Planetary Data System

Search Service

Software Requirements and Design Document (SRD/SDD)

Sean Kelly

2010-03-17
Version 0.1

Jet Propulsion Laboratory
Pasadena, California

CHANGE LOG

Revision	Date	Description	Author
0.1	2010-03-17	Initial draft.	S. Kelly

TABLE OF CONTENTS

1.0 INTRODUCTION

The PDS 2010 effort will overhaul the PDS data architecture (including, but not limited to, the data model, data structures, data dictionary, data etc.) and deploy a software system (including, but not limited to, data services, distributed data catalog, data etc.) that fully embraces the PDS federation as an integrated system while taking advantage of modern innovations in information technology (including, but not limited to, networking capabilities, processing speeds, and software breakthroughs).

This service provides the function of searching cataloged metadata.

1.1 Document Scope and Purpose

This document addresses the use cases, architecture, analysis, design, and implementation of the search service within the PDS 2010 data system. This document is intended for the reviewer of the service as well as the developer of the service (for implementation) and tester of the service (for quality assurance).

1.2 Method

This combined Software Requirements and Software Design Document (hereafter, SRD/SDD) represents the search service software by defining use cases (a form of requirements capture), requirements list (a second form of requirements capture), 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

Although use cases are presented in this document, a separate list of requirements in the form of declarative statements is also present in this document. These declarative statemnts are identified with a paragraph encoding scheme "**L*i*.*n*.*j***", where:

- L is the literal letter L (ell) signifying "level".
- *i* is a level number from 1 to 5 inclusive.
- *n* is a three letter category name and may either be QRY for "query" or SCH for "search".
- *j* is a requirement number (a natural number).

Following the text of a requirement may be a reference to an overarching requirement or use case from which it was derived; these references will be listed in parentheses. Indented text (in a smaller font) represent comments for additional insight or explanation of the requirement.

1.4 Controlling Documents

The following documents exert a controlling influence on this document:

[1] Planetary Data System (PDS) Level 1, 2 and 3 Requirements, 2006-08.

[2] Planetary Data System (PDS) 2010 Project Plan, 2010-02.

[3] Planetary Data System (PDS) 2010 System Architecture Specification, Version 1.0, 2010-02-14.

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

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

1.5 Applicable Documents

The following documents are applicable or associated with this document, but do not exert any controlling influence on this document:

[6] Distributed Infrastructure Design Team. *Search Service Summary*, 2009-05-31.

[7] Distributed Infrastructure Design Team. *PDS Search Services*, 2009-06-03.

1.6 Document Maintenance

The PDS anticipates that the design of the search service will evolve and mutate over time in order to accommodate ongoing updates to requirements and leverage pioneering technologies, and, similarly, this document should reflect such changes. This document will be limited to design content because the service specification content will be captured with the service documentation (including, but not limited to, an installation guide, an operations manual, etc.). This document will be stored within a configuration control system and any modifications shall be submitted to the PDS Management Council for approval.

2.0 SERVICE DESCRIPTION

The Search Service is a deployable server component that accepts queries for data and returns a set of matching results. Informally, queries pose the question of where certain data is, the search service satisfies that question with the answer of where to find such data. Figure 1 depicts the search service relative to the other components of PDS 2010.

