PDS 2010 High-Level Architecture

Dan Crichton

March 22-24, 2010

Agenda

- Engineering Overview/PDS3 Implementation
- Architectural Drivers
- PDS 2010 Project Overview
- High Level Architecture Concept
- Major Design Decisions

Architecture...what is it?

- Architecture: The fundamental organization of a system embodied in its <u>components</u>, their <u>relationships</u> to each other, and to the <u>environment</u>, and the <u>principles</u> guiding its design and evolution. (ANSI/IEEE Std. 1471-2000)

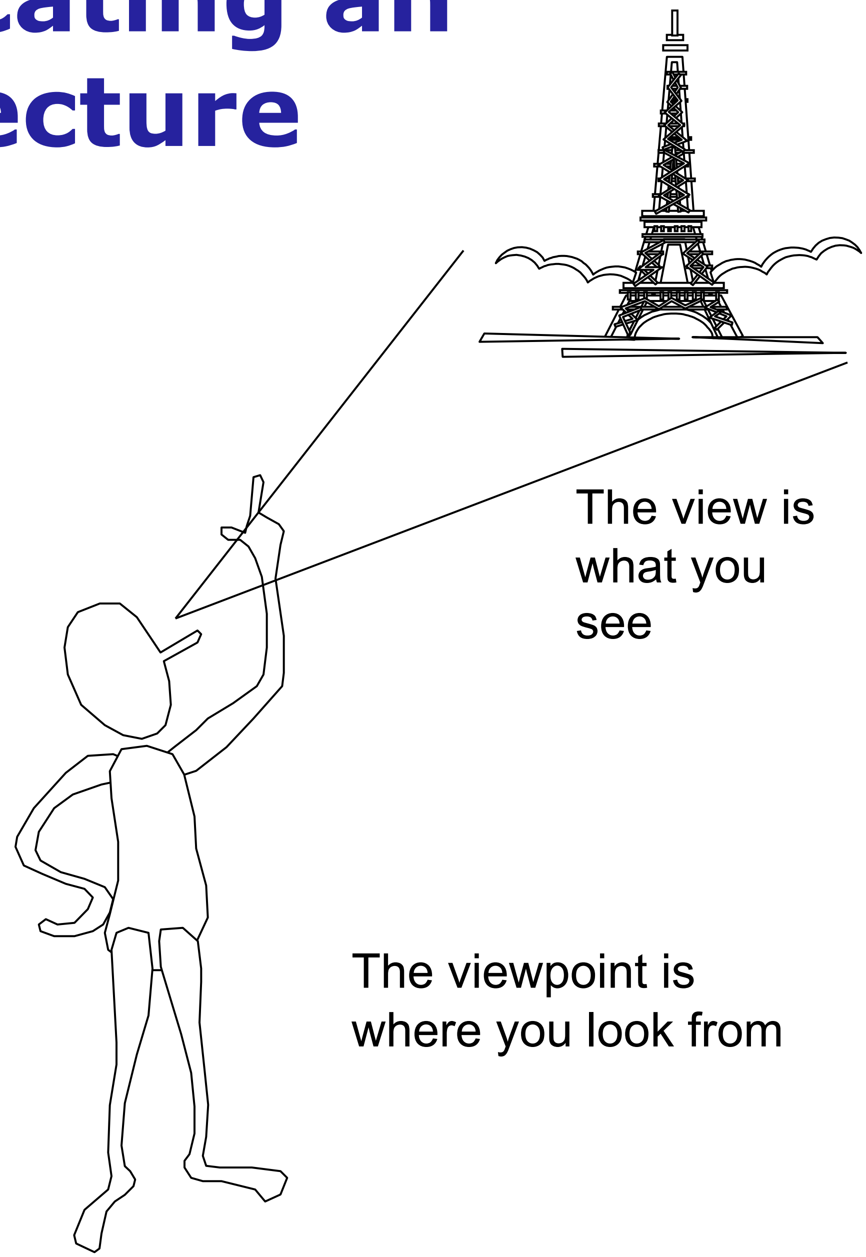
- PDS 2010 Reference System Architecture is decomposed into four core pieces:
  - Process Architecture
    - Describes the core processes PDS follows for its system
    - PDS examples: archive management, preservation planning, peer review, standards management, etc
  - Data Architecture
    - Describes the information models and data standards PDS follows for its system
    - PDS examples: PDS data model, PDS data dictionary, ODL (Grammar), etc
  - Application Architecture
    - Portals, tools, etc
  - Technology Architecture
    - Infrastructure elements

