PDS 2010 System Design

Technical Session

June 10-11, 2009

Distributed Infrastructure Design Team

Sessions

- Overview of System Design
  - Provide an overview of the system design showing the end-to-end flow and the supporting services, tools and applications that will be necessary.

- Implementation Approach
  - Provide another level of detail for each of the components in the system including their functionality and usage within the system.

Overview of System Design

Topics

- Design Team
- Design Principles/Goals/Constraints
- Service-Based Design
- Ingestion Scenario
- Distribution Scenario
- Reporting Scenario

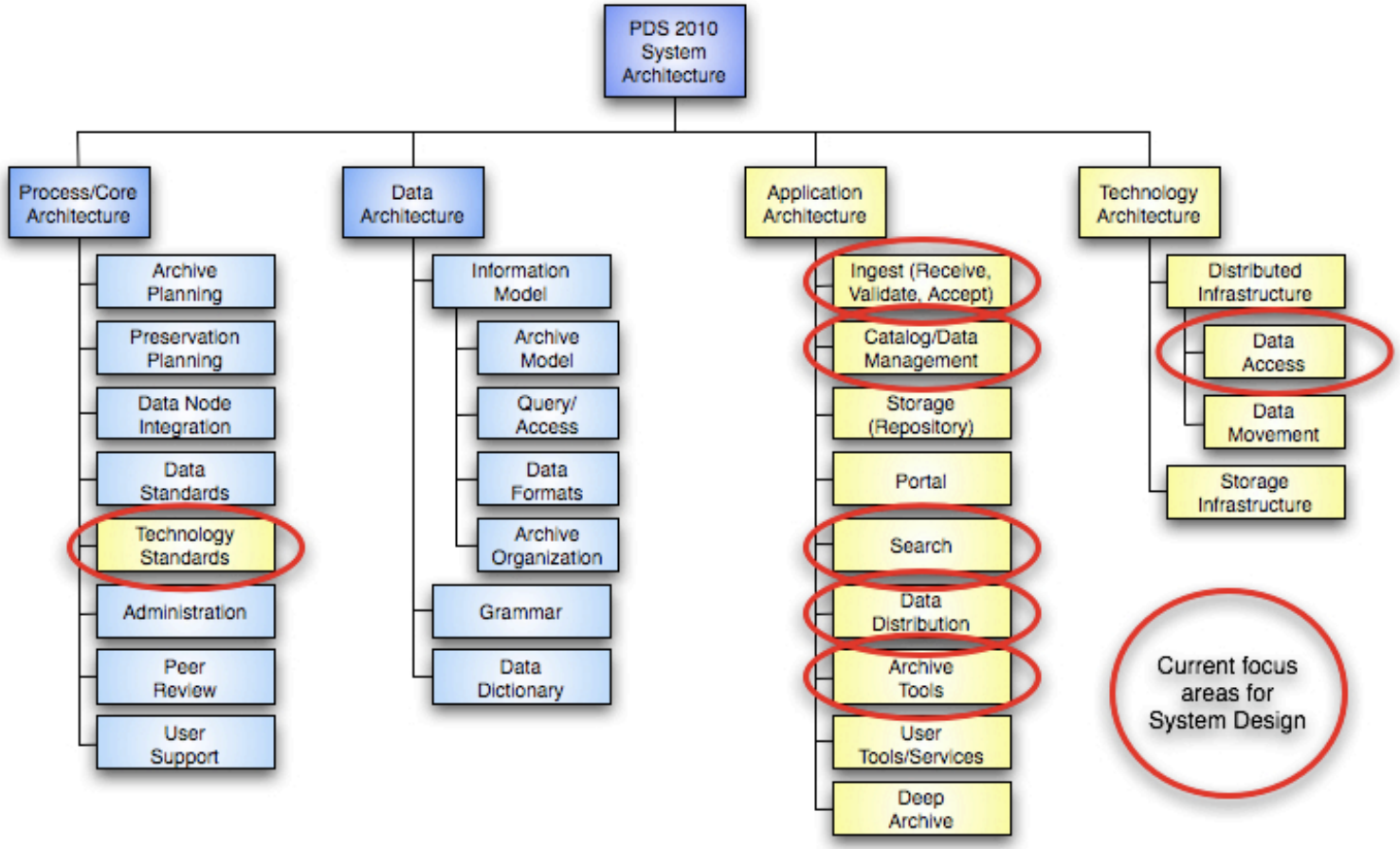
Design Team

- Formed the design team back in January, which consists of the following personnel:
  - Sean Hardman (Engineering)
  - Todd King (PPI)
  - Mike Martin (Management)
  - Paul Ramirez (Engineering)
  - Alice Stanboli (Imaging)
  - Tom Stein (Geosciences)
- Weekly teleconferences (more or less) are held on Tuesday mornings, formerly Thursday afternoons.
- Current artifacts are captured on the PDS Wiki and Engineering Node web sites:
  - http://oodt.jpl.nasa.gov/wiki/pages/viewpage.action?pageId=2600
  - http://pds-engineering.jpl.nasa.gov/index.cfm?pid=100&cid=134

Design Team
Sub-Project/Team Objectives

- Investigate and select the core technologies to be utilized in the development and operation of PDS 2010.
- Initiate development of some of the core services that will serve as building blocks for development of the system.
  - Core services include: Registry, Security, Report, Dictionary and Distributed Access Infrastructure.
- Recent focus has shifted towards defining ingestion and distribution functionality.
- Capture technology standards and service development guidelines for the PDS.

Design Team
Sub-Project/Team Focus



Design Team
Engineering Approach

- Prepare a brief white paper identifying the state-of-the-practice for each service and whether there are COTS or open source solutions available.
- Identify use cases and/or requirements for the service.
- Prepare a design for implementing the service from scratch or for integrating a COTS or open source solution.
- Implement/integrate the service per the design.
- Test the service against the requirements.
- Deploy the system to the target environment (e.g., DN, EN).

Design 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.

* Derived from Architectural Principles

Design Goals*

- Improve ingestion efficiency (catalog and data products).
- 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.

*Derived from PDS 2010 Drivers and Goals

Design Constraints

- 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 minimize unnecessary movement of data.
- Limited and/or trickle-in funding designated for PDS 2010, dictates a flexible and phased approach for development and deployment of the system.

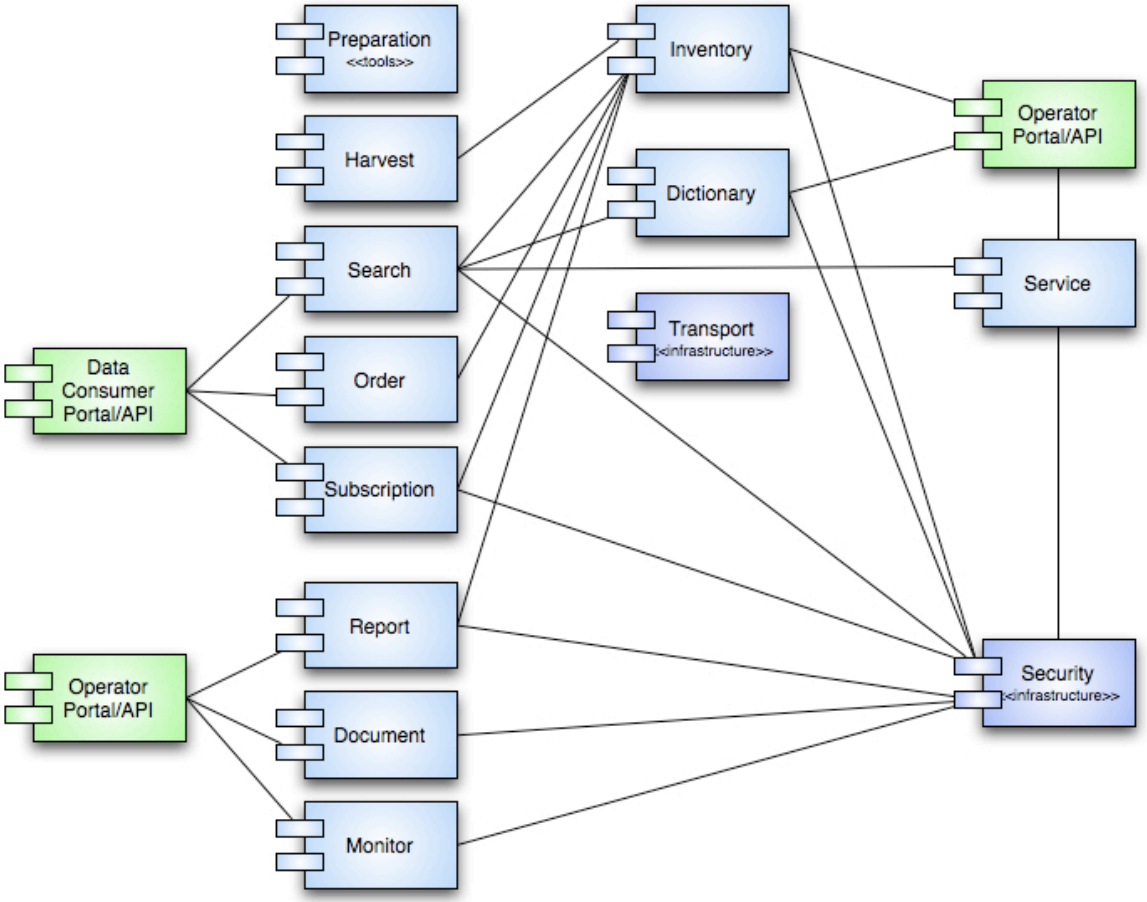
Service-Based Design

- There are several advantages to adopting a Service-Oriented Architecture (SOA):
  - Captures many of the best practices of previous architectures.
  - Well suited for a distributed system.
  - Promotes "loose coupling", "software reuse", "encapsulation" along with other hot buzz phrases in software development today.
  - A service-based architecture provides currency and timeliness for the system.
- Currently working towards a lightweight SOA solution that suits PDS.
- Service-based functionality will focus on search and retrieval of data.
- A tool-based approach is still appropriate for data preparation.

