

A horizontal banner image featuring a sequence of celestial bodies from left to right: Earth (blue and white), Mars (reddish-brown), and Jupiter (orange and white). The text "Planetary Data System" is overlaid in white on the right side of the banner.

Planetary Data System

# **System Architecture and Operations Concept Update**

PDS System Design Review II  
Greenbelt, Maryland  
June 21-22, 2011

Sean Hardman

# Topics

- Overview/Status
- Core Concepts
- System Components
- Operations Concept
- Wrap Up

# Overview

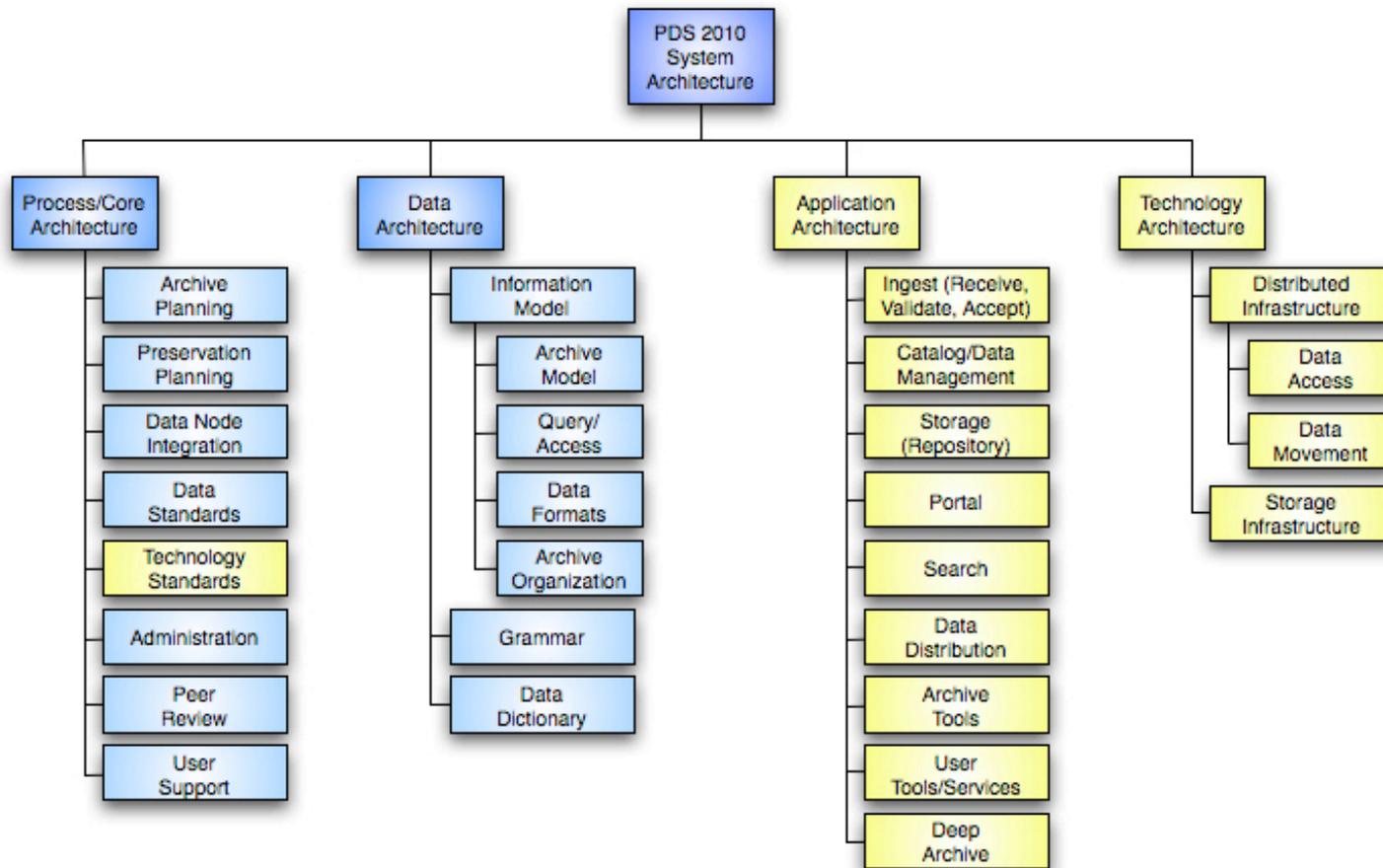
- The system architecture offers a balance between a distributed system and a centralized system.
- The system includes distributed services for product registration and search to be deployed at the Nodes along side their local data repositories.
- The system also includes centralized services for high-level search, tracking, reporting and monitoring to be deployed at the Engineering Node.