PDS Architecture Development Approach

- Identified the drivers and requirements
- Created an architectural description of PDS 2010
  - Identified stakeholders, concerns and associated models
- Identified core principles
- Separated the architecture into key viewpoints
- Created a decomposition of the system identifying the elements and mapping to the requirements
- Identified the high-level flows and analyzed from the process, information and technology perspectives
- Generated architectural models

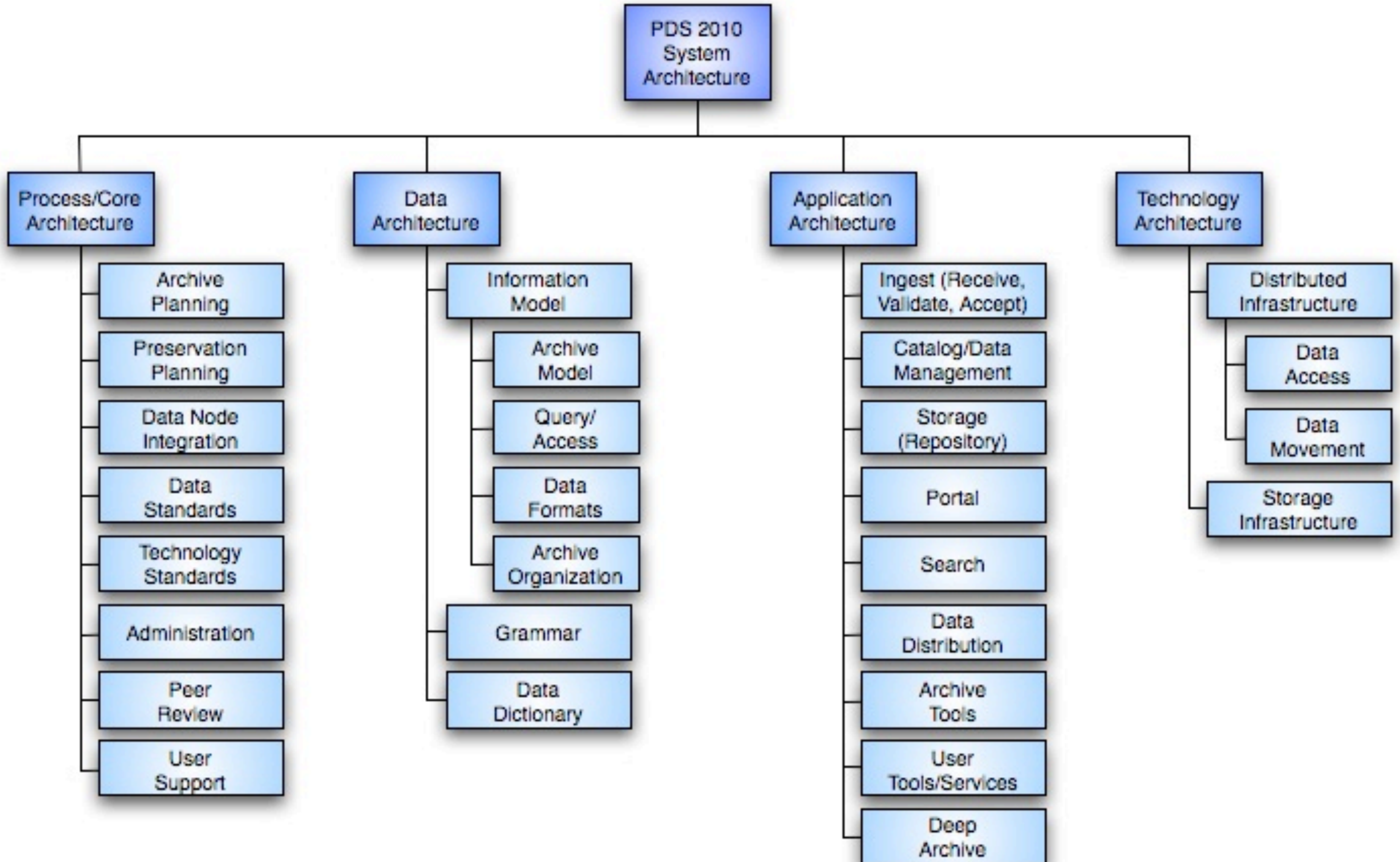
Communicating an Architecture

- One of the major challenges is communicating an architecture
  - Determine a useful view of the system for the stakeholder
  - Projects have suffered because a useful view wasn't provided
- Who are the PDS stakeholders that care about the architecture?
- How do we communicate their care-abouts?



(Management Council, System Engineers, Data Engineers, etc)

Decomposition of the Architecture



Elements derived from PDS 1,2,3 Requirements

Core Architectural Principles*

- Model driven
  - The system is based on the model
- Archiving is the priority
  - The system is designed with archiving as the priority
- Evolution of the system as elements
  - The system has a modular architecture allowing for independent evolution of elements
- Support for a distributed federation
  - Highly distributed allowing changes in federation structure and rules
- Use of standards
  - Standards are rigorously used. PDS adopts before developing, where possible
- Low cost of ownership
  - PDS ensures data providers and nodes can adopt and use tools with minimal resource impact
- Diversity
  - PDS is designed to suport diverse needs of providers, missions and planetary science community
- Scalability
  - PDS is designed to scale core functions of the system
- Explicit Design
  - Elements of the system are explicitly defined with unambiguous specifications
- International Adoption
  - Standards and tools are defined and implemented in order to allow for international adoption
- Integrity
  - Data integrity is architected into PDS processes and the system end-to-end
- Timeliness
  - PDS works with data providers as early as possible to adopt processes, standards and tools

