# Service Design

PDS 2010 System Review
March 22-24, 2010

Sean Hardman

# Topics

- Overview
- Design Decisions
- Services
- Wrap Up

# Overview

- Currently focusing on phase I components (Registry, Harvest, Security and Report).
  - Have also started looking into Search.
- Use cases, requirements and design are captured in a single document for each service.
- Requirements (Level 4 and 5) derived from Level 3 requirements (where applicable), use cases and existing functionality in the current system.

# Design Principles

## (Derived from Architectural Principles)

- Introduce common software, where appropriate, that is extensible to accommodate discipline-specific needs.
- Isolate technology choices from functionality to facilitate future upgrades.
- Minimize tight-coupling between components to facilitate phased deployment and component replacement.
- Simplify component and user interfaces to facilitate adoption and use of software.
- Utilize standard, open source and COTS solutions where appropriate.

# Design Goals

## (Derived from PDS 2010 Drivers and Goals)

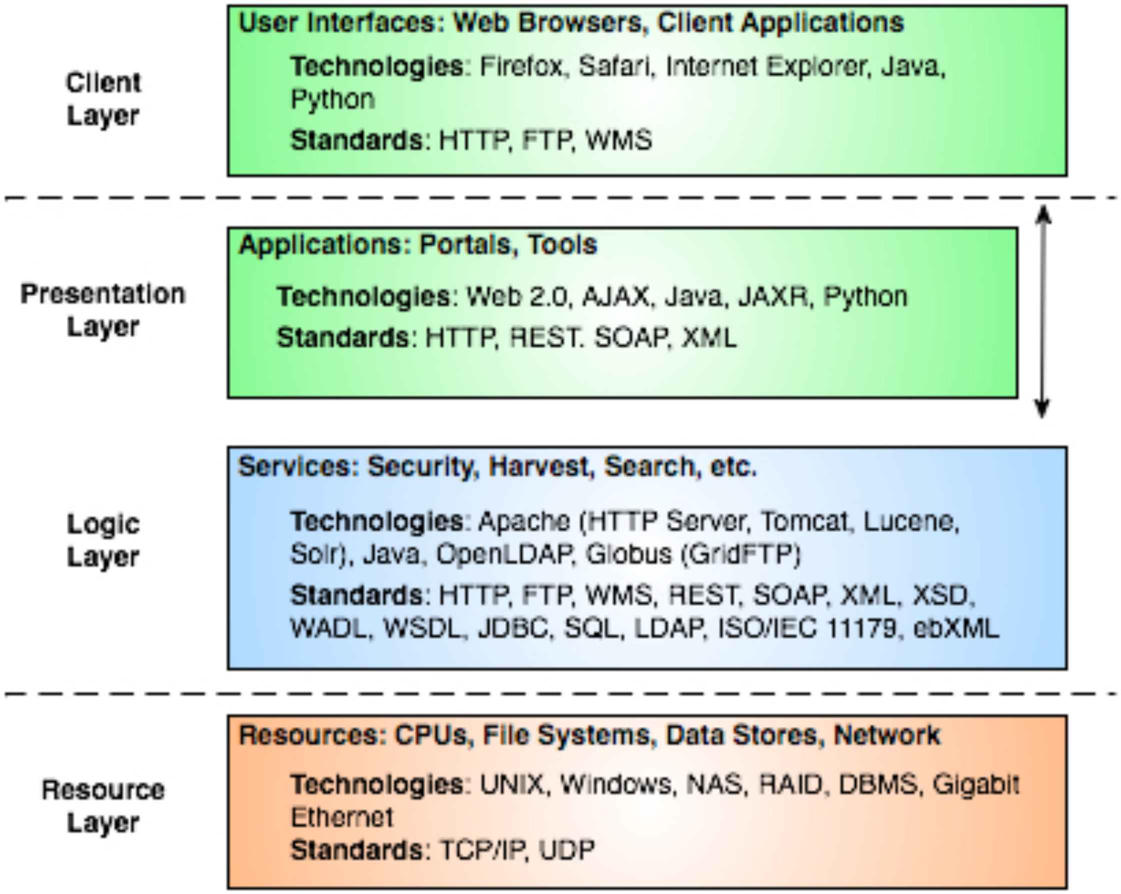
- Improve ingestion efficiency.
- Facilitate tracking and improve integrity of the archive.
- Facilitate data product search across nodes.
- Improve delivery of data to users and deep archive.
- Increase integration of software services across the Nodes and the system as a whole.
- Keep it simple.

# Design Considerations

- Local governance for data and metadata within the PDS system is retained by the Discipline Nodes.
- Current and proposed data volumes along with limited bandwidth suggest that the system should eliminate unnecessary movement of data.
- Majority of effort undertaken with in-guide funding, dictates a flexible and phased approach for development and deployment of the system.

# Technologies and Standards

## (Focus on Open Source Where Appropriate)



Client Layer

User Interfaces: Web Browsers, Client Applications

**Technologies**: Firefox, Safari, Internet Explorer, Java, Python

**Standards**: HTTP, FTP, WMS

Presentation Layer

Applications: Portals, Tools

**Technologies**: Web 2.0, AJAX, Java, JAXR, Python

**Standards**: HTTP, REST. SOAP, XML

Logic Layer

Services: Security, Harvest, Search, etc.

**Technologies**: Apache (HTTP Server, Tomcat, Lucene, Solr), Java, OpenLDAP, Globus (GridFTP)

**Standards**: HTTP, FTP, WMS, REST, SOAP, XML, XSD, WADL, WSDL, JDBC, SQL, LDAP, ISO/IEC 11179, ebXML

Resource Layer

Resources: CPUs, File Systems, Data Stores, Network

**Technologies**: UNIX, Windows, NAS, RAID, DBMS, Gigabit Ethernet

**Standards**: TCP/IP, UDP

# Topics

- Overview
- Design Decisions
- Services
- Wrap Up

# Design Decisions

- Service Interfaces
  - Implement REST-based interfaces where appropriate.
- Grammar Representation
  - Adopt XML as the data language/grammar where appropriate.

# Service Interfaces
## REST vs. SOAP

- REST is generally considered lightweight and simpler to implement than SOAP.
- Where web-based service interfaces are planned, a REST-based interface will be implemented.
  - REST stands for Representational State Transfer and is a style of software architecture based on HTTP.
  - This decision applies to the Registry and Search services.
- Other services that integrate COTS or open source solutions will utilize their provided interfaces (e.g., LDAP).

# Grammar Representation

## XML vs. ODL

- XML has been adopted by a number of science data and archive systems for capturing metadata.
- XML offers a larger community and a stable standard to build on for the future.
- The long-term benefits in cost reduction for development and maintenance outweigh initial transition costs.
- Although no official survey has been performed, it is believed that data providers and consumers would applaud the move to XML.

# Topics

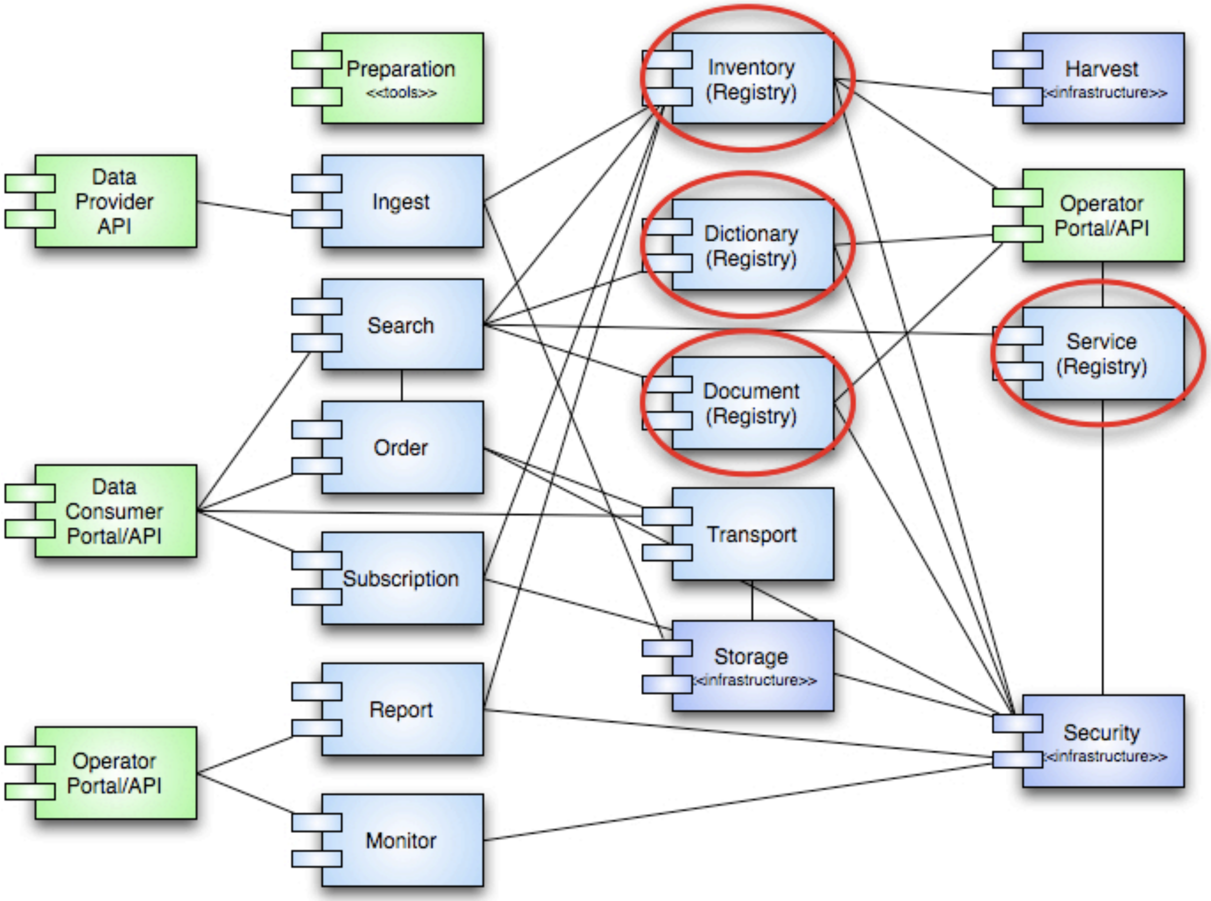
- Overview
- Design Decisions
- Services
- Wrap Up

