System Core Concepts

Technical Session
Pasadena, CA
February 28 – March 2, 2011

Sean Hardman

Topics

- Design Principles, Goal and Considerations
- Core Concepts
  - Service-Based Design
  - Enhanced Tool Suite
  - System of Registries
  - Common Search
- System Layering

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.

Core Concepts

- Service-Based Design
  - Support remote access to data and services to bring the federation together both for ingestion and distribution.
- Enhanced Tool Suite
  - A tool-based approach is still appropriate for certain functions.
- System of Registries
  - Adopt a system of registries to support improved tracking and access.
- Common Search
  - A publicly available layer facilitating search across PDS.

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.
  - A service-based architecture provides currency and timeliness for the system.
- Service-based functionality will focus on public interfaces for search, retrieval and value-added processing (science services) of data.

Service-Based Design
Service Interfaces

- 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 applies mainly to the Registry and Search services.
- Other services that integrate COTS or open source solutions will utilize their provided interfaces (e.g., LDAP).

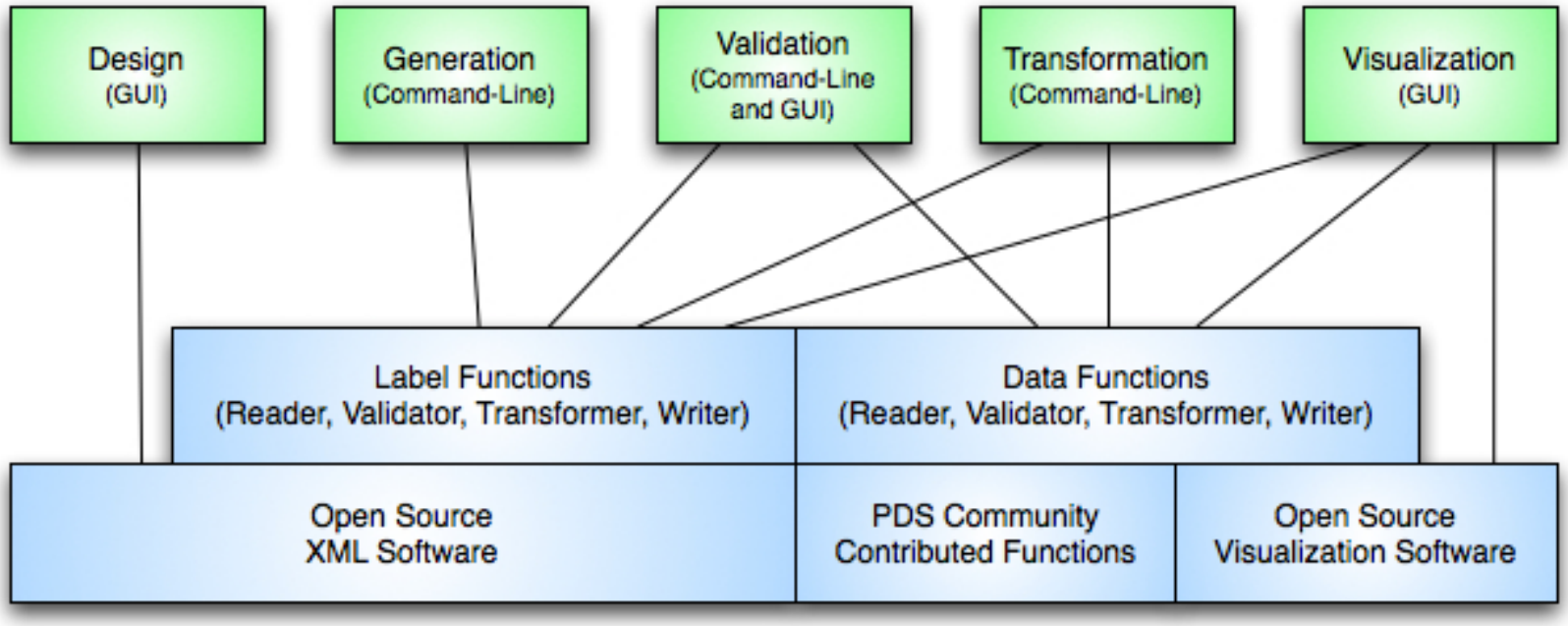
Enhanced Tool Suite

- There are several functions related to product labels and data where a tool is an appropriate interface:
  - **Design** of product label schemas
  - **Validation** of products and collections of products
  - **Generation** of product labels in a pipeline
  - **Transformation** of product formats (labels and data)
  - **Visualization** of image data
- Tool-based interfaces refers to standalone command-line and GUI applications executed on a user's desktop machine.