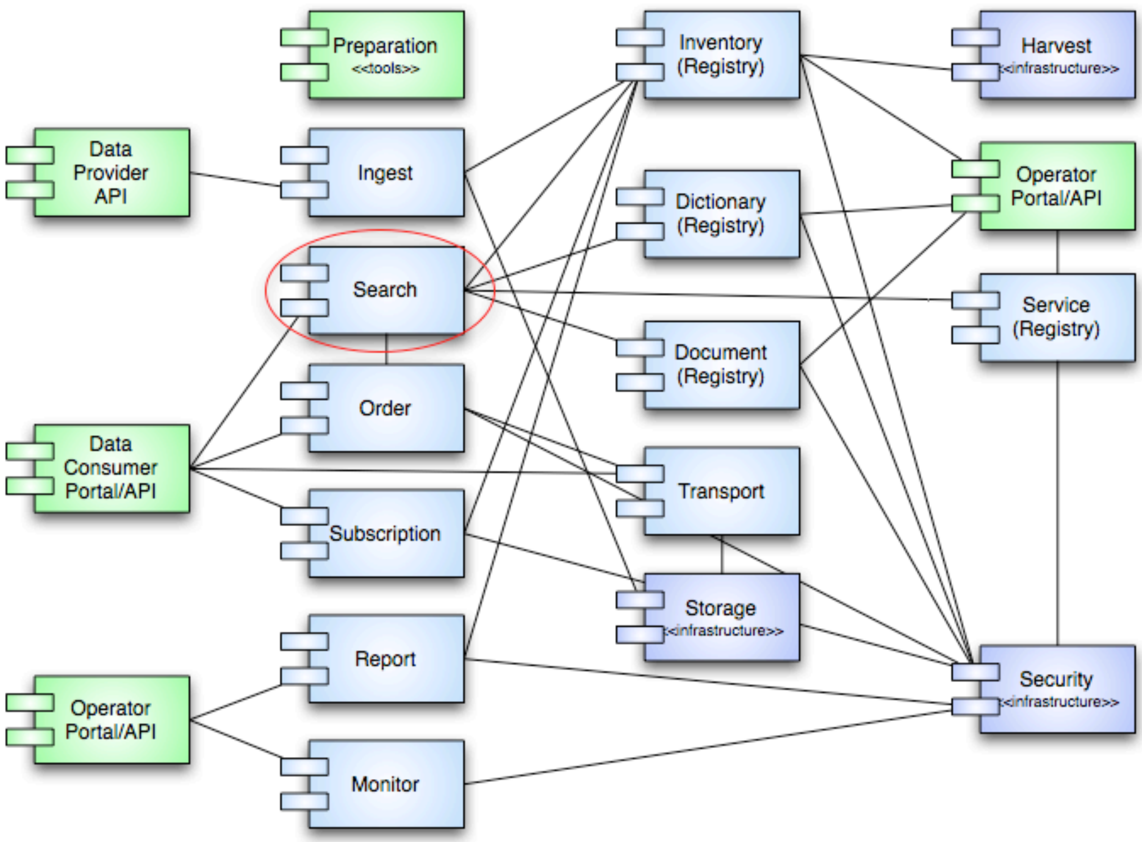

Figure 1. Context of the search service within the overall PDS 2010 component architecture.

A query in the sense of the search service means any type of declaration of desiderata, such as "Where are images of Mars taken by an infrared camera between 2009-01-01 and 2009-12-31 UTC?" or "What standards documents mentions the term 'interferometer'?" The desiderata necessarily require foreknowledge of what a particular Search Service curates in order for there to be any hope of an answer to a query.

A installation of the Search Service means a single running instance of the service, or—more specifically—a computing process that answers incoming TCP/IP connections on a set of network interfaces and on a single port. PDS nodes as well as other groups may deploy a Search Service on any compatible platform and use it to catalog particular sets of data and answer queries about that data. The PDS engineering node will run a "main" Search Service that provides answers to queries through the PDS public portal; the University of Maryland will run a backup service to perform the identical role should the engineering node's server fail. PDS nodes and other partners may run their own instances of the search service for discpline-specific needs.

Data in this sense means anything that the Search Service is capable of cataloging. It includes but is not limited to images, PDS-formatted images, PDS labels, radar altimetry raw data, engineering data records, derived temperature emissions, derived and reduced magnetometer datasets, plain text documents, PDF documents, XML files, HTML documents, and empty files.

Results to a query consist primarily of two things:

- First, a set of *identifiers* that tell where matching data may be found. Identifiers in this sense are Uniform Resource Identifiers (URIs) that, depending on user capability, may be resolved to get actual data.
- Second, a set of *metadata* that annotate each location with descriptors indicating its relevance and context. Such metadata serves to help guide the ultimate end-user in selecting the best result from a potentially large set of matching results.

2.1 Querying

Queries to the Search Service express a user's desiderata PDS data. The Search Service supports several formats of queries in order to better facilitate certain use cases. This section describes, in a high level, those formats.

2.1.1 Open

The open query format is the easiest to use and most familiar to regular users of the internet as it provides the identical capability that the intimately familiar Google search engine does. The open query format is merely a sequence of textual terms that the Service matches against its catalog (subject to certain processing and rules as described in the Implementation section). Matching results depend on the resources containing the terms mentioned.

Figure 2 shows an example of an existing “search engine” that allows the open format of query to be posed.

Figure 2. An example of a web-based "search engine" that supports "open" queries.