# Registry Service

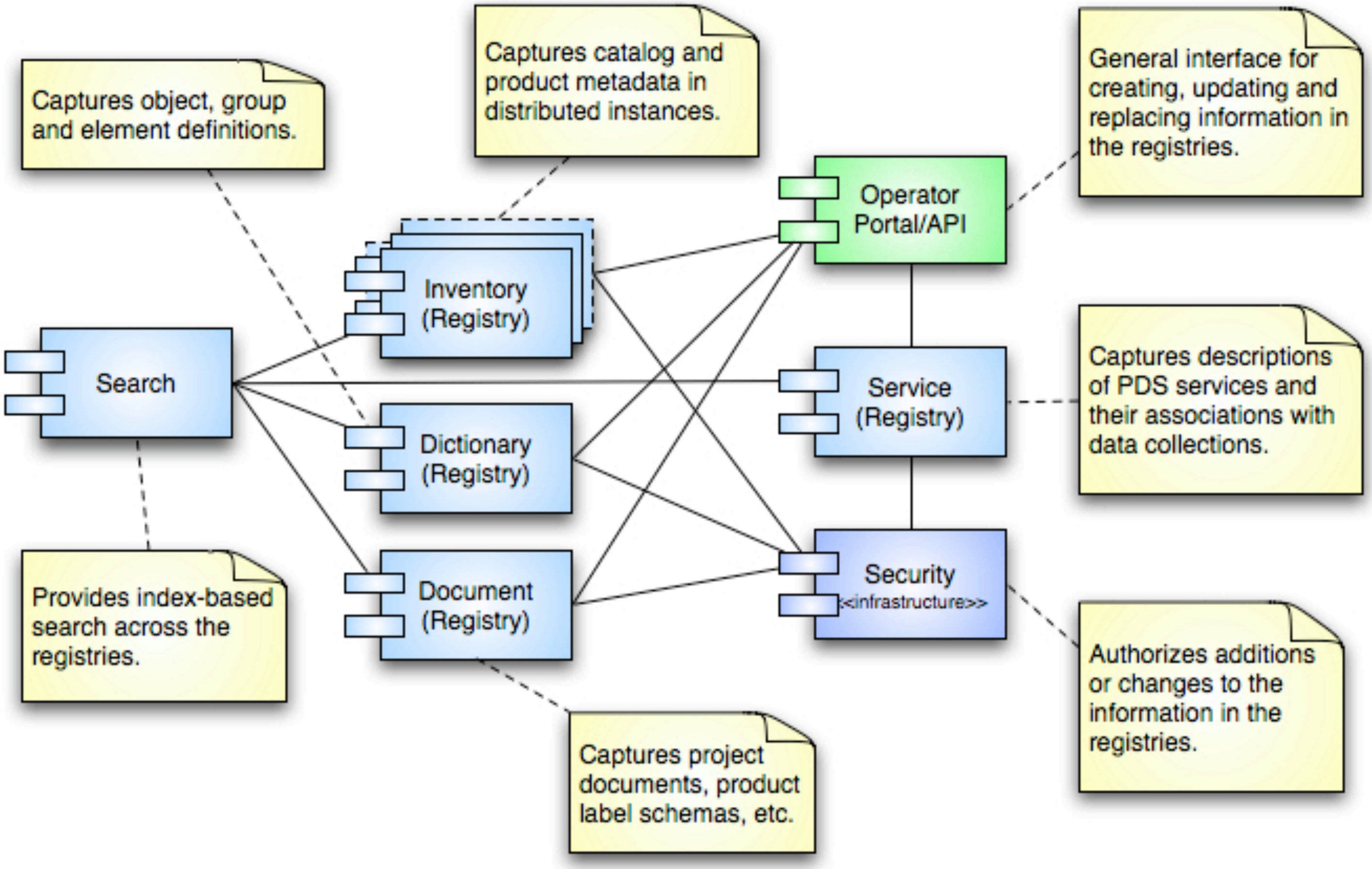
# Overview

- The design for PDS 2010 calls for multiple registries that are distinguished by their content.
  - These are the Inventory, Dictionary, Document and Service instances.
- The Registry service provides a common implementation for each of these services.
- Design based on the CCSDS effort:
  - Registry and Repository Reference Model, Draft White Book.
- The CCDSD effort leverages the ebXML standard.

# Context


Preparation
<<tools>>
Inventory
(Registry)
Harvest
<<infrastructure>>
Data
Provider
API
Ingest
Dictionary
(Registry)
Operator
Portal/API
Search
Service
(Registry)
Document
(Registry)
Order
Data
Consumer
Portal/API
Transport
Subscription
Storage
<<infrastructure>>
Report
Security
<<infrastructure>>
Operator
Portal/API
Monitor

# Detailed Context

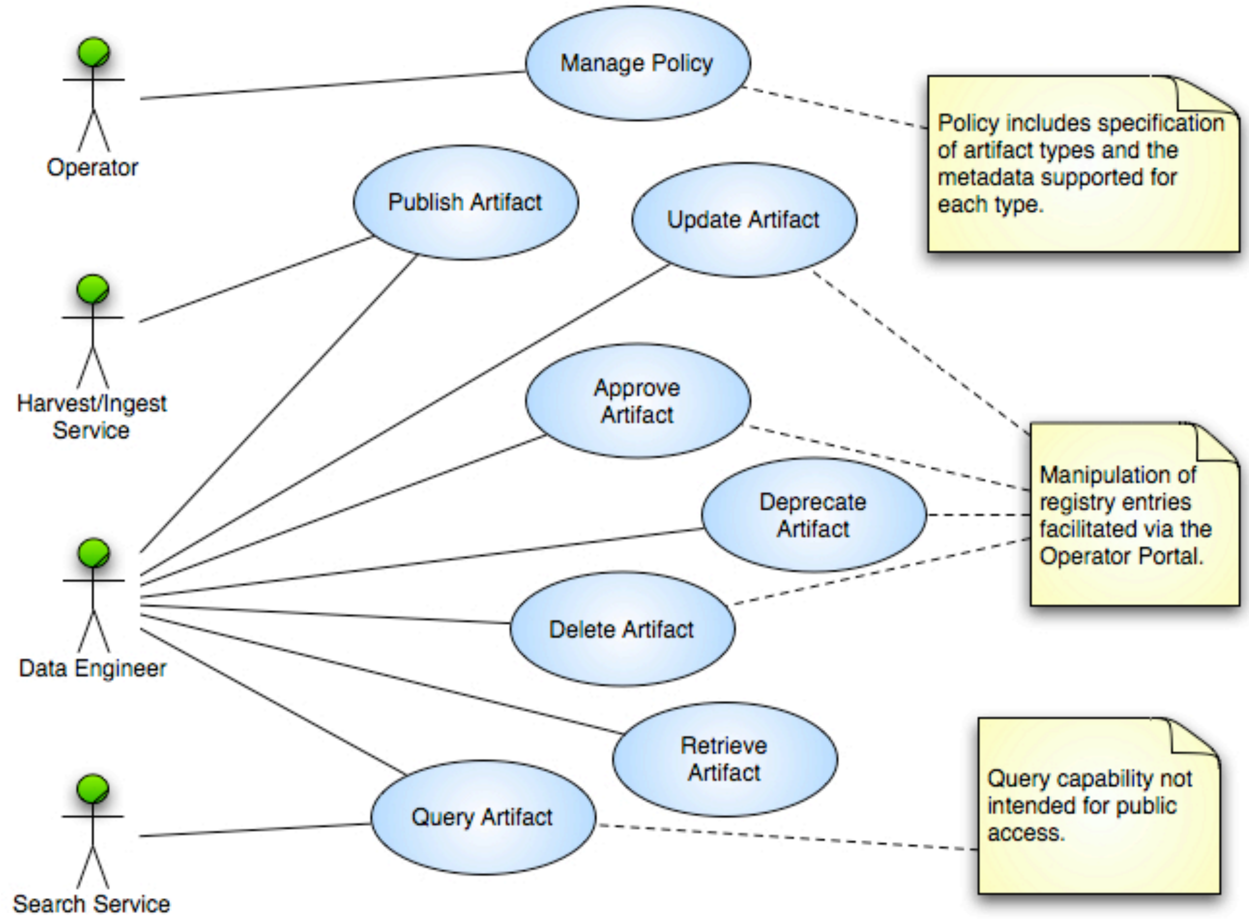

Captures object, group and element definitions.
Captures catalog and product metadata in distributed instances.
General interface for creating, updating and replacing information in the registries.
Operator Portal/API
Inventory (Registry)
Search
Service (Registry)
Captures descriptions of PDS services and their associations with data collections.
Dictionary (Registry)
Document (Registry)
Security <<infrastructure>>
Provides index-based search across the registries.
Authorizes additions or changes to the information in the registries.
Captures project documents, product label schemas, etc.

# Registry Entry Types

- Metadata Entry
  - Captures a description of a non-digital object.
  - Examples include missions, instruments, etc.
- Digital Object Entry
  - Captures a description of a digital object that is referenced by an URI.
  - Examples include products, product label schemas, etc.
- Relationship Entry
  - Captures an association between registered objects.
  - Associations are also typed (e.g., member of).

# Use Cases

## (Taken from CCSDS Specification)

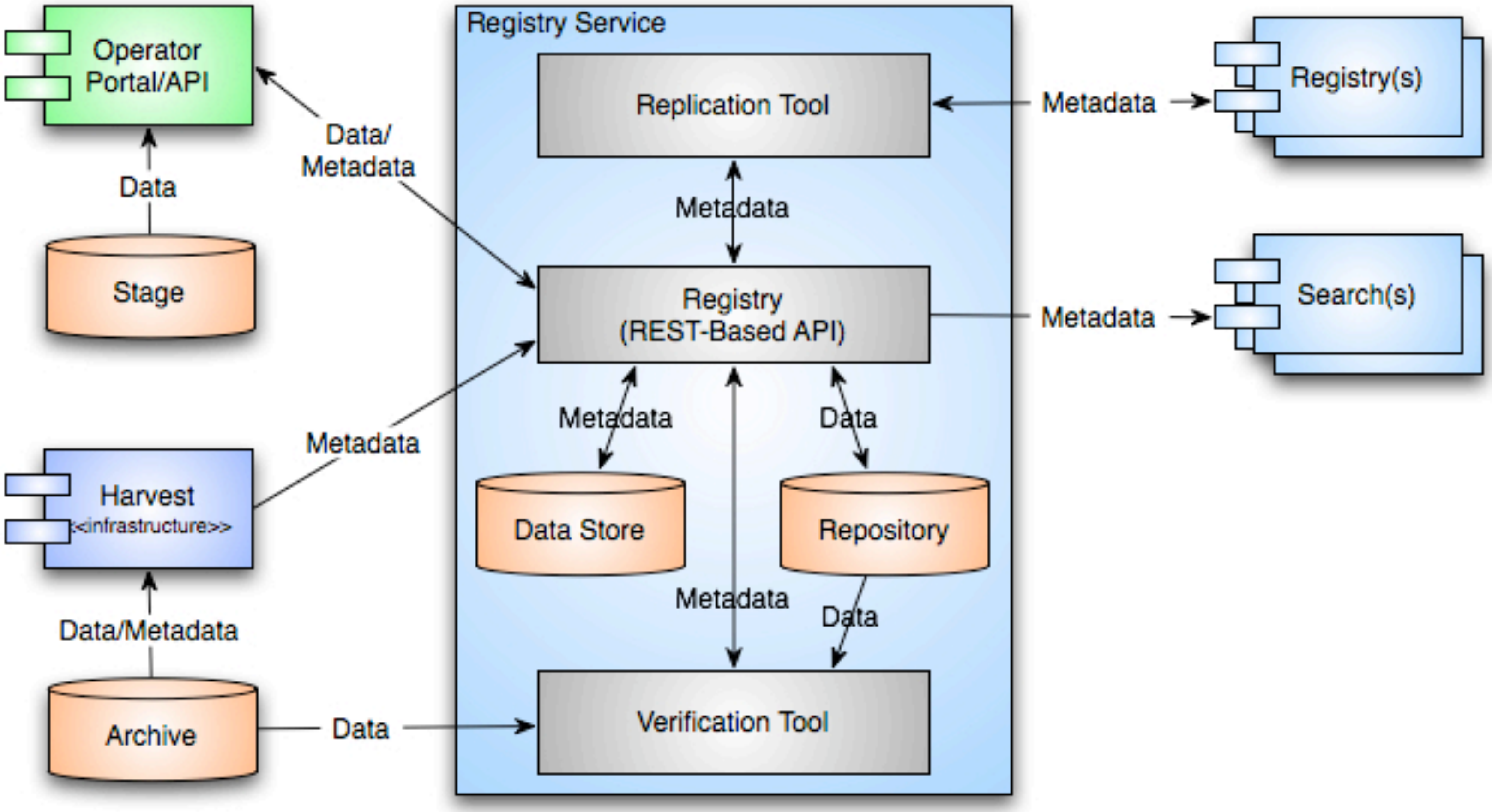


Policy includes specification of artifact types and the metadata supported for each type.