Enhanced Tool Suite
Approach

- These capabilities provided in the form of software libraries with command-line interfaces.
- Develop a common set of functions and make them available in Java-based and Python-based libraries.
- These functions are then available to the tools and services within the system.
- Portions of the library will be open for contributions from the PDS community.

Enhanced Tool Suite
Architecture



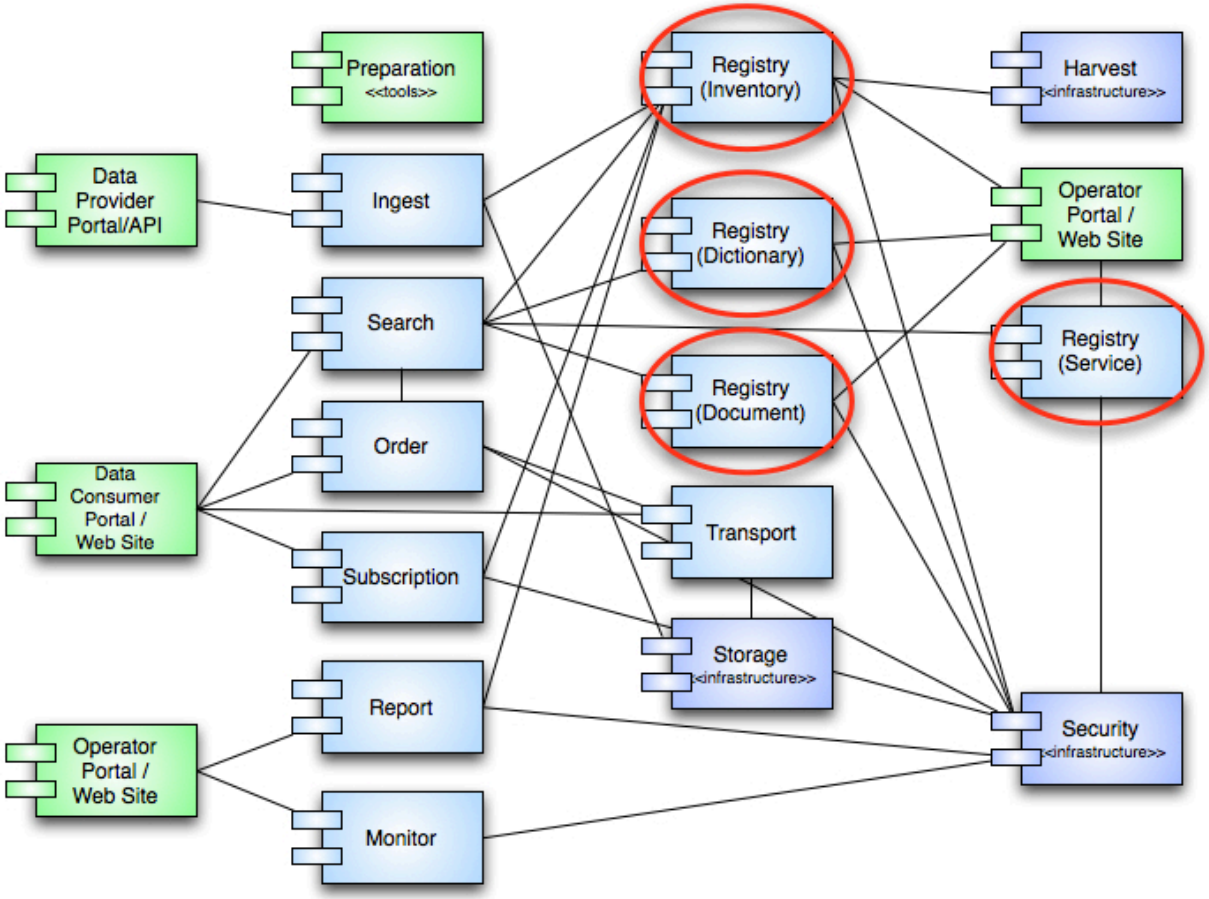
System of Registries

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

System of Registries
Registry Service

- Provides functionality for tracking, auditing, locating, and maintaining artifacts within the system.
  - Artifacts range from products consisting of data files and label files, schemas, dictionary definitions for objects and elements, service definitions, etc.
- Provides a common implementation for registry service instances.
- Design based on a CCSDS effort that builds on the ebXML standard.