* PDS Architecture Study Team

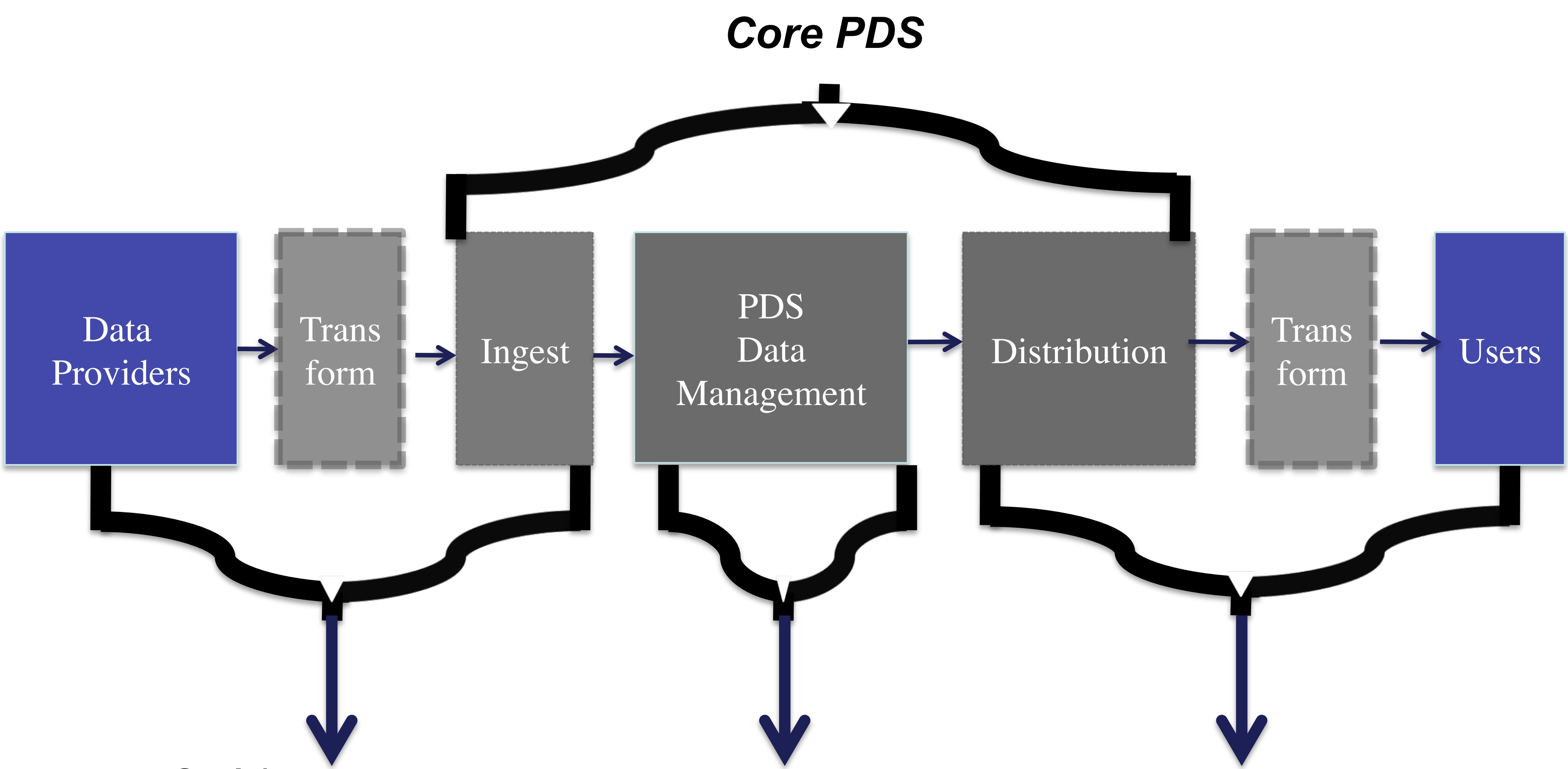
Three Major Functions of PDS

1. Delivery of Data to PDS
   - Provide data standards and software tools and services to improve the efficiency of delivery of high quality products to PDS

2. Management of Data within PDS
   - Preserve and ensure the stability and integrity of the PDS data

3. Distribution of Data from PDS
