perhaps with the use of a pop-up box. |
| **5** | **Request Changes Communication** – Need to be able to add a descriptor to the message so the requestor knows what needs to be changed. Need to be able to add a brief description to the subject line. The long explanation would still remain in the body of the communications message. |
| **6** | **Performance Improvements** – Improve load times for requests with a lot of amendments. Remove the request reload when switching between History/Communications pages and the request. Improve document handling and storage. |
| **7** | **Participant Tracking** – Indicate when a participant was added or deleted from a request. |

## User Profiles

There are three classes of users of the DART application. They are researchers, approving authorities, and a read-only administrator role. See Table 2: User Classes below for more details.

Table : User Classes

| **Type of User** | **Description** | **Access and Responsibilities** |
| --- | --- | --- |
| Primary Users | Researchers:   * DART Principal Investigator * DART User * DART Reviewer * DART Owner * DART Business Owner | Read/Write tools access to DART.  Responsible for submitting requests. |
| Primary Users | Approving Authorities:   * NDS Administrator * NDS Reviewer * Office of Research and Development (ORD) Reviewer | Read/Write access to DART.  Responsible for reviewing and approving requests. |
| Secondary Users | Administrators:   * DART Administrator | Read/View-only access to DART.  Responsible for viewing requests. |

# Background

## Overview of the System

The DART application is a secure, custom online application that is used to:

* Gather request data access requirements.
* Specify request locations and associated personnel.
* Determine and gather the appropriate required documentation.
* Distribute the documentation to the appropriate reviewers.
* Perform reviews and approve requests.
* Send out email notices to the appropriate parties during the process informing them of needed actions or information.
* Facilitate and capture text communications between study personnel and the approvers.

The DART application was developed to meet a perceived need of some set of potential researchers and operations staff with a streamlined workflow to facilitate transparent and timely decision responses.

The benefits of DART:

* It fosters better communication between researchers and operations staff.
* It provides tracking of the request approval process.

The DART Dashboard displays the status of each request.

The major users of the system:

* Customer (Requester) Group - Create DART request with list of participants and locations and uploaded documents.
* NDS Group - The NDS group performs the initial approval, selects the middle-level review groups (depending on the list of selected Data Sources) for inclusion in the approval process and performs the final approval.
* Privacy Group
* Security Group
* ORD Group
* OEF-OIF Group
* DART Surgery Group
* CAPRI Administrators

## Overview of the Business Process

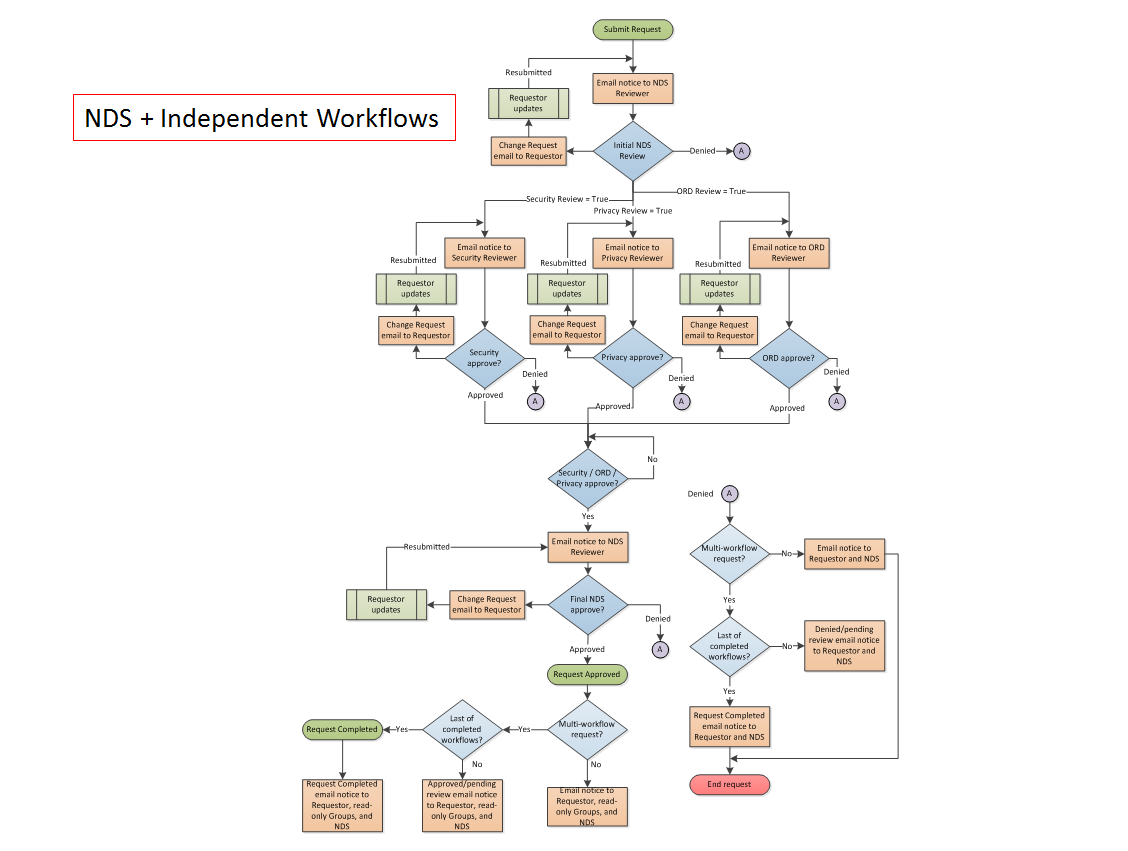


Figure : Business Process Diagram

Table : Business Process

| Business Process ID | Business Process Name | Type | Owner | Description |
| --- | --- | --- | --- | --- |
| 1 | NDS Workflow | Existing | NDS | Workflow for NDS approval |
| 2 | Independent Workflow | Modernized | Independent  Data Stewards | Workflow for Independent data stewards |

VINCI (E) DART Requirements Specification Document (RSD)

Above link will be updated once RSD is finalized (currently points to old RSD)

## Overview of the Significant Requirements

### Overview of Significant Functional Requirements

* NDS Reviewers do not want to see Independent Approver requests in their queue.
* Independent Approver Reviewers need to see all of their requests in their queue regardless of other workflows involved.
* Independent Approver Staffers need a read-only role to service requests.
* Requestors need to see all separate workflow DART requests individually in their queue.

The VINCI (E) BRD and RSD provide additional information on requirements and can be found at the DNS.

### Capacity

In accordance with VINCI standards, the DART application capacity will need to meet the criteria shown in Table 19 below per the VINCI Business Requirements Document Section 8.2.