2.1.2 Guided

The guided query format (also known as "faceted query") is highly interactive and is familiar to internet users who frequent online shopping sites. Guided query presents a series of high-level organizational categories along with a set of terms in each category. Selecting an item constrains what items appear in other categories or may open up entirely new categories. This method of searching allows a broad overview of available data while enabling customized narrowing of matching results through user-selected progressive disclosure.

Figure 3 shows an example of guided search from a web-based “online shopping” site.

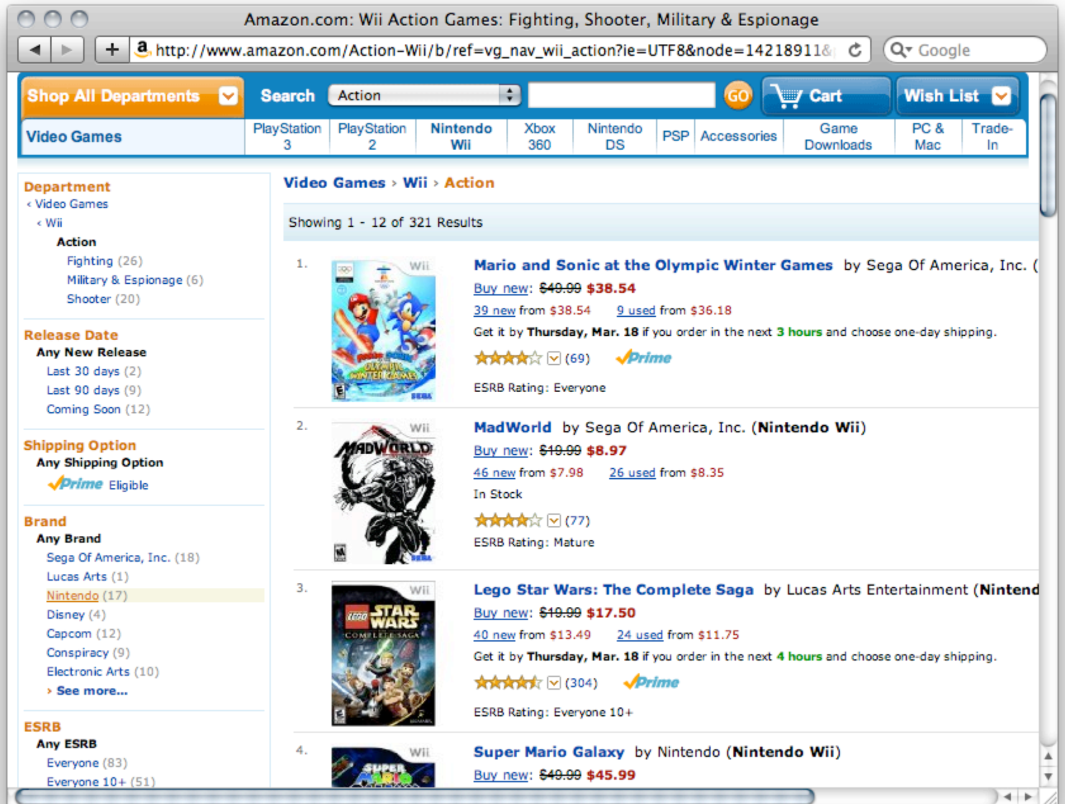

Figure 3. An example of a guided or "faceted" search in an online shopping context. Facets appear on the left of the window and change as categories are selected.

2.1.3 Constrained

The constrained query format enables the end user or client application to specify constraints on any of the searchable indexes within the search service. Such constraints may specify exact or relative values, such as a latitude equal to 23 degrees or a longitude greater than 32 degrees but less than 46 degrees, and so forth. This format requires that the client be intimately familiar with the searchable indexes, their data types, and the kinds of constraints that may be expressed on each. As a result, this query format is targeted towards expert users or client applications that can be programmed with such knowledge.

Figure 4 displays an example of a constrained search form with a (typical) plethora of controls.