# Architecture Status

- The System Architecture has remained relatively unchanged since the first System Review (Feb 2010).
- Planetary Data System (PDS) 2010 System Architecture Specification, Version 1.2, May 25, 2011.
  - Added a derived requirement for the Portal element. Updated the service diagrams to account for the aggregate Registry Service instance.

# Architectural Elements

(Elements derived from Level 3 Requirements)



# Topics

- Overview/Status
- Core Concepts
- System Components
- Operations Concept
- Wrap Up

# Core Concepts

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

# Service-Based Design

- Service-based functionality will focus on public interfaces for search, retrieval and value-added processing (science services) of data.
- Where web-based service interfaces are planned, a REST-based interface will be implemented.
- Other services that integrate COTS or open source solutions will utilize their provided interfaces (e.g., LDAP).

# System of Registries

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

# 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

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

# 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 data
- Tool-based interfaces include standalone command-line and GUI applications executed on a user's desktop machine.

# Enhanced Tool Suite

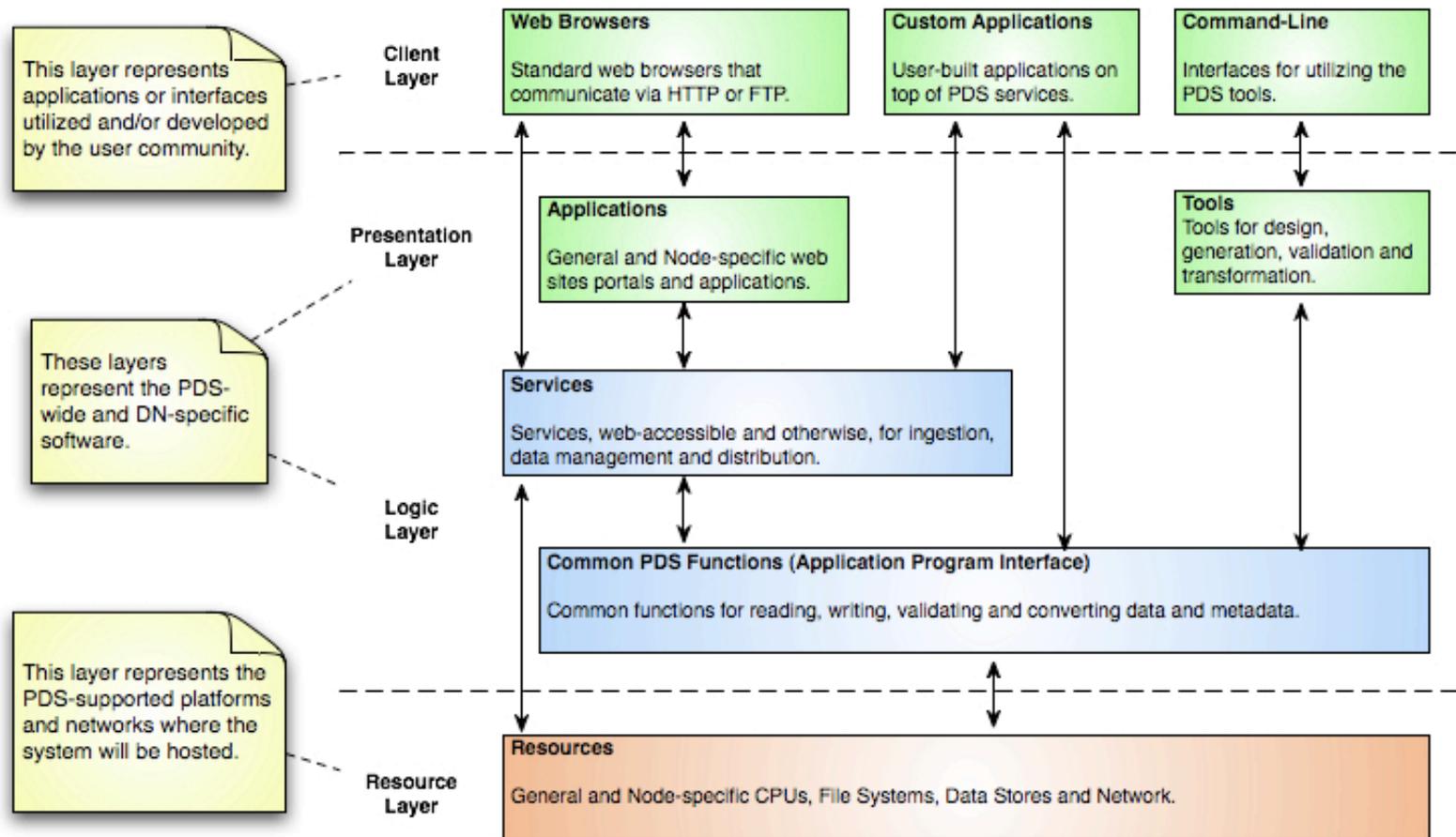
## Approach

- Capabilities provided in the form of software libraries with command-line interfaces.
- Develop a common set of functions and make them available in Java and Python-based libraries.
- 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.

# Topics

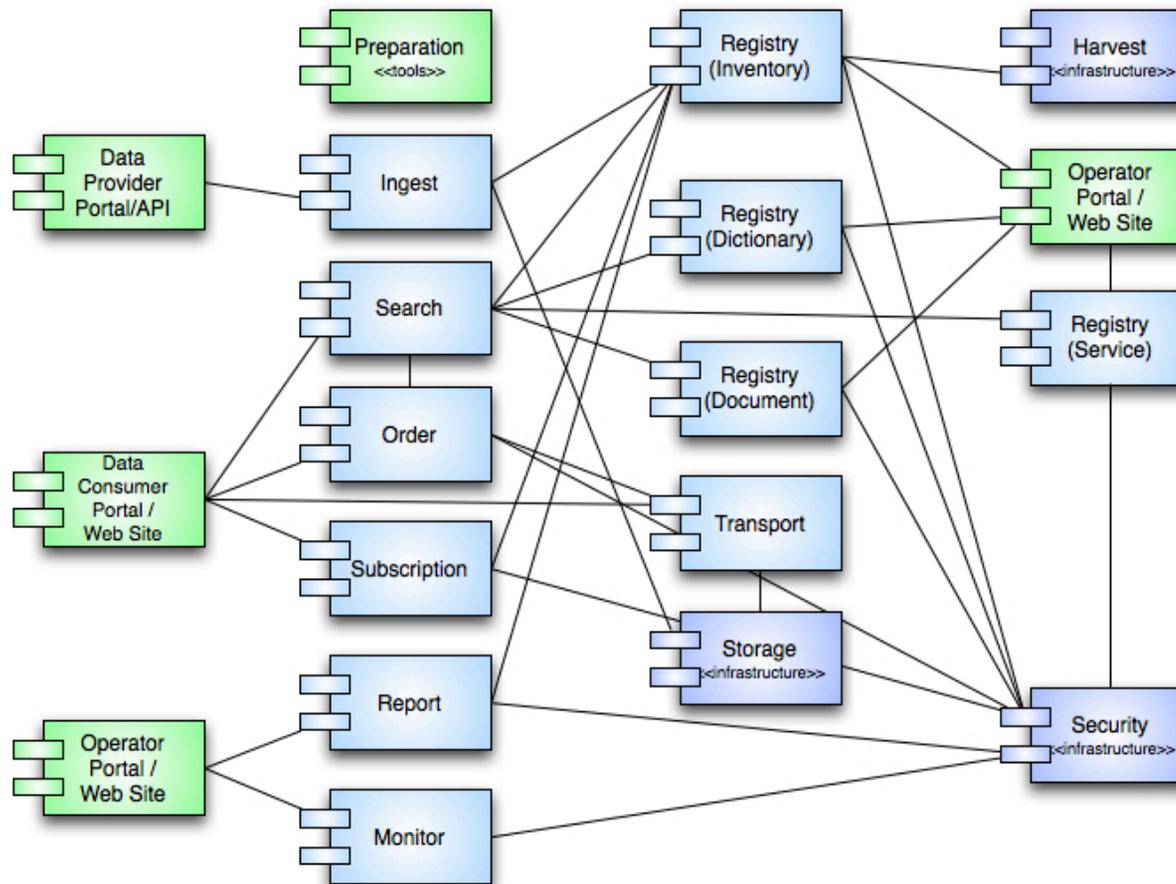
- Overview/Status
- Core Concepts
- System Components
- Operations Concept
- Wrap Up

