# Planetary Data System

# Security Service

# Software Requirements and Design Document (SRD/SDD)

Sean Hardman

March 13, 2010
Version 0.4

Jet Propulsion Laboratory
Pasadena, California

DRAFT

# 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 |

DRAFT

# TABLE OF CONTENTS

**1.0 INTRODUCTION ..... 4**
1.1 Document Scope and Purpose ..... 4
1.2 Method ..... 4
1.3 Notation ..... 4
1.4 Controlling Documents ..... 5
1.5 Applicable Documents ..... 5
1.6 Document Maintenance ..... 5
**2.0 SERVICE DESCRIPTION ..... 6**
**3.0 USE CASES ..... 8**
3.1 Manage User ..... 9
3.2 Manage Group ..... 10
3.3 Manage User/Group Relationship ..... 10
3.4 Authenticate User ..... 11
3.5 Authorize User ..... 11
3.6 Query User(s) ..... 12
3.7 Update User ..... 12
**4.0 REQUIREMENTS ..... 13**
4.1 Level 4 Requirements ..... 13
4.2 Level 5 Requirements ..... 13
**5.0 DESIGN PHILOSOPHY, ASSUMPTIONS, AND CONSTRAINTS ..... 15**
**6.0 ARCHITECTURAL DESIGN ..... 16**
6.1 Service Architecture ..... 16
6.2 External Interface Design ..... 17
6.3 Internal Interface Design ..... 17
6.4 Data Model ..... 17
**7.0 ANALYSIS ..... 20**
**8.0 IMPLEMENTATION ..... 21**
**9.0 DETAILED DESIGN ..... 23**
**APPENDIX A ACRONYMS ..... 24**

DRAFT

# 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.

DRAFT

## 1.4 Controlling Documents

[1] Planetary Data System (PDS) Level 1, 2 and 3 Requirements, August 2006.

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

[3] Planetary Data System (PDS) 2010 System Architecture Specification, Version 1.0, February 28, 2010.

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

[5] Planetary Data System (PDS) Service Software Requirements Document (SRD), TBD.

## 1.5 Applicable Documents

[6] NASA Procedural Requirements (NPR) Security of Information Technology, NPR 2810.1A, May 16, 2006.

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

[8] Enabling Virtual Federation With OpenSSO, Part 1: Introduction, January 16, 2009.

[9] Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, March 15, 2005.

[10] PDS4 Information Model Specification, PDS4 Information Model Specification Team, February 2010.