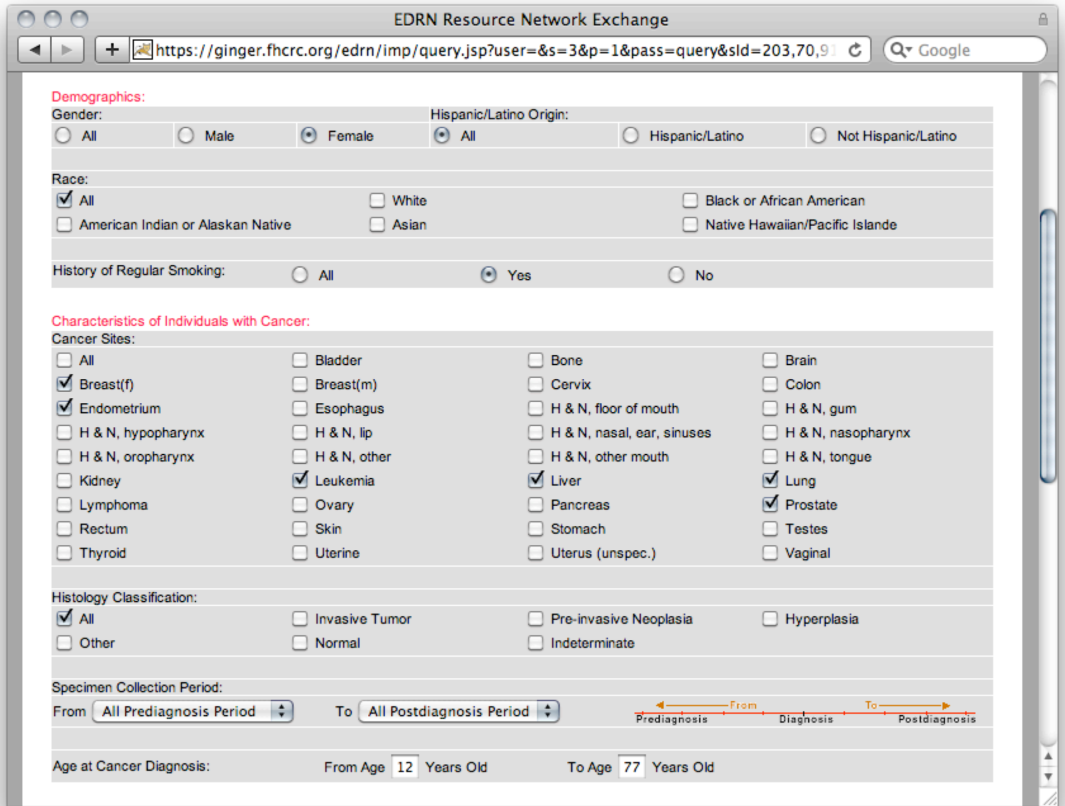

Figure 4. An example of a constrained search from a web-based medical application. Entry fields, check boxes, radio buttons, and drop-down menus are typical.

2.1.4 Exploratory

Exploratory queries are unlike the three others in that they begin with query results instead of an expression of desiderata. The purpose of exploratory queries are to provide results related to an existing set of results but were not purposely included in their original query formats. This enables the user to find interesting data that's "nearby" in terms of time, space, or other relative measure.

2.2 Results

Naturally, a search is nothing without results, which are what the end user is ultimately after. The search service provides a single format of results regardless of the format of the query. The results are a sequence of Uniform Resource Identifiers (URIs) that name and/or locate data that the search service has identified as being relevant to the query. The URIs are each reified with metadata that explain each match in order to provide the end user or client application with context that describes the data and why each URI was returned as a match.

The set of metadata is configurable for each installation of the search service in order to provide necessary context of results for a particular scientific or other application.

2.2.1 Ordering of Results

The search service offers a feature complementary to the acceptance of queries affecting the presentation of the sequence of matching URIs (and their metadata annotations). This feature enables the end user or client application to specify the the sort order of the results.

Unless otherwise specified, results are returned by relevance. The computation of relevance depends on several factors:

- For open searches, keyword frequency ratios to document size provide relative scoring of relevance between hits.
- Guided and constrained searches compute weights assigned to terms and values and leverage multidimensional metrics for relevance scoring.

2.2.2 Segmentation

For certain searches, the number of matching URIs may be enormous and exceed what an end user or client application can comfortably process. As an example, a constrained query for datasets produced after 1900 could well include over a million hits from a search service primed with every planetary, atmospheric, and geophysical dataset produced worldwide.

In order to overcome this problem the search service provides the capability for segmentation of hits. Along with the expression of the query's desiderata, the end user or client application may also specify

- A limit on the total number of acceptable hits, such as "1000 or less"
- A cardinal number identifying which of all of the hits to return, such as "the 42nd hit"
- A cardinality on the sequence of hits that include the number, such as "the 10 following hits following the 4nd"

These features enable segmentation of results for display, for example, over a series of web browser pages (where segmentation takes on the more specialized term "pagination"), or for batch processing of data by a scientific analysis program.