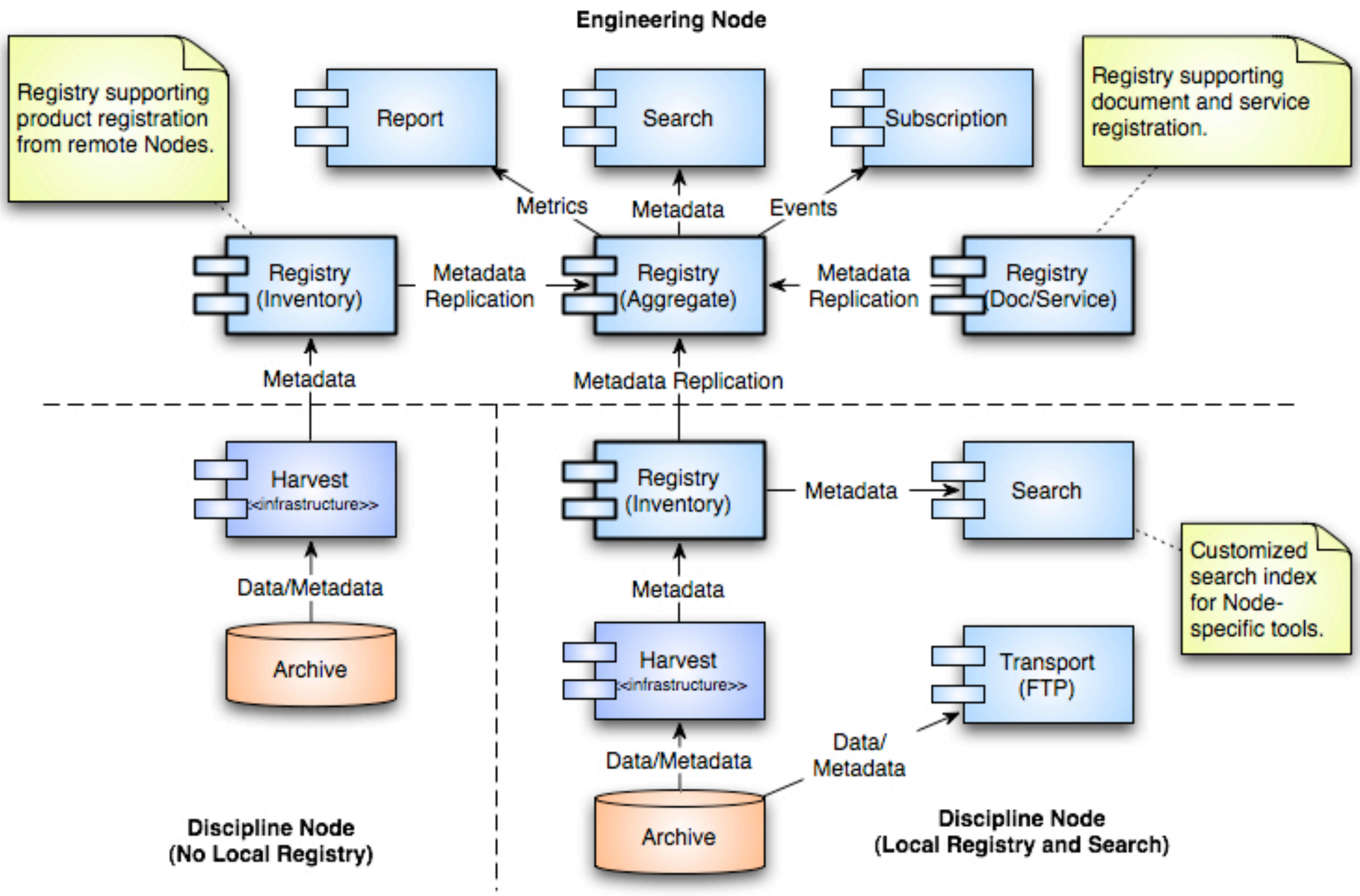
Manipulation of registry entries facilitated via the Operator Portal.

Query capability not intended for public access.

# Architecture
## (Stand-Alone Registry Capabilities)



# Architecture
## (Deployment Scenarios)



# Analysis

- Initially looked at related standards (UDDI and ebXML) and determined that ebXML better supports federated registries.
- Evaluated two software packages:
  - freebXML
    - Open Source package no longer maintained
    - Would require further development and upgrade
  - WellGEO RegRep from Wellfleet Software Corporation
    - COTS package developed by the main author of freebXML
    - Met requirements but not mature and exceeded budget constraints
- After these two evaluations, the team decided to implement the CCSDS reference model.

# Implementation

- Currently implementing the REST-based registry interface with these characteristics:
  - A URL assigned to every resource
  - Formulate URLs in a predictable manner
  - Use HTTP methods for actions on a resource (GET, POST, PUT and DELETE)
  - Leverage HTTP protocol headers and response codes where applicable
- Goals include:
  - Keep the service simple and refrain from adding too much functionality
  - Allow messaging in the form of XML or JavaScript Object Notation (JSON)
  - Allow for extensibility as new types of artifacts are defined

# Implementation Examples

- This interface delegates all functions involving an artifact.

  http://pds.nasa.gov/services/registry/artifacts/

  - GET: Retrieves a paged list of artifacts from the registry.
  - POST: Publishes an artifact to the registry.
- This interface acts on a specific artifact:

  http://pds.nasa.gov/services/registry/artifacts/{version}/{lid}/

  - GET: Retrieves an artifact from the registry.
  - PUT: Updates the artifact in the registry with the given artifact.

