Planetary Data System

Security Service

Software Requirements and Design Document (SRD/SDD)

Sean Hardman

June 11, 2011
Version 1.0

Jet Propulsion Laboratory
Pasadena, California

CHANGE LOG

Revision	Date	Description	Author
0.1	2009-11-01	Initial draft.	S. Hardman
0.2	2009-11-10	Incorporated comments from the SDWG and added some more content and references.	S. Hardman
0.3	2009-12-01	Incorporated comments from T. King and additional content.	S. Hardman
0.4	2010-03-09	Completed comment incorporation with updates to the requirements and data model. Also updated the architecture and implementation deployment.	S. Hardman
0.5	2010-09-15	Removed use cases and requirements associated with updating and querying user information. This information will now be managed in the Registry Service. Added a level 3 requirement for derivation purposes. Also replaced OpenSSO references with OpenAM.	S. Hardman
0.6	2010-09-28	Updated controlling document reference.	S. Hardman
1.0	2011-06-11	Updated document references, removed references to OpenAM and added LDAP group definitions.	S. Hardman

TABLE OF CONTENTS

1.0 INTRODUCTION

The PDS 2010 effort will overhaul the PDS data architecture (e.g., data model, data structures, data dictionary, etc) and deploy a software system (online data services, distributed data catalog, etc) that fully embraces the PDS federation as an integrated system while leveraging modern information technology.

This service provides the authentication and authorization functions for the system.

1.1 Document Scope and Purpose

This document addresses the use cases, requirements and software design of the Security service within the PDS 2010 data system. This document is intended for the reviewer of the service as well as the developer and tester of the service.

1.2 Method

This combined Software Requirements and Software Design Document (SRD/SDD) represents the software by defining use cases and requirements and by using architecture diagrams, functional descriptions, context diagrams and data flow diagrams for the high-level design. The detailed design will be illustrated using UML diagrams.

1.3 Notation

The numbering of the requirements in this document will be formatted as **LX.SEC.AA.X**, where:

- **LX** represents the requirements level where X is a number.
- **SEC** is an abbreviation representing the security requirements section for the specified level.
- **AA** is a two-letter abbreviation representing the requirement sub-category (optional).
- **X** is a unique number within the section and optional sub-category for the requirement.

Following the text of a requirement may be a reference to the requirement or use case from which it was derived. The reference will be in parenthesis. A paragraph following a requirement, which is indented and has a reduced font size, represents a comment providing additional insight for the requirement that it follows. This comment should is not part of the requirement for development or testing purposes.

1.4 Controlling Documents

[1] Planetary Data System (PDS) Level 1, 2 and 3 Requirements, March 26, 2010.

[2] Planetary Data System (PDS) 2010 Project Plan, February 2010.

[3] Planetary Data System (PDS) 2010 System Architecture Specification, Version 1.2, May 25, 2011.

[4] Planetary Data System (PDS) 2010 Operations Concept, February 2010.

[5] Planetary Data System (PDS) General System Software Requirements Document (SRD), Version 1.0, June 11, 2011.

1.5 Applicable Documents

[6] Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map, OpenLDAP Foundation, RFC 4510, June 2006.

1.6 Document Maintenance

The component design will evolve over time and this document should reflect that evolution. This document is limited to design content because the specification content will be captured in separate documentation (e.g., Installation Guide, Operation Guide, etc.). This document is under configuration control.

2.0 COMPONENT DESCRIPTION

The Security service provides the authentication and authorization functions for the PDS 2010 system (referred to as the "system" from this point forward). In addition to security, this service will include directory service functionality by utilizing the Lightweight Directory Access Protocol (LDAP) standard. The intent of this service is to control access to interfaces and services that require authentication and authorization (e.g., Monitor, Report, Registry interfaces, etc.). The following diagram details the context of the Security service within the system:

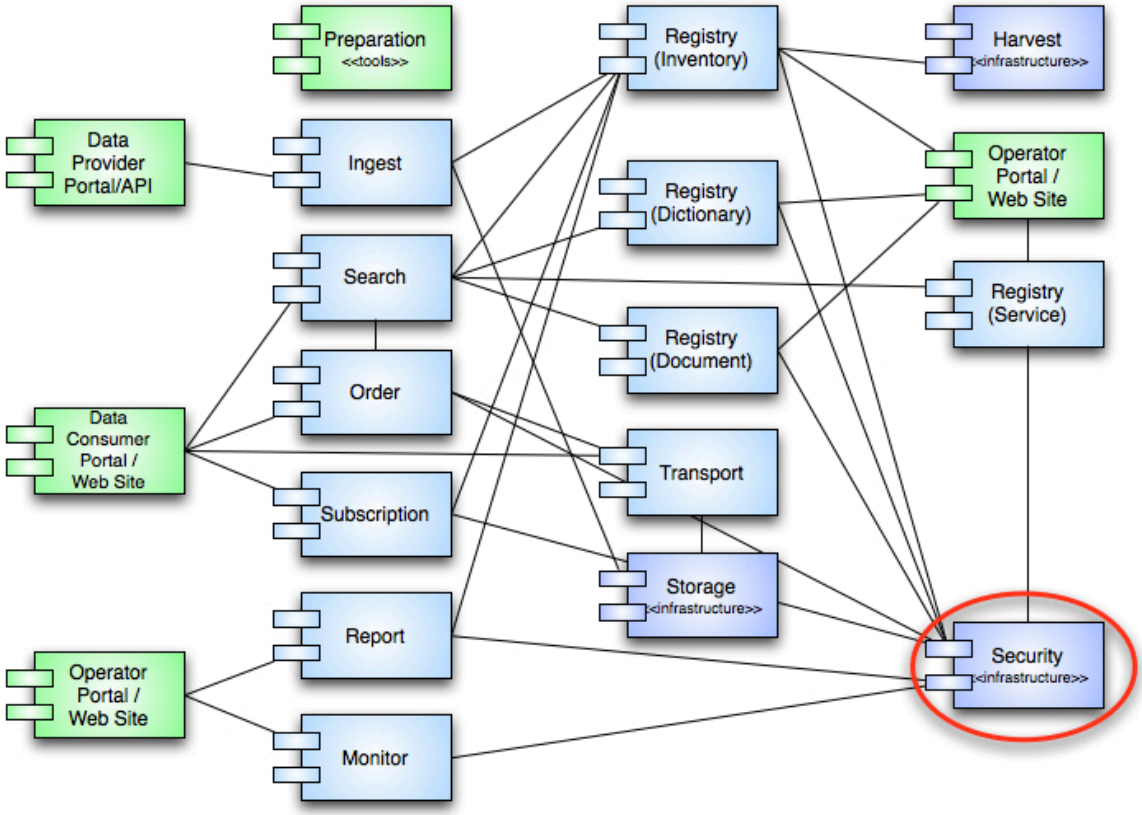

Figure 1: Security Service Context

Within the system, the Security service is an infrastructure service. This means that there will not be any external interfaces to the service. All interfaces will be with other components of the system and considered internal interfaces.

As depicted in the diagram above the Security service supports multiple interfaces to other services within the system. In general, these services will interface with the Security service as a means of obtaining authentication and authorization for their capabilities. In addition to the interfaces depicted in the diagram above, there will likely exist interfaces with portal components as a

means of authentication and authorization for access to those interfaces. The details regarding the service interfaces are provided in section 6.2.

Although the current PDS system does not have an organized Security service, there are interfaces and functions that require controlled access and homegrown solutions exist to control that access. The following are examples of capabilities that exist in the current system that require authentication and/or authorization:

Ingest Catalog Metadata
The Data Engineers at the Engineering Node perform ingestion of catalog metadata into the catalog database. The PDS 2010 Registry service will provide similar functionality for the catalog and product level metadata.

Update Catalog Metadata
The current system has a number of tools that allow Engineering Node and Discipline Node personnel to update certain portions of the catalog database. The PDS 2010 system will provide a portal to enable updates or maintenance of the majority of inventory entries.

Manage Subscriptions
The current Subscription application allows a user to login and manage their subscriptions. PDS 2010 will offer a similar service.

The service defined in this document will provide the PDS 2010 system with a single implementation of authentication and authorization capabilities for use by the other services and applications within the system.