Figure 5 gives an example of web-based pagination controls.

Figure 5. Navigation controls typical of pagination from a web-based content management system.

3.0 USE CASES

This section describes scenarios of the use of the search service in order capture the functional requirements. Use cases are a superior tool for capturing and describing requirements by explaining actual, specific scenarios of end user (or client application) goals within the system.

Figure 6 shows a UML diagram of the use cases that depict the requirements of the search service.

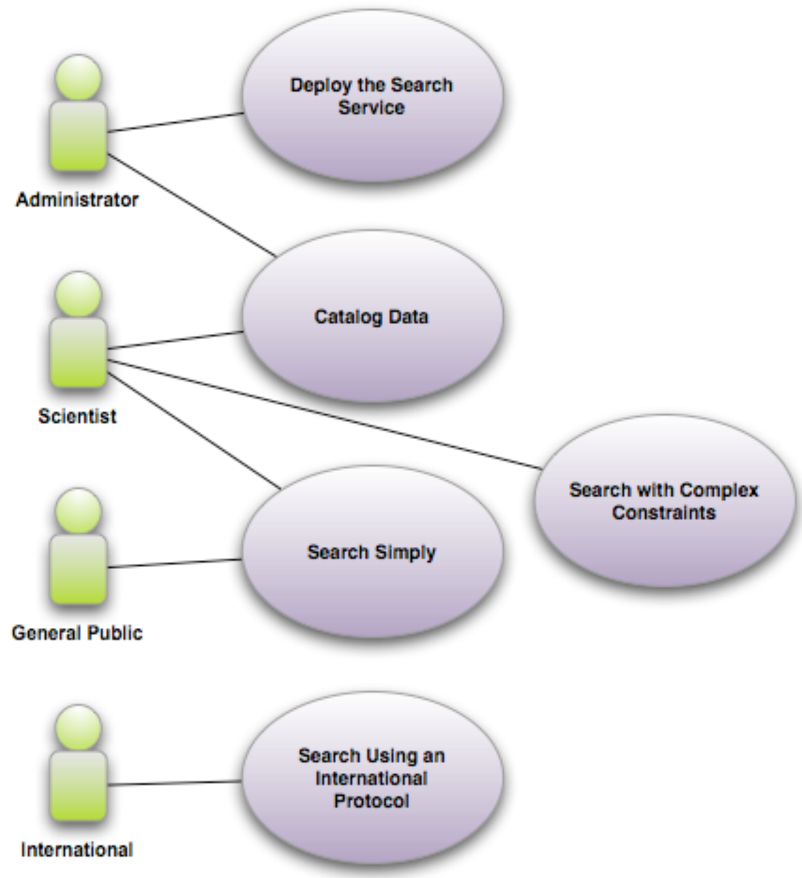

Figure 6. Use case diagram capturing requirements of the search service.

The remainder of this section gives an overview of each of the actors in the use cases, followed by a description of each use case.

3.1 Actors

This section gives an overview of the actors involved in the use cases. Actors initiate interaction with the search service in order to achieve a goal.

3.1.1 General Public

The general public's interest in scientific data may be driven by curiosity, a desire to learn, a need for new desktop background, or other reason. Regardless of the motivation, the general public as an actor comes to the search service without a comprehension of cataloged data, its organization, or its internal structure.

3.1.2 Scientist

Planetary, astrophysical, atmospheric, and other scientists require data to complete their work. We may subdivide scientists by their experience with the kinds of data that may be discovered via the search service. Novice scientists may not be intimate with the organization or structure of data, while advanced scientists (including those at PDS discipline nodes) could make "power queries"; the search service accommodates both.

3.1.3 International

Another actor we identify for the search service is the international user. Planetary research is not the exclusive domain of NASA. ESA, ISRO, JAXA, and a growing number of other international agencies are conducting and innovating in the field and require data from the search service in internationally agreed upon formats.

3.1.4 Administrator

A final actor is the system administrator. This person is responsible for instantiating and deploying the search service in a particular system context.

3.2 Cases

The use cases in this section describe an actor's interaction with the search service from start to finish in order to accomplish a goal.

3.2.1 Deploy the Search Service

In this case, a system administrator deploys the search service, enabling an instance of it to become operational.

The basic course of events is as follows:

1. The system administrator unpacks the archive software distribution of the search service.
2. The system administrator configures the service source code for the target system, compiles, and installs the system.
3. The system administrator initially starts an instance of the search service.