   - Provide software services, tools and standards to improve the usability of PDS and the data in the archive

Level 0 Conceptual Flow



Goal: Improve efficiency and support to deliver high quality science products to PDS

Goal: Preserve and ensure the stability and integrity of PDS data

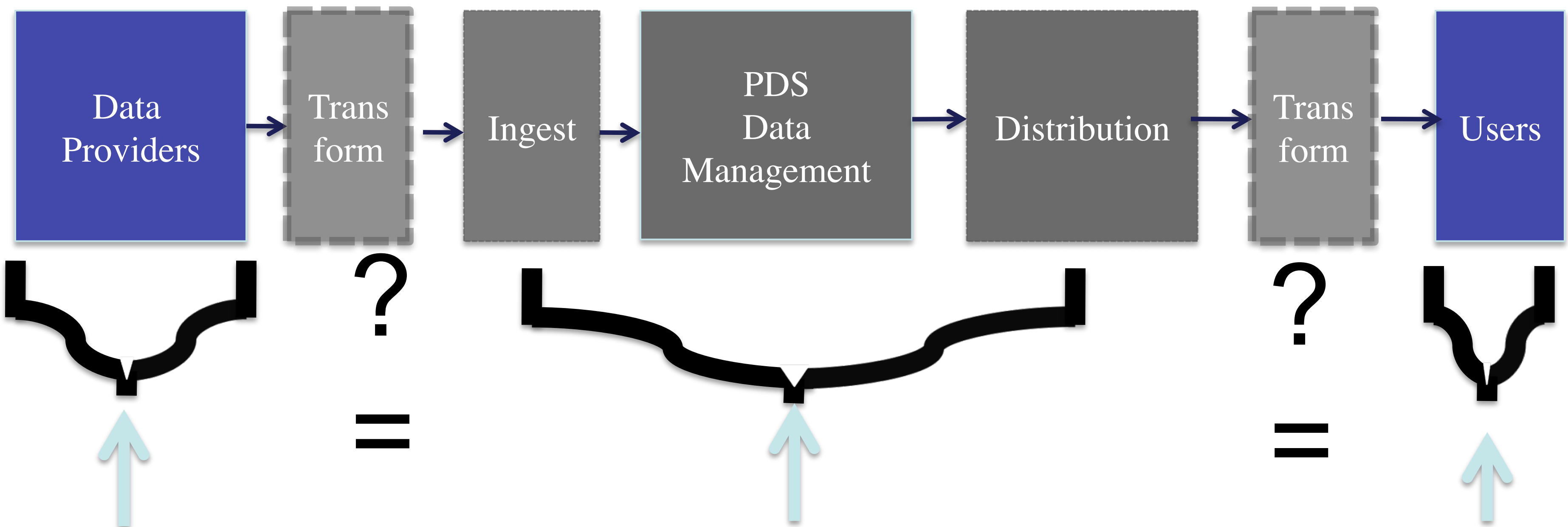
Goal: Improve user support and usability of the data in the archive