3.0 USE CASES

A use case represents a capability of the component and why the user (actor) interacts with the component. It should be at a high enough level so as not to reveal or imply the internal structure of the system. An actor is an object (e.g., person, application, etc.) outside the scope of the component but interacts with the component. This section captures the use cases for the Security service based on the description of the service from the previous section. These use cases will be used in the derivation of requirements for the service. The following diagram details the use cases:

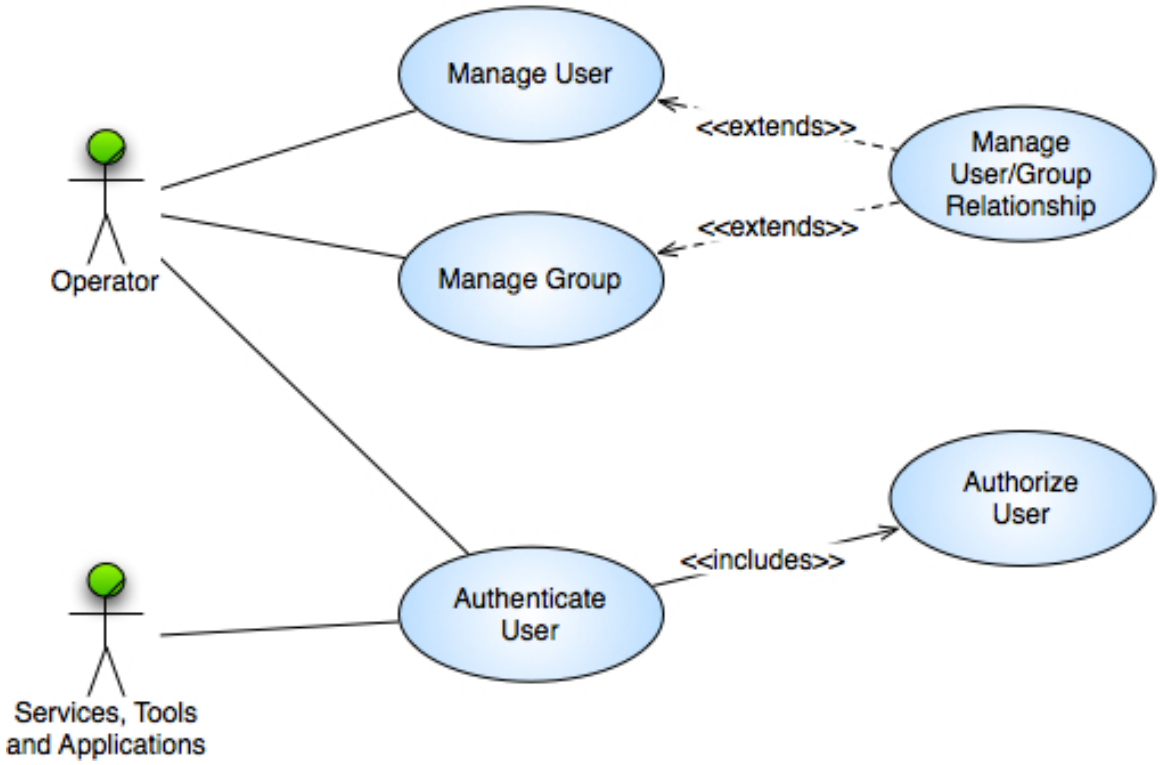

Figure 2: Security Service Use Cases

The above diagram identifies the following actors (represented as stick figures):

Operator
This actor represents a portion of the PDS Technical group that is responsible for configuring and monitoring the system.

Services, Tools and Applications
This actor represents the software within the system that will request authentication and authorization for access to its capabilities.

The following sections detail the use cases identified in the above diagram.

3.1 Manage User

A user's identity within the system must be managed including creation, update and deletion of that identity. This use case pertains to the Operator actor.

1. Operator receives a request to create, update or delete a user identity.
2. Operator accesses the Security service manager interface and performs the requested operation. Although not depicted in the use case diagram, the Operator requires authentication.
3. Security service accepts (verifies input against constraints) and commits (updates the underlying data store) the operation.

Alternative: Create Operation
At step 2, the operation is to create a new user identity.

a. Operator submits identifying information for the user.
b. Operator adds the user to one or more groups (extend Manage User/Group Relationship use case).
c. Return to primary scenario at step 3.