Table : Service Level Requirements (SLR) - Capacity

| NONF ID | SLR Question | SLR Criteria | Description |
| --- | --- | --- | --- |
| 1.15.1 | How many users will be on the system hourly? | DART - 50 | NA |
| 1.15.2 | How many transactions will each average user perform each hour? | DART - 5 | Predicted number of transactions is 250 data access requests per year. |
| 1.15.3 | What are the anticipated peak user times during the day? | Business day from 8AM ET through 5PM HA | NA |
| 1.15.4 | What is the anticipated peak transaction load (when do you think that there will be the most transactions being performed on the system) during the day? | Business Day from 9AM ET through 3PM HA | NA |
| 1.15.5 | How many new users will be added in one year? | DART - 1,000 | NA |
| 1.15.6 | How many more (if any) transactions will be added in one year? | DART - 200% | NA |
| 1.15.7 | What kind of information will be stored? (Specify average of each kind per month) | DART - Data user, study, and supporting documentation | NA |
| 1.15.8 | What kind of search capacity is required? | DART - Medium | Medium (11-1000 per hour) |
| 1.15.9 | What type of system(s) is/are required? | Intranet (All VA)  Internet (public) | NA |
| 1.15.10 | Is there a need for heavy application reporting? If yes, when? | DART - Weekly | NA |

### Availability

In accordance with VINCI standards, the DART application will need to meet the criteria for availability as shown in Table 20 below per the VINCI Business Requirements Document Section 8.1.

Table : Service Level Requirements (SLR) - Availability

| **NONF ID** | **SLR Question** | **SLR Criteria** | **Description** |
| --- | --- | --- | --- |
| 1.16.1 | How much time should the system be available (and how much down time is acceptable due to incident [unexpected] outage)? | DART - 3 business days downtime | Downtime does not include planned outages for maintenance which generally occur during non-business hours |
| 1.16.2 | When should the system be available (what will be the core operating hours of the system)? | DART - During business hours | DART is generally used by users during working hours. |
| 1.16.3 | How soon should the system fully recover from an outage? (Includes Mean Time to Restore [MTRS]) | DART - 1 business day | DART, VINCI Central, and VINCI Workspace systems employ redundancy and should not experience unavailability to end user. |
| 1.16.4 | How much data will be restored when outage is recovered? | DART - Full restoration | DART has real-time duplication of database to prevent loss. |
| 1.16.5 | What time period should be considered for maintenance periods? | Late Friday through midnight Sunday Eastern Time. | NA |
| 1.16.6 | What standard time zone will the system operate in? | Eastern, Central, Mountain, Pacific, and Hawaii-Aleutian (HA) | Encompasses all VA facilities throughout the US minus territories |

# Conceptual Design

## Conceptual Application Design

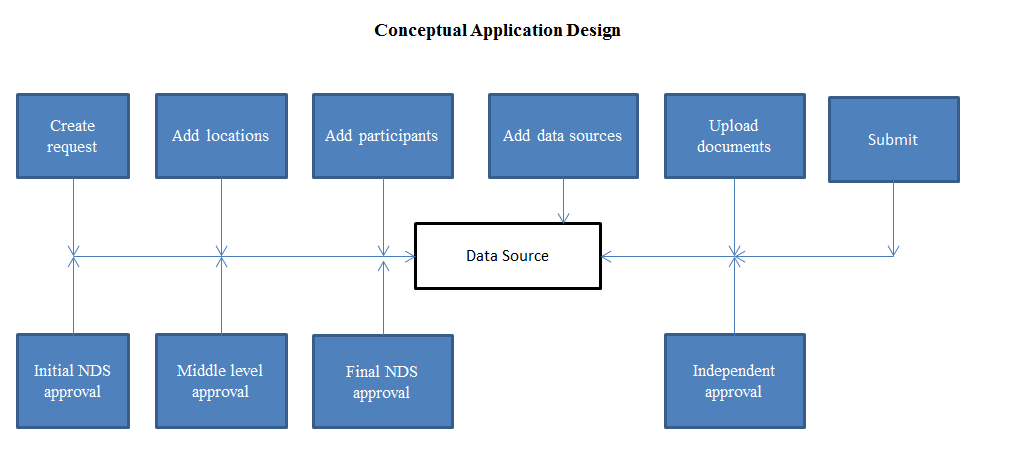


Figure : High-Level Application Design

Table : Objects in the High Level Application Design

Objects / Components to be Built or Modified

| ID | Name | Description | Service or Legacy Code |
| --- | --- | --- | --- |
| 1 | Create request | Create research request in DART application | DART Web application |
| 2 | Initial and final NDS approval | NDS admin group performs initial/final NDS approval | DART Web application |
| 3 | Middle level approval | Some admin groups perform middle level approval | DART Web application |
| 4 | Independent work flow approval | Admin group performs independent approval process without NDS approval | DART Web application |

Internal Data Stores

| ID | Name | Data Stored | Steward |
| --- | --- | --- | --- |
| 1 | Data Source | DART requests | VINCI |

.

### Application Context

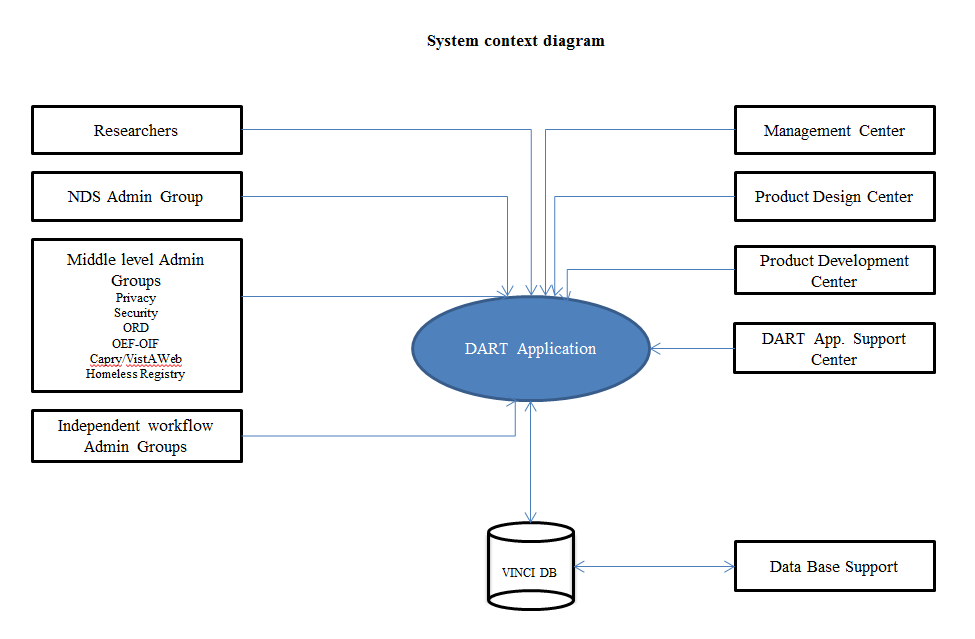


Figure : Application Context Diagram

Table : Application Context Description

| ID | Name | Description |
| --- | --- | --- |
| 1 | DART App. | Web application which provides the IRB Research data access request process used by National Data Systems (NDS). |
| 2 | Researchers | Customers who create requests in DART application |
| 3 | NDS Admin Group | National Data Systems oversees VA databases for Researchers |
| 4 | Middle level Admin Groups | Ensuring policies and procedures establishing requirements to authorize access to VA databases. |
| 5 | Independent workflow Admin Groups | Groups which provide independent, single approver workflow reviews based on data source selections |