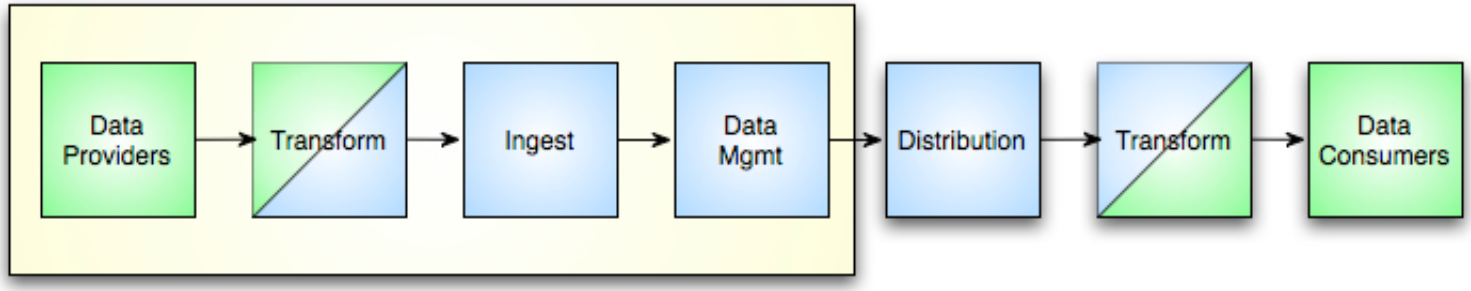
Service-Based Design
Service Vision

- Plans include developing a Service Specification to guide future service and component development for PDS personnel.
  - Will provide details on such things as interface and message content requirements.
  - Will facilitate development of node-specific services/ components (e.g., transformation) that can be integrated with PDS 2010 services.
- The goal is to design and build an extensible system that can grow and have functionality added to over time.

Service-Based Design
Component Identification

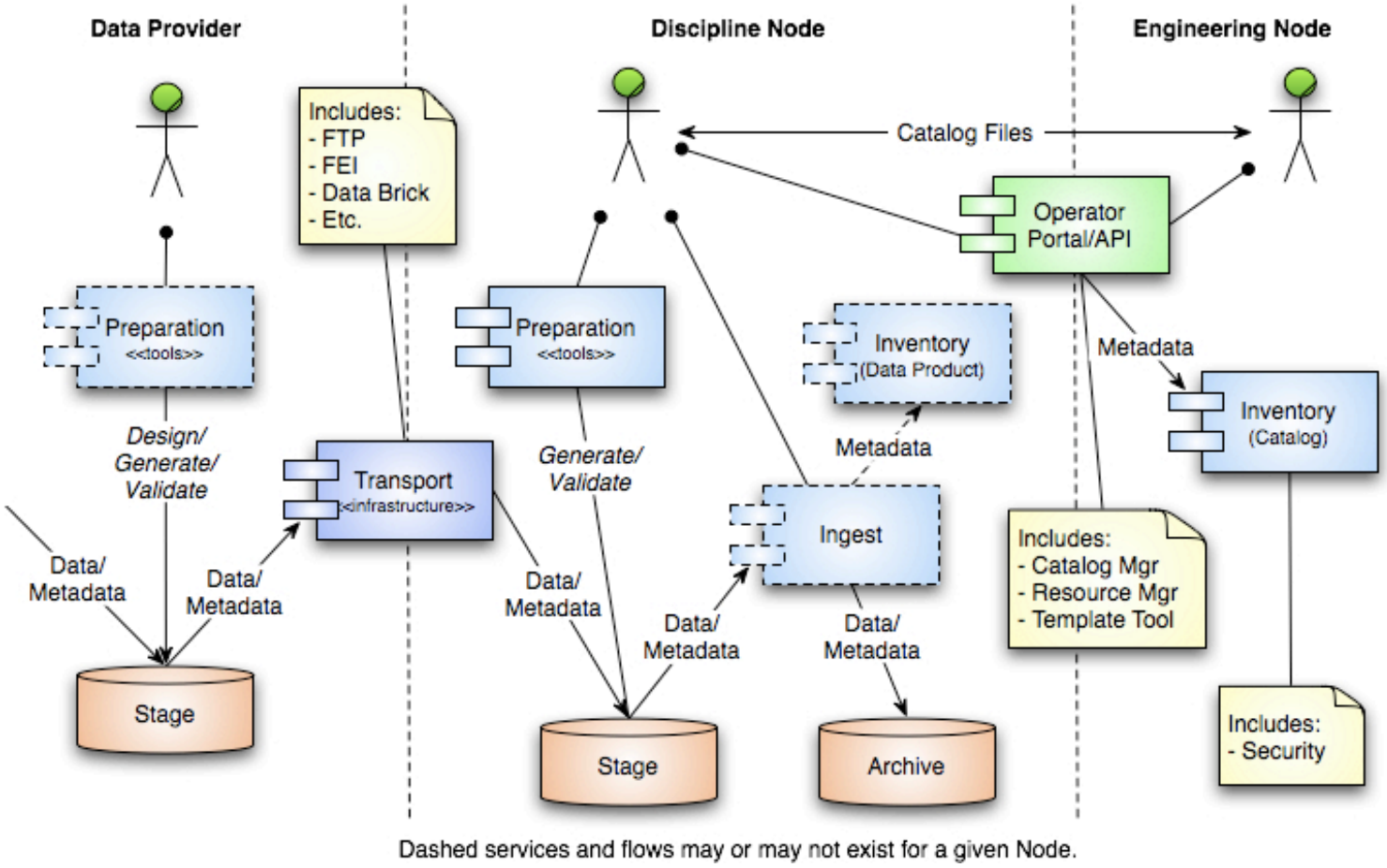


Ingestion Scenario

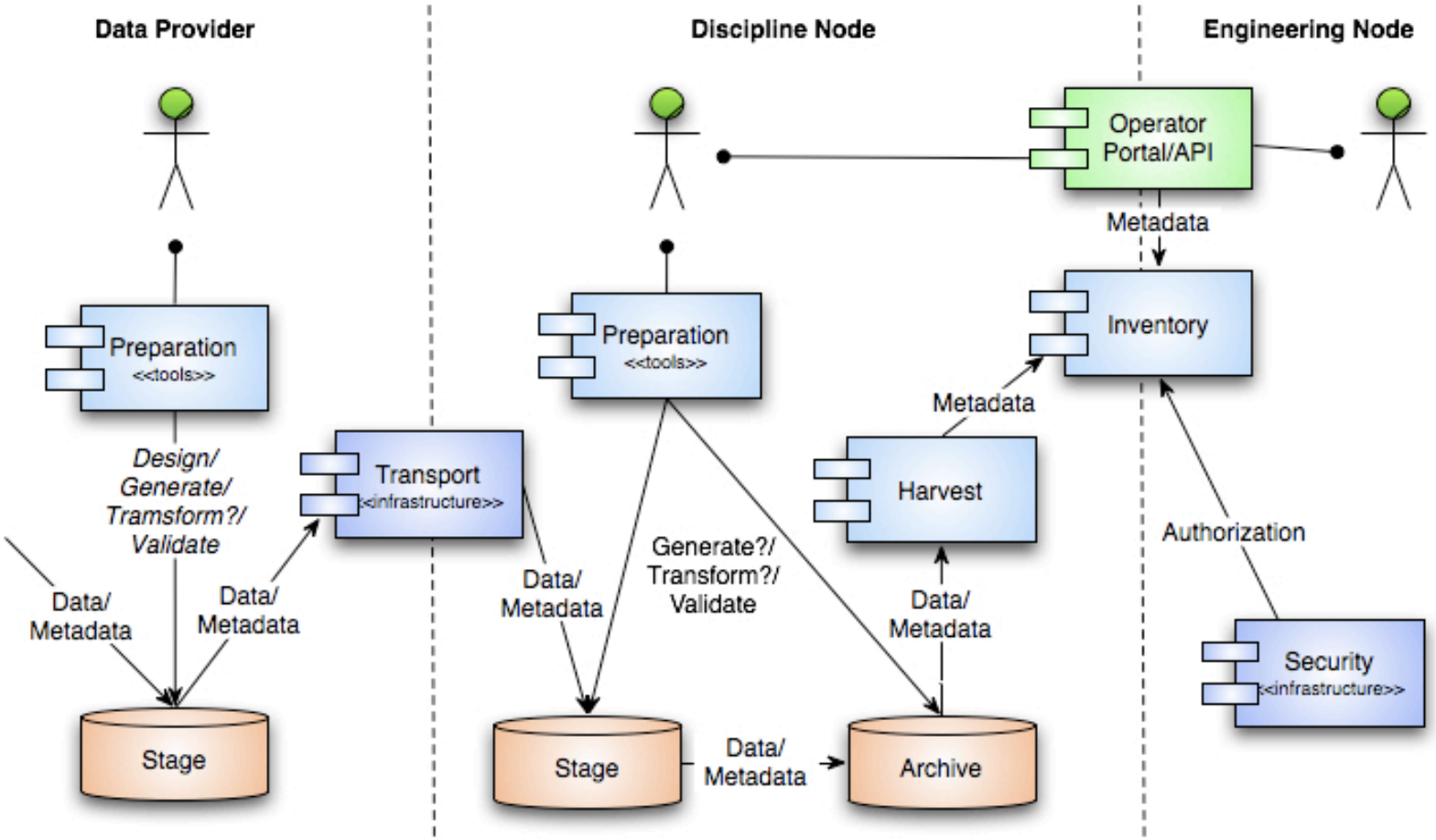


- The ingestion scenario covers ingestion of catalog and data product metadata into their respective Inventory services.
- The proposed ingestion design will be contrasted with the current design (using current naming convention) and focuses on the area in the end-to-end diagram above highlighted by the yellow box.