## 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.

DRAFT

## 2.0 SERVICE 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:

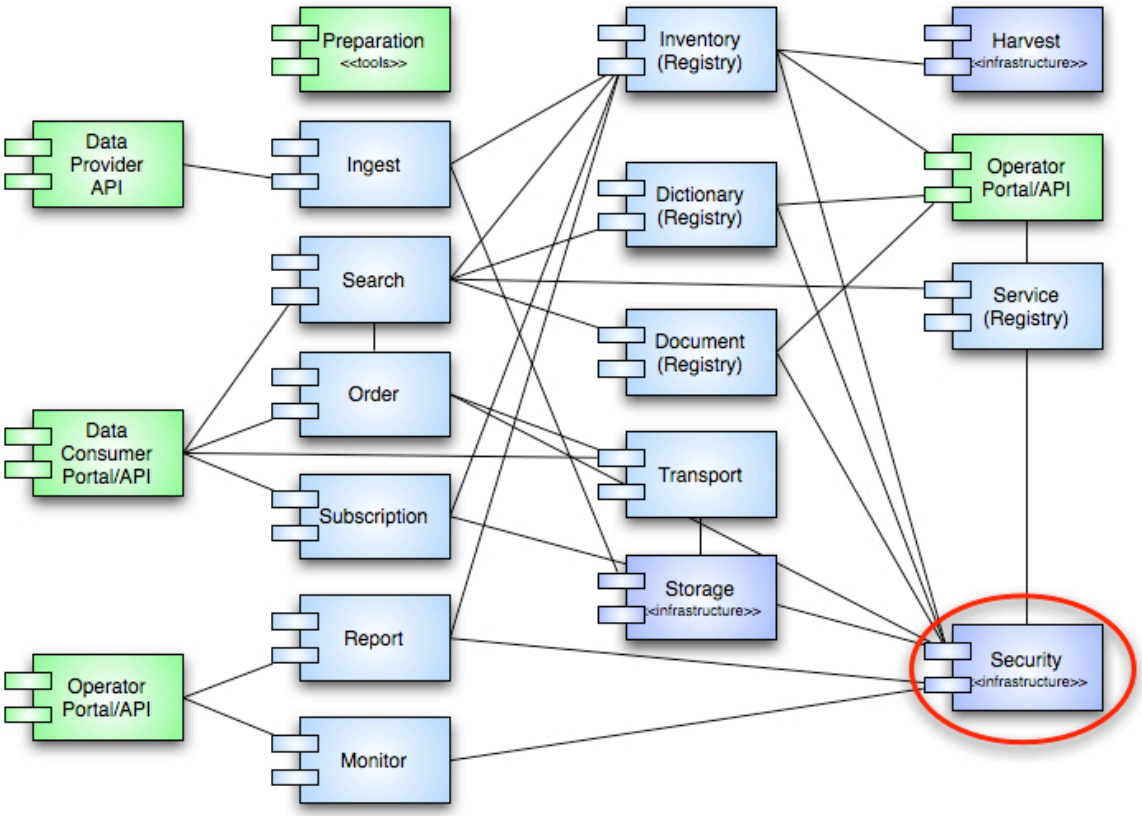


Figure 1: Security Service Context

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 Inventory 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.

**Phonebook**
The current Phonebook application lists contact information for PDS personnel. PDS 2010 will provide a similar application with the storage and management of those entries handled by this 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.

DRAFT

## 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 component from the previous section. These use cases will be used in the derivation of requirements for the component. The following diagram details the use cases:

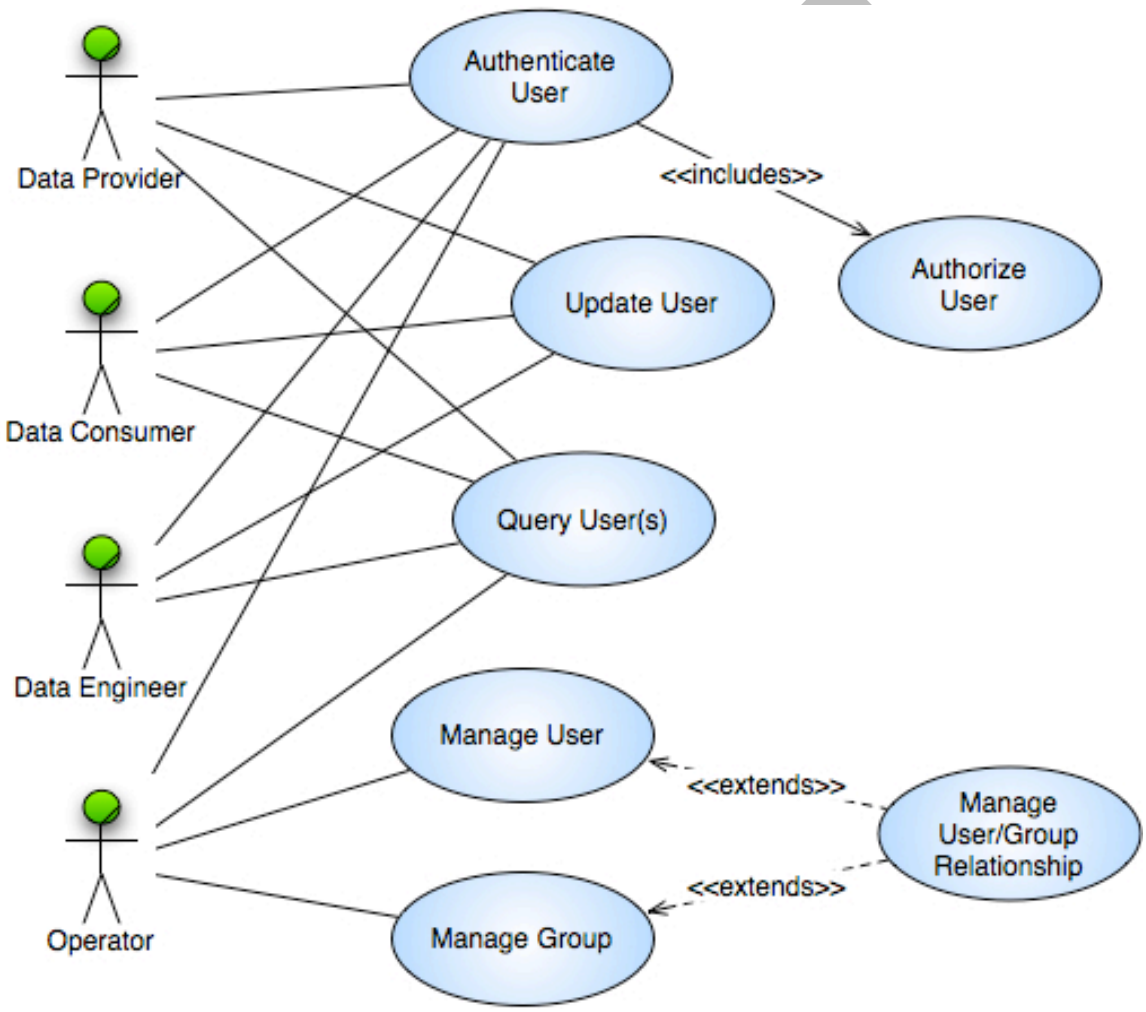


Figure 2: Security Service Use Cases

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

**Data Consumer**

This actor represents the Planetary Scientist, which includes those experienced with solar system exploration missions and those who are mission-naïve. They include graduate students.

**Data Engineer**
This actor represents a portion of the PDS Technical group that curates the data before and after it enters the PDS system.

**Data Provider**
This actor represents the mission, instrument team and NASA-funded researcher who are involved with delivering data to the PDS.

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

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.

<u>Alternative: Create Operation</u>
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.

<u>Alternative: Update Operation</u>
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.

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

DRAFT

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.

<u>Alternative: Create Operation</u>
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.

<u>Alternative: Update Operation</u>
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.

<u>Alternative: Delete Operation</u>
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.

DRAFT

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.

<u>Alternative: Authentication Failure</u>
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.

<u>Alternative: Authorization Failure</u>
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.

DRAFT

2. Security service determines whether the authenticated user belongs to an authorized group to access the requested service/application.

### 3.6 Query User(s)

The publically available information in a user's identity is discoverable to support a phonebook-like function. This use case pertains to all actors defined above and utilizes an application interface developed outside of the service.

1. User accesses the directory application from one of the PDS-provided portals and performs a query.
2. Security service accepts the query and returns information for one or more entries matching the criteria.

### 3.7 Update User

A user must have the ability to update their own non-critical identifying information, such as address, phone number, etc. This use case pertains to all actors defined above except for the Operator and utilizes an application interface developed outside of the service. Operators will use the manager interface to update their own information.

1. User accesses the directory application from one of the PDS-provided portals and performs the information update. Although not depicted in the use case diagram, the User requires authentication.
2. Security service accepts (verifies input against constraints) and commits (updates the underlying data store) the operation.

DRAFT

# 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 nor are there any related level 3 requirements [1] associated with security. In order to justify the functionality of this service, the capabilities of the existing system, as described in section 2.0 of this document, are referenced as well as any new functionality necessary to support planned services in the PDS 2010 system.

## 4.1 Level 4 Requirements

The level four requirements in PDS represent subsystem, component or tool 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. (Legacy)

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

## 4.2 Level 5 Requirements

The level five requirements in PDS represent subsystem, component or tool 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)

> This requirement facilitates the phonebook capability with the Personnel data model identifies captured attributes and any constraints on those attributes based on the type of user.

DRAFT

**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)

**L5.SEC.8** - The service shall allow a user to update a restricted set of the user's own identifying information. (L4.SEC.2, UC 3.7)

Users may not update their group membership.

**L5.SEC.9** - The service shall allow queries against the list of users with a restricted set of identifying information returned. (L4.SEC.2, UC 3.6)

The result set will not include attributes such as the user's password and any internal flags.

DRAFT

# 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 [7].

In addition to a directory service that supports LDAP, the PDS 2010 system will need software that facilitates the exchange of identifying information between client portals/applications and the services. This design specifies the OpenSSO [8], where SSO stands for single sign-on, software to satisfy this capability. Since OpenSSO suggests the use of OpenDS, where DS stands for Directory Service, this design specifies the use of this software component for the directory service.