### Application Locations

Table : Application Locations

| Application Component | Description | Location at Which Component is Run | Type |
| --- | --- | --- | --- |
| DART web application | Electronic approvals for data access | VA Windows Server | Java Web application. |
| VINCI data base | Contains all data for DART application | VA Windows Server | SQL Server data base. |

Table : Application Users

| Application Component | Location | User |
| --- | --- | --- |
| DART Application | Anywhere via the VA Intranet | Requestors, Approvers, and Read-only Administrators |

## Conceptual Data Design

### Project Conceptual Data Model

DART’s Conceptual Data Model as it currently exists should not be impacted by the current development cycle. This section will be updated if changes are identified during the agile development process. See Detailed Data Model Data Design Section in Section 5 for the realized Data model.

### Database Information

DART Database Servers

Currently, DART has four database instances (running SQL Server 2014).  These database server details are as follows:

1. Production:
   1. Server\Instance:  DNS.DNS
   2. Database:  VINCI
   3. Database:  DART\_Backup (used for DART Reporting)
2. UAT:
3. Server\Instance:  DNS.DNS \UAT
4. Database:  VINCI\_UAT
5. QA:
6. Server\Instance:  DNS.DNS \UAT
7. Database:  VINCI\_Test\_DevTeam
8. Dev:
9. Server\Instance:  DNS.DNS \Dev
10. Database:  VINCI\_Dev

The DART “Production” database contains all current production data, and is the storage location for all real DART work (i.e., creating a new DART request, modifying an existing DART request, reviewing a DART request).  This database contains DART request data and system events that are also used for DART Reporting.  The DART\_Backup database contains a mirror of the Production database, VINCI; the DART\_Backup database is refreshed nightly (SQL Server Agent Job) with the contents of the Production database.

The DART “UAT” database is typically used for demos of the DART application and User Acceptance Testing of new features before they are deployed to Production.

The DART “QA” database is typically used for QA testing of new DART features before they are ready for User Acceptance Testing.

The DART “Dev” database is typically used during the implementation and developer testing of new DART features.  DART developers share a database instance (the “Dev” database), so that new development takes place on one standardized database instance.

Table : Database Inventory

| Database Name | Description | Type | Steward |
| --- | --- | --- | --- |
| VINCI | This database contains list of tables, views and stored procedures which the DART application uses. | Modify | VINCI |

### User Interface Data Mapping

#### Application Screen Interface

##### Dashboard Page for Researchers

UI of create a new request popup window, used to create request. This version will add a second request type of “Preparatory to Research”.

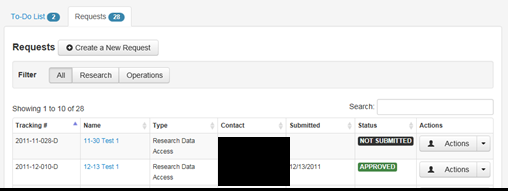


Figure : UI of Dashboard Page for Researchers

##### New Request

UI of create request popup window, used to specify the request type and official name.

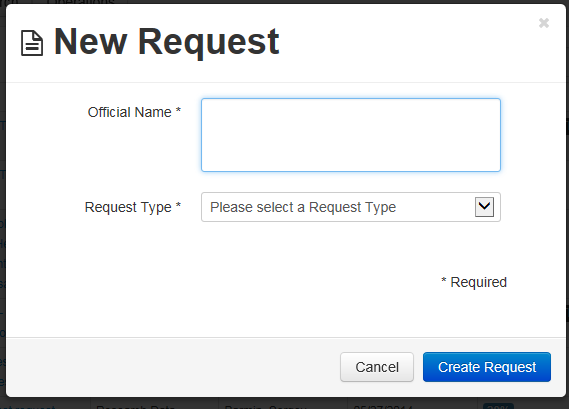


Figure : UI of "Information" tab for New Request

##### Activity Information Tab

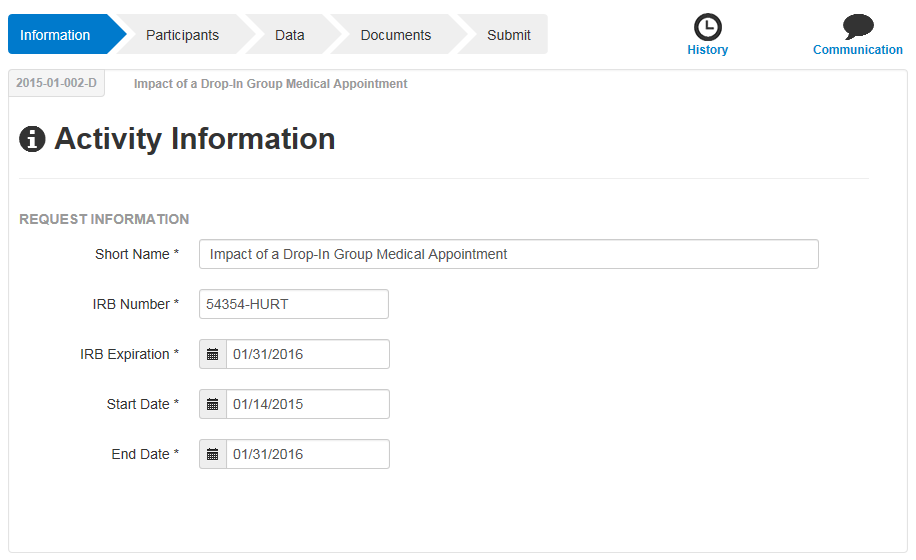


Figure : UI of Activity Information Tab

##### Participants Tab

UI of “Participants” tab to select participants.

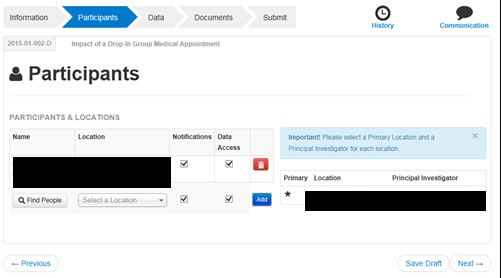


Figure : UI of “Participants” Tab

##### Data Sources Tab

UI of “Data” tab, used to select the desired Data Sources. This version in order to support “Preparatory to Research” workflow will provide a limited amount of Data Sources and remove SAS Grid from the current view.

