Effective Health Care Program
Research Reports

Number 26

Proof-of-Principle Evaluation of a Distributed Research Network

Jeffrey Brown, Ph.D.
John Holmes, Ph.D.
Beth Syat, M.P.H.
Kimberly Lane, M.P.H.
Ross Lazarus, M.B.B.S., M.P.H.
Richard Platt, M.D., M.S.

Research from the Developing Evidence to Inform Decisions about Effectiveness (DEcIDE) Network

AHRQ
Agency for Healthcare Research and Quality
Advancing Excellence in Health Care • www.ahrq.gov

June 2010

The DEcIDE (Developing Evidence to Inform Decisions about Effectiveness) network is part of AHRQ's Effective Health Care program. It is a collaborative network of research centers that support the rapid development of new scientific information and analytic tools. The DEcIDE network assists health care providers, patients, and policymakers seeking unbiased information about the outcomes, clinical effectiveness, safety, and appropriateness of health care items and services, particularly prescription medications and medical devices.

This report is based on research conducted by the DEcIDE (Developing Evidence to Inform Decisions about Effectiveness) Centers at the HMO Research Network and the University of Pennsylvania under contract to the Agency for Healthcare Research and Quality (AHRQ), Rockville, MD (Contract No. HHSA29020050033I TO5). The AHRQ Task Order Officer for this project was Scott R. Smith, Ph.D.

Suggested citation:

Brown J, Holmes J, Syat B, et al. Proof-of-principle evaluation of a distributed research network. Effective Health Care Research Report No. 26. (Prepared by the DEcIDE Centers at the HMO Research Network and the University of Pennsylvania Under Contract No. HHSA29020050033I T05.) Rockville, MD: Agency for Healthcare Research and Quality. June 2010. Available at: http://effectivehealthcare.ahrq.gov/reports/final.cfm.

Contents

Author affiliations:

Jeffrey Brown, Ph.D.[a]
John Holmes, Ph.D.[b]
Beth Syat, M.P.H.[a]
Kimberly Lane, M.P.H.[a]
Ross Lazarus, M.B.B.S., M.P.H.[a]
Richard Platt, M.D., M.S.[a]

[a]Department of Ambulatory Care and Prevention, Harvard Medical School and Harvard Pilgrim Health Care
[b]Center for Clinical Epidemiology and Biostatistics, University of Pennsylvania

Abstract

This report describes the implementation and evaluation of a proof-of-principle demonstration of selected functions of a distributed research network designed to support research on the comparative effectiveness and safety of marketed therapeutic agents. Key specifications of the demonstration included: (1) a distributed architecture; (2) strong local control of data uses; and (3) federated querying.

The demonstration was designed to illustrate (1) functions of a central portal, (2) menu-driven distributed querying, and (3) secure distribution and remote execution of analytic computer programs (SAS code) and aggregation of the results.

Development of the user portal and menu-driven query relied on a rapid prototyping approach. It illustrated real-time federated querying using two identical, synthetic databases stored on physically remote servers. This work demonstrated the possibilities of a distributed research network, the potential of a strong central portal design, and some of the development challenges and successful approaches in building a network. Many of the features included in the demonstration are directly applicable to the development of a permanent network.

For distribution and execution of analytic programs, we partnered with the grid research team of the Centers for Disease Control and Prevention's (CDC) National Center for Public Health Informatics (NCPHI). The demonstration illustrated the secure distribution of a SAS program to two network nodes. It also drew attention to the complex demands of conforming to each data owner's security and other requirements, particularly with regard to their allowing incoming queries.

Lessons learned from this proof-of-principle demonstration include the likely success of an incremental approach to software development and network implementation, building on the activities described in this report. Additionally, our experience illustrates the value of developing a system that allows data holders to poll a central site for requests (a "publish and subscribe" model), rather than requiring data holders to accept even a small security risk associated with allowing incoming queries through their system firewalls. As expected, the additional functionality needed includes strong security, autonomy, and auditing features.

