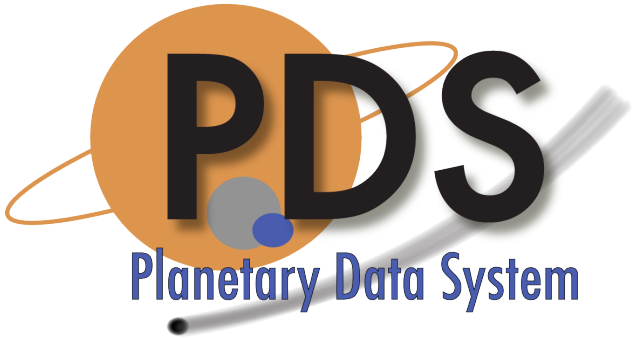



# Data Ingestion and Registration

PDS Technical Session
Pasadena, California
September 21-23, 2016

Sean Hardman

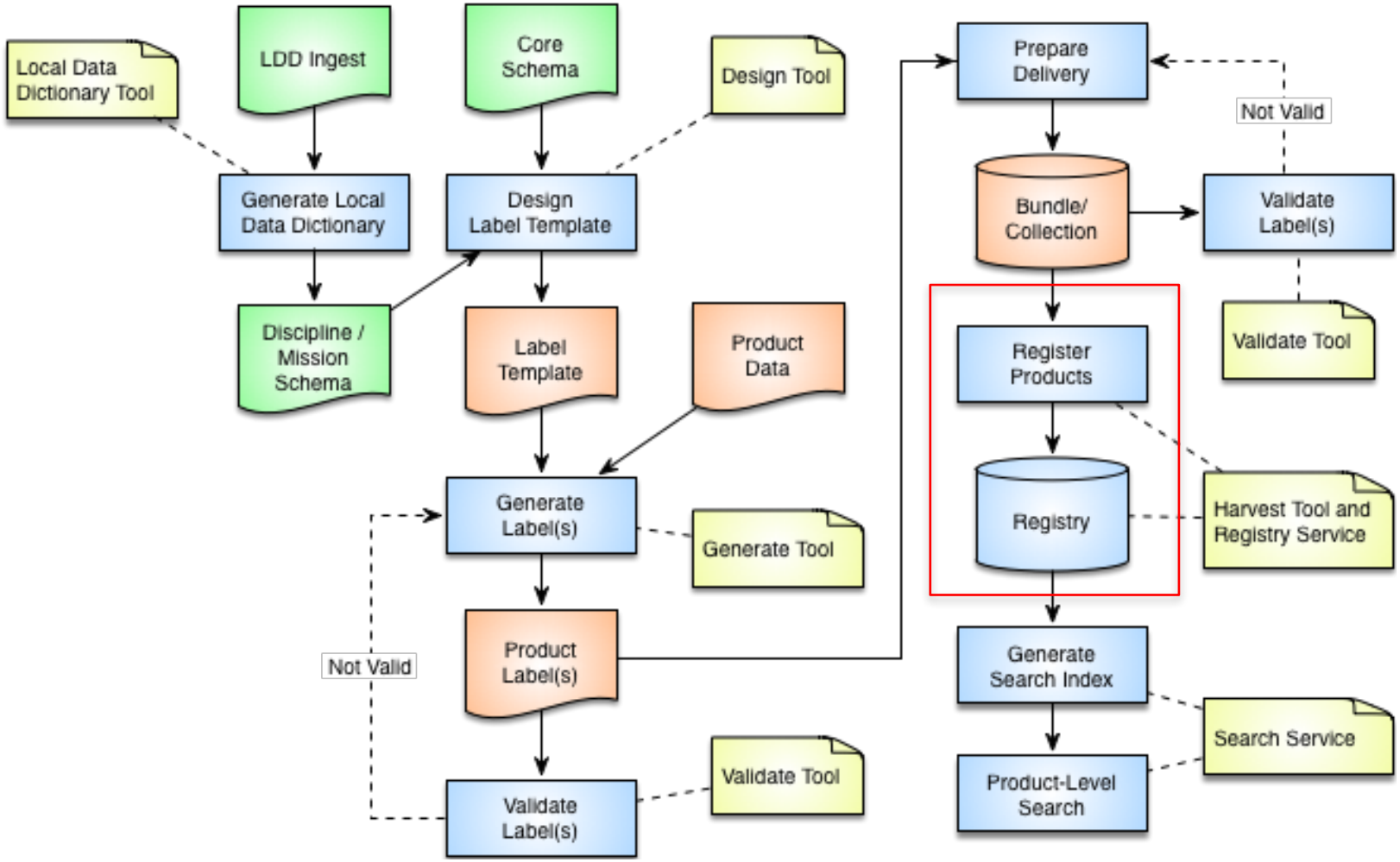
# Topics

- Overview
- Lifecycle
- Harvest Tool and its Configuration
- Registry Service and it Configuration
- Ingestion Process
- Service Status/Usage

# Overview

- Ingestion involves harvesting and registration of PDS products.
- The Harvest and Registry components facilitate ingestion of these products.
  - Although they support PDS3 products, this presentation focuses on PDS4 products.
- The following product lifecycle diagram details where the components fit into the process.