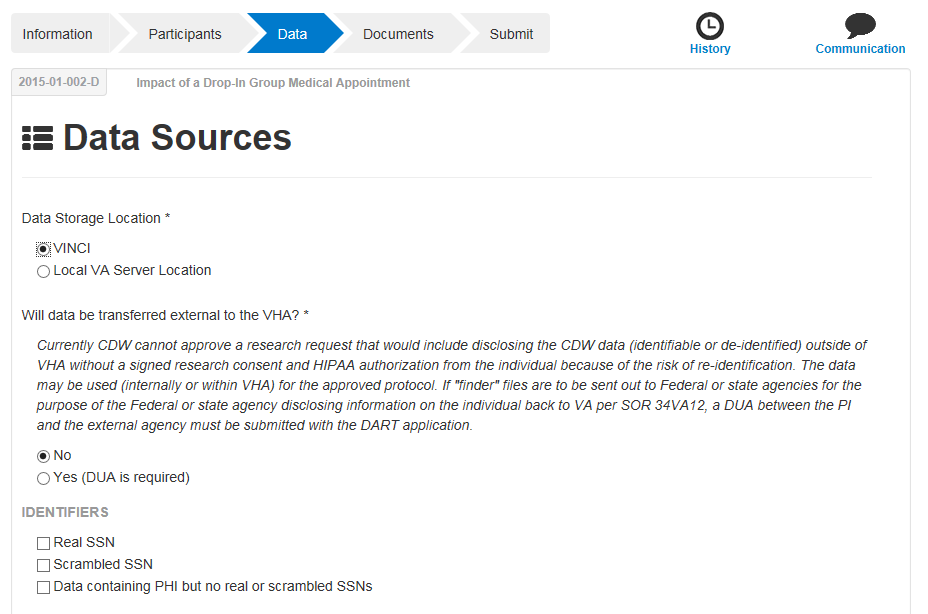


Figure : UI of “Data” tab, used to select the desired Data Sources

##### Documents Tab

UI of “Documents” tab’ used to upload and view documents that are necessary when creating a new data access request.

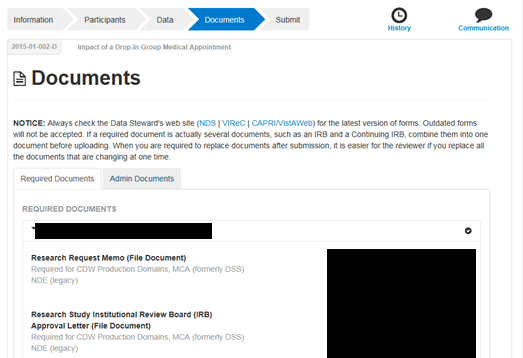


Figure : UI of “Documents” tab’ used to upload and view documents

##### Submit Tab

UI of “Submit” tab, used to submit a request for review by the appropriate Admin group(s).

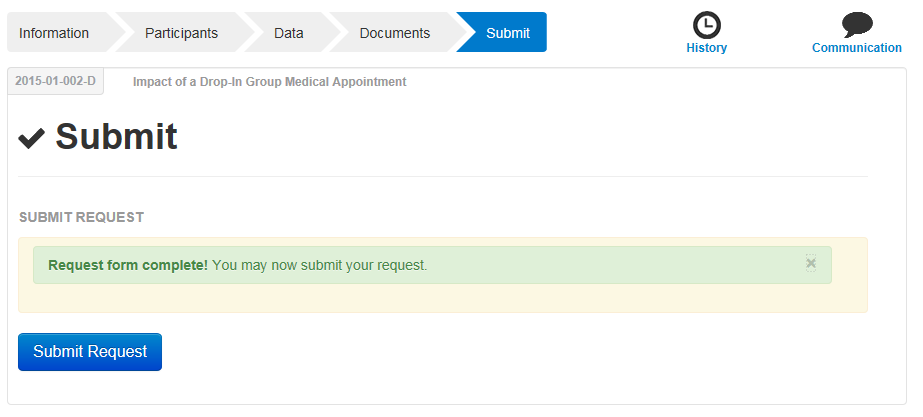


Figure : UI of “Submit” tab, used to submit a request for review by the appropriate Admin group(s)

##### Administration

To be completed in upcoming development agile sprints.

#### Application Report Interface

DART reporting functionality for Preparatory to Research and Research requests is available for Administrators by clicking the “DART Reporting” option on the right hand side of the screen when displaying the Administrators Dashboard. Currently there are seven Reports available:

* Change Request Report - Provides a list of Requests that have been in a change request status grouped by 1-90 days, 90-120 days and greater than 120 days.
* Data Request Summary Report (Data sources) – provides a list of Data Sources and totals for Requests and Participants per Data Source for a given timeframe.
* Data Request Detail Report (Data sources) - provides a list of Data Sources with a detailed list of requests and participants per data source for a given timeframe.
* Data Status Summary Report (Events) – provides a list of DART Events and totals for events by event type for a given timeframe.
* Data Status Detail Report (Events) – provides a list of DART Events with a detailed list of requests per event for a given timeframe.
* Data Source Summary Report (Approved Requests) – provides a list of DART Requests in an Approved Status and totals for Request for a given timeframe.
* Data Source Detail Report (Approved Requests) provides a list of DART Requests in an Approved Status with a detailed list of those requests for a given timeframe.

#### Unmapped Data Element

No unmapped data elements are being used for this project.

## Conceptual Infrastructure Design

Existing VINCI infrastructure will be utilized in this project.

### System Criticality and High Availability

DART application is available for customers 24/7. Any time that the system is not accessible (downtime) can be due to the failure of hardware, software, the network, or any other factor (such as loss of power) that causes the system to be down. Mean Time To Repair (MTTR) – System down time shall be a maximum of 24 hours depending on the nature of the failure. For example, if the servers that are allocated to go down (for whatever reason) it is likely they could be brought back on line in a 24 hour period because the problem would be relegated to the servers themselves and not some kind of environmental catastrophe. If the failure is due to a more widespread problem, such as a fabric failure at AITC - Austin, the down time could be greater than 24 hours.

Accuracy – The system shall return entries exactly as they were passed with no modification.

Defect repair – It is expected that all defects related to the implementation get resolved during initial development and subsequent UAT testing.

Additional information available in the System Security Plan (SSP) which is only accessible by authorized users.

### Special Technology

No special technology used for this project.