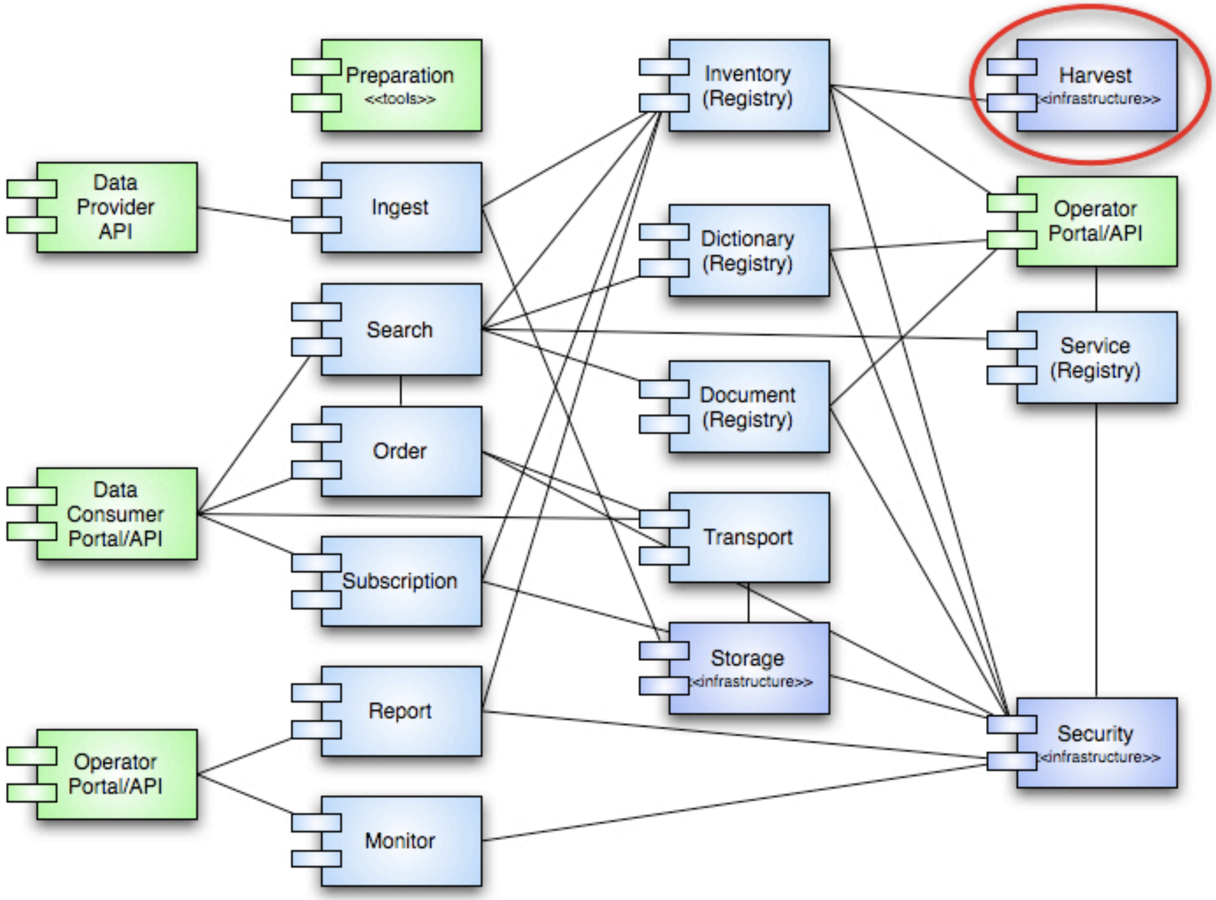
# Harvest Tool

# Overview

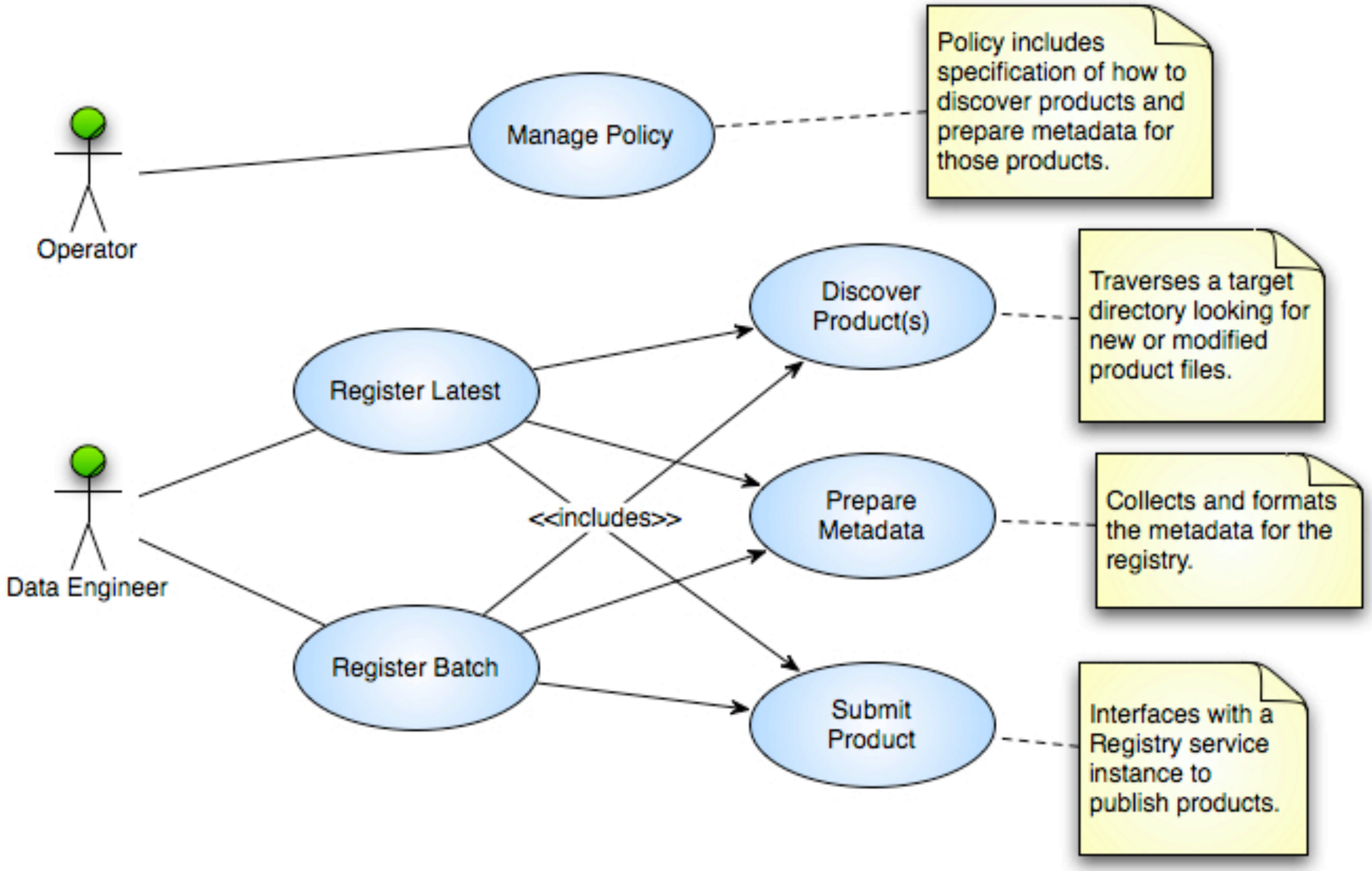
- Provides functionality for capturing and registering product metadata.
- Crawler-based tool configurable to support PDS3 and PDS4 archive directory structures.
- Designed to integrate well with existing Node operations.
- Provides the first line of metadata harvesting within the system in order to facilitate tracking of and access to products.

# Context

## (Infrastructure Component Limited to Internal Interfaces)



# Use Cases



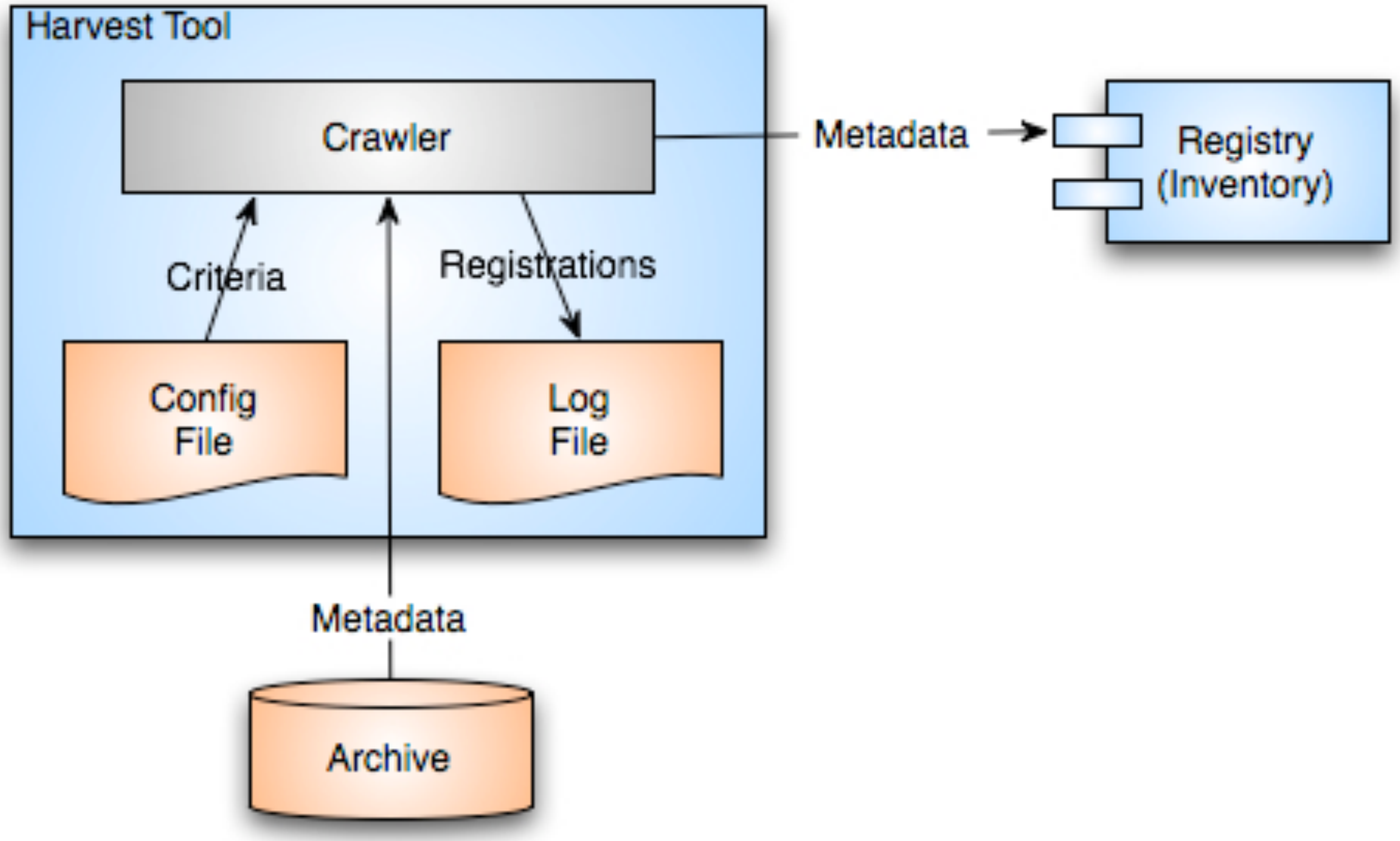
Policy includes specification of how to discover products and prepare metadata for those products.

Traverses a target directory looking for new or modified product files.

Collects and formats the metadata for the registry.

Interfaces with a Registry service instance to publish products.

# Architecture



# Implementation

- The tool offers a command-line interface and supports execution from a scheduler application (e.g., cron).

- Plan to investigate two operational crawler-based implementations from JPL as a starting point for this tool. They come from:
  - Physical Oceanography Distributed Active Archive Center (PO.DAAC)
  - Orbiting Carbon Observatory (OCO) / NPP Sounder PEATE