Ingestion Scenario
Current Design



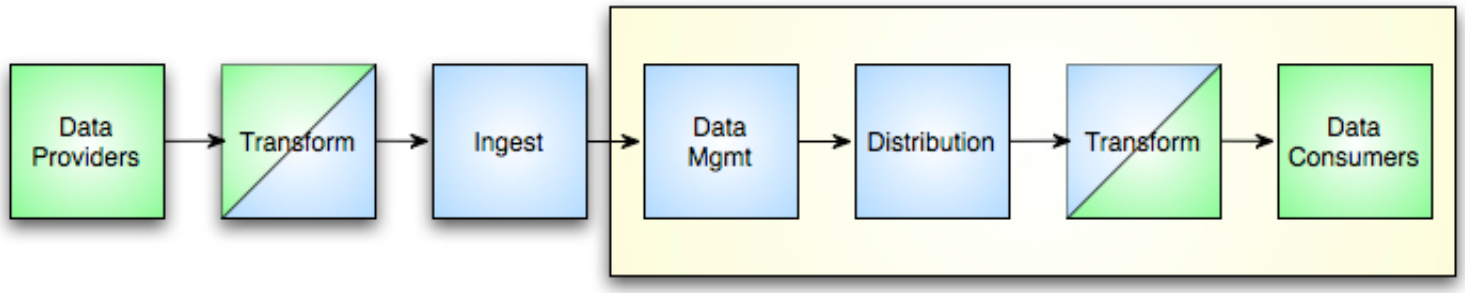
Ingestion Scenario
Proposed Design



Ingestion Scenario
Design Differences

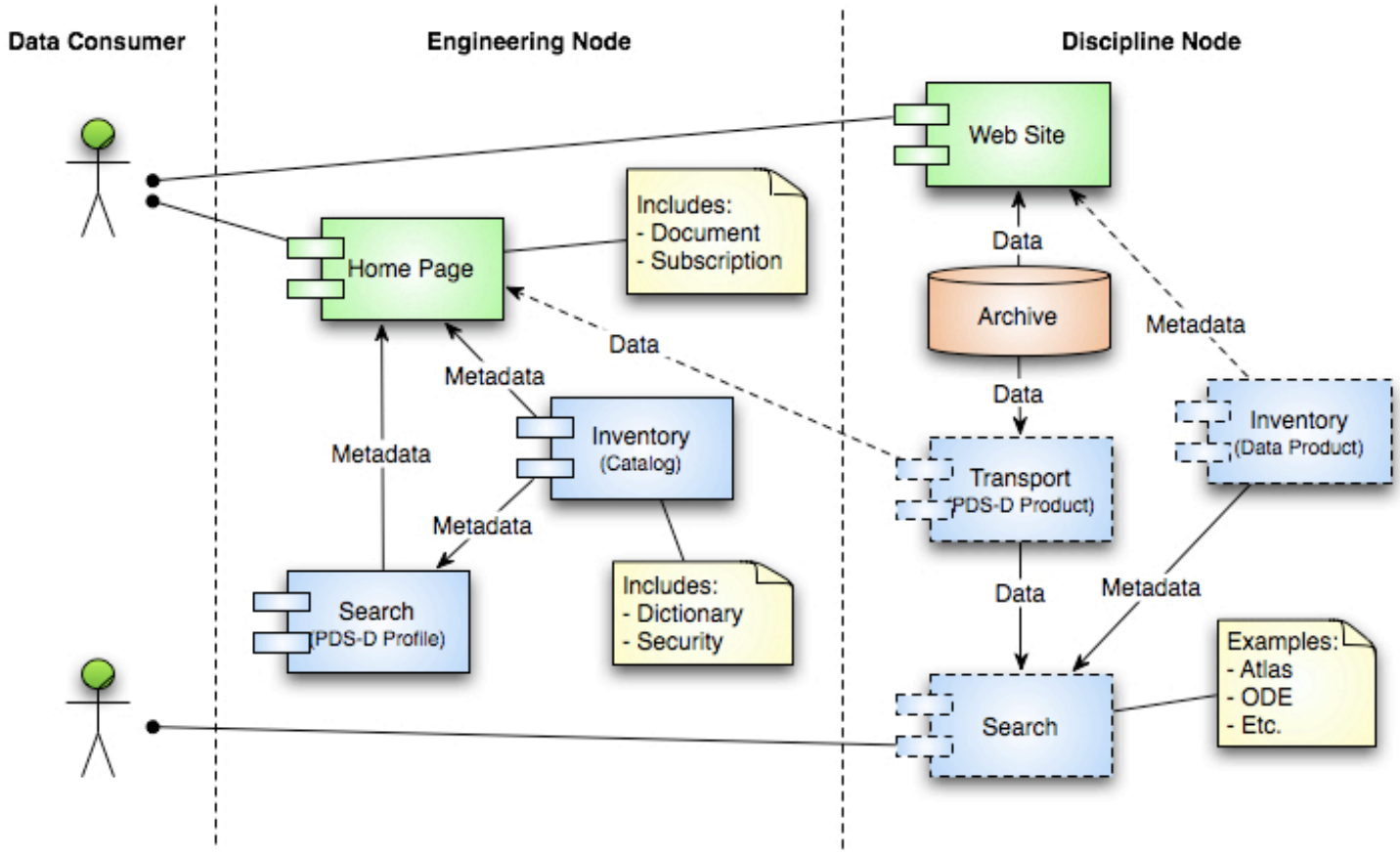
- Transformation of incoming data/metadata is shown as a possible function for the Data Provider or the Discipline Node via a tool.
- A Harvest Service is introduced for capturing and registering data product and catalog-level metadata.
  - A portal/API can also be utilized for submission and maintenance of data product and catalog-level metadata.
- An Inventory Service is introduced for tracking data product submissions and catalog-level metadata.

Distribution Scenario



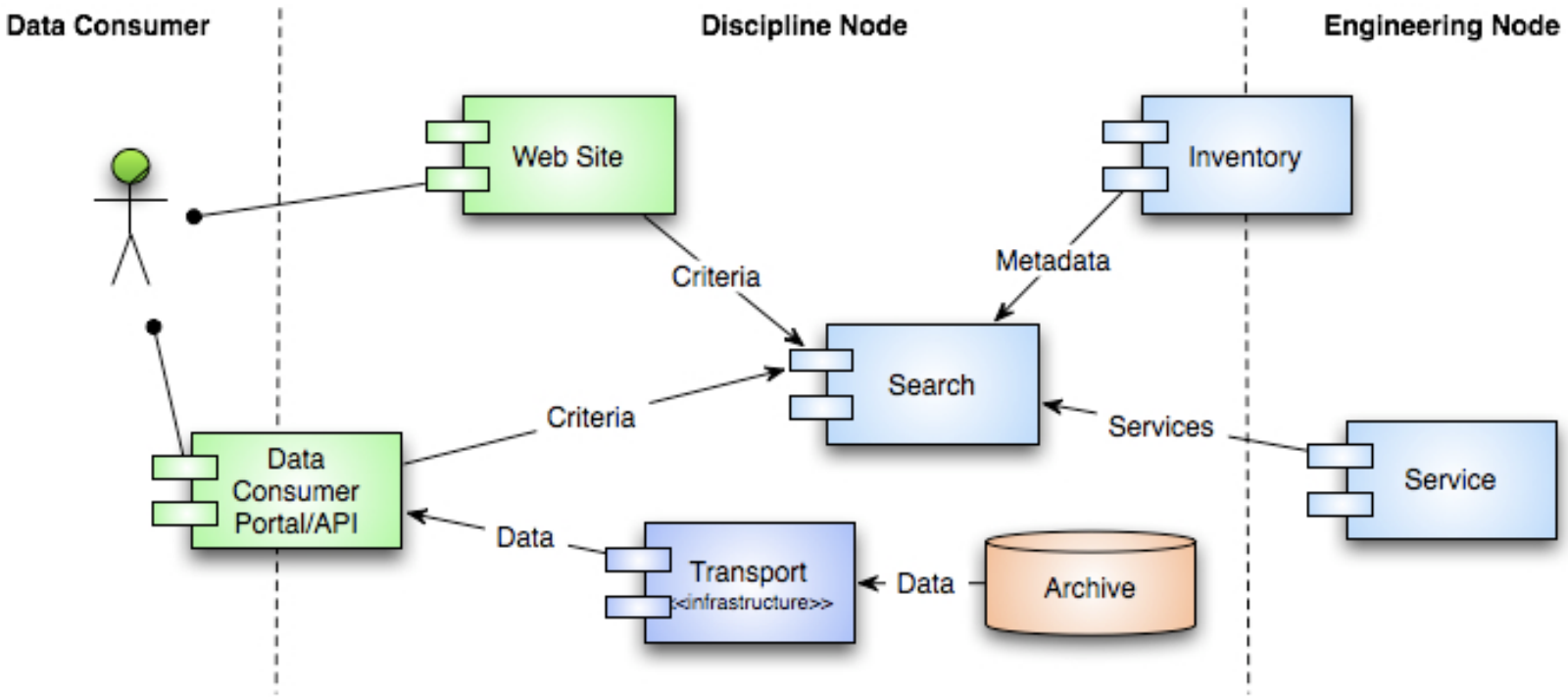
- The distribution scenario covers search of the catalog and data product metadata and distribution of associated data.
- The proposed distribution design will be contrasted with the current design (using current naming convention) and focuses on the area in the end-to-end diagram above highlighted by the yellow box.
- The proposed design includes scenarios for DN and EN initiated searches.

Distribution Scenario
Current Design

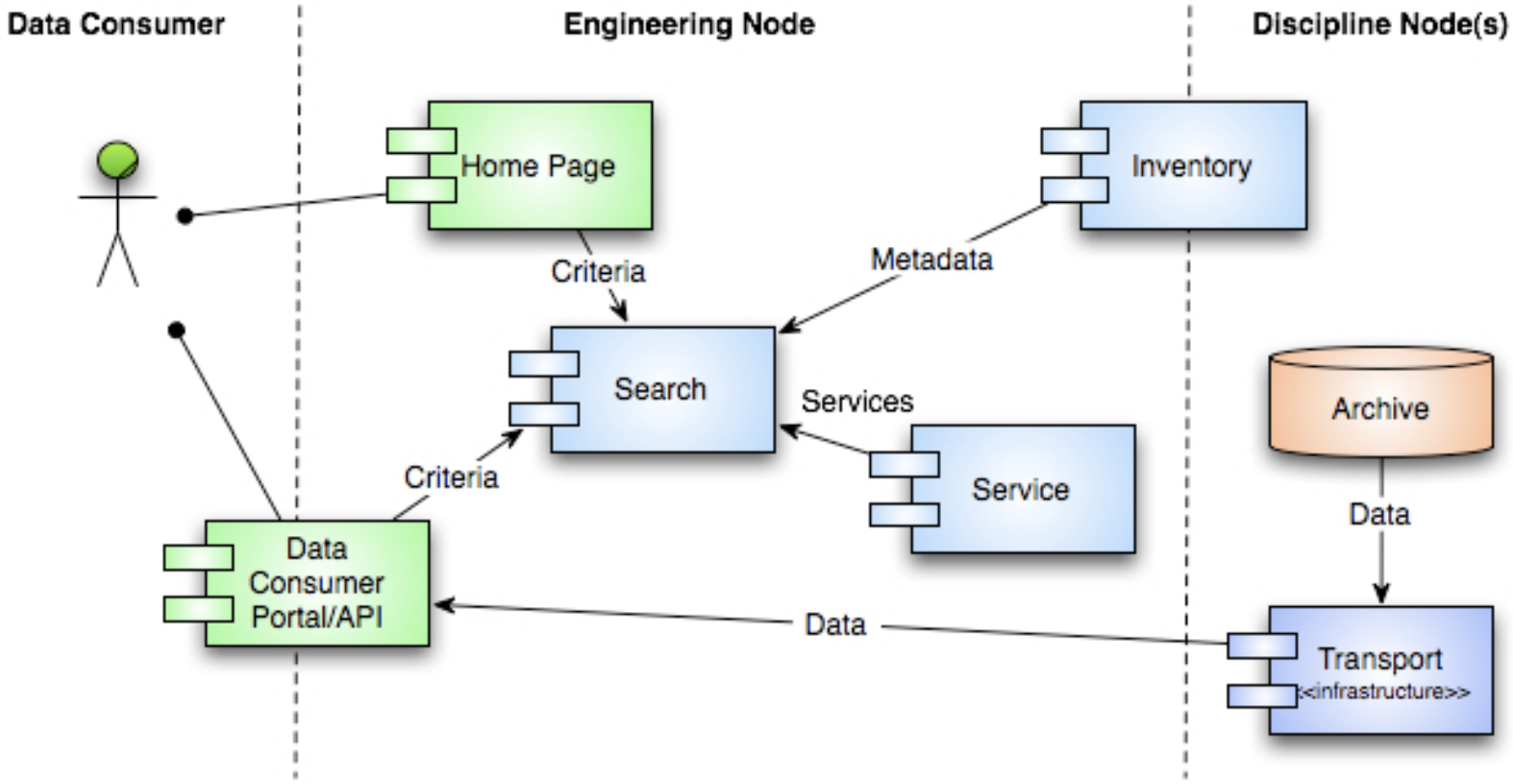


Dashed services and flows may or may not exist for a given Node.