# Component Layering



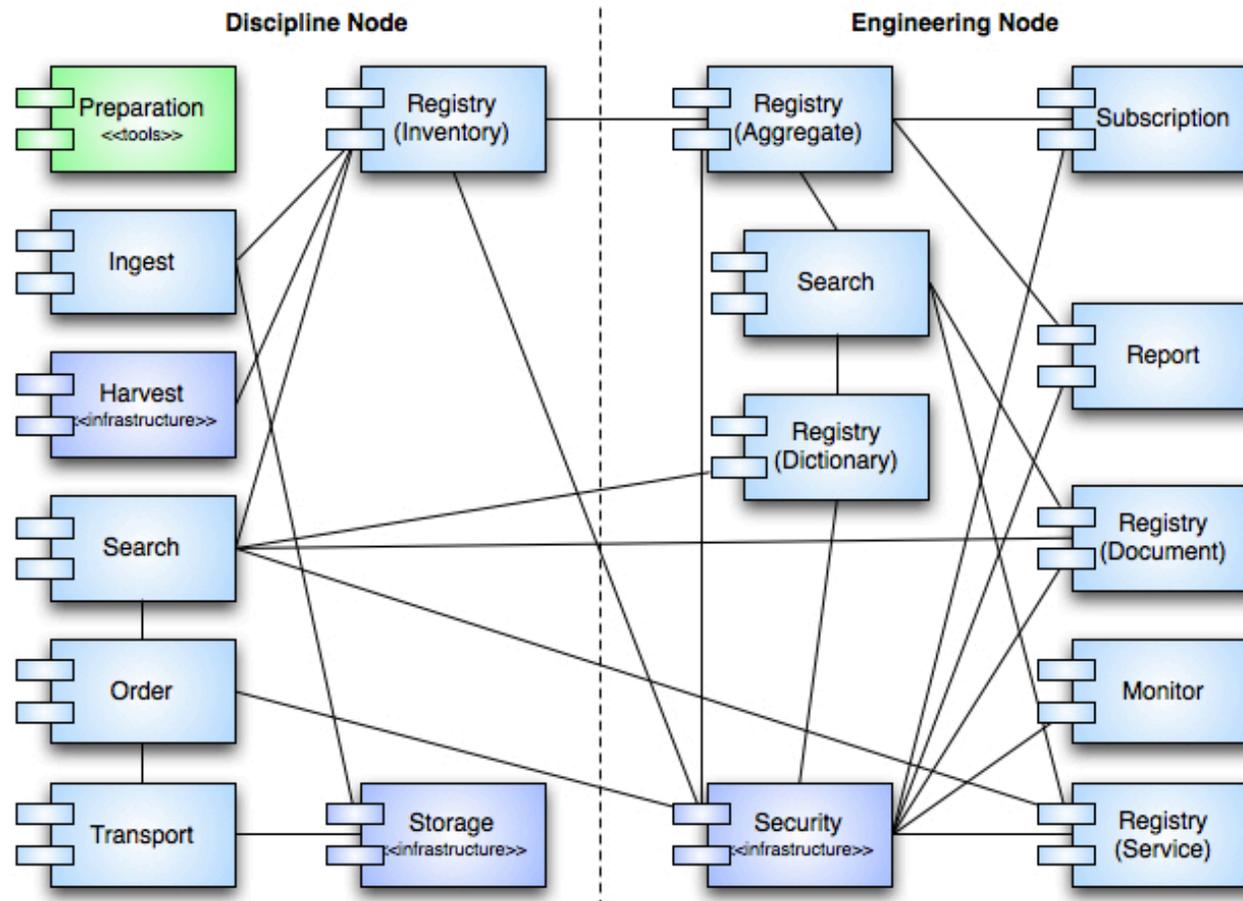
# Component Identification

## (Components derived from Architectural