Alternative: Update Operation
At step 2, the operation is to update an existing user identity.

a. Operator submits updates to the identifying information for the user.
b. Operator optionally adds or removes the user to/from one or more groups (extend Manage User/Group Relationship use case).
c. Return to primary scenario at step 3.

Alternative: Delete Operation
At step 2, the operation is to delete an existing user identity.

a. Operator removes the user from one or more groups (extend Manage User/Group Relationship use case).
b. Operator deletes the identifying information for the user.
c. Return to primary scenario at step 3.

3.2 Manage Group

A group's identity within the system must be managed including creation, update and deletion of that identity. This use case pertains to the Operator actor.

1. Operator receives a request to create, update or delete a group identity.
2. Operator accesses the Security service manager interface and performs the requested operation. Although not depicted in the use case diagram, the Operator requires authentication.
3. Security service accepts (verifies input against constraints) and commits (updates the underlying data store) the operation.

Alternative: Create Operation

At step 2, the operation is to create a new group identity.

a. Operator submits identifying information for the group.
b. Operator adds one or more users to the group (extend Manage User/Group Relationship use case).
c. Return to primary scenario at step 3.

Alternative: Update Operation

At step 2, the operation is to update an existing group identity.

a. Operator submits updates to the identifying information for the group.
b. Operator optionally adds or removes one or more users to/from group (extend Manage User/Group Relationship use case).
c. Return to primary scenario at step 3.

Alternative: Delete Operation

At step 2, the operation is to delete an existing group identity.

a. Operator removes the users from the group (extend Manage User/Group Relationship use case).
b. Operator deletes the identifying information for the group.
c. Return to primary scenario at step 3.

3.3 Manage User/Group Relationship

The user/group relationship within the system must be managed including adding and removing a user from a group. The Manage User and Manager Group use cases extend this use case. This use case pertains to the Operator actor.

1. Security service receives a request to add or remove a user from an existing group.
2. Security service accepts (verifies input against constraints) and commits (updates the underlying data store) the operation.

3.4 Authenticate User

A user of a system service/application requires authentication where appropriate. This use case pertains to all actors defined above.

1. User requests access to a restricted service/application.
2. Service/Application challenges the user for authentication credentials (user name and password).
3. User securely submits authentication credentials to service/application.
4. Service/Application securely submits authentication credentials to Security service.
5. Security service verifies authentication credentials.
6. Security service determines authorized access (include Authorize User use case).

7. Service/Application receives authentication/authorization from Security service and grants access to the user.

Alternative: Authentication Failure
At step 5, the user's authentication credentials did not match a valid user.

a. Security service returns an exception to the service/application.
b. Service/Application challenges the user to reenter authentication credentials.
c. Return to primary scenario at step 3 unless the maximum retry count has been exceeded. In that case, the Security service returns an exception to the service/application.

Alternative: Authorization Failure
At step 6, the user does not belong to an authorized group to access the requested service/application.

a. Security service returns an exception to the service/application.

3.5 Authorize User

A user of a system service/application requires authorization where appropriate. This use case is included as part of the Authenticate User use case and is only exercised if the user is authenticated. This use case pertains to all actors defined above.

1. Security service receives valid authentication credentials.
2. Security service determines whether the authenticated user belongs to an authorized group to access the requested service/application.

4.0 REQUIREMENTS

The architecture definition phase of the PDS 2010 project resulted in the decomposition of the system into several elements [3]. The Security service does not derive directly from any of those elements but is derived from requirement 2.10.3 of the PDS Level 1, 2, and 3 Requirements document [1]. The following level 3 requirement is relevant to this service:

> **2.10.3** PDS will ensure that appropriate mechanisms are in place to prevent unauthorized users from compromising the integrity of PDS systems and data

In addition to the level 4 and 5 requirements specified below, the Security service must also comply with the general service-based requirements found in the General System SRD document [5].