Distribution Scenario
Proposed Design (DN Search)



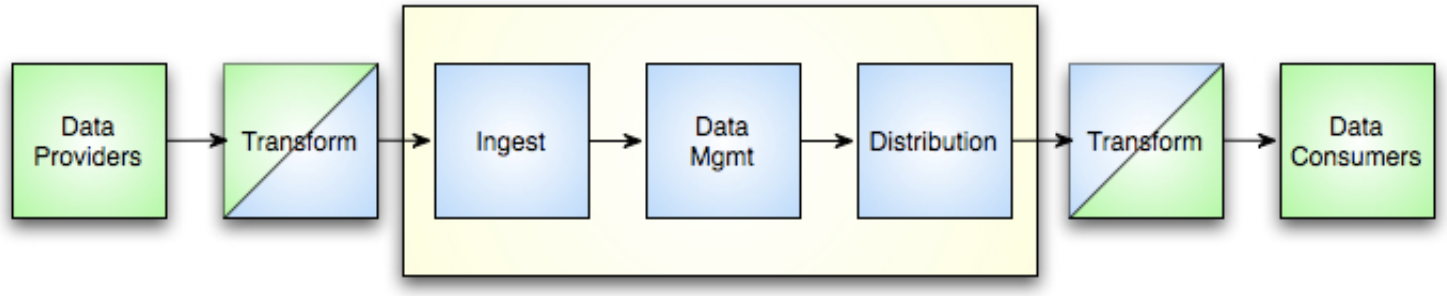
Distribution Scenario
Proposed Design (EN Search)



Distribution Scenario
Design Differences

- Utilization of a common Search Service for interfacing with the services hosting catalog-based and Node-specific data product metadata.
- Extensible but common, Transport Service is introduced to facilitate access and usability (i.e., transformation) of data products.
- Introduction of a Data Consumer portal/API for discovering and retrieving data/metadata.
  - The portal can be a Node developed application or user developed.

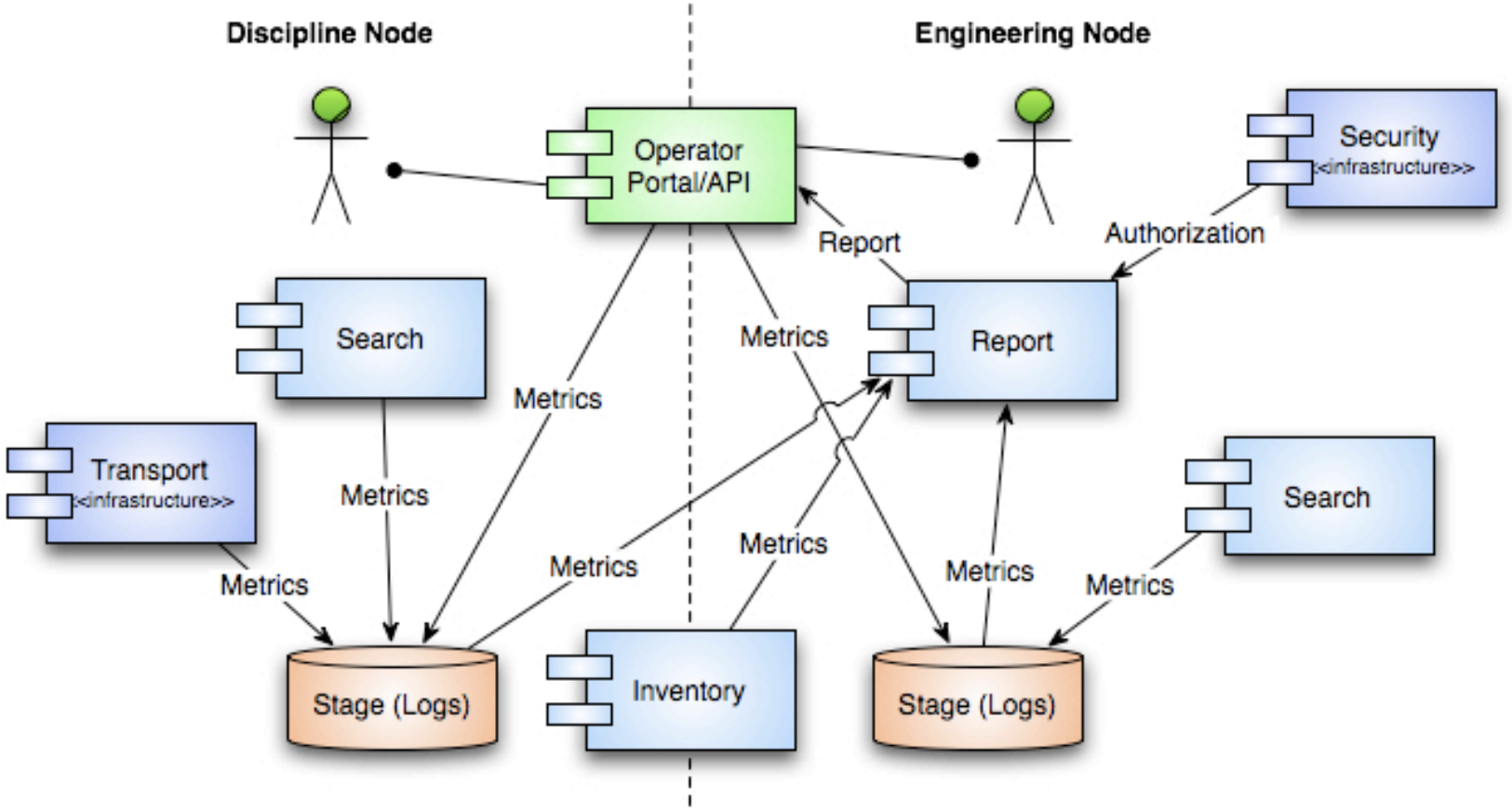
Reporting Scenario



- This scenario details a centralized Report service that consolidates metrics from various sources in the system and provides report generation.

- The proposed reporting design focuses on the area in the end-to-end diagram above highlighted by the yellow box.

Reporting Scenario
Proposed Design



Reporting Scenario
Design Specifics

- Non-registry services (e.g., Portals, Search and Transport) will generate local logs detailing activity.
  - At a minimum this includes standard web and FTP logs.
  - Logs will be pulled periodically (e.g., daily) by the Report service.
- Registry-based services will be queried on demand using their standard interfaces.
  - Queries would be performed periodically to extract metrics.
- This service is a good candidate for a COTS solution.

Questions/Comments

Tool and Service details are forthcoming in the next session.

Implementation Approach

Topics

- Overview
- Federated Registries
- Technology Architecture
- Service and Tool Details
- Design Goal Evaluation
- Plans

Overview

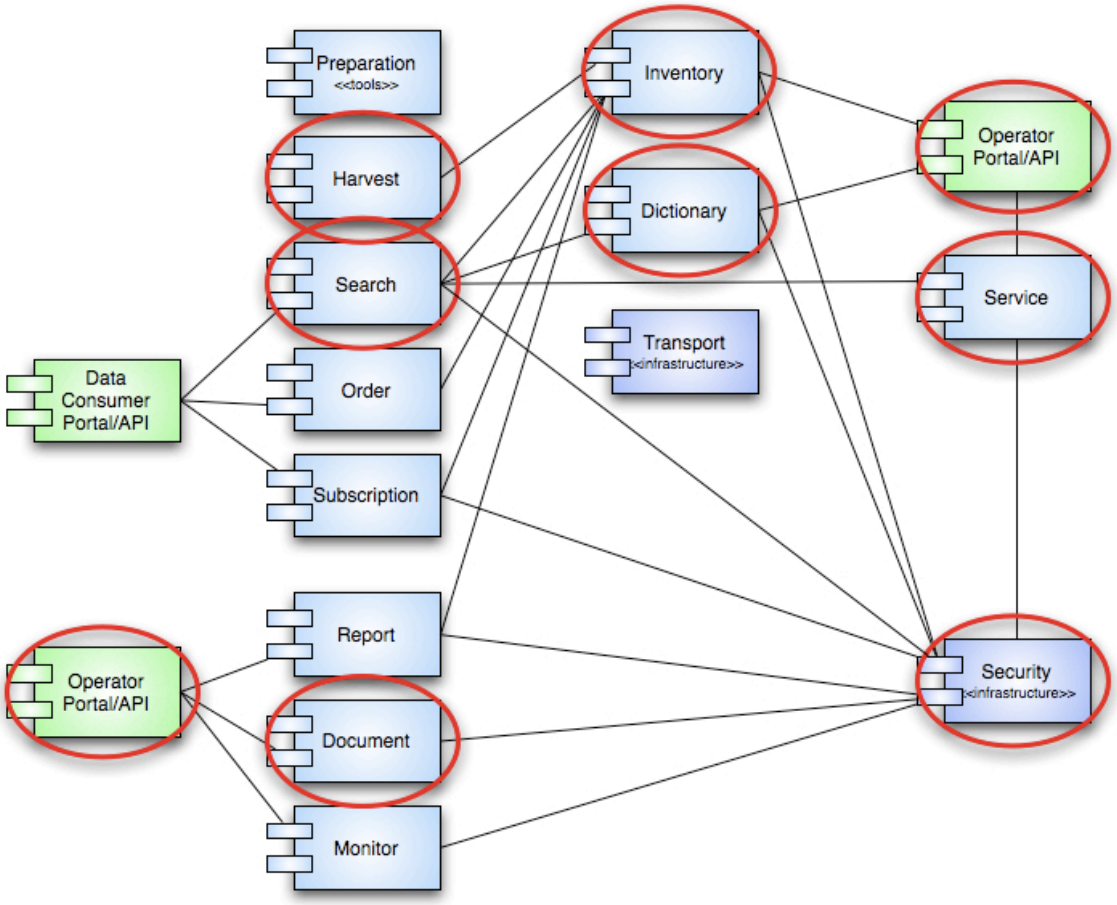
- Provide another level of detail for each of the components in the system including their proposed functionality and usage within the system.
- The level in this session includes detailing the likely technologies and standards to be utilized by each component.
- This session starts with the discussion of a design concept that is referenced throughout the session.