1. Introduction

1.1. Objectives and Goals of Distributed Research Network Project

The overall objective of this project is to design a scalable, distributed health information network architecture that will support secure data analyses on the risks and benefits of therapeutics. The two key network design products are:

- Specifications of network architecture and research network cooperative (Report #1). This product has been completed and included the technical design, key infrastructure components, and organizational structure required for a network to support large-scale, population-based studies on the risks and benefits of therapeutics.
- Proof-of-principle demonstration and evaluation (Report #2—this report). This product includes a proof-of-principle implementation of a network prototype using some of the design features described in Report #1.

The system architecture complies with all privacy, security, and legal requirements, including current state and federal laws.

1.2. Purpose and Outline of This Report

This report, the second of four for this project, describes the implementation of a network proof-of-principle to demonstrate selected functions of a distributed research network. Further, it evaluates two proof-of-principle demonstrations and characterizes the needs, challenges, and barriers to creation of a distributed research network.

Section 2 of this report delineates the proof-of-principle specifications and evaluation goals for the demonstrations. Section 3 outlines the implementation of the two proof-of-principle demonstrations. Section 4 describes an evaluation of the proof-of-principle demonstrations and describes the key implementation challenges faced during development of the demonstrations. Finally, Section 5 details an evaluation of another proof-of-principle activity that occurred in parallel with implementation of the proof-of-principle demonstrations.

2. Proof-of-Principle Specifications and Evaluation Goals

The specifications for the demonstrations were developed with substantial input from the Agency for Health Care Research and Quality (AHRQ). The design focused on several key elements of the proposed network and did not attempt to illustrate design features (e.g., redundant security, fine-grained permissions, secure messaging, and authentication) that are well-established within existing software and information technology systems.

2.1. Summary and Key Specifications of the Proof-of-Principle Demonstrations

The key features included:

- **Distributed architecture.** Data remain under control of the data holder. Analysis code is distributed to the data holder for execution and results returned only with the data holder's approval.
- **Strong local control of data uses.** Data holders must be able to control access to, and uses of, the data they hold and to have access to audit trails/logs of all data uses.
- **Federated querying.** A single *network portal* is used to develop and distribute queries, aggregate and distribute results, and maintain centralized logs of network usage.

Two distinct demonstrations were designed based on these general specifications. One was designed to illustrate menu-driven distributed querying and select central portal functions. The other to illustrate the secure distribution and remote execution of analytic computer programs (SAS code) and return and aggregation of the results; this functionality represents a novel application of federated architecture within this environment.

2.2. Specifications

2.2.1. Menu-Driven Query and User Portal: Overall Requirements and Assumptions

The menu-driven query and portal design focused on the following functions:

- Assembly of a simple query using a menu-driven user interface
- Issuance of the query
- Distribution of the query to multiple network servers
- Execution of the query locally against a test dataset
- Aggregation of the results and presentation as a single results set

2.2.2. Secure Distribution and Execution of SAS Code: Overall Requirements and Assumptions

The objective of the second demonstration was to allow an authorized user to distribute a SAS program to data holders and to have the system return results to the user. Specifically, the use-case included the following steps:

- An authorized user authenticates to a central portal
- A SAS program is distributed to each data holder (node); the data owner allows or denies the request for the program to run
- The SAS program is executed at each node, and a standard results set is returned
- The results are aggregated and made available to the authorized user
- A log of site activity for each node is generated

The SAS programs were limited to simple frequencies with structured results sets. This approach was selected to maintain focus on the key design features of a distributed research network.

3. Proof-of-Principle Implementation

3.1. Introduction

This section describes the implementation of the two demonstrations specified in Section 2.

3.2. Menu-Driven Query and User Portal

This demonstration was developed in collaboration with Lincoln Peak Partners, a privately-held information technology services company located in Westborough, MA.

The high-level network architecture for this demonstration is illustrated in Figure 1. The architecture shows a central Hub (or portal) and the proposed functionality on the left side and a data holder "data mart" on the right side. Brief descriptions of the portal features are in Table 1.