4. The system administrator arranges for automatic startup of the search service on each target system boot.
5. The system administrator adjusts the available catalog indexes and metadata as necessary for the local application.
6. The system administrator arranges for instances of the registry service to catalog their contents with the new search service instance.

3.2.1.1 Additional Information

Version	0
Goal	Operationalize an instance of the search service.
Summary	The system administrator deploys the search service, making an instance of it operational.
Actors	System administrator
Stakeholders	End users and application developers
Notes	The most essential steps are the last two. The fifth step tells an instance of the search service what indexes to generate for cataloged content while the sixth actually causes the service to actually catalog content. Without these steps, the search service has no catalogs to answer queries for.

3.2.2 Catalog Data

In this case, an authorized user submits data to the search service to be cataloged, enabling it to be later searched for and discovered.

The basic course of events is as follows:

1. The user or client application accesses the search service's web-based catalog interface.
2. The search service confirms the authentication and authorization of the client with the security service.
3. The user or client application sends an identifier (URI) of the data and the data to be cataloged to the search service.
4. The search service analyzes the data and and updates its indexes based on the analysis.

3.2.2.1 Additional Information

Version	0.

Goal Update the indexes maintained by the search service with presented data.

Summary The end user or client application submits data to be cataloged to the search service which indexes it.

Actors Scientist.

Stakeholders All.

Notes This use case must occur before any searches can return results.

3.2.3 Search Simply

In this case, a user (any kind) visits the search service's human-consumable web interface and enters an open search (as described in 2.1.1). Matching results are displayed 20 at a time in relevance order with result segmentation navigation controls, i.e., "pagination" controls.

The basic course of events is as follows:

1. User visits search service's human-consumable web page.
2. User types terms into prominent open text search box.
3. User presses Search button.
4. User views results.

Optionally, if presented with more than 20 results, the user may press the "Next Page" link to see the next segment of matching results.

3.2.3.1 Additional Information

Version 0.

Goal Support a simple search.

Summary The user executes a search using free text keywords.

Actors Any.

Stakeholders Any.

Notes None.

3.2.4 Search with Complex Constraints

In this case, a Scientist visits the search service's human-consumable web interface and selects the Advanced search feature. After s/he fills out the selected values for various constraints and results ordering and segmentation, s/he presses the Search button. Matching results are displayed a page at a time in the selected order with result segmentation navigation controls, i.e., "pagination" controls.

The basic course of events is as follows:

1. Scientist visits search service's human-consumable web page.
2. Scientist clicks the Advanced Search link.
3. Scientist specifies search criteria by entering constraints on available indexes.
4. Scientist specifies results presentation by selected results sort ordering and number of results per page.
5. User views results.

3.2.4.1 Additional Information

Version 0

Goal Support a complex constrained search with a potentially complex set of specifications.

Summary The user executes a search using free text keywords.

Actors Any.

Stakeholders Any.

Notes None.

3.2.5 Search Using an International Protocol

In this case, an external application developed by an international third party makes use of the web-based programmatic interface of the search service that is compatible with PDAP. This interface, recommended by the International Planetary Data Alliance, enables searches from both within and without the PDS from applications that can "speak" PDAP.

The basic course of events is as follows:

1. The external application formulates a PDAP query.
2. The external application sends the query to the search service.
3. The search service returns a VOTable of matching results.
4. The external application digests the VOTable in a way appropriate to it.

3.2.5.1 Additional Information

Version 0

Search Service SRD/SDD

Goal Support queries via PDAP.

Summary The search service accepts PDAP queries (keyword/value pairs) and returns PDAP responses (XML-formatted VOTables).

Actors International.

Stakeholders Any.

Notes None.

4.0 REQUIREMENTS

During the architecture definition phase of PDS 2010, we decomposed the system into various elements that included the search service, which was derived from requirement 3.1 of the Level 1, 2, and 3 Requirements document [1]. Although use cases are an excellent tool for capturing requirements, we re-state the requirements of the search service in an alternative form in this section.

The following system requirements from [1] relate to the search service and are quoted here for convenience:

3.3.1 PDS will develop and maintain online interfaces allowing users to search the archive.

3.3.2 PDS will develop and maintain online interfaces for discipline-specific searching

3.3.3 PDS will allow products identified within a search to be selected for retrieval.

4.1 Level 4 Requirements

The level 4 requirements in PDS represent subsystem, component, or tool requirements, but at a high level. The following list of level 4 requirements are relevant to the search service:

L4.QRY.1 The system shall provide the capability to search for and identify artifacts across the entirety of the PDS (3.3.1).