Federated Registries

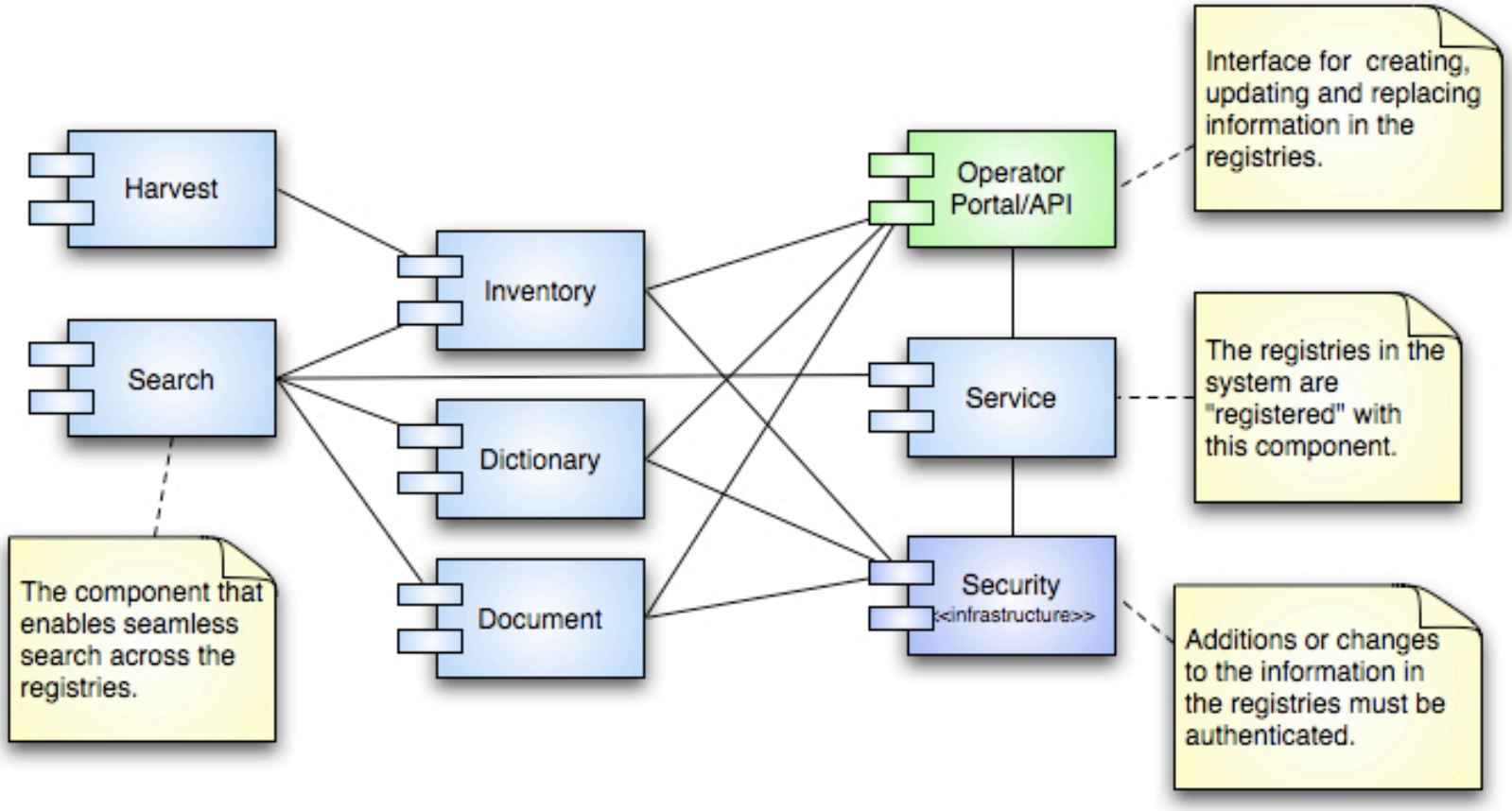
- A registry provides services for sharing content and metadata.
- A federated registry allows cooperating registries to appear and act as a single virtual registry.
- A query into the federation returns results from all cooperating registries.
- A federated registry:
  - Provides seamless information integration and sharing
  - Preserves local governance

Federated Registries
Features

Federated Registries
Related Components



Federated Registries
Focused View



Federated Registries
Registry Content

- Service
  - Service Descriptions
- Inventory
  - Mission, Instrument, Target, Data Set, Data Product Descriptions, etc.
- Dictionary
  - Object/Group Definitions
  - Element Definitions
- Document
  - PDS Documents (e.g., APG, PAG, etc.)
  - Software Packages
  - Schema Documents

Federated Registries
Registry Standards

- There are two prevailing registry standards in common use:
  - UDDI (Universal Description Discovery & Integration)
    - One of the standards from the WS-*(Web Services) stack of standards.
    - Promotes a service registry or "yellow pages" of available services.
  - ebXML (Electronic Business using eXtensible Markup Language)
    - A modular suite of specifications enabling business of the Internet.
    - Promotes a registry as an information repository.
    - Supports registration of different objects based on an ebRIM profile per object type.
- Although they both facilitate a SOA, the ebXML standard better facilitates the federated registry concept.

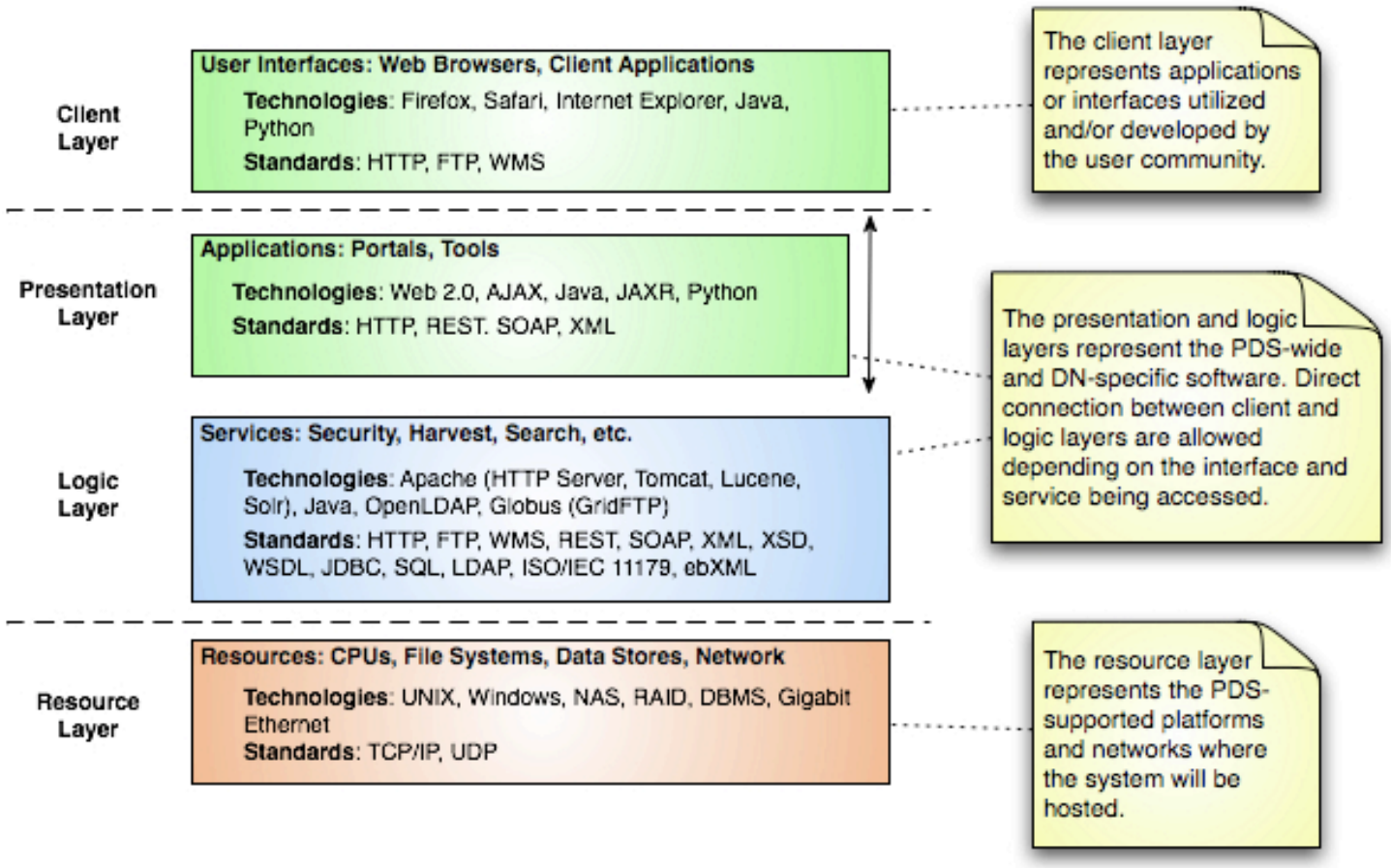
Federated Registries
Implementation Options

- Open Source Solutions
  - freebXML
    - A downloadable package that has support for several object types built-in.
    - Would most likely require continued development on our part.
  - Cultural, Artistic and Scientific knowledge for Preservation, Access and Retrieval (CASPAR) Project
    - Offer a Registry/Repository package based on freebXML.
- COTS Products
  - WellGEO RegRep from Wellfleet Software Corporation
    - Developed by the main author of freebXML.
    - We are currently in contact with this individual.
- Home Grown
  - Implement an ebXML-compliant solution from scratch.
  - Tailor existing components to provide an ebXML interface (e.g., OODT Profile/Product servers).