Development of the menu-driven query and user portal relied on a rapid prototyping approach. The first phase included development of a high-level architecture (Figure 1) and a series of "wire-frame" specifications for the central portal. The wire-frames included rough sketches of each page of the portal and the functionality specified for the page. After completion of the wire-frames, a preliminary version of the portal was implemented for review and comment, followed by updates and additional rounds of review and revision.

This demonstration illustrated real-time federated querying using two synthetic databases stored on physically remote servers. The demonstration illustrated the following network functions:

- User log-in (username and password)
- Menu-driven query formulation
- Query distribution
- Query monitoring, return, and aggregation
- Query results page
- Data owner auditing and monitoring on access to the databases
- Implementation of select policy rules
    - User-based permissions
    - Query formation rules (limited to fewer than 10 unique query items, such as drugs, drug classes, and diagnoses)
    - Query results viewing rules (requirement of two responses before a user could view)

The implementation was built using the Microsoft .Net platform using the C# programming language. The web application components utilize ASP.NET, with service components arranged in a service-oriented architecture (SOA) utilizing the SOAP protocol for communications between components. The Hub database is built on SQL

Server 2005; Data Mart components use ADO.NET and ODBC for connectivity to remote databases.

A recording of the demonstration is available at https://btconferencing.webex.com/btconferencing/playback.php?FileName=www.btconferencing.com/webex/K0107189.wrf.

3.3. Secure Distribution and Execution of SAS Code

This demonstration was performed in partnership with the grid research team of the Centers for Disease Control and Prevention's (CDC) National Center for Public Health Informatics (NCPHI). NCPHI leads a broad initiative to create the Public Health Grid (PHGrid), which provides distributed computing capabilities to support a wide array of public health needs http://sites.google.com/site/phgrid/.

The demonstration was designed to transfer a (SAS) program to each data holder's site, execute the program on an existing dataset stored at the sites, and send the results securely to a central site for aggregation and viewing. The demonstration used SAS datasets supplied by the project team; the datasets contained no protected health information.

3.3.1. Installation and Implementation of Grid Nodes

Based on input from NCPHI, and in consultation with the Informatics team, the implementation was built using the Globus Toolkit,® available from the Globus Consortium (http://globus.org), a set of software tools that are the foundation for NCPHI's PHGrid initiative. Globus is a collection of infrastructure components providing authentication and access control, remote job execution, secure file transfer, and other basic communication services. Layered on these basic services are specialized applications that use these infrastructure components to provide higher level applications. Specifically, a combination of the Globus Toolkit® PHGrid Node VMWare software version 0.2 (2008.12.22), Secure Simple Transfer Service, SAS v9.1, and Unix shell scripting was used in this implementation. Details can be found on the DRN wiki: http://sites.google.com/site/phgrid/Distributed-Research-Network.

Five demonstration sites initiated the local processes and approval procedures needed to permit the demonstration, including installation of the Globus grid node using the Virtual Data Toolkit (VDT). Four of the five partner health plans that attempted to gain the necessary approvals for the demonstration were not able to do so within the time allotted; therefore, the final network configuration used for the demonstration included two sites: one of our originally intended partners and a development node staged within the NCPHI development lab. A separate server acted as the central node (workstation) for the demonstration. The reasons why the other health plan partners were not able to participate in the demonstration are described below.

Architecture

The PHGrid components included: (1) the PHGrid node appliance, which allows security to be configured individually at each site; (2) secure simple transfer service, which is a Java-based grid web service that is hosted within the PHGrid node and uses SSL/TLS encryption; and (3) automation programs written as Linux, Windows, and Perl scripts. The architecture of the demonstration is illustrated in Figure 2.

A synthetic dataset was provided to sites, to be accessed via the Globus node.