4.1 Level 4 Requirements

The level four requirements in PDS represent subsystem or component requirements at a high level. The following requirements pertain to the Security service:

L4.SEC.1 - The system shall authorize access to system interfaces that allow for ingestion or modification of data contained within the system. (2.10.3)

L4.SEC.2 - The system shall maintain a list of authorized users. (2.10.3)

4.2 Level 5 Requirements

The level five requirements in PDS represent subsystem or component requirements at a detailed level. The following requirements pertain to the Security service:

L5.SEC.1 - The service shall authenticate a user given identifying credentials for that user. (L4.SEC.1, UC 3.4)

L5.SEC.2 - The service shall encrypt the transmission of identifying credentials across the network. (L4.SEC.1, UC 3.4)

L5.SEC.3 - The service shall authorize an authenticated user for access to a controlled capability. (L4.SEC.1, UC 3.5)

L5.SEC.4 - The service shall allow an operator of the system to create, update or delete a user identity. (L4.SEC.2, UC 3.1)

L5.SEC.5 - The service shall capture identifying information associated with a user identity. (L4.SEC.2, UC 3.1)

L5.SEC.6 - The service shall allow an operator of the system to create, update or delete a group identity. (L4.SEC.2, UC 3.2)

> A group will be associated with a controlled capability within the system. A user's membership within a group will determine authorized access.

L5.SEC.7 - The service shall allow an operator of the system to add or remove a user from a group. (L4.SEC.2, UC 3.3)

5.0 DESIGN PHILOSOPHY, ASSUMPTIONS, AND CONSTRAINTS

The intent of the Security service is to provide a simple solution for authorizing access to certain interfaces within the system. The service will utilize an Access Control List (ACL) security model where users are assigned to groups with authorized access to capabilities within the system based on group membership. Before authorization, the user is authenticated by securely passing their identifying credentials (user name and password) across the network to this service for authentication. To accomplish this the design utilizes prevailing standards and open source software that satisfy the requirements.

The prevailing standard for services of this type (e.g., directory service) is the Lightweight Directory Access Protocol (LDAP). LDAP is an application protocol for querying and modifying directory services running over TCP/IP. Its current version is LDAPv3, which is specified in a series of Internet Engineering Task Force (IETF) Standard Track Requests for Comments (RFCs) as detailed in RFC 4510 [6].

6.0 ARCHITECTURAL DESIGN

The architectural design covers the component breakdown within the service, external/internal interfaces and the associated data model.

6.1 Component Architecture

The following diagram details the architecture for the Security service including interfaces between the various components:

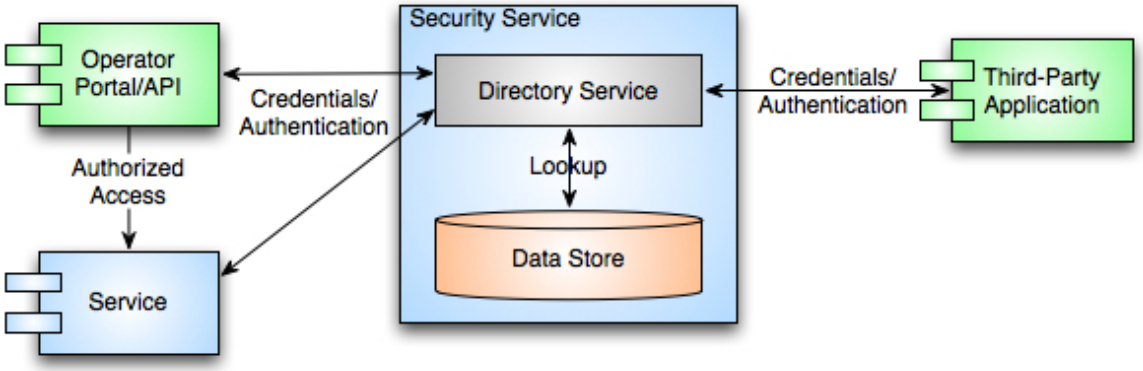

Figure 3: Security Service Architecture

The service architecture provides for three different scenarios for authenticating and authorizing a user's access to capabilities within the system:

Portal/Application Access to Service
This scenario represents a portal/application that offers a capability requiring controlled access to a system service. The portal/application interfaces directly with directory service by passing the authentication credentials and receiving authorization in the form of a cookie. The portal/application sends the authorization to the service with each subsequent request.

Portal/Application Access
This scenario represents a portal/application that offers a capability requiring controlled access but does not interface with one of the system services. The portal/application interfaces with the directory service by passing the authentication credentials and receiving authorization. Although there may be others, the interface that supports the Manage User and Manage Group use cases will certainly fit in this scenario.