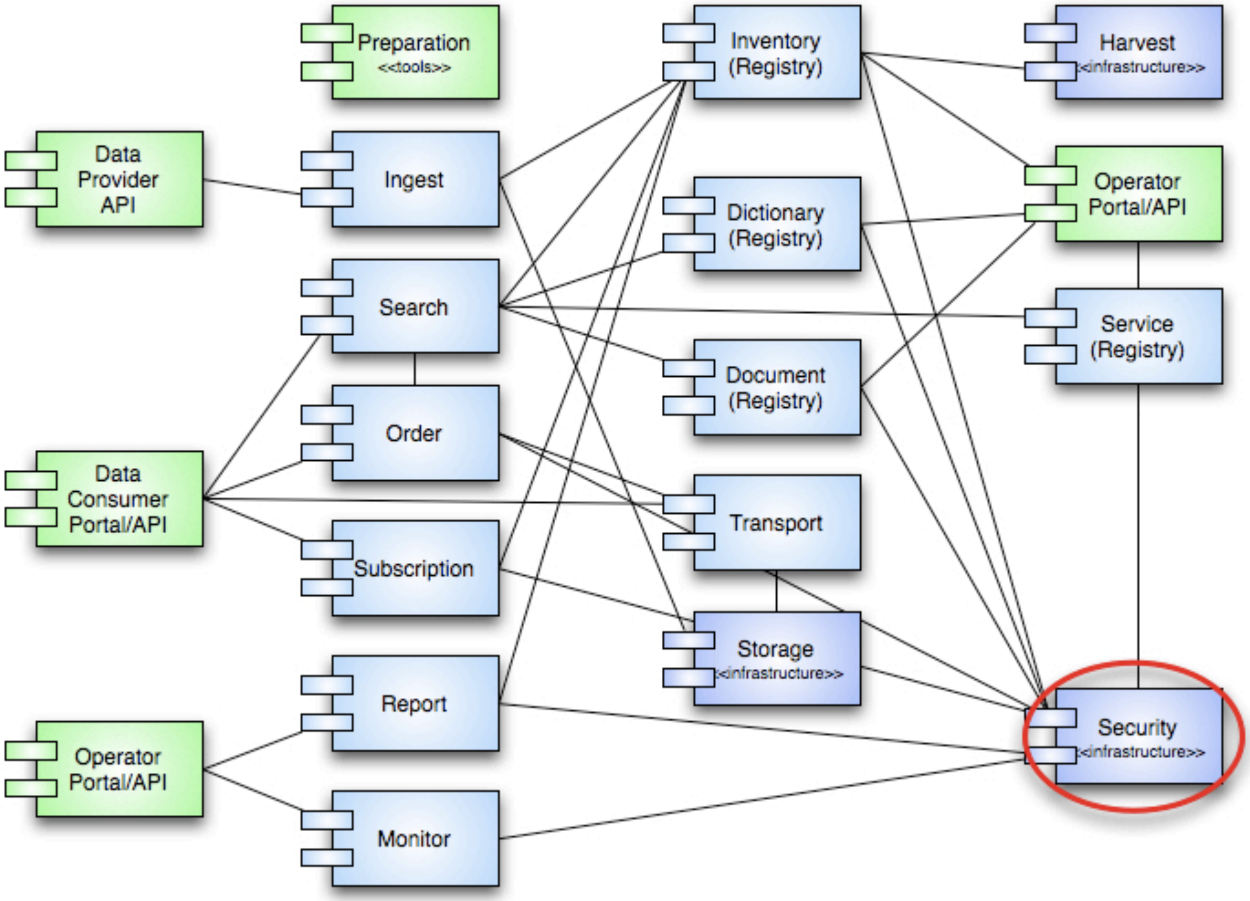
# Security Service

# Overview

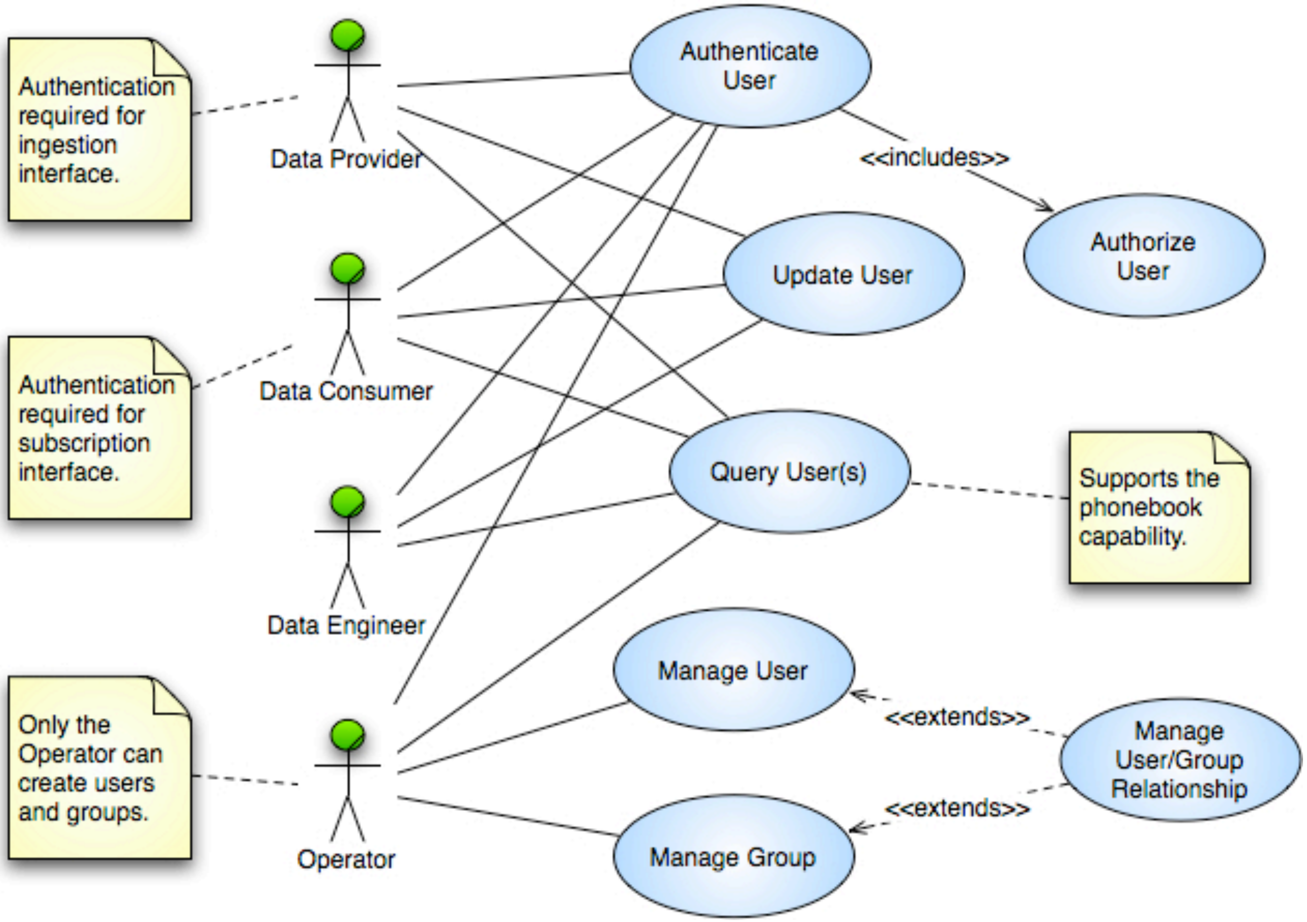
- Provides the authentication and authorization functions for the system.
- Control access to interfaces and services (e.g., Monitor, Report, Registry instances etc.).
- Also supports for the phone book (personnel directory) capability.
- Looking to satisfy the requirements with Open Source software supporting the Lightweight Directory Access Protocol (LDAP).
  - Using Sun's Open Single Sign-On (OpenSSO) package

# Context

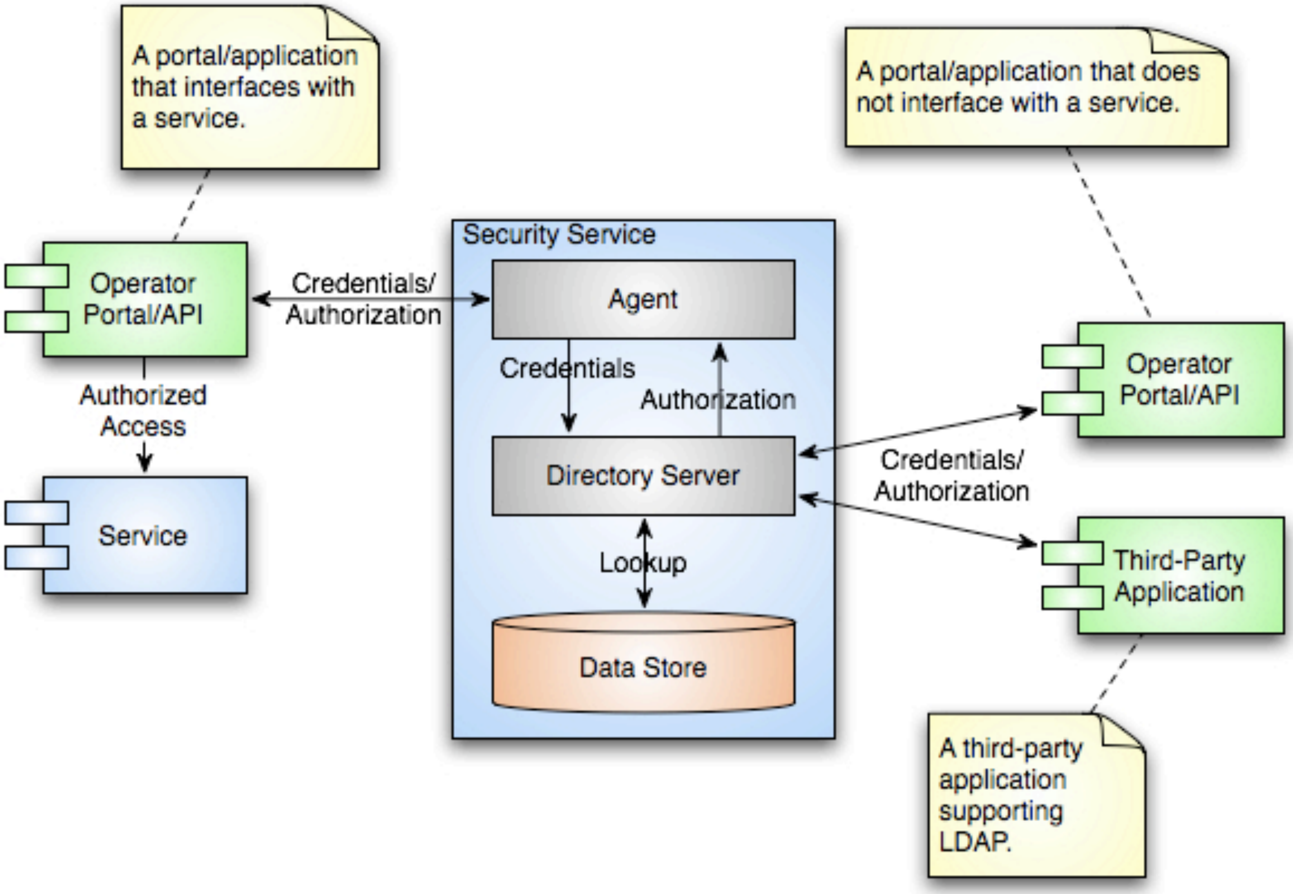
## (Infrastructure Component Limited to Internal Interfaces)



# Use Cases


Authentication required for ingestion interface.
Data Provider
Authentication required for subscription interface.
Data Consumer
Data Engineer
Only the Operator can create users and groups.
Operator
Authenticate User
<<includes>>
Authorize User
Update User
Query User(s)
Supports the phonebook capability.
Manage User
<<extends>>
Manage User/Group Relationship
<<extends>>
Manage Group

# Architecture
## (Support for Three Controlled Access Scenarios)


A portal/application that interfaces with a service.
Operator Portal/API
Authorized Access
Service
Credentials/ Authorization
Security Service
Agent
Credentials
Authorization
Directory Server
Lookup
Data Store
A portal/application that does not interface with a service.
Operator Portal/API
Credentials/ Authorization
Third-Party Application
A third-party application supporting LDAP.