Elements)



# Component Provisioning

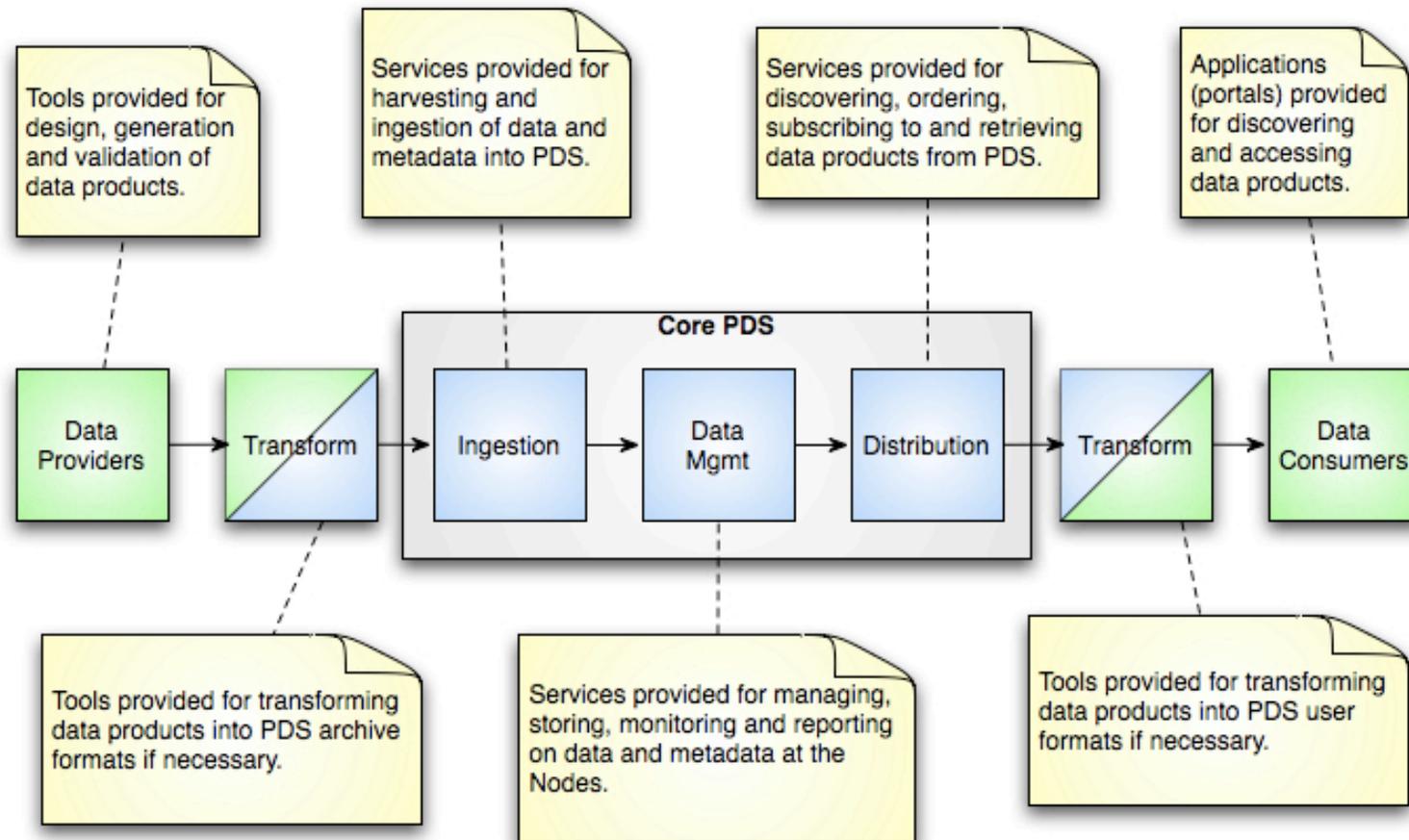
(Balanced between Centralized and Decentralized)