DRAFT

# 6.0 ARCHITECTURAL DESIGN

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

## 6.1 Service Architecture

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

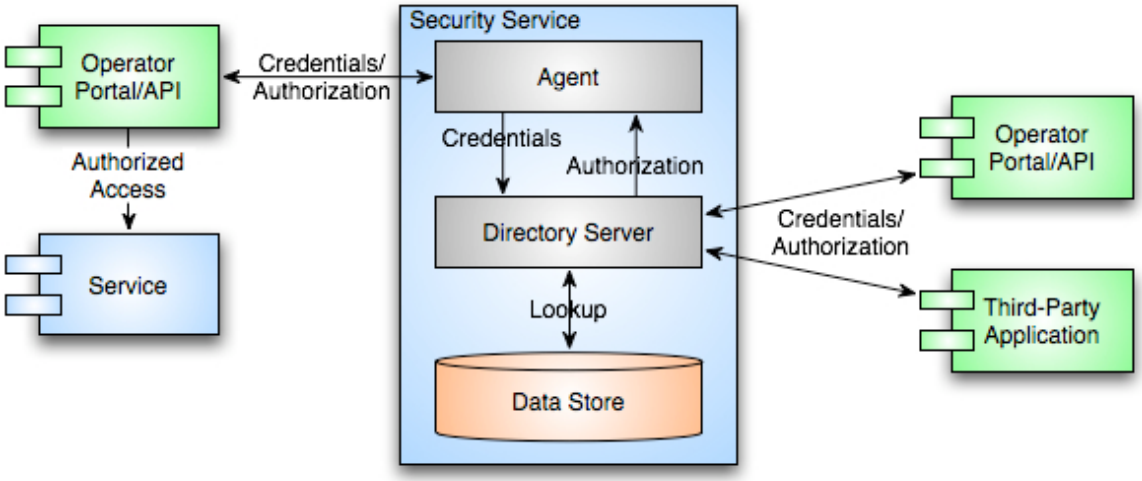


Figure 3: Security Service Architecture

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 Agent intercepts the request to the service and handles the authentication/authorization of the user by accepting and passing the authentication credentials to Directory Server. The Agent returns the authorization to the portal/application 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. In this scenario, the portal/application interfaces directly with Directory Server 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 Server by passing the authentication credentials and receiving authorization.

In the scenarios above, the Agent represents the OpenSSO software package and the Directory Server represents the OpenDS software package.

## 6.2 External Interface Design

The interface between the Operator Portal and OpenSSO uses Secure Attributes Exchange (SAE). The SAE interface is specific to the OpenSSO software [8]. It provides a mechanism for one application to communicate identity information to a second application in a different domain. In essence, SAE provides a secure gateway that enables legacy applications to communicate user attributes used for authentication without having to deal specifically with federation protocols and processing.

The interfaces that interact with the Directory Server follow the Lightweight Directory Access Protocol (LDAP). The Technical Specification Road Map for LDAP [7], provides the details for this protocol.

## 6.3 Internal Interface Design

The main internal interface within the service occurs between OpenSSO agents where the Security Assertion Markup Language (SAML) is utilized. SAML [9] is an XML-based standard for exchanging authentication and authorization data between security domains.

## 6.4 Data Model

The data model pertains to the organization of the data stored in the data store of the Directory Server. The following diagram represents the data model:

DRAFT

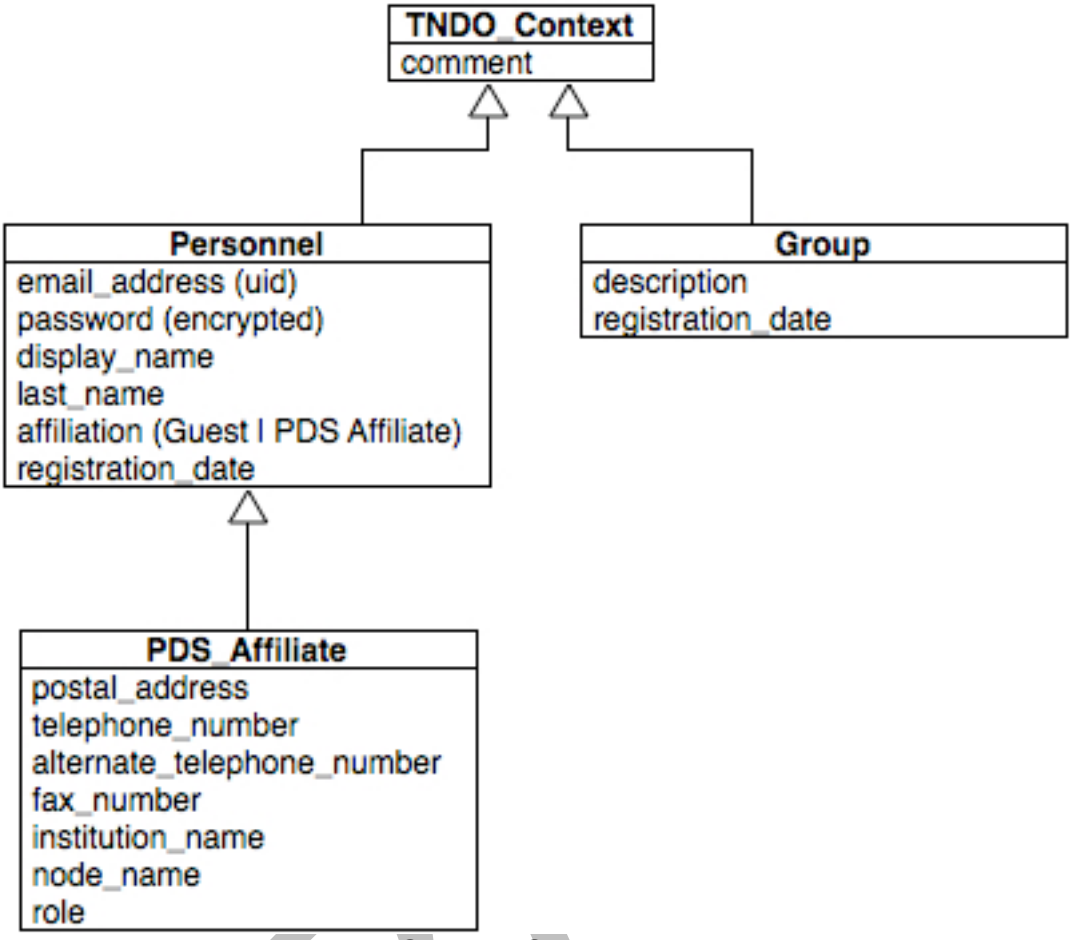


Figure 4: Security Service Data Model

The class definitions are as follows:

**TNDO_Context**
The Tagged NonDigital Object (TNDO) Context class is an abstract class for the context class hierarchy.

**Group**
The Group class provides a description of a group. Membership in a given group grants certain permissions to PDS resources.

**Personnel**
The Personnel class provides a description of a person who has an association with the planetary science community.

**PDS_Affiliate**
The PDS Affiliate class provides a description of a person who has an association with the planetary science community and has access to PDS resources.

See the PDS4 Information Model Specification [10] for additional details on the data model. The following diagram represents the namespace designation utilized in the Directory Server related to the data model depicted above:

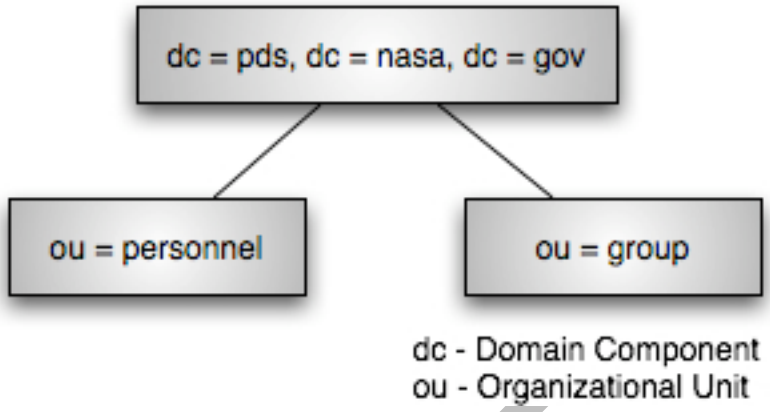


Figure 5: Security Service Namespace

DRAFT

# 7.0 ANALYSIS

The choice of LDAP is based on its prevalence in industry. The choice of OpenSSO was based on the experience of the Lunar Mapping and Modeling Project (LMMP) at JPL who are implementing similar capabilities as the PDS 2010 system.