Third-Party Application Access
This scenario represents third-party applications that will interface directly with the directory service by passing the authentication credentials and receiving authorization.

In the scenarios above, the directory service represents the OpenDS component.

6.2 External Interface Design

The interfaces that interact with the directory service follow the Lightweight Directory Access Protocol (LDAP). The Technical Specification Road Map for LDAP [6] provides the details for this protocol.

6.3 Internal Interface Design

There are no internal interfaces specified for the Security service.

6.4 Data Model

The data model pertains to the organization of the data stored in the data store of the directory service. The following list of attributes represents the minimum set specified for each personnel entry:

- **uid** represents the unique identifier of the individual. This could be an email address or some other uniquely identifying string of characters.
- **sn** represents the surname or last name of the individual.
- **cn** represents the common name or full name of the individual.
- **userPassword** represents the secret word or phrase of the individual.

The following diagram represents the namespace designation utilized in the directory service:

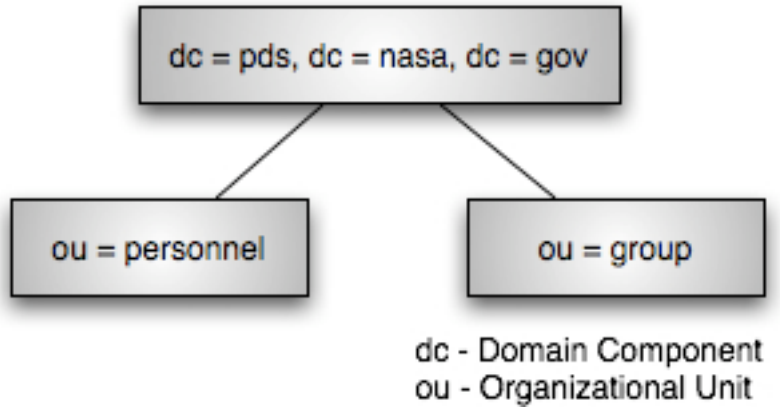

Figure 4: Security Service Namespace

The Security service will utilize group membership to determine authorized access to a given component capability. The following groups have been defined:

pds-guest
This group contains members of the PDS user community. Members of this group will be able to access the Subscription and Order services. This would include the ability to manage and store any preferences associated with these services.

pds-staff
This group contains members of the PDS Node staff and selected affiliates. Members of this group will be able to access PDS internal services like the Report and Monitor services. This would include the ability to view information but not necessarily modify anything.

pds-admin
This group contains members of the Engineering Node who will act as operators for the system services. Members of this group will be able to access all services. This would include the ability to modify content associated with these services.

<node>-admin
These groups contain members of staff from a specific PDS Node. Group names would be *atm-admin*, *geo-admin*, etc. Members of this group will be able to access all services under the specific Node's purview. Unlike the *pds-staff* group, this would include the ability to modify content associated with these services.

7.0 ANALYSIS

The choice of LDAP is based on its prevalence in industry. The Engineering Node development staff also installed and tested OpenAM. The demonstration consisted of an Apache Tomcat with OpenAM deployed to that server and configured to utilize the JPL LDAP server for authentication. The staff also configured a second Tomcat server with the OpenAM policy agent and the sample web application. The developer accessed the web application and was prompted for a user name and password. Once entered, the developer was authenticated against the JPL LDAP server and was granted access to the application.

Although the demonstration was simple, it did demonstrate one of the scenarios for authentication within the PDS 2010 system. After using the OpenAM software in a PDS 2010 system deployment after the prototype build it was determined that the complexity in configuring the software was not worth the effort. The decision was then made to keep it simple only utilize an LDAP-based directory service. OpenDS was the directory service of choice mainly because it was the software that was packaged with the OpenAM software.

8.0 IMPLEMENTATION

The PDS 2010 system is a phased implementation with increasing capabilities delivered in three planned builds. The builds are as follows:

- **Build 1** – This build consists of the Ingestion subsystem including the Security, Harvest, Registry (Inventory, Dictionary, Document, Service) and Report components along with the Data Provider tool suite.
- **Build 2** – This build consists of the Distribution subsystem including the Search and Monitor components along with a revised web site and general portal applications.
- **Build 3** – This build consists of enhanced user capabilities include the Order and Subscription components along with integration of Discipline Node applications and science services.