System of Registries
Registry in Context



System of Registries
One Implementation, Multiple Uses

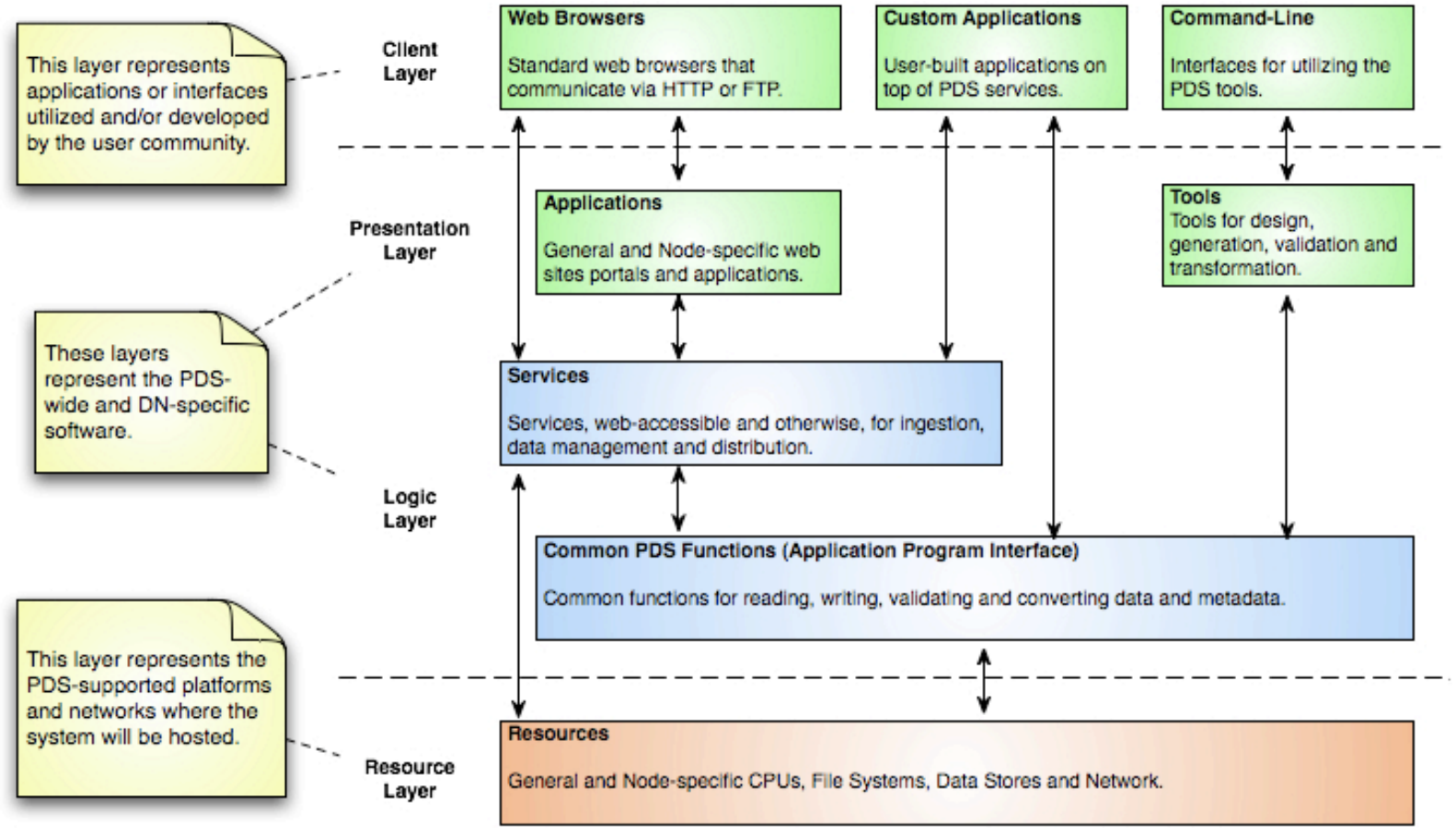
- Inventory
  - Captures catalog and product metadata in a number of distributed instances deployed at the Engineering and Discipline Nodes.
  - Tracks catalog and product artifacts from mission delivery to deep archive.
- Dictionary
  - Captures the data dictionary, which consists of object/element definitions and their associations.
- Document
  - Captures project documents, product label schemas, etc.
- Service
  - Captures descriptions of PDS services and their associations with data collections.

Common Search

- The Search Service serves as the publicly available interface for information contained in the Registry Services.
- The metadata in the registries represents the contents of the PDS archives.
- This service allows that metadata to be annotated for the purposes of search.
  - For example, updates to geometry, feature identification, etc.
- Defines a common search protocol that facilitates parameter passing and search integration.

System Layering

(Layering improves reuse, maintainability and scalability.)



Wrap Up

- The design and development of the system is guided by the design principles, goals and considerations adopted early on in the project.
- The core concepts realize the system architecture which lays the groundwork for the software system.
- The design retains the spirit of the PDS as a loosely coupled federation.

Questions/Comments