# Product Lifecycle



# Harvest Tool

- Crawler-based tool for capturing and registering product metadata.
- Allows for periodic or on-demand registration of products.
- Configurable to support registration of products residing in PDS3 and PDS4 archives.
- 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.

# Harvested Metadata

## Identification Area

- The logical identifier and version identifier become the unique identifier for the product.
- Title becomes the display name.
- Product class is used to classify the object type.

```
<Identification_Area>
  <logical_identifier>
urn:nasa:pds:example.dph.sample_archive_bundle:data:tablechar.vg2-j-pls-5-summ-ele-mom-96.0sec-v1.0
  </logical_identifier>
  <version_id>1.0</version_id>
  <title>
Voyager Electron density and moment temperature Plasma Experiment
  </title>
  <product_class>
Product_Observational
  </product_class>
…
</Identification_Area>
```

# Harvested Metadata

## Context and Observation Areas

- These areas contain most of the queryable metadata.

```
<Observation_Area>
  <Time_Coordinates/>
  <Primary_Result_Summary>
    <purpose>Science
    </purpose>
    <processing_level>Raw
    </processing_level>
    …
</Primary_Result_Summary>
  <Investigation_Area/>
  <Observing_System/>
    <Observing_System_Component/>
  <Target_Identification/>
</Observation_Area>
```

# Harvested Metadata

## Reference List Area

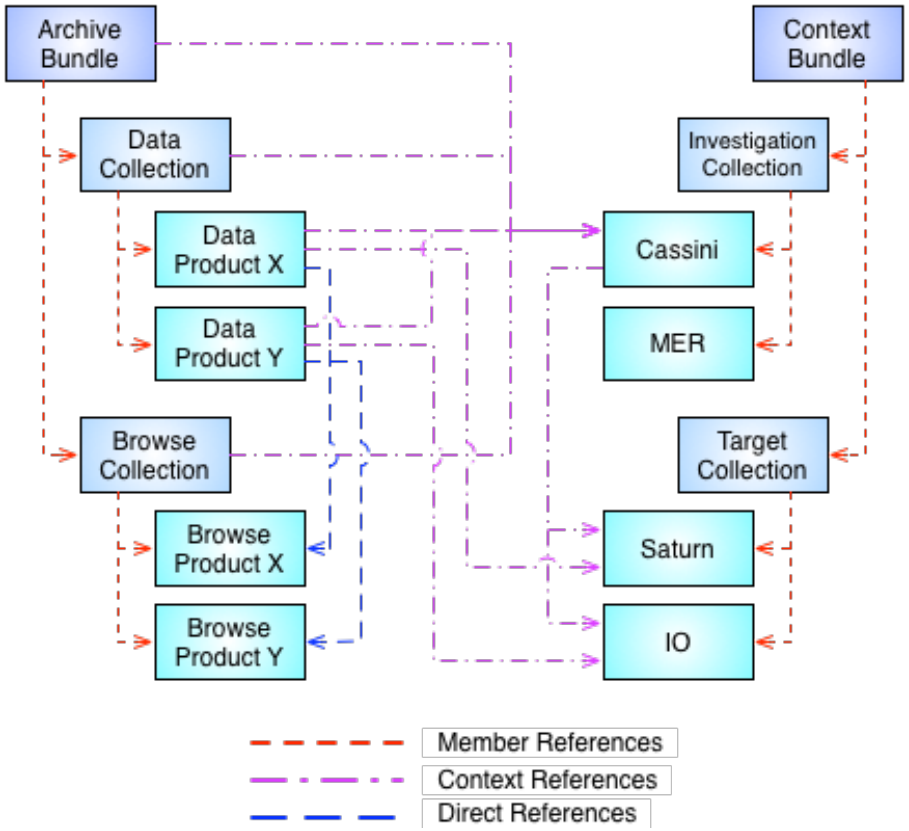
- An internal reference entry becomes a “reference” slot in the registry.
- Relates two registered objects with a reference type.
- These are used later when the search index is generated.

```
<Reference_List>
  <Internal_Reference>
    <lid_reference>
urn:nasa:pds:context:node:node.ppi
-ucla
    </lid_reference>
    <reference_type>
data_curated_by_node
    </reference_type>
  </Internal_Reference>
</Reference_List>
```

# Harvested Metadata

## Reference Types

- Member
  - Represents the logical/physical tree of the archive.
- Context
  - Informational to facilitate search.
  - Realized at bundle, collection and/or data product level.
- Direct
  - Represents ancillary information.