Federated Registries
freebXML Evaluation

- Downloaded and installed the software at the Engineering Node.
  - Not exactly a straightforward process.
  - The certificate-based authentication has limited access to outside of JPL.
- Currently working on exporting object descriptions (in ebRIM format) from the data model for import into the registry.
- The current version is based on version 3.0 of the standard while version 4.0 is currently in draft form and has favorable features:
  - Support for LDAP-based authentication/authorization.
  - More support for REST-based interfaces.

Technology Architecture

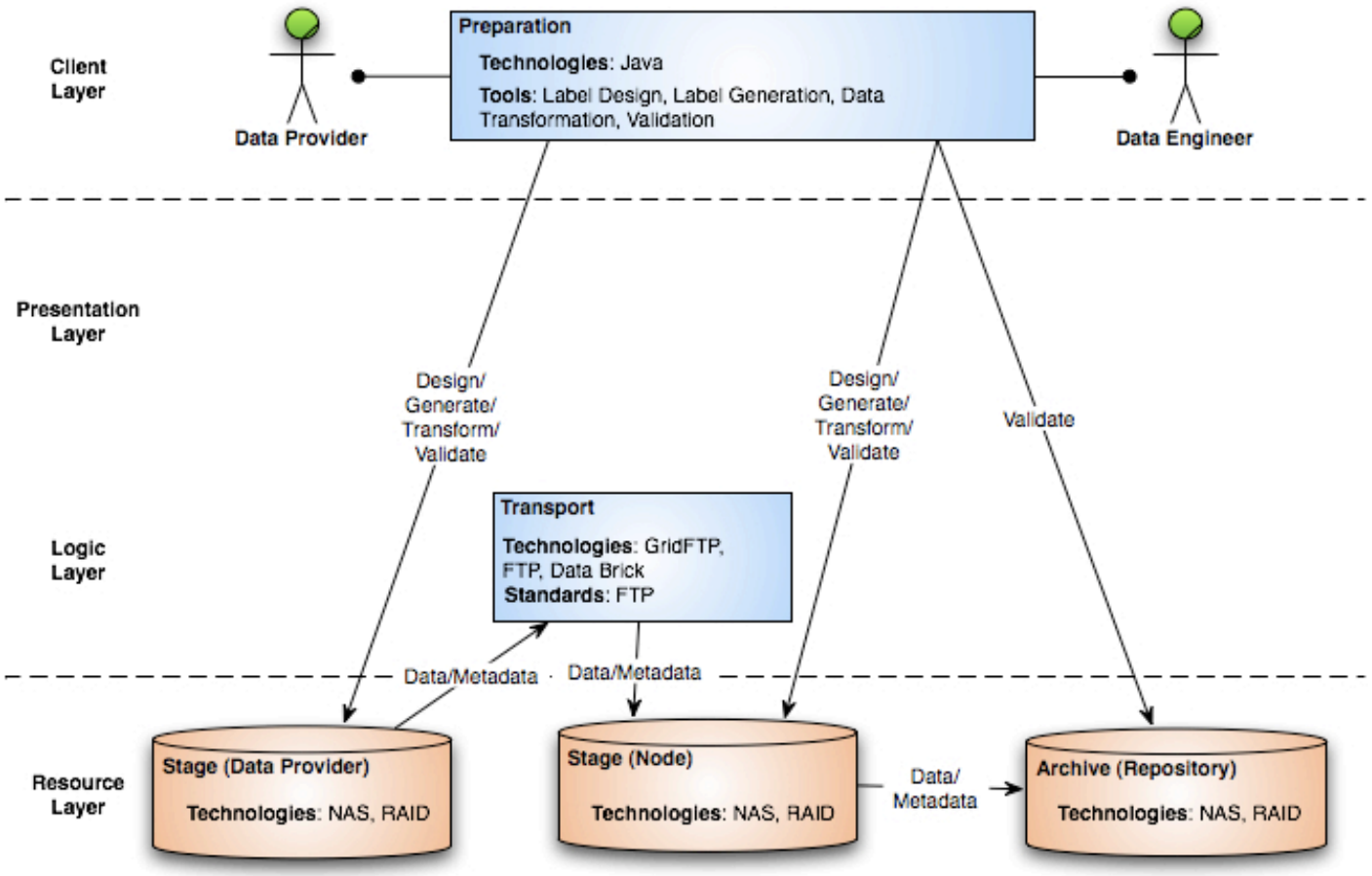


Service and Tool Details

Tools and services will be addressed according to their roles in the following scenarios:

- Data Product Preparation
  - Preparation (Tools) and Transport
- Dictionary Maintenance
  - Dictionary, Security and Operator Portal
- Catalog Ingestion
  - Ingest, Inventory, Security and Operator Portal
- Catalog Search
  - Search, Registry, Data Consumer Portal
- Data Product Ingestion
  - Ingest, Inventory, Security and Operator Portal
- Data Product Search
  - Search, Transport and Data Consumer Portal

Data Product Preparation



Data Product Preparation
Preparation

- This component consists of a suite of tools for preparing data/metadata for archive.
- Providing a suite of tools simplifies the interface between data providers and PDS.
- Allows for flexibility in pipeline integration.
  - Functions like generation and transformation can be performed by the Data Provider or the Node depending on the agreement.
- The tools will be built on a common library facilitating future tool development.
  - Similar to the current VTool model but will also incorporate data access functionality for data manipulation.
- Related to the Data Product, Grammar and Packaging (Volume) models.

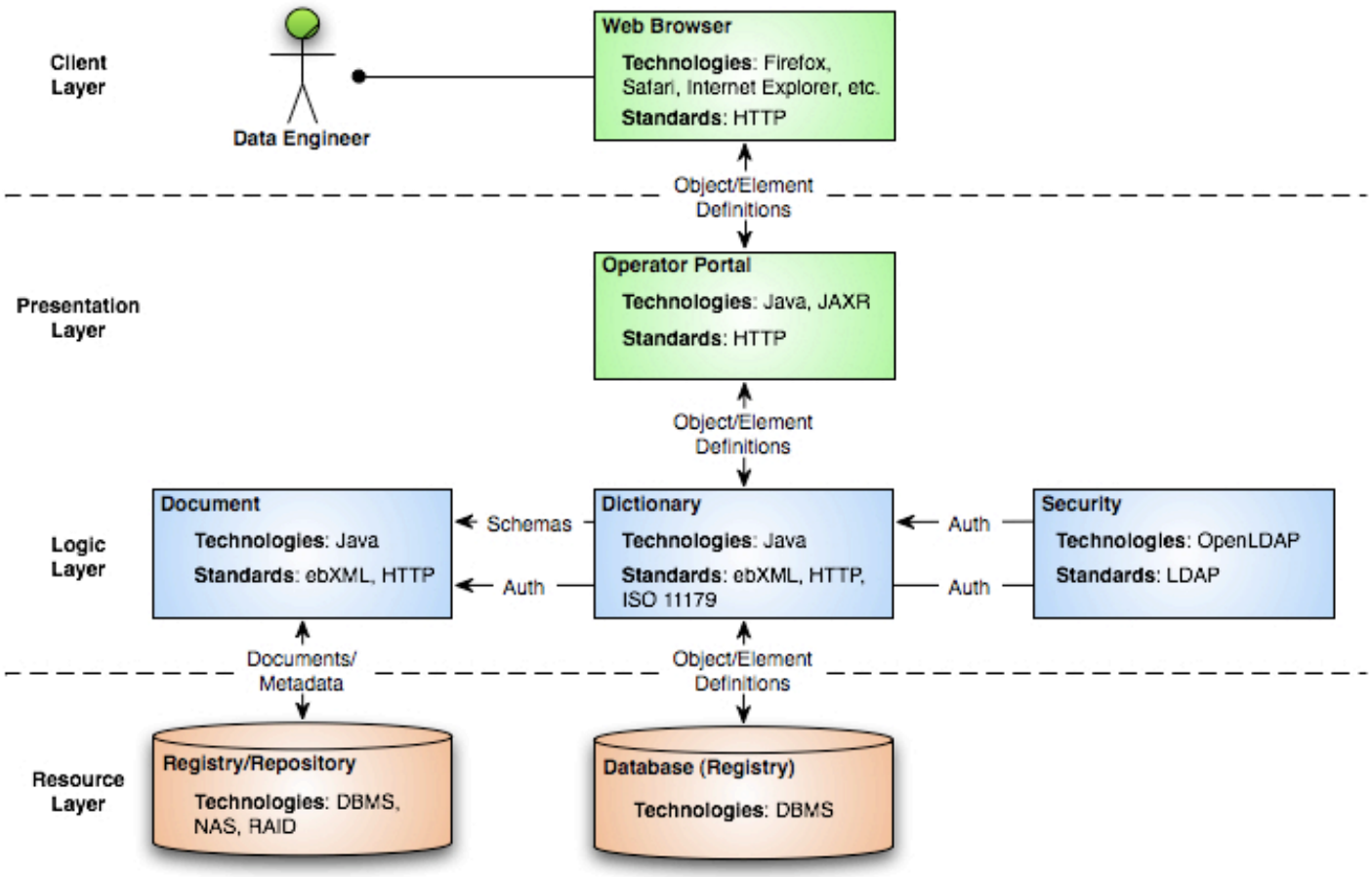
Data Product Preparation
Preparation (Tools)

- Label Design
  - This is essentially the Label Template Design Tool (LTDTool).
  - This tool's capabilities would be extended to the design of schemas and not just templates.
- Label Generation
  - This functionality is currently offered by a few Nodes.
  - The intent is to build a common function for label generation utilizing the LTDTool schemas and incorporating existing Node capability.
- Data Transformation
  - This tool provides transformation of incoming data to archive formats or from the archive to user formats.
  - Could also include packaging of data to support user requests or delivery to the deep archive.
  - Design a plug-in capability to allow for new format support to be added over time.
- Validation
  - This is essentially the Validation Tool (VTool).
  - This tool's capabilities would be extended to data and package validation.

Data Product Preparation
Transport

- Provides transport mechanisms for delivering data product(s) to the Node.
  - Data delivery mechanisms may include HTTP, FTP, GridFTP, Data Brick, etc.
- There are no direct model relations for transportation.

Dictionary Maintenance



Dictionary Maintenance
Dictionary

- A registry supporting Create, Read, Update and Delete (CRUD) functions based on authorized access.
- Metadata for registered object/element definitions are stored in a local database.
  - Most likely implemented or adopted in accordance with the ISO 11179 standard.
- Communication with the registry is facilitated via an API (e.g., Java API for XML Registries (JAXR)).
- Related to the Data Dictionary model.