### Technology Locations

The physical location of the VINCI assets is:

AITC – VA Departmental Offices

1615 Woodward Street

Austin, TX 78772-7830

Table : Technology Locations

| Technology Component  Production 1 | Location | Usage |
| --- | --- | --- |
| Physical Workstations | Nationwide |  |
| Virtual Workstations | AITC |  |
| Interface Processors | AITC |  |
| Mainframe | AITC |  |
| Application Server | AITC |  |
| Databases | AITC |  |

### Conceptual Infrastructure Diagram

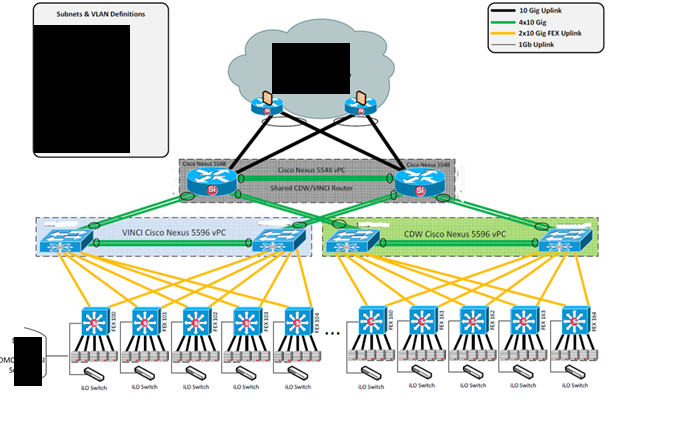


Figure : Conceptual Infrastructure Diagram

#### Location of Environments and External Interfaces

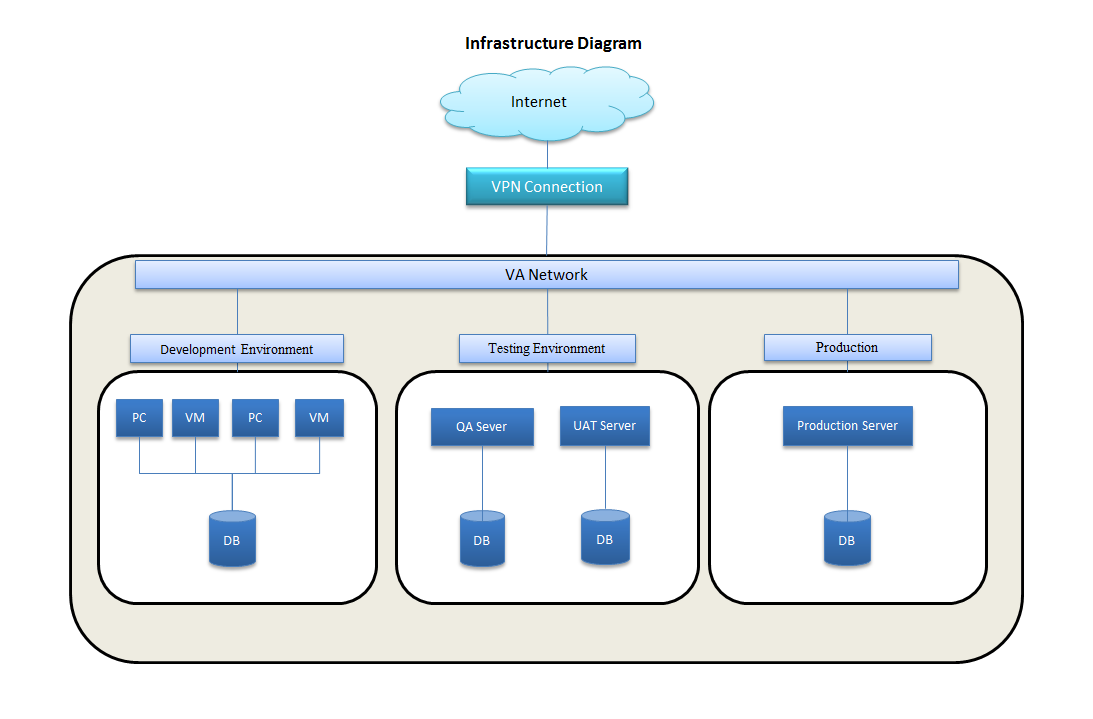


Figure : VINCI DART Infrastructure Diagram

#### Conceptual Production String Diagram

Conceptual Production String Diagram not applicable for this project.

# System Architecture

## Hardware Architecture

The DART application is deployed on purchased components that make up the VINCI AITC infrastructure. The following hardware is in the process of being purchased in parallel to this project:

* HP 16x BL460c Gen9 around (2x 2.5ghz 12 cores; 512gb of RAM; 2x 240gb SSD; 2x IB/Ethernet cards add-on; 2x C7000 Chassis with IB add-on)
* FortiGate 1500D Firewall
* DDN SSDs for 3 spare pool drives and to add another storage pool of 10 SSD drives
* DL560 Gen8 Server (HP DL560 Gen8 CTO Server (HP 16GB 2Rx4 PC3-14900R-13 Kit; HP 300GB 6G SAS 15K 2.5in SC ENT HDD; HP 800GB 12G SAS HE 2.5in EP SC SSD; HP 10GbE 2P 560FLR-SFP+ Adptr FIO Kit; HP 2GB FBWC for P-Series Smart Array; HP Ethernet 10Gb 2P 560SFP+ Adptr; HP TPM Module Kit; HP 81E 8Gb SP PCI-e FC HBA; HP BLc 10Gb SR SFP+ Opt)
* 3PAR 4-node 7400c with AO x (Tier-0 SSD with tier-2 NL drives; 12K20D-16FC with 5x SS8460 enclosures, 20x HMSHMS-3M cables, power and ICL cables, rail kits, licenses for SFA OS, dual UPSs and ATSs; 3TB 7,200 RPM 6Gb/s SAS; 800GB Read Intensive SSD 6Gb/s SAS drive module (read/write ratio 90/10), for less than 3 full writes /day)

### DART Application Servers

Currently, DART has four application servers that provide the DART client application for the end user via the web.  These application server and client details are as follows:

1. Production
2. UAT
   1. https://DNS .DNS :PORT/vinci\_dart\_client/dart9/index.html
3. QA
   1. http:// DNS .DNS :PORT/vinci\_dart\_client
4. DEV
   1. http:// DNS .DNS :PORT/vinci\_dart\_client