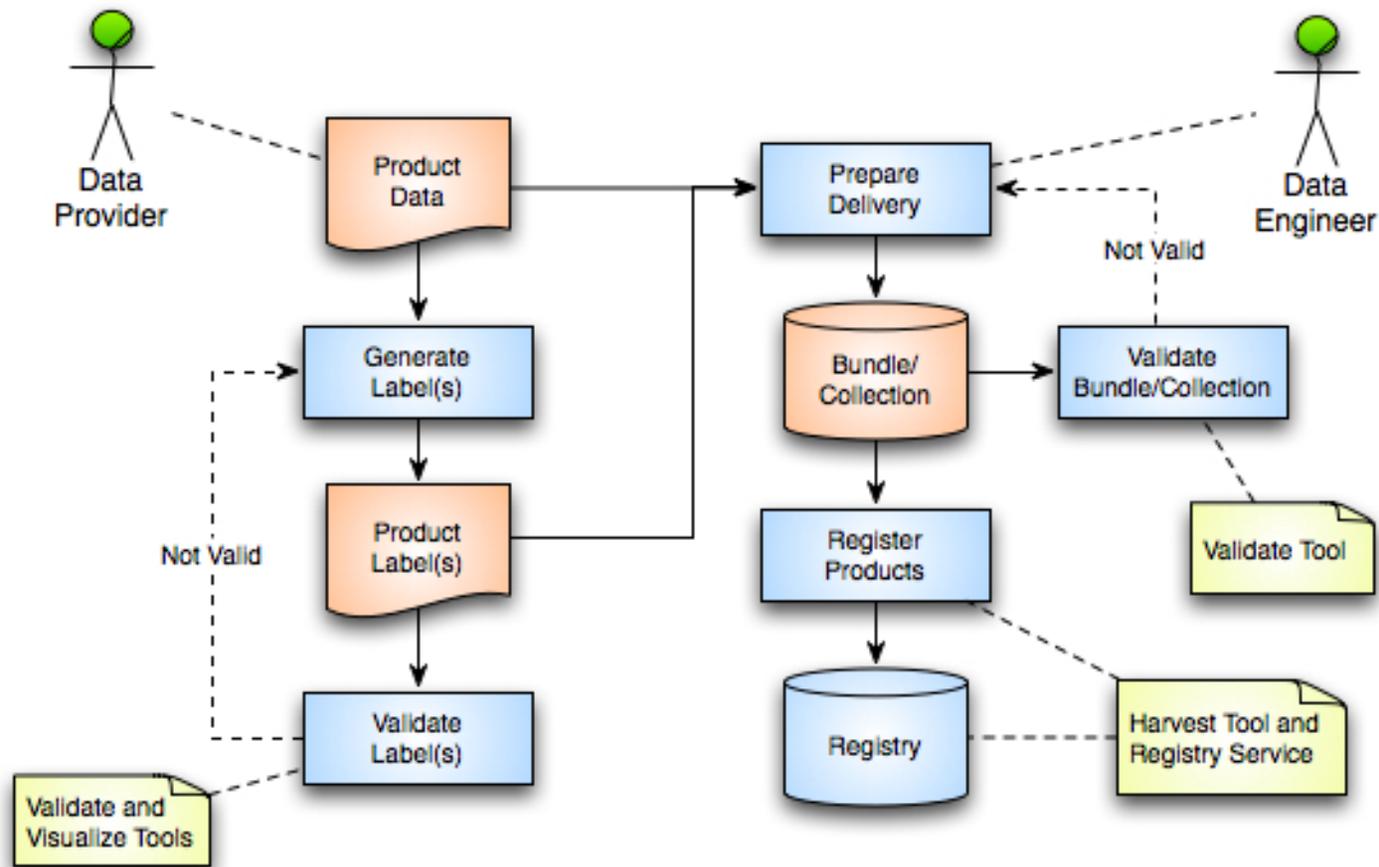
# Topics

- Overview/Status
- Core Concepts
- System Components
- Operations Concept
- Wrap Up