The DRN nodes existed in a virtual demilitarized zone ("DMZ"), protected from unauthorized traffic from the internal network as well as from the Internet. Query results were passed from the DMZ into the central NCPHI site through an external firewall at each site. The sites were responsible for opening ports 139 and 445 to allow files to pass through the internal firewall to the DRN node. A NCPHI representative worked directly with the IT contacts at the demonstration sites to address security concerns and allow access for port 8443 from the central NCPHI site.

Local Constraints and Security Requirements

Although this limited demonstration did not involve identifiable patient data, the sites insisted on conforming to their established local policies and procedures regarding security and data access. The local policy reviews identified several areas of concern for the health plans' data privacy and security teams, mainly related to system security and a reluctance to "open" ports for the demonstration. This was the case despite creation and deployment of a secure messaging service that required only a standard secure web browser port (443 used for the secure HTTPS internet protocol) to be opened at the host firewall for the specific external machine used as the query source and result destination. Permitting only access from one specific remote SSL secured machine is generally regarded as a highly secure approach, particularly to a server on a special isolated network (i.e., DMZ) inside the institutional firewall.

Although each of the health plans raised various questions regarding the data privacy, security, and required infrastructure (e.g., software and hardware requirements), none refused to consider installation or raised roadblocks that could not be addressed through more detailed discussions and perhaps enhancements to the proposed network architecture. In the end, the lengthy and complex process needed to address the health plans' concerns and obtain all of the necessary approvals proved insurmountable within our timeframe for all but one of our health plan partners.

QueryInterface

A simple command-line interface was used. This allowed the user to submit a query by referencing a file that contains the query code. A graphical user interface was designed and developed to augment the command-line interface; however, development delays did not permit implementation of the interface in time for the demonstration.

Implementing the Query

Both participating nodes received the query sent from the NCPHI client workstation, manually executed a series of steps to allow the query to run against the SAS datasets, and returned the results to the central workstation for aggregation.

3.4. Summary

The demonstrations achieved their main objectives to: (1) demonstrate a menu-driven query interface for distributed querying and the functionality of a central portal; and (2) create a secure network to transmit executable SAS programs to remote nodes and return the results for aggregation and viewing.

Partners indicated concern with data autonomy and security, and they expressed interest in fine-grained permissions, security, and authorization and strong authentication. The health plan partners expressed a desire to review all network requests as they arrive and before the results leave their organizations. These partners also felt that detailed auditing and active monitoring of network use would be valuable. The health plans differed regarding their internal policies and procedures for evaluating, approving, and implementing the proposed system, and there were some differences in the availability of the necessary IT expertise to implement and manage the required system architecture. Additional details of the health plans' reactions to the secure distributed querying demonstration are provided in Section 4.

4. Evaluation

This section includes an evaluation of the demonstrations and describes the key implementation challenges faced during development of the demonstrations.

4.1. Menu-Driven Query and User Portal

4.1.1. Policies and Procedures That Contributed to Successful Aspects of the Demonstrations

Specific factors that contributed to the success of the menu-driven query and user portal demonstration are listed below.

- Use of a simple data model and synthetic data
- Use of an existing query interface
- Rapid-cycle development and testing
- Standardized query language (i.e., SQL)
- Use of servers controlled by our software partner (no issues with access to the servers or configuration)
- Centralization of the network logic in Hub (portal), thereby keeping the Data Marts simple
- Clear and efficient decision-making process during the development phase
- Clear and ongoing communication with stakeholders

4.1.2. Factors That Contributed to Unsuccessful Aspects and How These Should Be Modified

The menu-driven query and user portal demonstration achieved all of the specified aims.

4.2. Secure Distribution and Execution of SAS Code

4.2.1. Policies and Procedures That Contributed to Successful Aspects of the Demonstrations

Use of synthetic data and simple SAS programs and output contributed to the success by helping to maintain focus on the technical challenges of secure distribution

and execution of a software suite that is commonly used for the kinds of analyses envisioned for the distributed network.