## Software Architecture

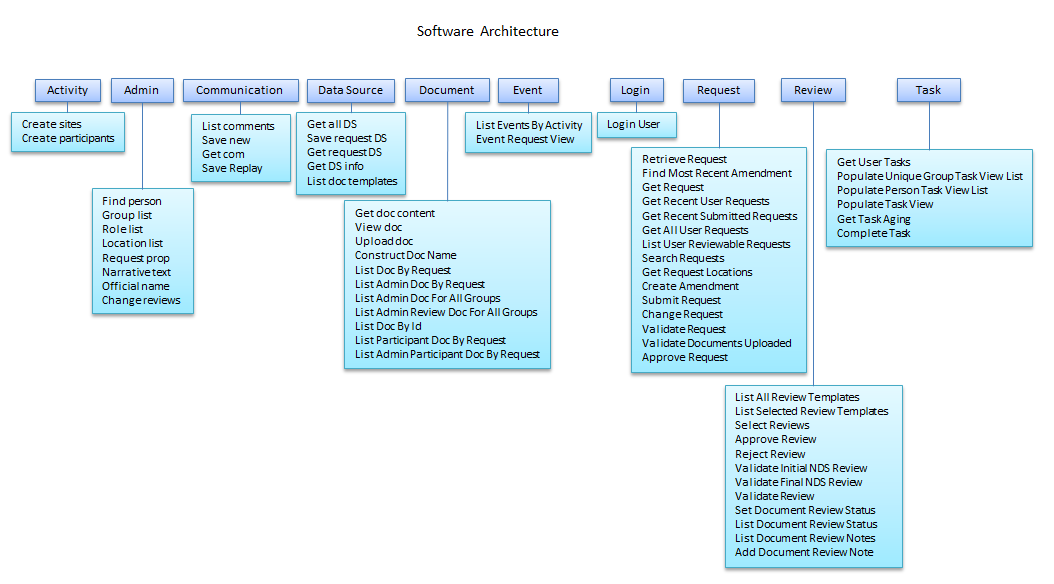


Figure : Software Architecture

### Process View

DART operates as a single process within the confines of the environment provided by the web server context. Additionally the Spring Framework provides sessioning and the controllers that are part of Spring MVC provide specific access points into DART which are both serializable and thread safe.

All threads within DART are Java standard threads provided by the JDK 1.7.0\_13. Both Spring and Hibernate provide thread pool mechanisms to manage and maintain these Java threads.

There is no inter-thread communication between these standard Java threads. Anywhere that thread operations could interfere with each other, appropriate synchronization mechanisms are used.

### Deployment View

DART is deployed to a single node or machine within a web server containing a JVM. In addition, the SQL Server database can be deployed either locally or remotely on a different machine; the database connection is configured using standard JDBC and jTDS configuration settings.

There are no specialized networking equipment or setup requirements.

The configuration and hardware requirements for the DART node is 1 virtual CPU, a minimum of 4GB of RAM, and a minimum of 60 GB of disk space. If the database is deployed locally and contains only the DART database, these requirements are sufficient. Deployed remotely, the database node’s size would start as an equivalent VM/node.

Due to DART being a web-based application, this deployment needs to be performed only once in production. DART does not need to be installed on each individual end-user’s computer. The DART server needs to be installed just once inside the VA network and accessible to these users’ computers via a standard web browser.

### Implementation View

This figure shows the overall structure of the DART system including the primary layers.

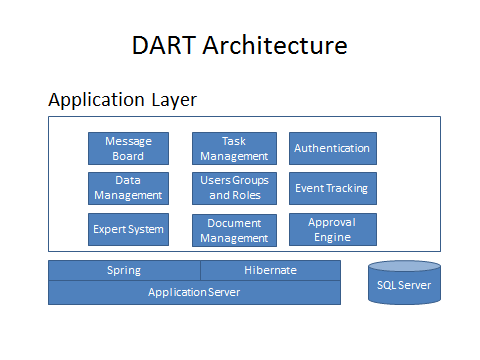


Figure : DART Architecture (Application Layer)

The application layer contains the major functional modules of DART. As is typical the application layer rides on top of supporting layers – in this case, the Spring Framework, Hibernate, and Apache Tomcat (Application Server) and accesses the SQL Server 2014 database via standard JDBC mechanisms.

### Modules

* The DART system is made up of a single WAR file that contains: JARs, HTML, CSS, JavaScript, and other web application dependencies and configuration files. The WAR file contains everything necessary to run the DART web application.
* The source code for the system is Java source arranged in proper Java class files. Where the name of the source file matches the name of the primary class contained within that file. The Java source code follows standard naming conventions and is organized by layers and responsibilities, including: models, controllers, views, business objects, data access objects, etc.

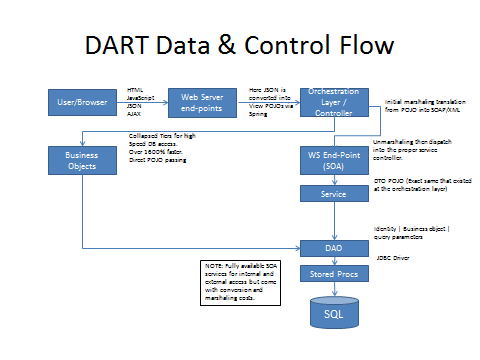


Figure : DART Data & Control Flow

## Network Architecture

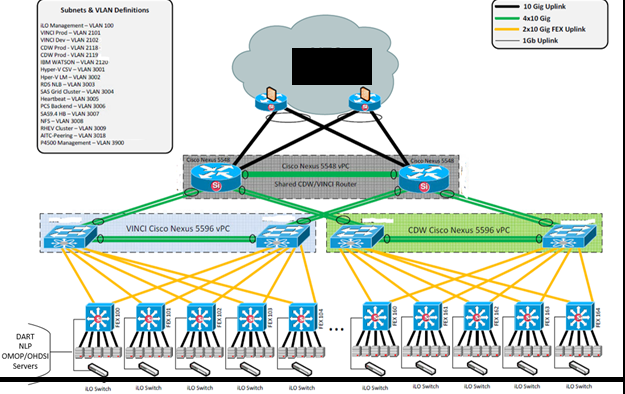


Figure : Conceptual Infrastructure Diagram

## Service Oriented Architecture / ESS

The application does not expose web services. The application contains business services that are embedded within the application.

## Enterprise Architecture

The adherence to the TRM and Standards Profile (SP) will be followed and described in more details when all the technology components for this project are selected. DART will promote interoperability, portability, adaptability within systems, quality assurance and will utilize current technology to provide a framework for IT application and infrastructure development.

| **Technology** | **Version** | **TRM** |
| --- | --- | --- |
| **Infrastructure** |  |  |
| Apache Tomcat | 8.x | Approved w/ Constraints |
| Microsoft SQL Server | 2014 | WAIVER IN PLACE for CDW/VINCI |
| Oracle Java | 1.8.x | Approved w/ Constraints |

DART has been submitted to the TRM, request 14608.

# Data Design

The solution will store both structured and unstructured data. Unstructured data (MS Word documents, PowerPoint presentations, PDFs) will be assigned metadata to facilitate searching and retrieval operations. Structured data captured in the DART application and web forms will be stored in MS SQL Server with associated metadata for retrieval and analytical reporting. This section outlines the design of the database management system (DBMS).