# Implementation

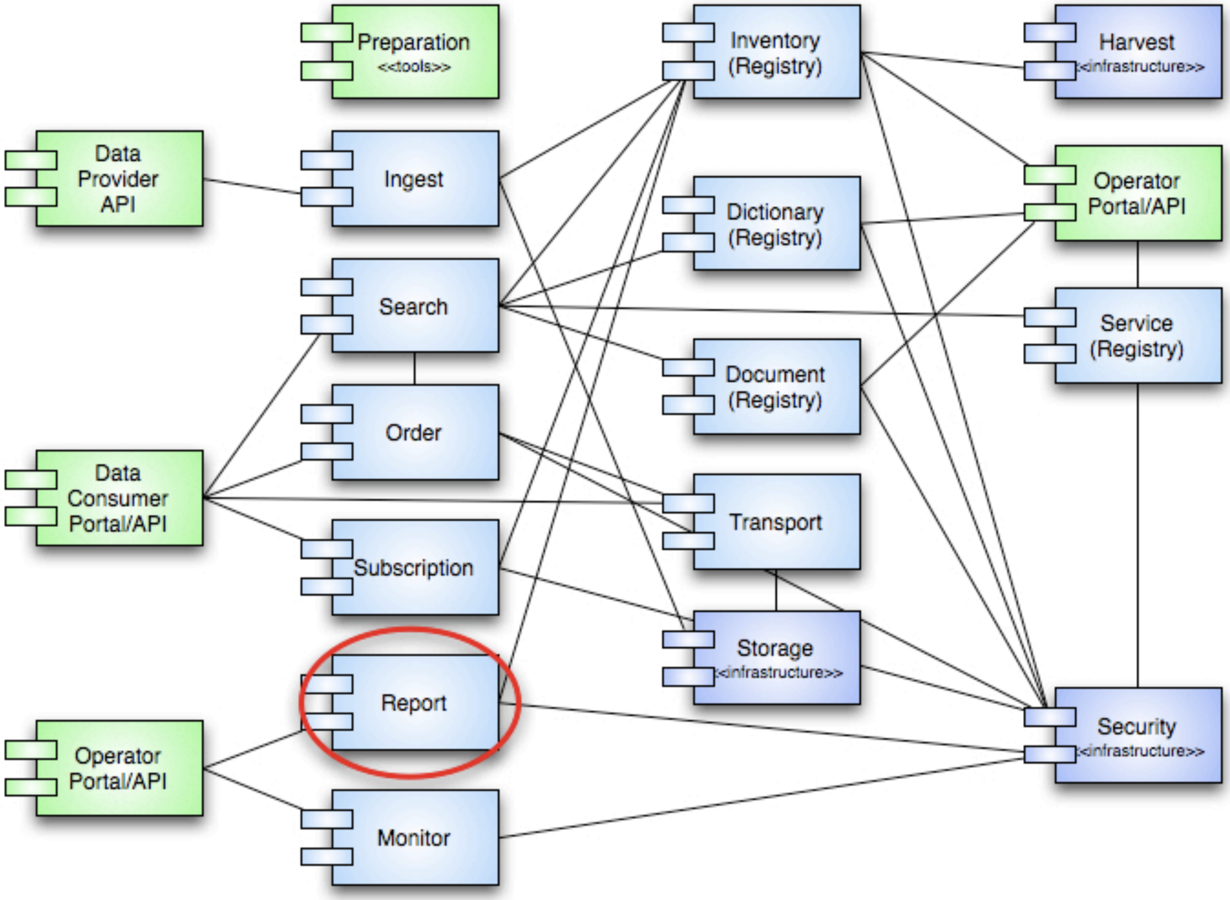
- As previously mentioned, OpenSSO and OpenDS from Sun have been selected.
  - OpenDS is a directory service from Sun based on the LDAP standard.
- We have installed the software on a couple of platforms and have tested the demonstration.
  - Currently integrating with the Registry service interface.
- The next step is to finalize the associated data model and populate from the current phonebook for further testing.

# Report Service

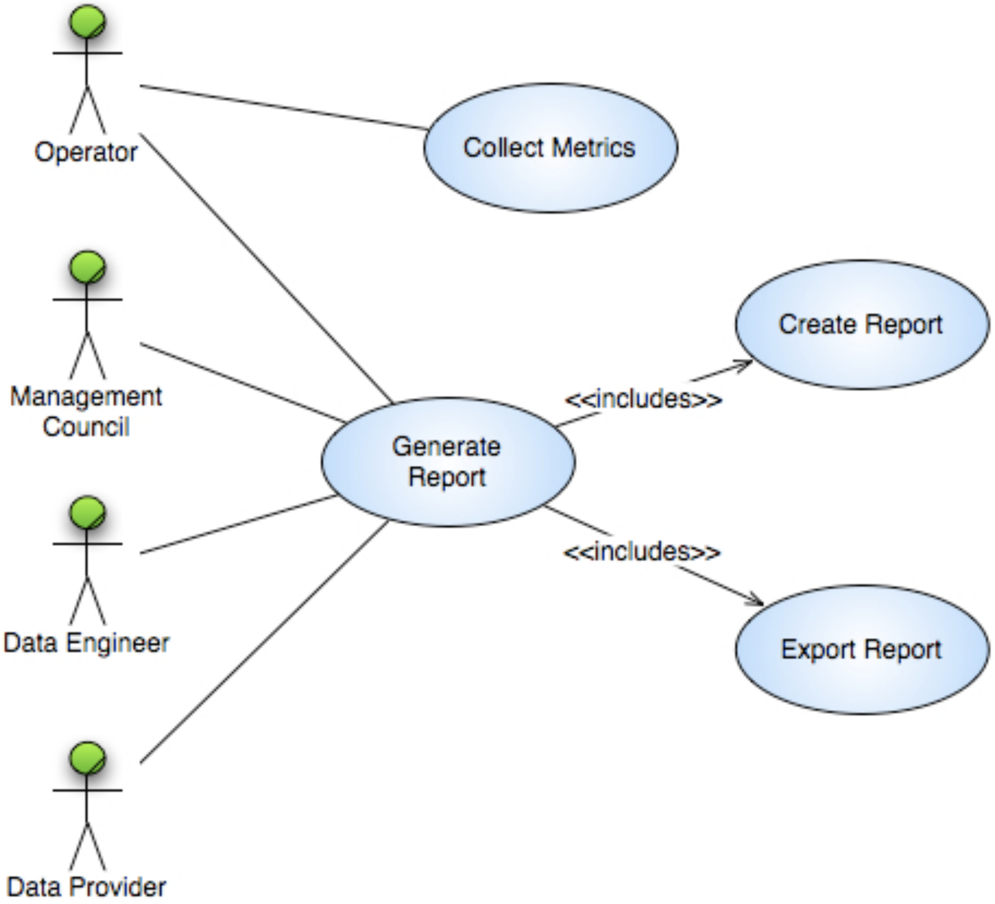
# Overview

- Provides functionality for capturing and reporting metrics.
- Not limited to metrics generated by PDS 2010 services.
  - Includes metrics from the FTP and web logs from the Nodes.
  - As well as any other commonly generated metric.
- Looking to satisfy the requirements with a COTS package.

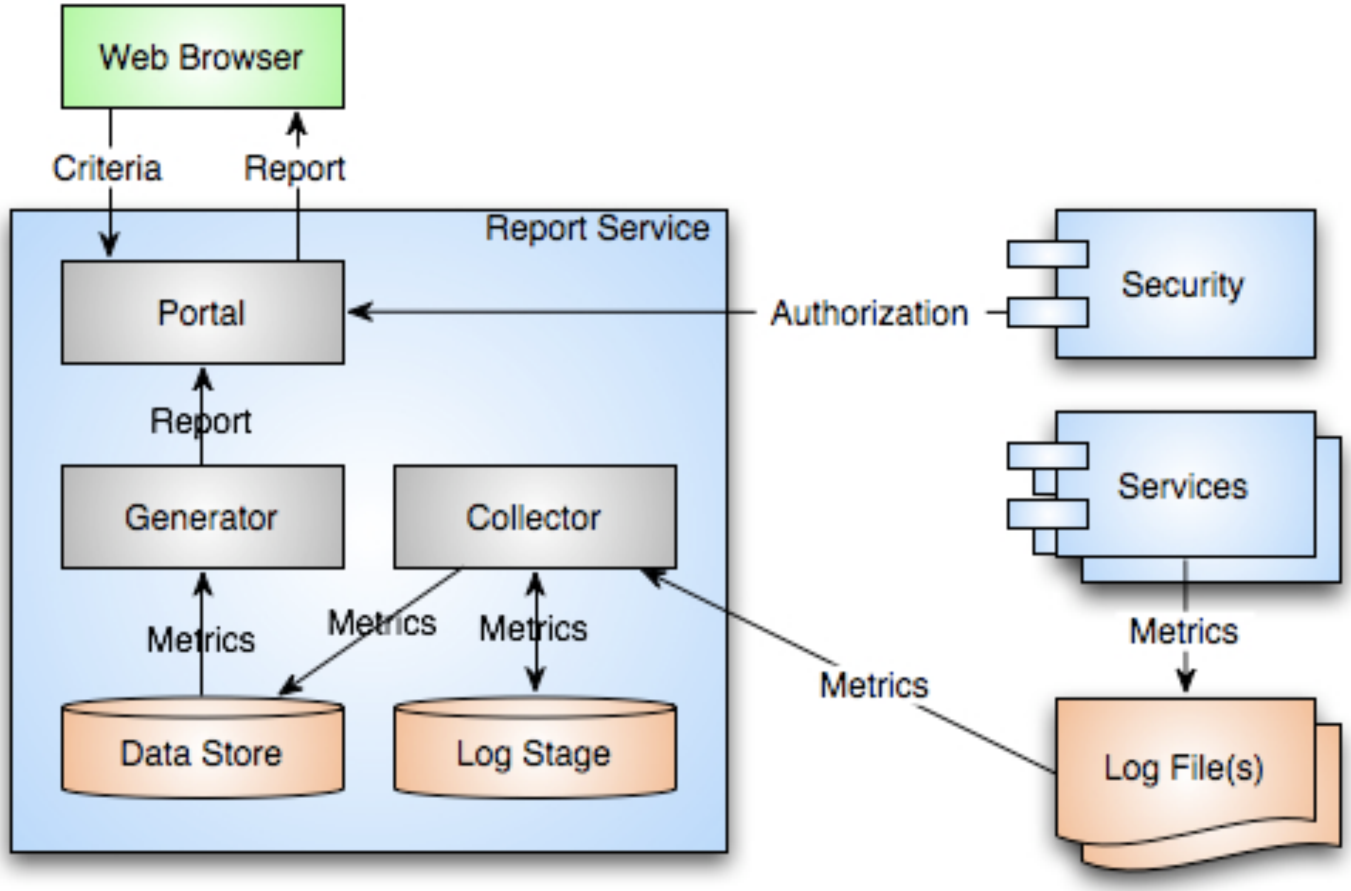
# Context


Preparation
<<tools>>
Inventory
(Registry)
Harvest
<<infrastructure>>
Data
Provider
API
Ingest
Dictionary
(Registry)
Operator
Portal/API
Search
Service
(Registry)
Document
(Registry)
Order
Data
Consumer
Portal/API
Transport
Subscription
Storage
<<infrastructure>>
Report
Security
<<infrastructure>>
Operator
Portal/API
Monitor

# Use Cases


Operator
Management Council
Data Engineer
Data Provider
Collect Metrics
Generate Report
<<includes>>
<<includes>>
Create Report
Export Report

# Architecture



# Implementation

- Plan to evaluate a couple of commercial software packages based on the service requirements.

# Search Service

# Overview

- This service is a deployable component that accepts queries for data and returns a set of matching results.
- Provides the public interface to the metadata contained in the federated registries.
- Provides the second line of metadata harvesting within the system in order to facilitate discovery of products.
- Generation of search indices from registry metadata supports multiple query formats and is tailor-able for customized search interfaces.