# Operations Concept

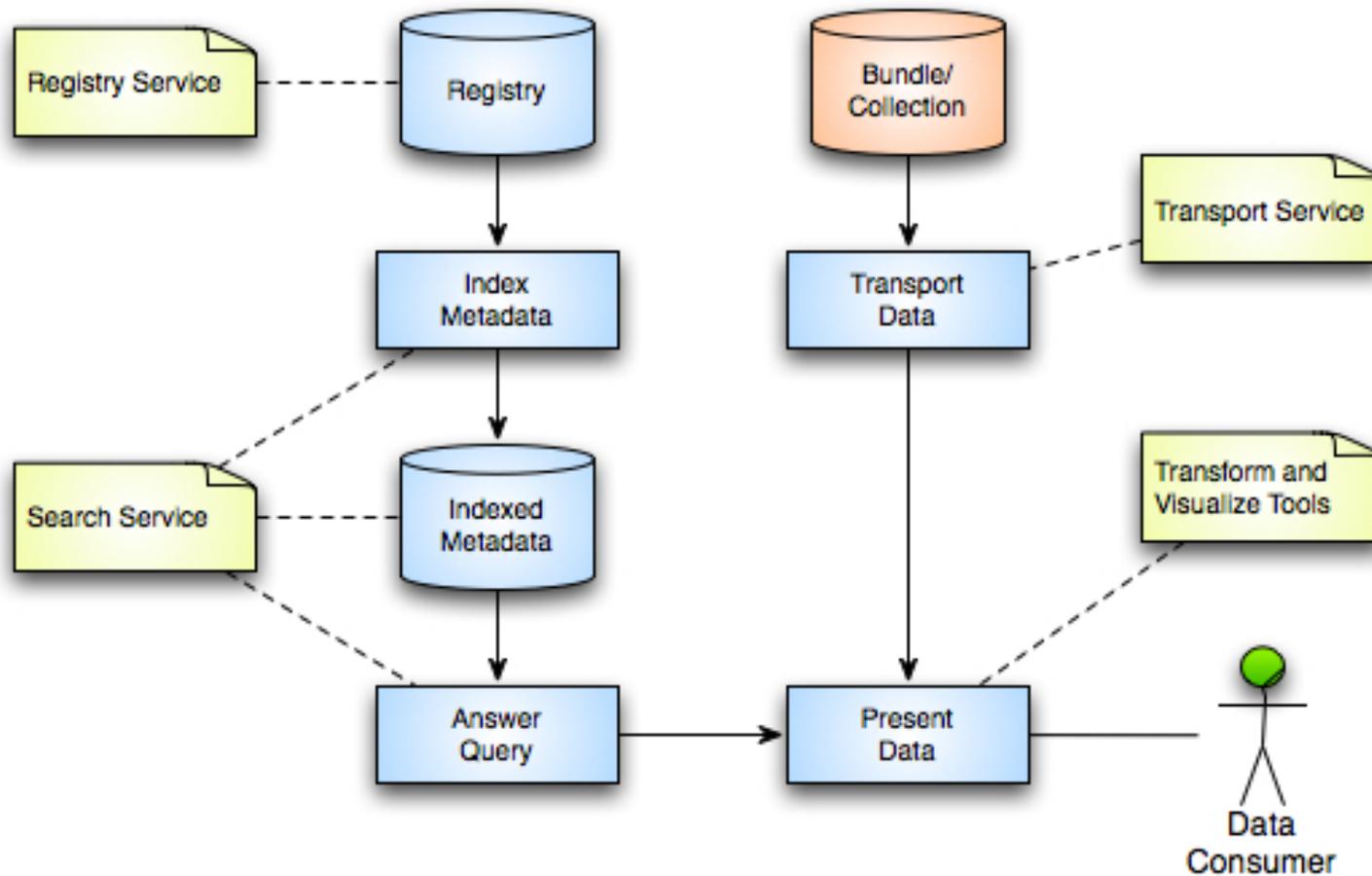


# Operations Concept Product Lifecycle (Ingestion)



# Operations Concept

## Product Lifecycle (Distribution)



# Topics

- Overview/Status
- Core Concepts
- System Components
- Operations Concept
- Wrap Up

# Wrap Up

- The system architecture retains the spirit of the PDS as a loosely coupled federation.
- The operations concept works with existing PDS policies and procedures and strives to minimize the impact on the Nodes.

# Questions / Comments