* Note: In the CD/DVD era of PDS, these weren't really split

Data Architecture Approach

- Introduce a major missing piece in PDS: identification of standard data structures/formats for archiving
  - Separate "ARCHIVE" format from "USER" format (User WG Recommendation)
  - Ultimately reduces software costs because we have predictable structures
  - Increase stability of the long-term archive
  - Difficult to satisfy systems requirements as BOTH an archive and a data system if there is no clear separation
- Re-architect data dictionary and keyword management
- Increase consistency and integrity of the archive by linking products all the way back to the information model
  - Deliver template (schemas) to users to improve both validation/ ingestion and access

Level 0 Conceptual Flow (Data Model View)

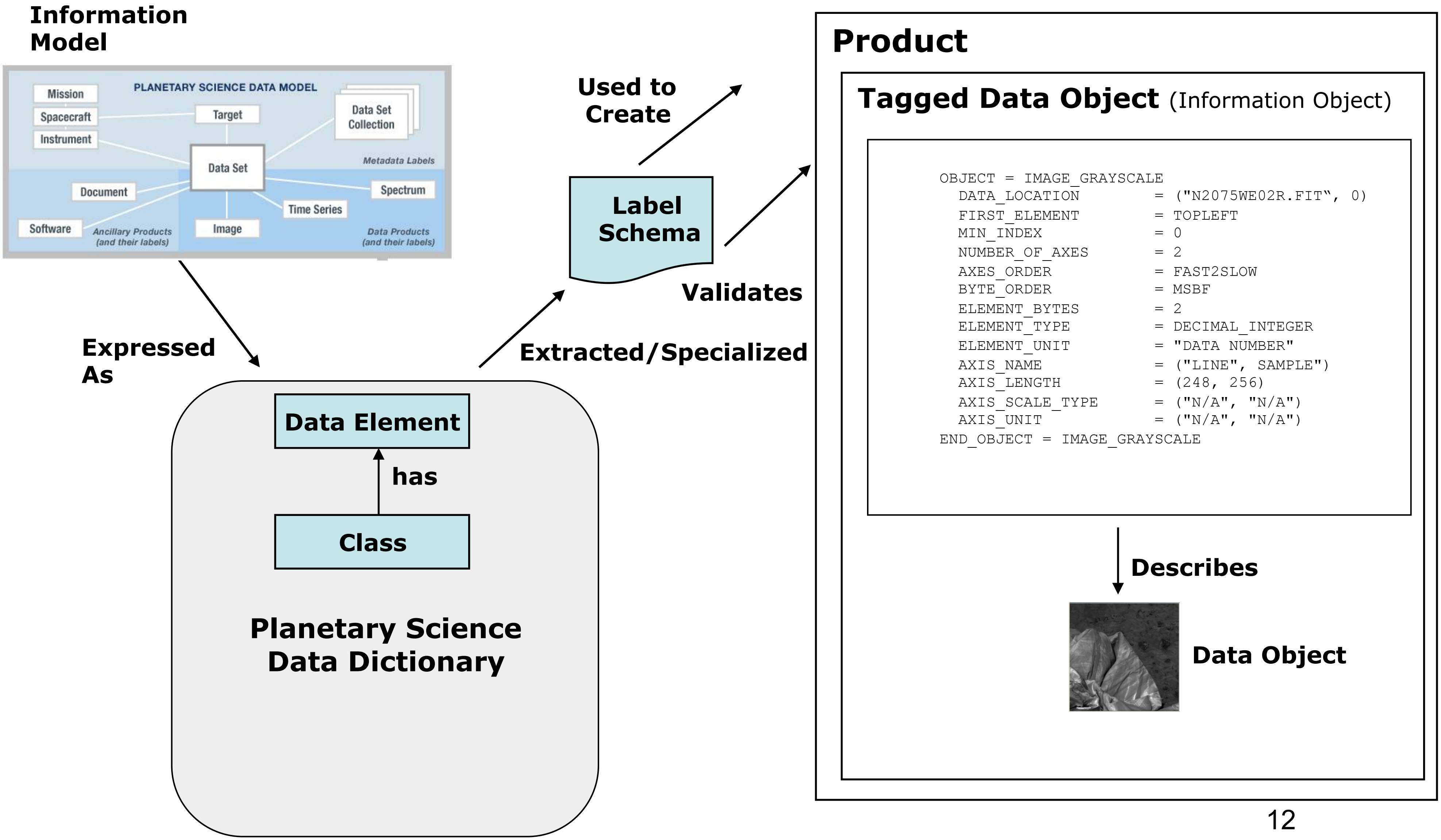


Data Provider Model (DF1): Data formatted based on data producer pipeline

Archive Model (DF2): Data formatted for long-term archive

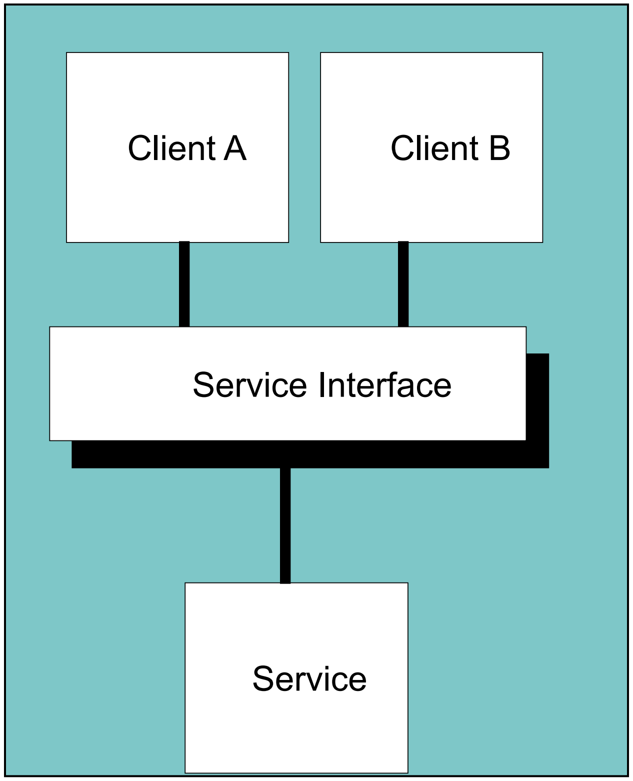
User Model (DF3): Data formatted for user use