The following agreements were reached regarding the installation process:

- Server in the Service Net was only available during pre-identified testing times
- Limited IP addresses were available from the Server in the Service Net
- Proof-of-principle test time was closely monitored by the site's IT representative

Once the server was ready (offline), network connectivity was requested in the site's DMZ. The site's network team set up this environment insisting on as little outside connectivity as possible. A machine to host SAS was configured and deployed, using an available SAS license, and ActivePerl was downloaded and installed. Several rounds of technical modification within the host health plan environment were required and completed in consultation with NCPHI. The NCPHI and health plan teams worked closely to achieve the necessary connectivity for the demonstration, often involving several hours of real-time debugging to make progress.

4.2.2. Factors That Contributed to Unsuccessful Aspects and How These Should Be Modified

Obtaining Institutional Approval for Participation in the Proof-of-Principle

Local IT policies and priorities made it difficult for the sites to obtain the approvals necessary to participate in the demonstration. Only one site was able to obtain the necessary approvals and install the required software to host a PHGrid node. As noted, four of the five demonstration sites were not able to install the Globus grid node in time for the demonstration. Each of the health plans followed their established IT and security guidelines for vetting the demonstration request. This process identified several barriers that contributed to the lack of success at these sites. These barriers included:

- Extended internal vetting processes, involving multiple departments
- Policies regarding password controls (password change policies, minimum password length, account lock-out policies, and session time-out rules)
- Concerns regarding the security of an open port
- Policies regarding auditing of network use
- Availability of the technical expertise, software, and hardware required for installation of the Globus Toolkit®

These health plan questions and policies led to extended technical discussions between the health plans and the Informatics team, including NCPHI, and multiple rounds of inquiry. These discussions, along with the lengthy internal procedures, combined to delay the IT and security decision-making processes enough to exclude the possibility of participation.

Performing Queries and Returning Results

During development of the demonstration it became clear that automation of the SAS program execution steps, although feasible, would not be possible within the necessary timeframe. The demonstration was able to show the secure transmission of SAS code from the central workstation managed by NCPHI (i.e., the portal) to the two

site nodes (Geisinger and NCPHI). Upon receipt, the sites initiated several manual steps that enabled the program to execute against the SAS datasets and securely return results to the central node. The manual steps consisted of the sequential execution of Perl scripts that were developed as part of the demonstration to: (1) execute the SAS program in the proper local environment; and (2) securely return the results table to the central node for aggregation.

Unanticipated Need for Custom Applications

It was anticipated that use of existing infrastructure (PHGrid services) would limit the need for extensive software development. As the project progressed, it became clear that many of the unique aspects of this project—specifically, the need to remotely invoke SAS in different computing environments and transmit the results back to a central node—required more software development efforts than anticipated. For example, the Perl scripts necessary to perform the project tasks proved more complicated than anticipated.

Summary of Challenges

This work clearly demonstrated the possibilities of a distributed research network, the potential of a strong central portal design, and some of the development challenges and successful approaches in building a network. This demonstration could be used to continue development of a network prototype by facilitating the illustration and implementation of new network features and functions. These additional functions could include more sophisticated authorization and permission policies, additional nodes, and a more flexible query interface. In addition, the system could replace the "push" mechanism that was used to send queries with a "pull" mechanism in which data holders are notified of waiting queries and retrieve them from the central portal for execution. Switching to a "pull" mechanism, also described as publish-and-subscribe or polling, would obviate many of the security concerns that limited the implementation of the second phase of the demonstration, which required access to data behind data holders' outermost firewalls.

Many of the features included in this demonstration are directly applicable to the development of a permanent network. The general hub-and-spoke design is consistent with the proposed architecture of a distributed network, and the menu-driven interface could be applied to many types of medical data. The demonstration touched on most of the system components of the high-level architecture illustrated in Figure 1 and described in Table 1. In fact, a few relatively minor modifications to the prototype would allow secure querying of health plan information of the same type as was included in the demonstration. The central portal design and menu-driven query interface could be used to develop queries and distribute them to health plans provided that the health plans adhered to the same data model as the demonstration. In this case, the health plan could then execute the query and upload the output to the central portal, at which time the portal would aggregate the queries and display the aggregated results in the same way as was done in the demonstration.