Dictionary Maintenance
Document

- A registry supporting Create, Read, Update and Delete (CRUD) functions based on authorized access.
- Metadata for registered documents are stored in a local database with the document files stored in a local repository.
- Communication with the registry is facilitated via an API (e.g., Java API for XML Registries (JAXR)).
- No direct model relations.

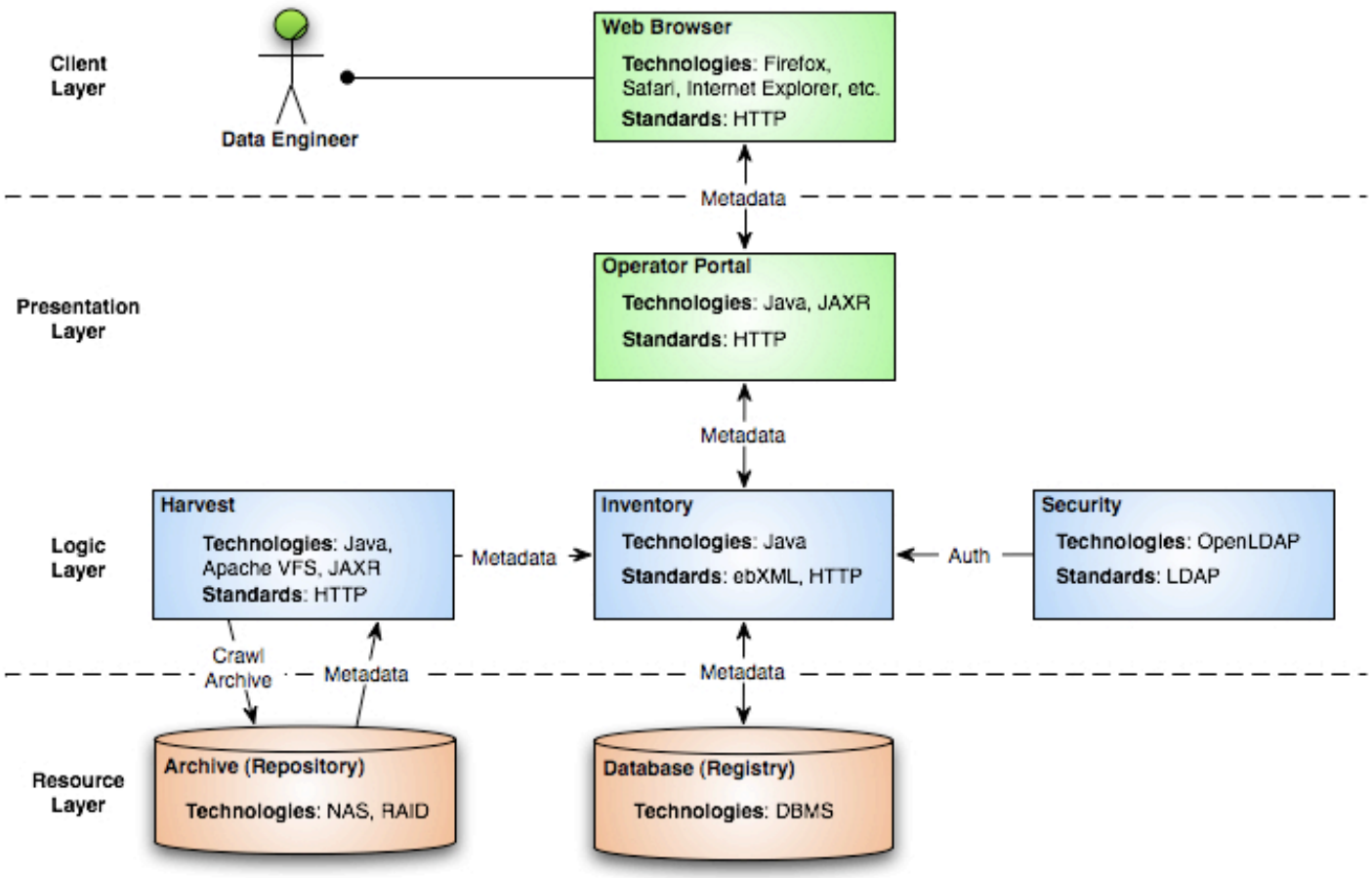
Dictionary Maintenance
Security

- An LDAP-based (Lightweight Directory Access Protocol) service providing authentication and authorization for CUD functions with the Dictionary service.
- Related to the Personnel model.

Dictionary Maintenance
Operator Portal

- A specialized web service for interacting with the Dictionary service.
- No direct model relations.

Catalog Ingestion



Catalog Ingestion
Inventory

- A registry supporting Create, Read, Update and Delete (CRUD) functions based on authorized access.
- Metadata for registered objects (e.g., Mission, Instrument, Target, etc.) are stored in a local database.
  - Most likely implemented with a customized database schema similar to the current catalog database.
- Communication with the registry is facilitated via an API (e.g., Java API for XML Registries (JAXR)).
- Related to the Mission, Instrument, Target, etc. models.

Catalog Ingestion
Harvest

- Specifics regarding Harvest will be discussed in the Data Product Ingestion portion.
- If catalog files persist in PDS 2010, this service could be used to extract metadata from the catalog files and register that information with the Inventory service.

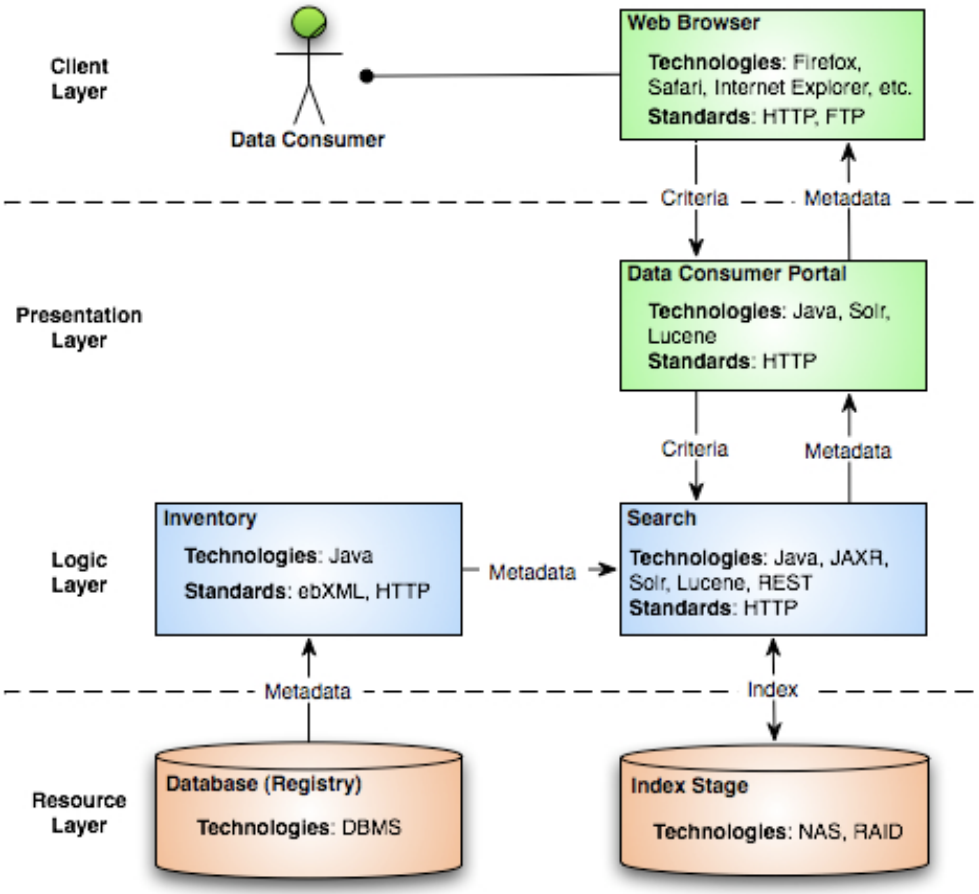
Catalog Ingestion
Security

- An LDAP-based (Lightweight Directory Access Protocol) service providing authentication and authorization for CUD functions with the Inventory service.
- Related to the Personnel model.

Catalog Ingestion
Operator Portal

- A specialized web service for interacting with the Inventory service.
- Could be the main interface for ingesting and maintaining catalog-level metadata.
  - Whether or not catalog files persist.
- No direct model relations.

Catalog Search



Catalog Search
Inventory

- A registry supporting Create, Read, Update and Delete (CRUD) functions based on authorized access.
- Metadata for registered objects (e.g., Mission, Instrument, Target, etc.) are retrieved from a local database.
  - Most likely implemented with a customized database schema similar to the current catalog database.
- Communication with the registry is facilitated via an API (e.g., Java API for XML Registries (JAXR)).
- Related to the Mission, Instrument, Target, etc. models.