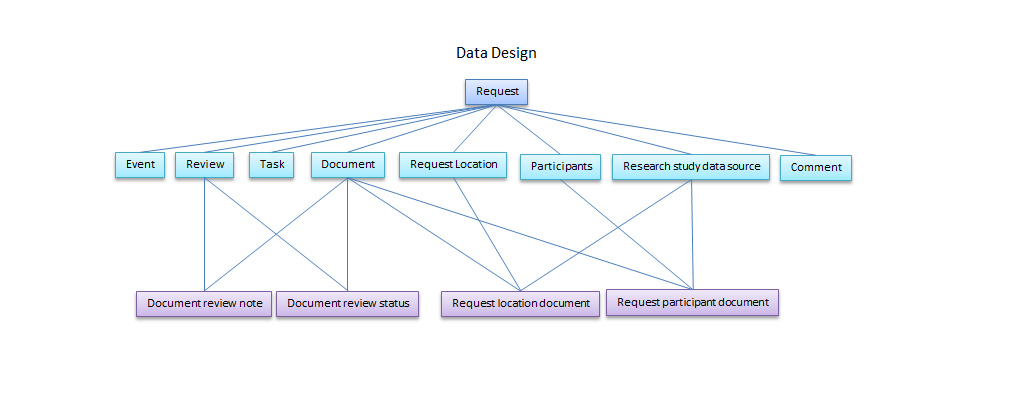


Figure : Data Design Diagram

## DBMS Files

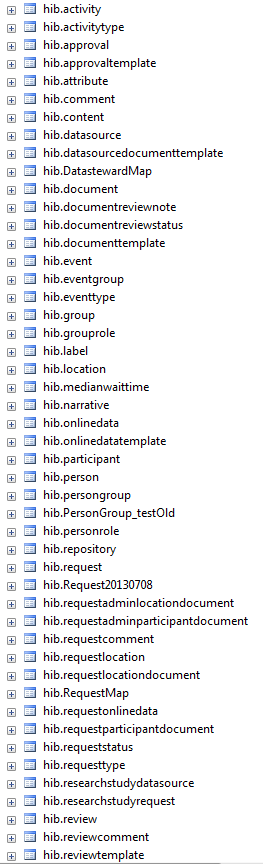


Figure : DBMS File Structure

## Non-DBMS Files

Non-DBMS Files is not applicable for this project.

## Data View

# Detailed Design

## Hardware Detailed Design

VINCI program level hardware will be utilized. DART as an application already has a Production and Pre-Production environment that will be utilized. No additional hardware is expected for the DART development effort.

## Software Detailed Design

DART’s software enhancement design will occur during the development sprints. This document will be updated once the designs are complete and prior to MS 2.

### Conceptual Design

#### Product Perspective

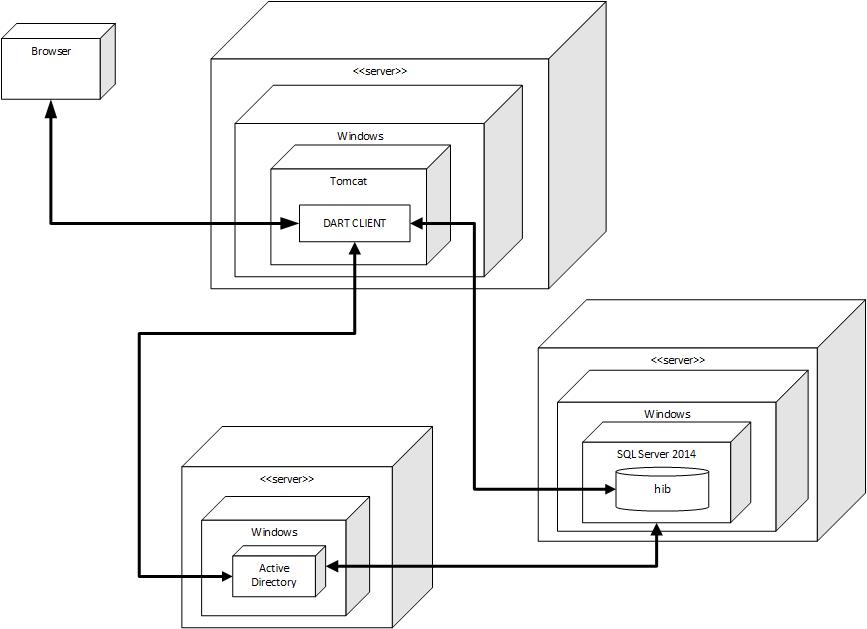


Figure : DART Component Diagram

#### User Interfaces

N/A

#### Hardware Interfaces

DART will leverage AITC resources via a virtual machine environment within the VINCI segment. Please see Hardware Architecture

#### Software Interfaces

The following describes the software interfaces that DART will utilize in delivery of the system architecture. Several of these software interfaces will be provided as either COTS products or open source technical solutions.

Apache Tomcat

Apache Tomcat is an open-source web application server developed by the Apache Software Foundation (ASF). Tomcat implements several Java Enterprise Edition (JEE) specifications including Java Servlet, JavaServer Pages (JSP), Java EL, and WebSocket, and provides a HTTP web server environment for Java code to run in. Tomcat is considered a lightweight container and is normally leveraged when a fully compliant JEE application server is not required. Tomcat is TRM approved.

Microsoft SQL Server 2014

Microsoft SQL Server is database server and will host the physical data required and generated by DART. This database provides a solution for horizontally scaling the physical data storage mechanism to meet increased demands on systems and is approved via waiver on the TRM

Active Directory

Active Directory is a component of the Windows Server line of operating systems that provides a set of processes and services for management of users, roles, and groups for access control and directory management.

#### Communications Interfaces

See Figure 19: DART Component Diagram

#### Memory Constraints

Memory constraints for DART shall reflect memory constraints of the software and applications installed on those systems. These constraints will be determined based on the requirements of the software being hosted by the systems and the deployment configuration of the applications being hosted on the systems.

#### Special Operations

N/A

#### Product Features

See Figure 3: Application Context Diagram and Figure 2: High-Level Application Design in section 3 above.

#### User Characteristics

The characteristics of the end-user community are office professionals familiar with logging into secured workstations and online web sites, operating the Microsoft office product suite, e-mail systems, Voice over Internet Protocol (VOIP) telephone systems, document management systems such as Microsoft SharePoint, and searching for and collecting online information through internal and external websites using web browsers such as Microsoft’s Internet Explorer, Mozilla’s Firefox, or Google Chrome.

The characteristics of the technical community are system administrators and developers familiar with remote administration of cloud and/or virtualized systems, implementing security measures in accordance with VA regulations, 24x7 system monitoring, system backups, familiarity with administering Windows servers, Microsoft Dynamics CRM, SQL Server, SQL Server Reporting Services, integration with Active Directory Federated Services and IP based phone systems, and supporting web services.

#### Dependencies and Constraints

Access to Active Directory. System Administrators. Software security patches. Hardware replacements. Regulatory issues and VA policies. Software Licensing.

### Specific Requirements

#### Database Repository

See existing Data design in section 5 above.