The Security service is scheduled for delivery in Build 1. This initial delivery will support test collection generation and registration. Additional capabilities are planned for follow-on deliveries as testing progresses and the data model matures.

Implementation of the Security service is limited to configuration and integration of the selected open source components with the other components in the system.

The scenario for deployment is to run a centralized instance of the directory service (OpenDS) at the Engineering Node. The following diagram depicts this deployment scenario:

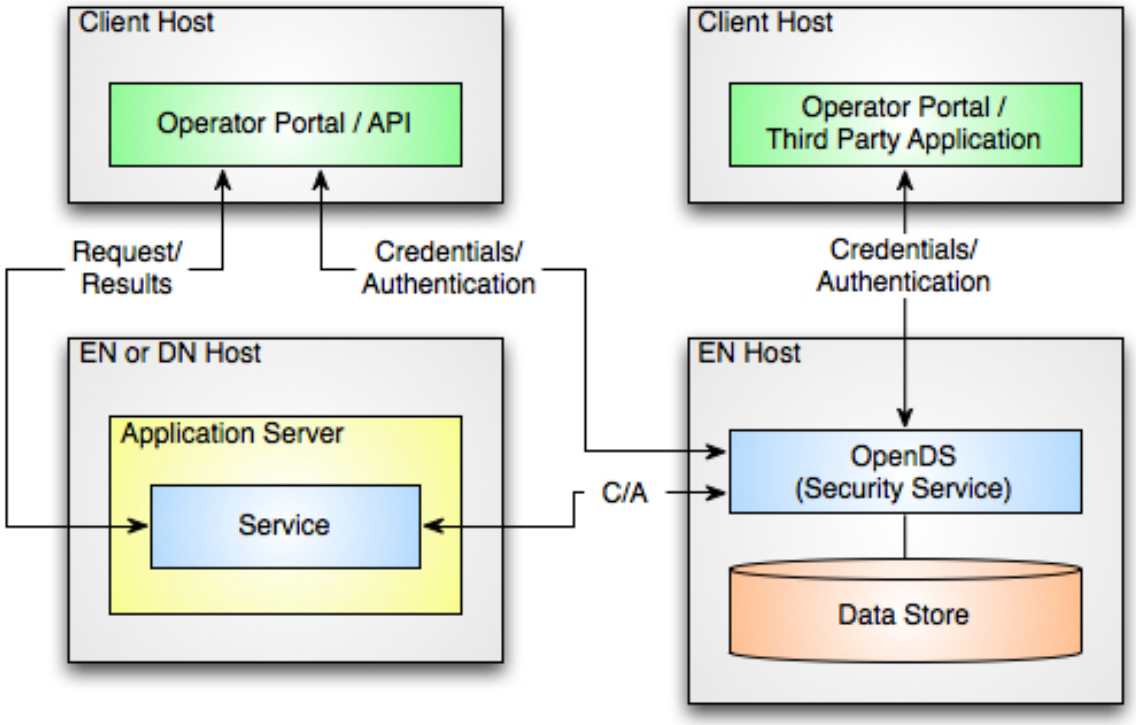

Figure 6: Security Service Deployment

Interfaces with the directory service (OpenDS) utilize LDAP as the communication protocol. Where controlled access to a service is required, the interface will utilize a service's REST-based interface over the Hypertext Transfer Protocol Secure (HTTPS) communication protocol.

9.0 DETAILED DESIGN

Since we are integrating an off-the-shelf software package, there is no detailed software design for this component.

APPENDIX A ACRONYMS

The following acronyms pertain to this document:

ACL	Access Control List
API	Application Programming Interface
COTS	Commercial Off-The-Shelf
DNS	Domain Name Service
DS	Directory Service
IETF	Internet Engineering Task Force
JPL	Jet Propulsion Laboratory
HTTPS	Hypertext Transfer Protocol Secure
LDAP	Lightweight Directory Access Protocol
NASA	National Aeronautics and Space Administration
PDS	Planetary Data System
REST	Representational State Transfer
RFC	Request For Comment
SDD	Software Design Document
SRD	Software Requirements Document
UC	Use Case