The Engineering Node development staff also installed and tested the OpenSSO demonstration software. The demonstration consisted of an Apache Tomcat with OpenSSO deployed to that server and configured to utilize the JPL LDAP server for authentication. The staff also configured a second Tomcat server with the OpenSSO 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. Installation and test of the demonstration occurred on the developers Mac PowerBook as well as on the target operational machine running Linux (Red Hat 5). Along with the simple authentication example provided in the demonstration, one can protect Web applications from unauthorized access through OpenSSO's security services. Several other options for protection also exist, such as by means of the client SDK or identity services that take advantage of the Simple Object Access Protocol (SOAP) and Representational State Transfer (REST) interfaces.

DRAFT

# 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. There is no planned phasing with regard to the implementation with all planned capabilities available in Build 1.

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 Server (OpenDS) at the Engineering Node with an OpenSSO Agent configured for each Apache Tomcat server that hosts a service with controlled access. The following diagram depicts this deployment scenario:

DRAFT

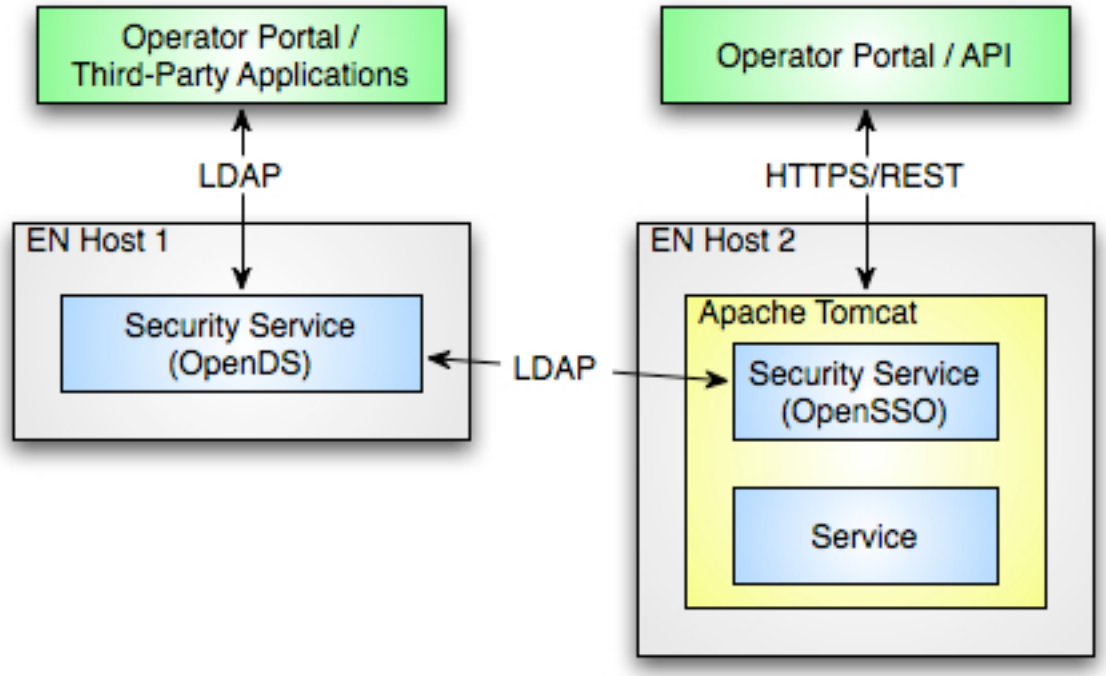


Figure 6: Security Service Deployment

Interfaces with the Directory Server (OpenDS) utilize LDAP as the communication protocol. Where controlled access to a service is required, the interface will utilize the service's REST-based interface over the Hypertext Transfer Protocol Secure (HTTPS) communication protocol. In this scenario, the OpenSSO Agent intercepts the request to the service and requires the client to submit authentication credentials.

DRAFT

# 9.0 DETAILED DESIGN

More information on this topic will be forthcoming in a subsequent version of this document.

DRAFT

# 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 |
| NPR | NASA Procedural Requirements |
| PDS | Planetary Data System |
| REST | Representational State Transfer |
| RFC | Request For Comment |
| SAE | Secure Attributes Exchange |
| SAML | Security Assertion Markup Language |
| SDK | Software Development Kit |
| SOA | Service-Oriented Architecture |
| SOAP | Simple Object Access Protocol |
| SDD | Software Design Document |
| SRD | Software Requirements Document |
| SSO | Single Sign-On |
| UC | Use Case |

DRAFT