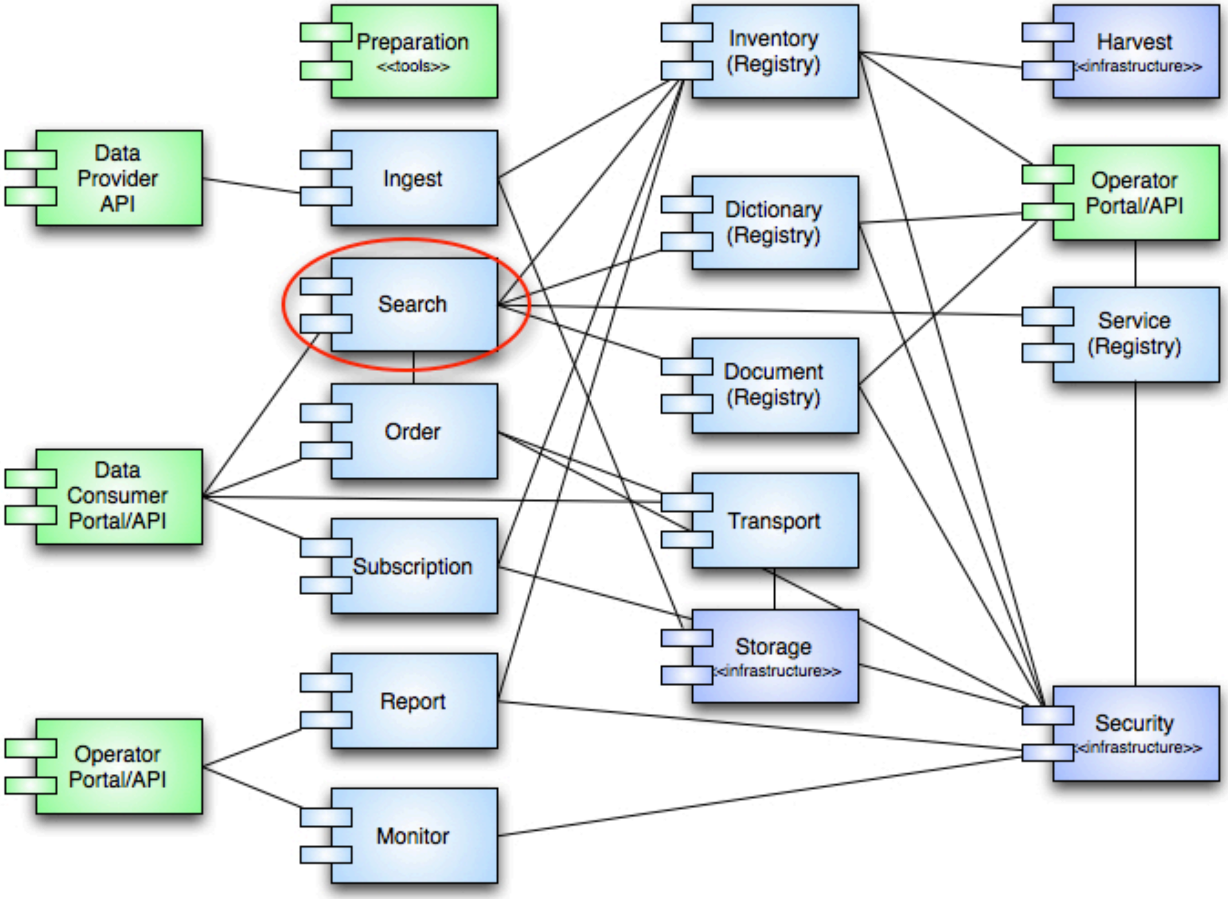
# Supported Query Formats

- Open
  - This is most commonly referred to as a Google-like search or text-based search.
  - User provides a sequence of textual terms that the service matches against its search indices.
- Guided
  - Known as faceted search and is highly interactive.
  - Presents a series of high-level organizational categories along with a set of terms in each category.
  - Enables customized narrowing of matching results through user-selected progressive disclosure.

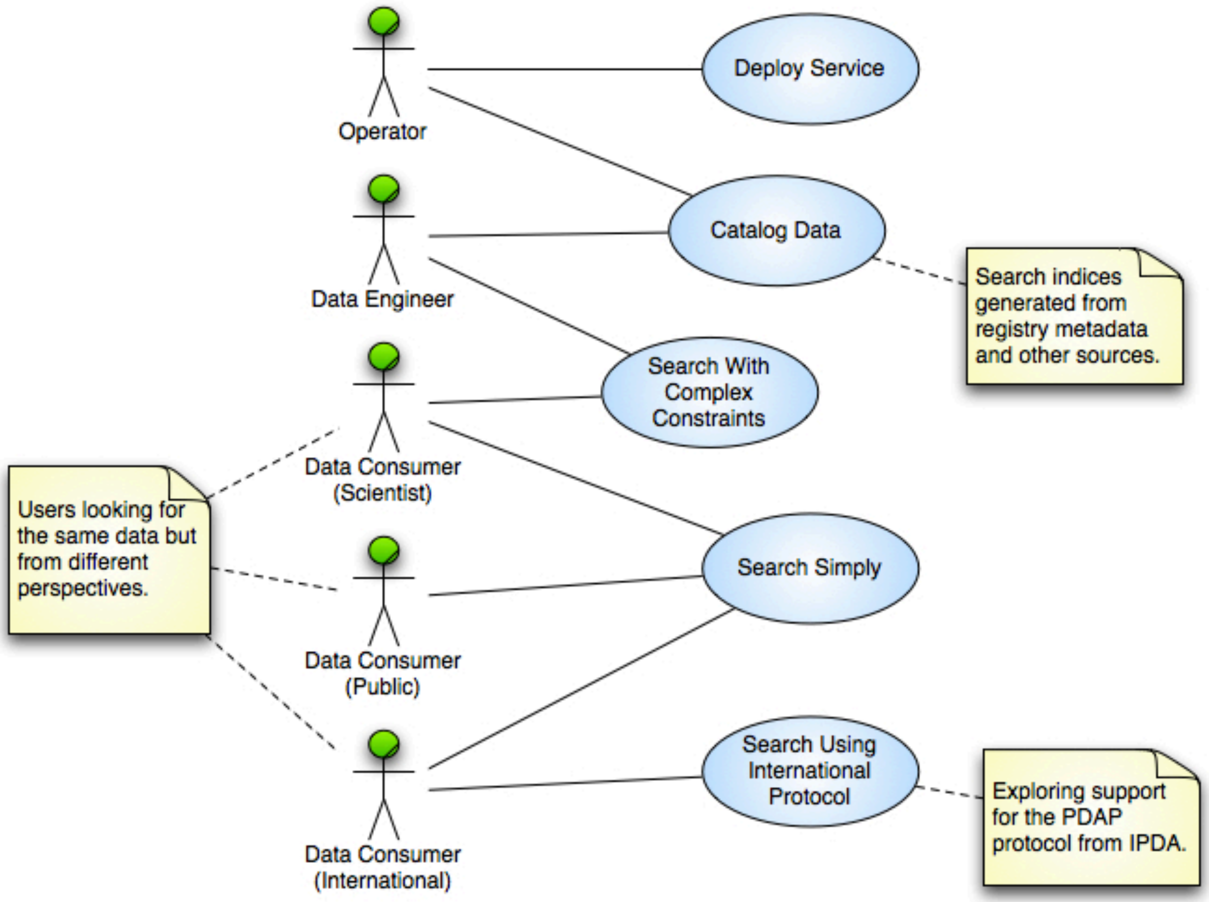
# Supported Query Formats (cont)

- Constrained
  - Referred to as form-based search allowing the user to specify query criteria from a defined set of constraints.
  - Requires intimate knowledge of the constraints and is targeted towards the expert user.
- Exploratory
  - This query format begins with the query results instead of a query expression.
  - Enables the user to find interesting data that's "nearby" in terms of time, space, or other relative measure.

# Context


Preparation
<<tools>>
Inventory (Registry)
Harvest
<<infrastructure>>
Data Provider API
Ingest
Dictionary (Registry)
Operator Portal/API
Search
Service (Registry)
Document (Registry)
Order
Data Consumer Portal/API
Transport
Subscription
Storage
<<infrastructure>>
Report
Security
<<infrastructure>>
Operator Portal/API
Monitor

# Use Cases


Operator
Data Engineer
Data Consumer (Scientist)
Data Consumer (Public)
Data Consumer (International)
Deploy Service
Catalog Data
Search With Complex Constraints
Search Simply
Search Using International Protocol

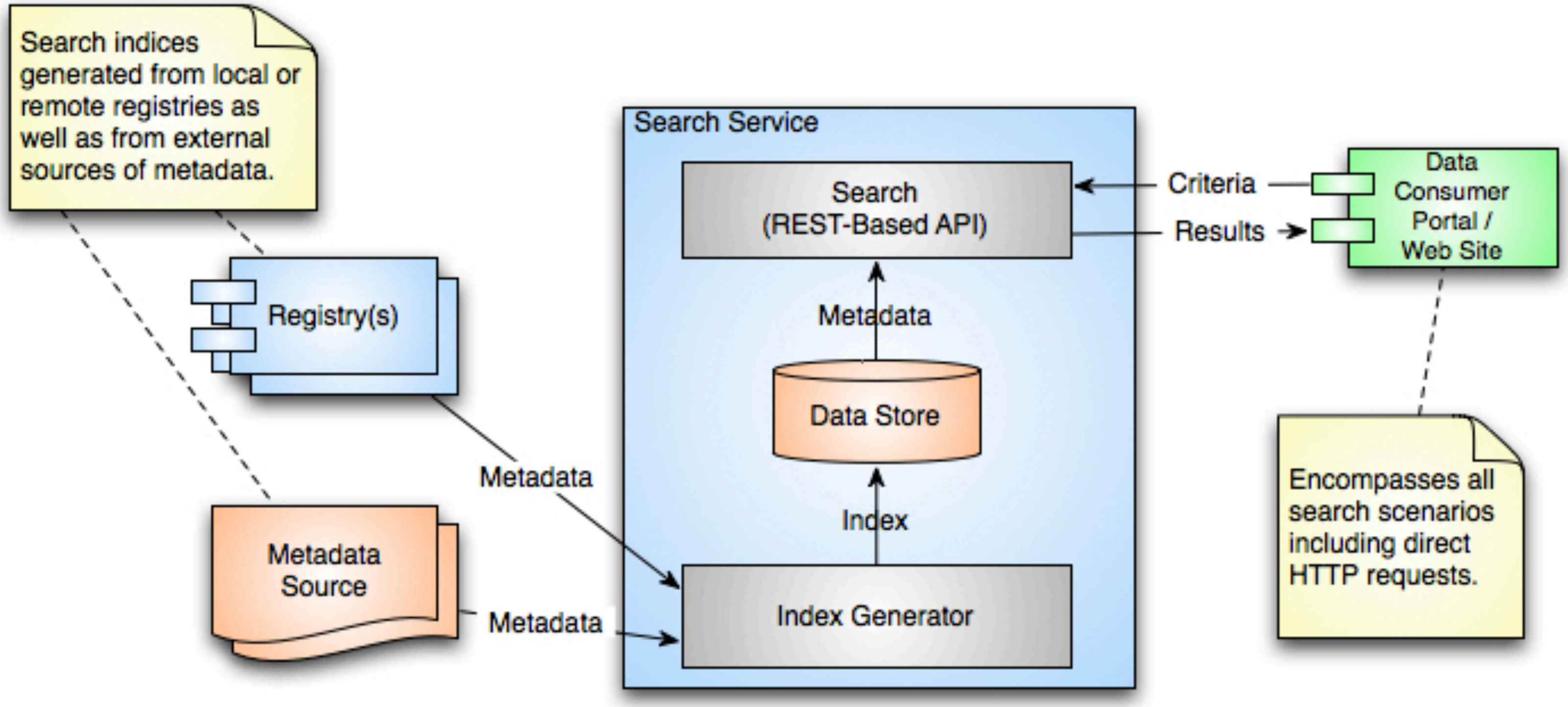
Search indices generated from registry metadata and other sources.