# Harvested Metadata

## Mission and Discipline Areas

- These areas can be harvested on demand and are likely Node/Bundle specific.
- Depends on the search requirements for the local deployment.
- Harvest can be configured to extract specific elements from the product label and place them into slots in the registry.

# Label Element to Registry Slot Mapping

- Long before there were Query Models and Property Maps, we used a spreadsheet to track the mapping of fields from the label to the Registry Service and then to the Search Service.
- These mappings are expressed in the Harvest Tool and Search Core configuration files.
- Future development will capture these in the Information Model so that default configurations can be somewhat automated.

# Harvest Configuration

- The tool comes preconfigured for harvesting the following product types:
  - Product_Attribute_Definition, Product_Bundle, Product_Class_Definition, Product_Collection, Product_Context, Product_DIP, Product_DIP_Deep_Archive, Product_SIP, Product_SIP_Deep_Archive, Product_*_PDS3
  - The preconfigured types can be overwritten.
- The configurations for the other products types (Product _Observational, etc.) can be found in the example configuration files.

# Harvest Configuration Example

```
<policy>
  <registryPackage>
    <name>Harvest Package Example Bundle Run</name>
    <description>This is a Harvest run of the example bundle.</description>
  </registryPackage>

  <collections>
    <file>$HOME/dph_example_archive_VG2PLS/browse/Collection_browse.xml</file>
    <file>$HOME/dph_example_archive_VG2PLS/context/Collection_context.xml</file>
    <file>$HOME/dph_example_archive_VG2PLS/data/Collection_data.xml</file>
    <file>$HOME/dph_example_archive_VG2PLS/document/Collection_document.xml</file>
    <file>$HOME/dph_example_archive_VG2PLS/xml_schema/Collection_xml_schema.xml</file>
  </collections>

  <directories>
    <path>$HOME/dph_example_archive_VG2PLS</path>
    <fileFilter>
      <include>*.xml</include>
    </fileFilter>
  </directories>
  …
</policy>
```

# Registry Service

- Provides functionality for tracking, auditing, locating, and maintaining artifacts within the system.
- Provides a common implementation for registry service instances based on the CCSDS Registry and Repository Reference Model.
- Provides a REST-based external interface via HTTP.
- Provides a metadata store interface for supporting different databases.

# Registry Configuration

- The information model identifies the PDS product types.
- It also identifies the common metadata elements for each of the product types.
- This information can be exported in a form to facilitate Registry Service configuration.
- Allows the information model to exert some semblance of control of the contents of the registries.

# Ingestion Process

## Harvest Tool to Registry Service

1. Harvest registers a package with the registry.
2. Harvest crawls the target directory and registers each product encountered.
   - Metadata is extracted from the product label.
   - Checksums are verified if supplied.
   - The product is registered as its specified type.
   - Product_File_Repository products are registered for each associated file.
   - Associations are registered to associate each of the above products with the package.
3. Harvest exits with a report.

# Ingestion Process

## Post Harvest

- In order to make the package and its associated products visible to Search Core, it must be approved.

- This can be accomplished via the Registry UI or the REST-based interface (e.g., curl).

# Registered Content

- Now that we have products ingested in the registry, we are ready to populate the search index.
- The Registry Service query interface is more tailored to metadata extraction than it is to user queries.
- Early on we were querying the registry directly in some of our interfaces. All interfaces now query the Search Service.

# Service Status

- The Registry Service instances deployed for PDS.
  - http://pds.nasa.gov/services/registry-pds3/
  - http://pds.nasa.gov/services/registry-pds4/
- The Registry Service instance deployed for IPDA.
  - http://planetarydata.org/registry/

# Service Usage

- The Registry Service can be queried for the purpose of metadata harvesting.
  - PDS3 instance contains the PDS3 catalog objects (data sets, missions, instruments, etc.).
  - PDS4 instance contains the PDS4 catalog objects (investigations, instruments, etc.) along with bundle and collection products.
  - IPDA instance contains tools/services and PSA data sets.

# Questions/Comments