L4.QRY.2 The system shall provide the capability to search for and identify artifacts within a single discipline (3.3.2).

4.2 Level 5 Requirements

The level 4 requirements in PDS represent subsystem, component, or tool requirements as the level 4 requirements do, however they do so at a far more detailed level. The following list of level 5 requirements are relevant to the search service:

L5.SCH.1 The service shall be deployable to its target system in a way that's comfortable and familiar to system administrators (L4.QRY.1, L4.QRY.2).

L5.SCH.2 The service shall provide an attractive user interface for entering of queries and display of search results accessible from a standards-compliant web browser (L4.QRY.1, L4.QRY.2).

L5.SCH.3	The service shall degrade gracefully on browsers that lack modern features and not depend on them for operation (L4.QRY.1, L4.QRY.2).
L5.SCH.4	The service's browser-based user interface shall be Section 508 compliant and adhere to WCAG level A (or better) standards for accessibility.
L5.SCH.5	The service shall provide a programmatic interface for entering of queries and return of search results that communicates over HTTP for use by client applications developed by PDS, PDS nodes, and others (L4.QRY.1, L4.QRY.2).
L5.SCH.6	The service shall provide a programmatic interface for cataloging of data that communicates over HTTP for use by instances of the registry service or by other applications (L4.QRY.1, L4.QRY.2).
L5.SCH.7	The service shall support searching by providing a sequence of open text keywords (L4.QRY.1, L4.QRY.2).
L5.SCH.8	The service shall support searching by providing a series of values for constraints on specified indexes (L4.QRY.1, L4.QRY.2).
L5.SCH.9	The service shall support narrowing of additional indexes results based on specifications of terms and/or values on indexes (L4.QRY.1, L4.QRY.2).
L5.SCH.10	The service shall provide results related to a set of presented results (L4.QRY.1, L4.QRY.2).
L5.SCH.11	The service shall support the ordering of results based on specified criteria including relevance and specified indexes (L4.QRY.1, L4.QRY.2).
L5.SCH.12	The service shall provide results to a search as a sequence of matching URIs to resources that contain search desiderata (L4.QRY.1, L4.QRY.2).
L5.SCH.13	The service shall annotate each URI of a result with metadata describing the URI (L4.QRY.1, L4.QRY.2).
L5.SCH.14	The service shall provide a programmatic interface for entering of queries and return of search results that communicates over HTTP and adhering to international standards for data search and retrieval (L4.QRY.1, L4.QRY.2).

L5.SCH.15 The service shall support configuration on the kinds of indexes maintained on cataloged data, including indexes that differ by data type, by data conversion, by index generation methodology, and by metadata maintenance for result annotation (L4.QRY.1, L4.QRY.2).

L5.SCH.16 The service shall enable cataloging of data only by selected principals by authenticating and authorizing them through the security service (L4.QRY.1, L4.QRY.2).

5.0 DESIGN PHILOSOPHY, ASSUMPTIONS, AND CONSTRAINTS

The intent of the search service is to make it easy to find PDS data. The search service itself does not contain PDS data, but instead *catalogs* them by being presented with a set of PDS data to process. Processing examines the data and adds to the search service's indexes. Queries scan the search service's indexes of previously cataloged data to find potential matches. The service returns a sequence of URIs to the originally processed data that match, as well as any additional metadata kept in the indexes.

Due to the variability of discipline-specific data, the search service does **not** enforce a specific set of catalog indexes. Rather, it enables each installation to configure the kinds of indexes needed to support searches. Indexes may:

- Be of various data types (such as date & time or floating point or HTML text).
- Require conversion or processing (such as removal of HTML tags, or ignoring of "stop words" like definite articles, prepositions, and the like).
- Display in search results as metadata annotations.

The search service does not *ingest* or *store* data, but instead maintains indexes *about* the data and *identifiers* (in the form of URIs) to them. Results of a search are the identifiers, plus metadata annotations about the identifiers. It is then up to the client program or end user to examine the metadata annotations and choose whether to retrieve the identified data. We expect that in most if not all cases the URIs will take the form of URLs that enable retrieval of data directly by virtually any browser or with some rudimentary data movement software.