Based on the lessons learned from this proof-of-principle demonstration, an incremental approach to software development and network implementation would likely be the most reasonable way to enlist support from data holders. An incremental approach,

in terms of network size (i.e., number of data owners) and functionality, coupled with strong security, autonomy, and auditing features, would be the preferred approach to building a distributed network.

5. Additional Proof-of-Principle Activity

The Harvard Shared Health Research Information Network (SHRINE; http://catalyst.harvard.edu/shrine/) was evaluated in parallel with implementation of the proof-of-principle demonstrations described above. The overall SHRINE design is based on Informatics for Integrating Biology and the Bedside (i2b2) architecture (www.i2b2.org) that is similar to other clinical data repositories used to identify patient cohorts from electronic medical record information. The DRN Informatics team partnered with Griffin Weber, M.D., Ph.D., Chief Technology Officer at Harvard Medical School and a lead architect of the SHRINE. During the evaluation, Dr. Weber:

- Discussed the pros and cons to the SHRINE approach as it relates to the DRN project's needs for distributed analytics
- Suggested a high-level SHRINE architecture to address the needs of distributed analytics
- Demonstrated a menu-driven query across multiple clinical data repositories located behind different firewalls

The Harvard SHRINE is designed as a peer-to-peer network that will allow federated queries across local Boston hospitals that have i2b2 systems installed. The intent is to facilitate simple, menu-driven queries of clinical data repositories and return aggregated results to the user. The initial implementation plan calls for a limited set of data elements to be available and will only permit aggregated results; no patient-level data will be available for research purposes.

Some potential benefits and drawbacks of the SHRINE approach for implementation of a distributed research network are described in Table 2.

Three potential approaches to using the SHRINE architecture for a distributed network were identified and presented. One approach is to use SHRINE as a messaging protocol, but to replace the i2b2 software with custom software and database model for each data holder. This approach, which would require substantial development effort, would leverage the messaging system but allow for more flexibility in the type of queries available to the user.

Another approach is to modify the i2b2 "cells" by changing data schemas and internal workflow to accommodate the needs of the distributed network. This approach would require the least amount of effort; however, it would eliminate the possibility of using future i2b2 updates because the system would no longer be compatible with the standard i2b2 structure.

A third option is to develop new "cells" for the i2b2 "hive" that would amount to expanding the functionality of i2b2 to accommodate the needs of the network.

6. Resources

Websites of Interest:

The Agency for Health Care Research and Quality (AHRQ):
http://effectivehealthcare.ahrq.gov/

i2b2:
https://www.i2b2.org

Harvard Shared Health Research Information Network (SHRINE):
http://catalyst.harvard.edu/shrine/).

Globus®:
http://www.globus.org/

Lincoln Peak Partners:
www.lincolnpeak.com

NCPHI's public health projects in CDC priority areas:
http://sites.google.com/site/phgrid/

Tables and Figures

Table 1. Description of system components

Component	Description
Access Control Manager	Manages all aspects of security for Hub including authentication, session management, policies, group permissions, user permissions, and access rights.
Query Manager	Manages query entry, routing, and distribution.
Results Manager	Manages receipt, organization, assembly, merging, and aggregation of result sets.
Workflow Manager	Manages workflow (e.g. for query approval) including request routing, alerting and notification, approval management, and tracking.
User Manager	Manages user accounts.
Audit Manager	Provides auditing functions including activity and error logging.
Researcher User Interface	User interface for research users, including menu driven and ad hoc query entry, query management, result status, and result set management.
Data Mart Admin User Interface	User interface for Data Mart administrators including data mart setup and configuration, access control management, and workflow management.
System Admin User Interface	User interface for System administrators including Hub setup and configuration, access control management, and user management.
Hub API	Application Programming Interface (e.g. web service API) for the DRN Hub. Exposes Hub functions for remote applications including query retrieval and results submission.
Hub Database	Database for the Hub.
Message Protocol	Protocol for messaging between the Hub and external applications including the Data Marts.
Data Mart Security Manager	Manages all aspects of security for Data Mart including authentication, session management, policies, group permissions, user permissions, and access rights.
Query Execution Manager	Manages execution of queries including queue management, query translation, query engine interface, and results handling.
Data Mart API	Application Programming Interface (e.g. web service API) for the Data Mart. Exposes functions for remote applications including query submission and results retrieval.
Data Mart Audit Manager	Provides auditing functions including activity and error logging.
Data Source Manager	Manages exchange of data between the Data Mart database and source systems.
Data Mart Database	Database for the Data Mart.