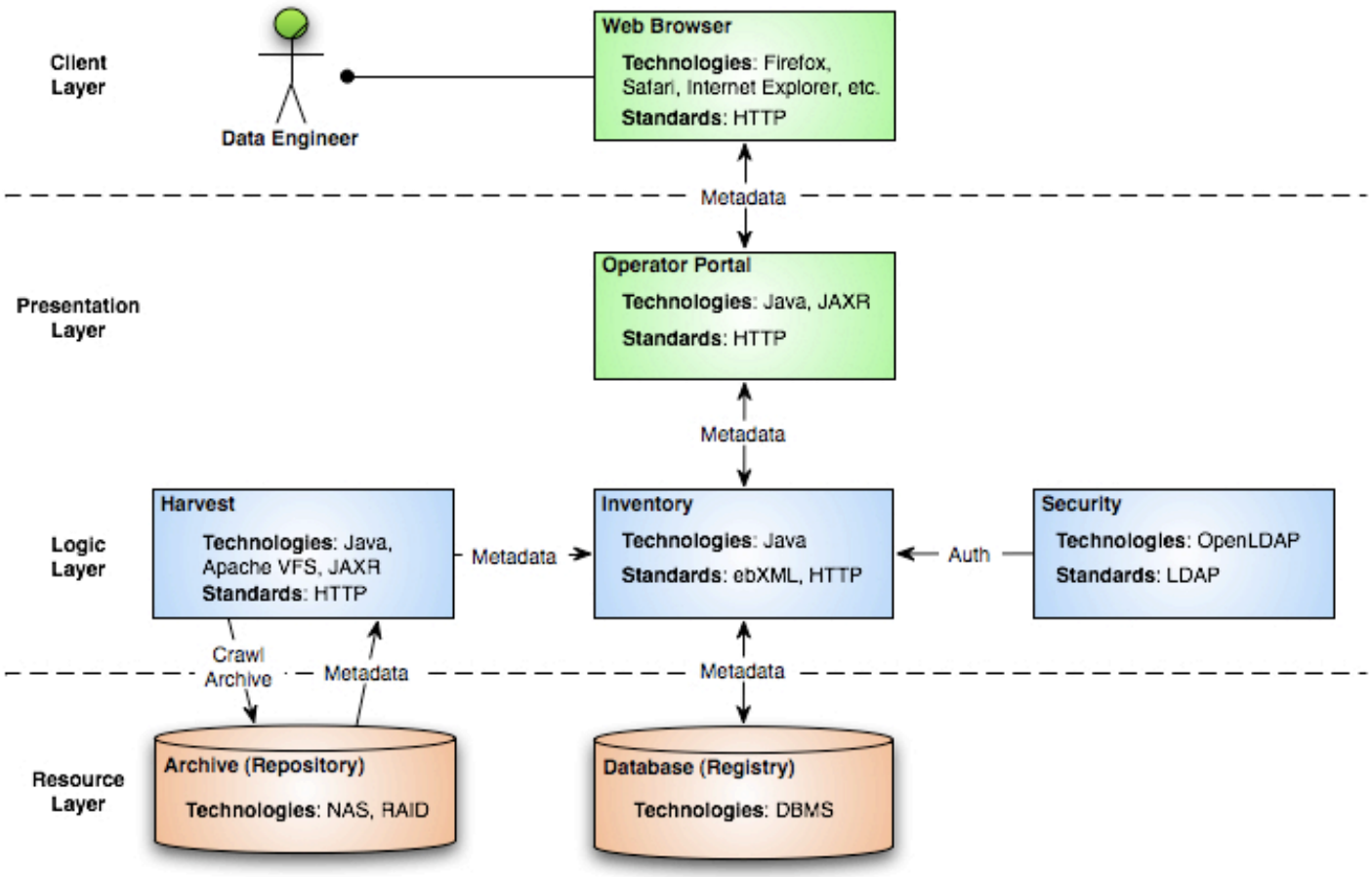
Catalog Search
Search

- Provides periodic extraction of metadata from the Inventory for index creation to facilitate indexed search.
  - Indexed search offers a significant performance increase over querying registries, specifically distributed registries, directly.
  - Allows the metadata to be captured and organized to facilitate user queries as they evolve over time without having to modify the underlying registry model.
- Provides a REST-based query interface of the catalog-level metadata against the index.
- The protocol for the REST-based query will be based on the PDS Query model.
- Related to the Query model.

Catalog Search
Data Consumer Portal

- A generic web service for interacting with the Search service.
- No direct model relations.

Data Product Ingestion



Data Product Ingestion
Harvest

- Could consist of a configurable crawler function possibly built on Apache (Virtual File System) VFS enabling local or remote access to archive.
- Can be run as a daemon process, periodically waking up to check for new or modified data products.
- Metadata are extracted from a data product and registered with the Inventory service.
  - Initial support for common elements.
  - Extensible by the Node for discipline-specific elements.
- This approach is backward compatible, allowing PDS3 and prior data products to be registered.
- Related to the Packaging (Volume) and Grammar models.

Data Product Ingestion
Inventory

- A registry supporting Create, Read, Update and Delete (CRUD) functions based on authorized access.
- Metadata for registered data products are stored in a local database.
  - Initial support for common elements.
  - Extensible by the Node for discipline-specific elements.
- The data files associated with the registered data products remain in a separate archive repository.
- Communication with the registry is facilitated via an API (e.g., Java API for XML Registries (JAXR)).
- Related to the Data Product model.

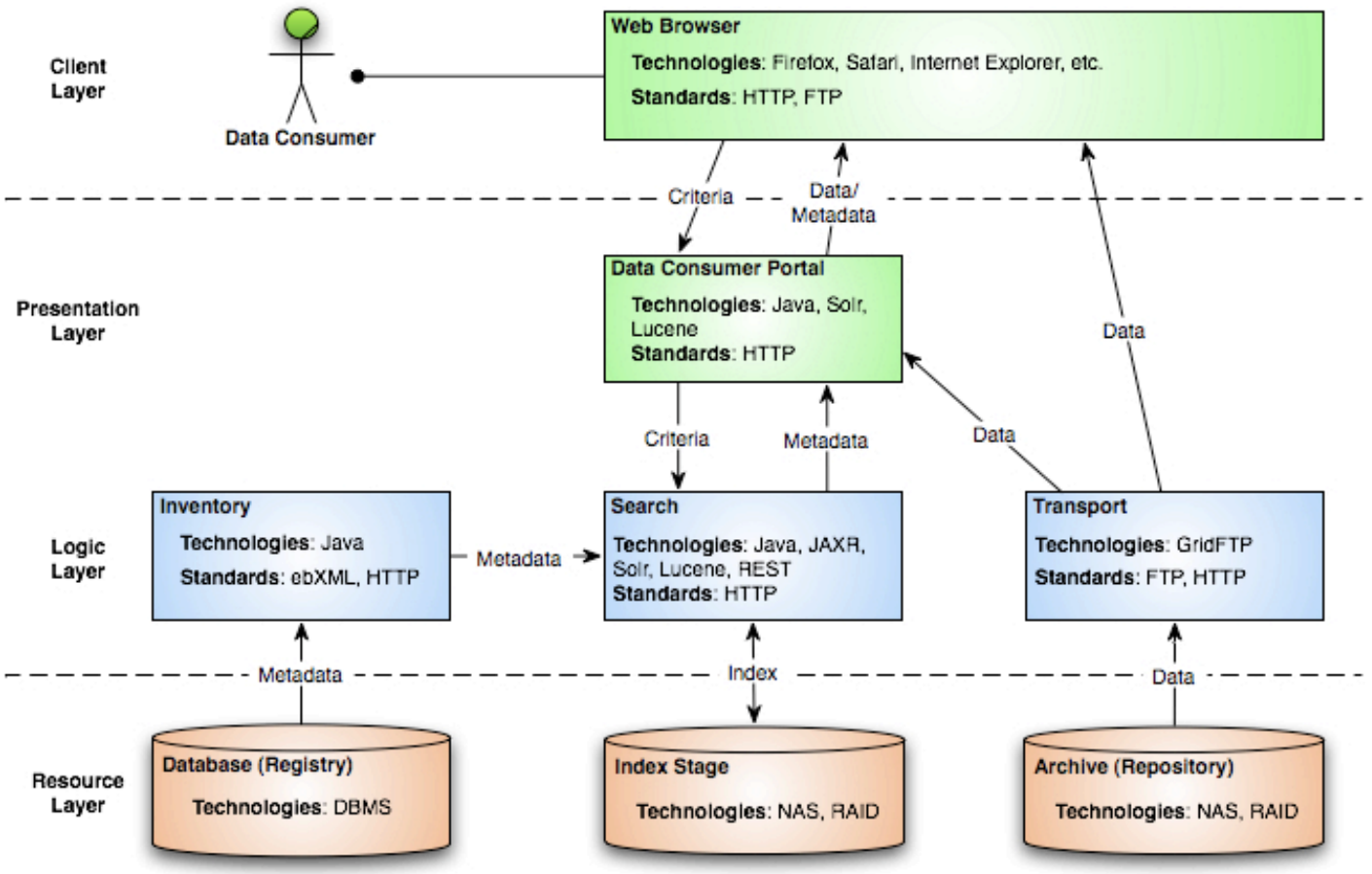
Data Product Ingestion
Security

- An LDAP-based (Lightweight Directory Access Protocol) service providing authentication and authorization for the create, update and delete functions with the Inventory service.
- Related to the Personnel model.

Data Product Ingestion
Operator Portal

- A generic web service for interacting with the Inventory service.
- Could be an interface for maintaining data product metadata.
- No direct model relations.

Data Product Search



Data Product Search Inventory

- A registry supporting Create, Read, Update and Delete (CRUD) functions based on authorized access. Focused on Read.
- Metadata for registered data products are retrieved from a local database.
  - Initial support for common elements.
  - Extensible by the Node for discipline-specific elements.
- Communication with the registry is facilitated via an API (e.g., Java API for XML Registries (JAXR)).
- Related to the Data Product model.

Data Product Search
Search

- Provides periodic extraction of metadata from the Inventory for index creation to facilitate indexed search.
  - Indexed search offers a significant performance increase over querying registries, specifically distributed registries, directly.
  - Allows the metadata to be captured and organized to facilitate user queries as they evolve over time without having to modify the underlying registry model.
- Provides a REST-based query interface of the data product metadata against the index.
- The protocol for the REST-based query will be based on the PDS Query model.
- Related to the Query model.

Data Product Search
Transport

- Provides transport mechanisms for returning discovered data product(s) to the data consumer.
  - Data retrieval mechanisms include HTTP, FTP, GridFTP, etc.
- Depending on the mechanism, transformation or packaging of data product(s) will be performed per the request.
- There are no direct model relations for transportation but transformation is related to the Data Product model.

Data Product Search
Data Consumer Portal

- A Node-specific web service for interacting with the Search service.
- No direct model relations.

Design Goal Evaluation

- Improve ingestion efficiency (catalog and data products).
  - Introduced a streamlined mechanism for ingestion of data products.
  - Catalog ingestion is facilitated through an online interface alleviating the need to pass around catalog files.
- Facilitate tracking and improve integrity of the archive.
  - Population of data product registries enables detailed tracking and assuming we capture checksum and file size information, integrity checking can also be facilitated.

Design Goal Evaluation (continued)

- Facilitate data product search across nodes.
  - Local data product registries at each Node (using common metadata), facilitate high-level search across the Nodes.
  - Extensible metadata capture in these registries facilitates correlation of like data.
- Improve delivery of data to users and deep archive.
  - Facilitate new mechanisms for packaging and transporting data.

Design Goal Evaluation (continued)

- Increase integration of software services across the Nodes and the system as a whole.
  - Introduce common services deployed centrally as well as locally at the Nodes.
- Keep it simple
  - Implement a lightweight SOA solution.
  - Minimize the impact on existing interfaces and processes.

Plans

- Capture use cases and requirements for each component.
- Generate a detailed design for each component and begin implementation for core components (e.g., Security, Service, Report and Dictionary).
- Work with the Data Design WG to develop a query model and a corresponding search protocol

Questions/Comments