Users looking for the same data but from different perspectives.

Exploring support for the PDAP protocol from IPDA.

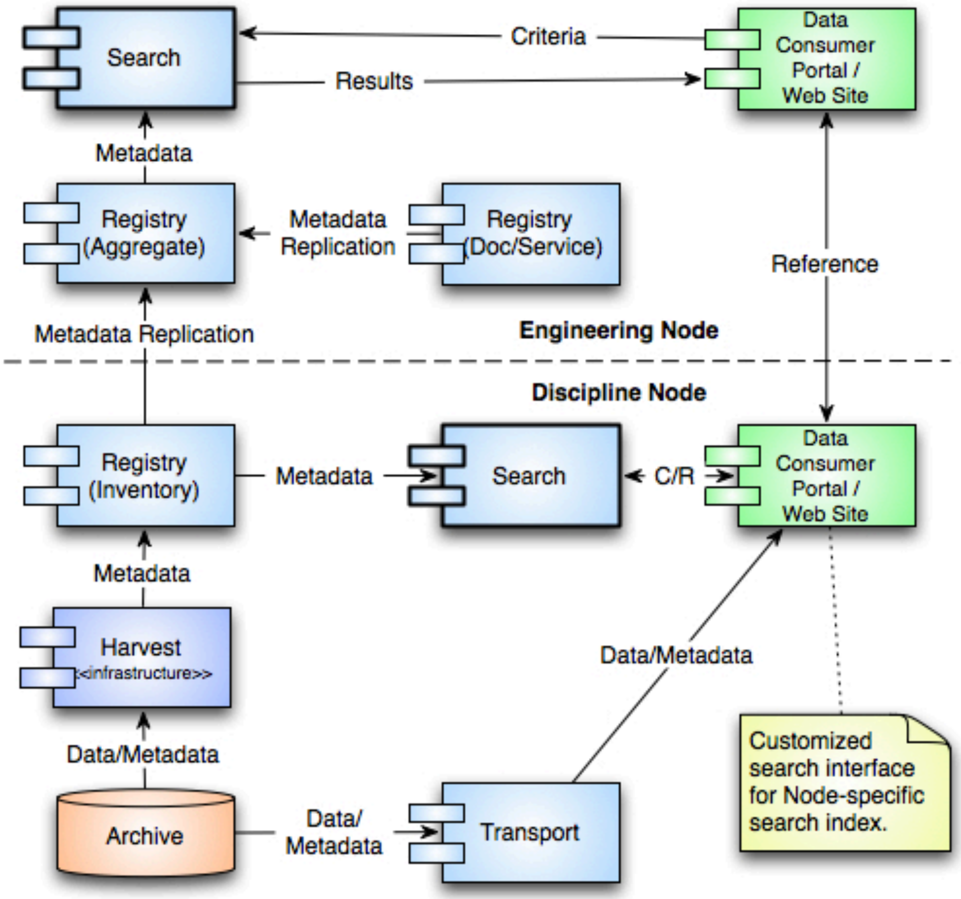
# Architecture

## (Stand-Alone Search Capabilities)



Search indices generated from local or remote registries as well as from external sources of metadata.
Registry(s)
Metadata Source
Metadata
Metadata
Search Service
Search (REST-Based API)
Metadata
Data Store
Index
Index Generator
Criteria
Results
Data Consumer Portal / Web Site
Encompasses all search scenarios including direct HTTP requests.

# Architecture
## (Deployment Scenarios)



# Analysis/Implementation

- Beyond desiring a REST-based interface and support for IPDA's PDAP query protocol, the design for this service has just started.

- It is anticipated that technologies such as Lucene and Solr will be utilized in the development of the service.

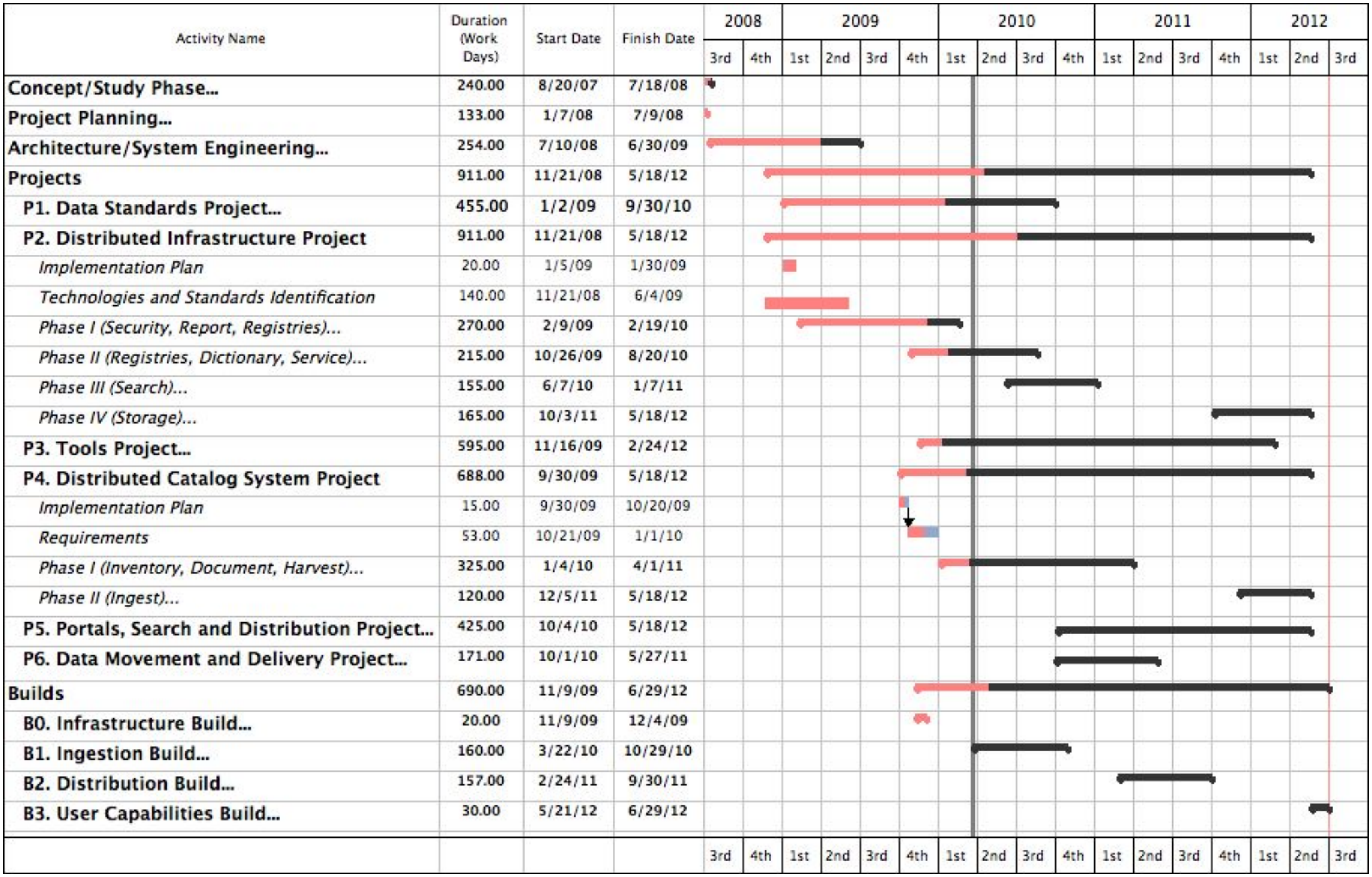
# Topics

- Overview
- Design Decisions
- Services
- Wrap Up

# Wrap Up

- Continue to work out the details for each service with periodic design reviews by the System Design Working Group.
- Design material posted to the PDS Engineering Node web site and made available to the PDS Technical Staff for review as well.
- Initiate implementation/integration for each of the core services.

# Schedule - Projects 2 and 4



| Activity Name | Duration (Work Days) | Start Date | Finish Date |
|---|---|---|---|
| **Concept/Study Phase...** | 240.00 | 8/20/07 | 7/18/08 |
| **Project Planning...** | 133.00 | 1/7/08 | 7/9/08 |
| **Architecture/System Engineering...** | 254.00 | 7/10/08 | 6/30/09 |
| **Projects** | 911.00 | 11/21/08 | 5/18/12 |
| **P1. Data Standards Project...** | 455.00 | 1/2/09 | 9/30/10 |
| **P2. Distributed Infrastructure Project** | 911.00 | 11/21/08 | 5/18/12 |
| *Implementation Plan* | 20.00 | 1/5/09 | 1/30/09 |
| *Technologies and Standards Identification* | 140.00 | 11/21/08 | 6/4/09 |
| *Phase I (Security, Report, Registries)...* | 270.00 | 2/9/09 | 2/19/10 |
| *Phase II (Registries, Dictionary, Service)...* | 215.00 | 10/26/09 | 8/20/10 |
| *Phase III (Search)...* | 155.00 | 6/7/10 | 1/7/11 |
| *Phase IV (Storage)...* | 165.00 | 10/3/11 | 5/18/12 |
| **P3. Tools Project...** | 595.00 | 11/16/09 | 2/24/12 |
| **P4. Distributed Catalog System Project** | 688.00 | 9/30/09 | 5/18/12 |
| *Implementation Plan* | 15.00 | 9/30/09 | 10/20/09 |
| *Requirements* | 53.00 | 10/21/09 | 1/1/10 |
| *Phase I (Inventory, Document, Harvest)...* | 325.00 | 1/4/10 | 4/1/11 |
| *Phase II (Ingest)...* | 120.00 | 12/5/11 | 5/18/12 |
| **P5. Portals, Search and Distribution Project...** | 425.00 | 10/4/10 | 5/18/12 |
| **P6. Data Movement and Delivery Project...** | 171.00 | 10/1/10 | 5/27/11 |
| **Builds** | 690.00 | 11/9/09 | 6/29/12 |
| **B0. Infrastructure Build...** | 20.00 | 11/9/09 | 12/4/09 |
| **B1. Ingestion Build...** | 160.00 | 3/22/10 | 10/29/10 |
| **B2. Distribution Build...** | 157.00 | 2/24/11 | 9/30/11 |
| **B3. User Capabilities Build...** | 30.00 | 5/21/12 | 6/29/12 |

# Questions/Comments