Table 2. Pros and cons of a SHRINE approach to implementation of a distributed research network

Potential Benefits	Potential Drawbacks
• Existing open source platform • Large user community • Commercial support • Federated queries • Standard messaging format • Auditing and logs • Flexible options for local nodes (choice of ontology, database, non-i2b2 systems) • Integration with Harvard, other Clinical and Translational Science Awards (CTSAs)	• The data "sheriff" (a human reviewer) approves topics, not individual queries • Initial support for aggregate queries, planned support for limited data sets, no plans to support distribution and execution of SAS or other analytic code • Login limited to Active Directory or username/password in database table • i2b2 data structure may not accommodate all necessary data elements

Figure 1. System architecture

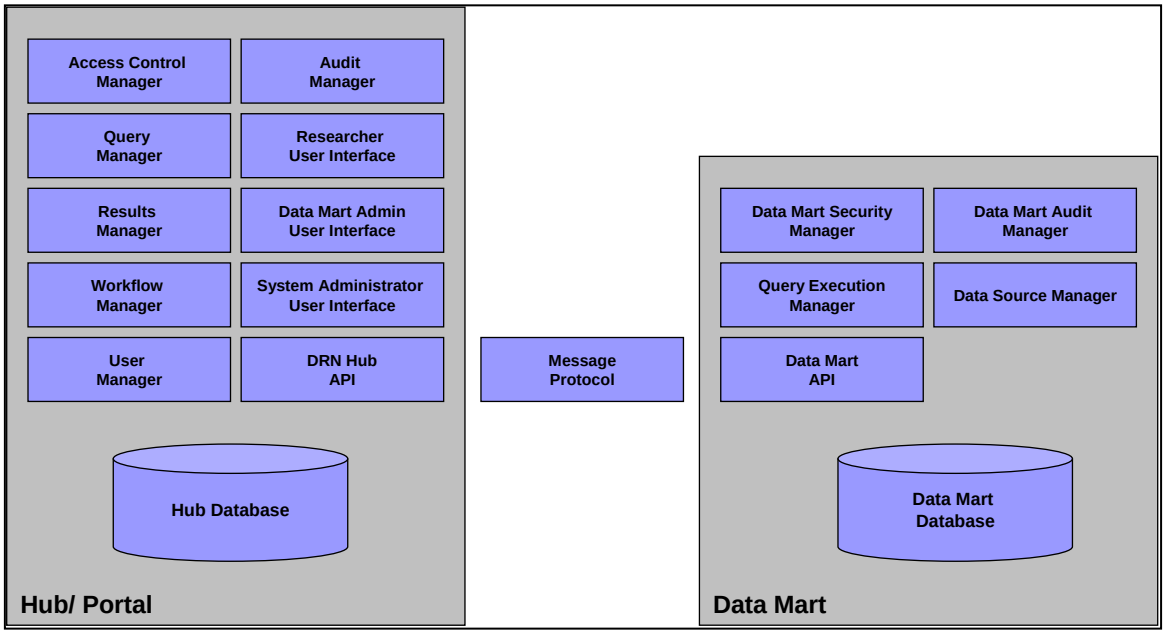

Figure 2. Network architecture