#### System Features

The DART Application has several user interfaces. The specific user interface the user will be exposed to is based upon their role in the system and the path they followed to arrive within this application. The six main areas of functionality include:

* Request Representative DART functionality
* DART Administrator functionality
* NDS Administrator functionality
* Privacy Administrator DART functionality.
* Security Administrator DART functionality.
* ORD Administrator DART functionality.
* Independent workflow Administrator DART functionality.

See existing UI design in User Interface Data Mapping above.

##### Design Element Tables

Storage requirements are based on historical usage and incoming data request to determine our future growth. Database Administrators (DBA) carefully monitor usage and provide future growth projections.

##### Routines (Entry Points)

There is only one html page for customers to start a new request or continue to work with an existing request, as well as for administrators to process an existing request during the approval process.

The page http|https:<application server host>:<application server port>/vinci\_dart\_client/dart9/index.html is the main entry point for the DART application which is dependent on the environment on which it is being accessed please see the DART Application Servers section below for more details. The same page can be used to start a new amendment for completed request.

##### Templates

N/A

##### Bulletins

N/A

##### Data Entries Affected by the Design

N/A

##### Unique Record(s)

N/A

##### File or Global Size Changes

N/A

##### Mail Groups

N/A

##### Security Keys

N/A

##### Options

N/A

##### Protocols

The following protocols are used: HTTPS and Java Database Connectivity (JDBC – TCP/IP)

##### Remote Procedure Call (RPC)

N/A

##### Constants Defined in Interface

N/A

##### Variables Defined in Interface

N/A

##### Types Defined in Interface

N/A

##### GUI

N/A

##### GUI Classes

N/A

##### Current Form

N/A

##### Modified Form

N/A

##### Components on Form

N/A

##### Events

N/A

##### Methods

N/A

##### Special References

N/A

##### Class Events

N/A

##### Class Methods

N/A

##### Class Properties

N/A

##### Uses Clause

N/A

##### Forms

N/A

##### Functions

N/A

##### Dialog

N/A

##### Help Frame

N/A

##### HL7 Application Parameter

N/A

##### HL7 Logical Link

N/A

##### COTS Interface

N/A

## Network Detailed Design

VINCI through the use of Virtual Machines has a process in place to allocate workspaces for its user community. Access to the VA network through hardline connections or VPN is required. Dart as a product is already has a Production and Pre-Production environment that will be utilized. No additional network changes are expected for the DART during this development effort.

## Security and Privacy

### Security

DART will be dependent upon services available from the IAM group at the time of implementation, with focus on Active Directory Federated Services. DART that will adhere to all VA security requirements in accordance to Federal Information Processing Standard (FIPS) 199 and National Institute of Standards and Technology (NIST) SP 800-60, recommended Security Categorization.

The Security Categorization will drive the initial set of minimal security controls required for the information system. Minimum security control requirements are addressed in NIST SP 800-53 Revision 4 and VA Handbook 6500 Revision 4, Appendix D.

Smart-card enabled login can be accomplished using either (1) the existing Spring Security framework currently utilized by the DART application, or (2) integration with IAM SSOi. Further analysis to be performed by the development team. A working prototype exists, but has not been deployed into Production environment.

Meeting held with IAM and selection of SSOi was made.

A new Policy was implemented on August 31, 2015 which dictates that all systems will require PIV access. This will not be addressed in the current development activity but will be planned for DART Enhancement Phase II, tentatively scheduled to begin in March 1016.

### Privacy

Protecting the privacy of data that DART will be managing whether it is transactional, unstructured, or meta-data is of utmost importance to VINCI system design and functionality, and there are both privacy and data security constraints that must be followed. DART utilizes role-based and group-based authorization in addition to VA Active Directory authentication, to determine what data is available to a specific DART user. Each researcher’s queue contains only the requests that were created by that researcher, and those requests that include the researcher as a participant. Each reviewer’s queue contains only those requests that have been sent for review by their Admin group. Review actions, such as approving or denying a request, can only be performed by DART users who have successfully authenticated their VA Active Directory credentials and belong to the appropriate Admin group.

DART will adhere to all proposed VA Privacy, Identity Management and Security requirements including VA Handbook 6500 Revision 4 and NIST SP 800-53 Revision 4 Privacy Controls. Efforts that involve the collection and maintenance of individually identifiable information must be covered by a Privacy Act system of records notice.

## Service Oriented Architecture / ESS Detailed Design

The application does not expose web services. The application contains business services that are embedded within the application.

# External System Interface Design

This application does not expose any external interfaces. The application only provides a web application interface to an end user available via a web browser

## Interface Architecture

N/A

## Interface Detailed Design

N/A

# Human-Machine Interface

The DART User interface (UI) enables researchers and VA administrative staff a work flow process to manage and maintain requests from researches to access VA maintained data. Dart does not control the access to the data just the management of the process to gain access, which is a manual process outside the scope of DART.

## Interface Design Rules

The user interface (UI) design of DART screens follows other VA web-based interfaces. All web pages in the DART application must be 508 compliant.

The following sections list the guidelines for user interface design.

### Visibility of System Status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

### Match between System and the Real World

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

### User Control and Freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

### Consistency and Standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

### Error Prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

### Recognition Rather Than Recall

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

### Flexibility and Efficiency of Use

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

## Inputs

Users interact with the system through a web browser. They enter data from the keyboard and submit it via HTML forms to customize drug interactions and associated information.

Access to the system is strictly controlled by role-based access. Security is handled through the use of VA Active Directory (LDAP).

## Outputs

DART is an application that provides management of researcher requests. There are no formal/automated outputs of the application.

## Navigation Hierarchy



# Attachment A – Approval Signatures

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

The Business Sponsor and Project Manager are required to sign.

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

Signed: PII Date:

< Business Sponsor >

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

Signed: PII Date:

< Project Manager >

1. Additional Information
   1. Identification of Technology and Standards

VA Informatics and Computing Infrastructure (VINCI)

DART - Data Access Request Tracker

* 1. Constraining Policies, Directives and Procedures

All software development by VINCI developers shall conform to technology standards as defined in the One-VA Technical Reference Model (TRM). As well as meet all VA/VHA security policies, regulations, and 508 compliance.

User access authentication will be accomplished through integration with the VA Active Directory. This shall ensure that only authorized users may access the applications and data.

The software architecture will need to be analyzed and modified for optimal performance on VINCI infrastructure.

Agile development methodologies will be utilized for development.

[VA Releases Open Source Policy Memorandum](http://www.openhealthnews.com/hotnews/va-releases-open-source-policy-memorandum)

* 1. Requirements Traceability Matrix

The RTM will be available at the DNS.

* 1. Packaging and Installation

This application is built using Maven which creates a web application archive (WAR) that will be distributed to a server that hosts a Java application server, namely Tomcat.

* 1. Design Metric