6.0 ARCHITECTURAL DESIGN

The architectural design includes the components that comprise the search service as well as both internal and external interfaces with the service and the data model of which it takes advantage.

6.1 Service Architecture

Figure 7 shows a UML class diagram that depicts the architecture of the search service.

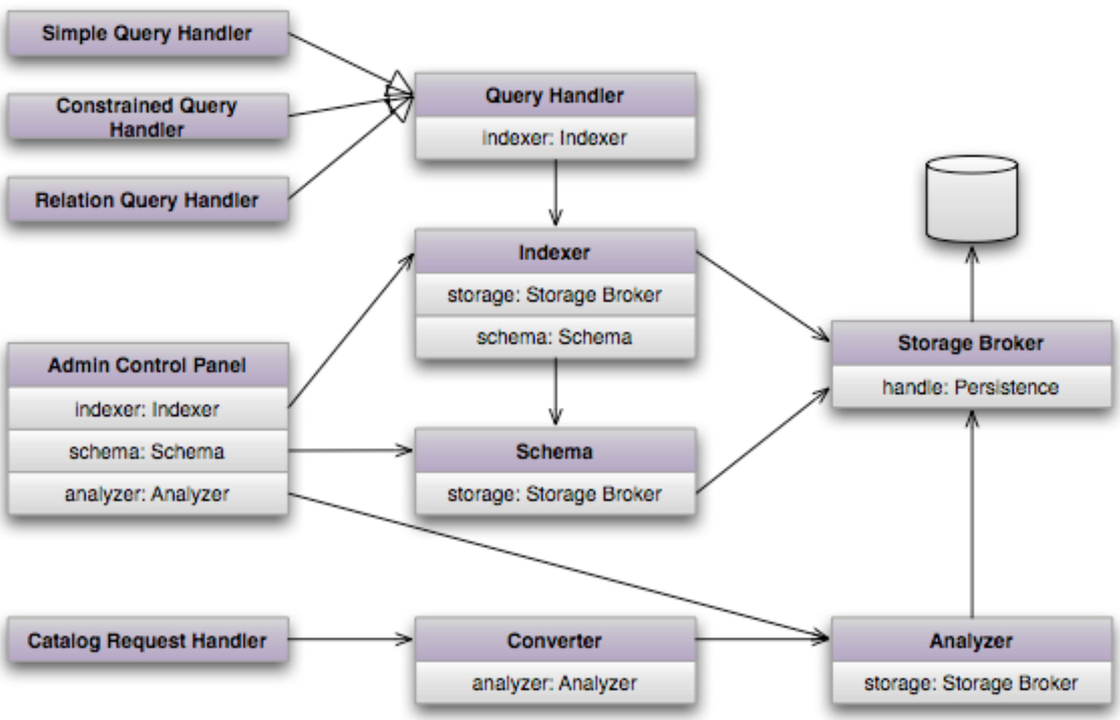

Figure 7. Overall architecture of the search service.

6.2 External Interface Design

The external interfaces for the search service are REST-based or REST-like. For now, they are captured in the OODT Wiki. Please see the following for details on the REST-based or REST-like API:

http://oodt.jpl.nasa.gov/wiki/display/pdscollaboration/Search+Interface

6.3 Internal Interface Design

TBD.

6.4 Data Model

TBD.

7.0 ANALYSIS

TBD.

8.0 IMPLEMENTATION

TBD.

9.0 DETAILED DESIGN

TBD.

9.1 Server Process

TBD.

9.2 Request Dispatching

TBD.

APPENDIX A ACRONYMS & ABBREVIATIONS

The following acronyms made an appearance in this document:

ESA	European Space Agency
HTML	Hypertext Markup Language
IP	Internet Protocol
ISRO	Indian Space Research Organization
JAXA	Japan Aerospace Exploration Agency
JPL	Jet Propulsion Laboratory
NASA	National Aeronautics and Space Administration
OODT	Object Oriented Data Technology
PDAP	Planetary Data Access Protocol
PDF	Portable Document Format
PDS	Planetary Data System
QRY	Query component
REST	Representational State Transfer
SCH	Search component
SDD	Software Design Document
SRD	Software Requirements Document
TBD	To Be Determined
TCP	Transport Control Protocol
UML	Unified Modeling Language
URI	Uniform Resource Identifier
UTC	Coordinated Universal Time
VOTable	Virtual Observatory Table
WCAG	Web Content Accessibilty Guidelines
XML	Extensible Markup Language