Data Architecture Concepts

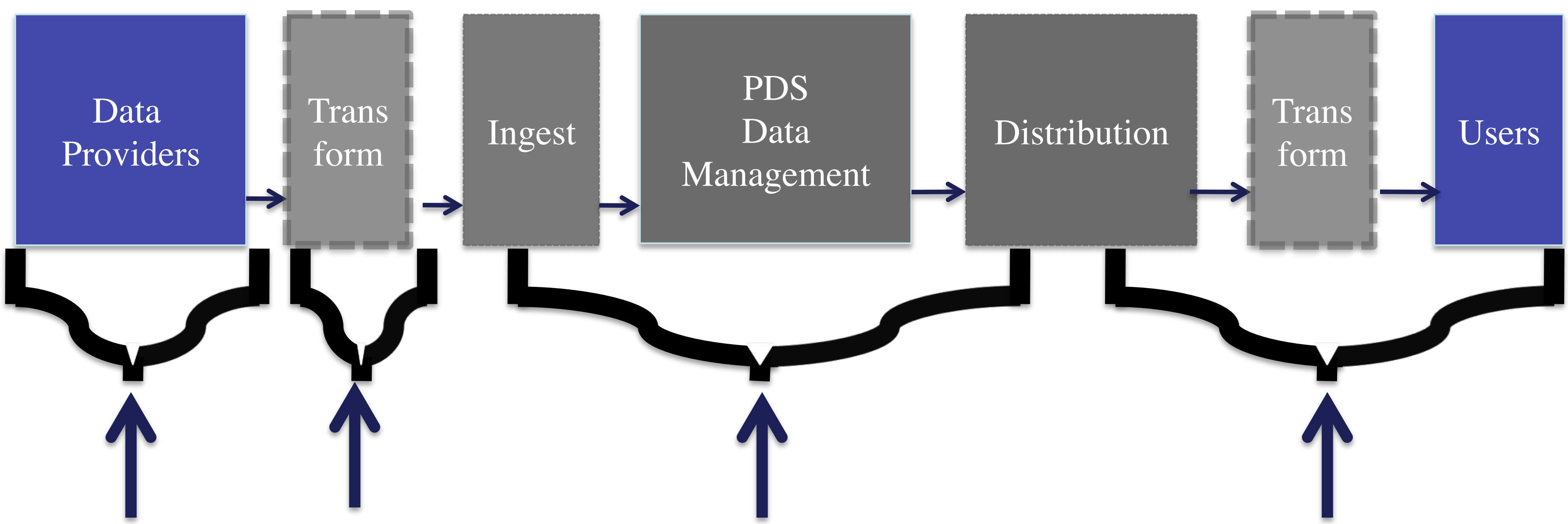


System Design Approach

- Based on a distributed information services architecture (aka SOA-style)
  - Allow for common and node specific network-based services
- System includes services, tools and applications
- Implement Distributed Services that bring PDS forward into the online era of running a national (and international) data system
  - With good data standards, they become critical to ultimately improving the usability of PDS
  - Support "on-the-fly" transformation to/from PDS
- Move towards increased sharing of libraries across PDS
- Adopt open source software, where possible



Level 0 Conceptual Flow (Software Mapping)



Tools: Generation, Design, Validation

Tools: ?

Tools: Validation, Ingest (Submission), etc

Services: Core Infrastructure, Inventory Management

Services: Search, Data Discovery, Data Access, *fromPDS,* DN services

Agenda

- Engineering Overview/PDS3 Implementation
- Architectural Drivers
- PDS 2010 Project Overview
- PDS 2010 Project Structure
- Major Design Decisions

Implementation Decisions

- There are multiple implementation choices that are part of the PDS 2010 design

- Critical ones include:
  - Grammar Decision
  - Data Dictionary Structure
  - PDS Distributed Infrastructure Standards (Registry, Data Dictionary, Security, etc)
  - Data Access Standards and Patterns
  - Format Transformation Tools and Services

PDS Technical Session June 11-13, 2009

- Focused on PDS 2010 and PDS4 data standards
- Covered data design, system design, examples, data dictionary, grammar/labels, standards documents, transition plans, migration, deployment, and next steps
- Actions, recommendations and issues were captured and are being worked in all the above areas
- No major show stoppers identified
  - Actions largely focused on capabilities and project suggestions

Benchmarking

System/ Software	Common Info Model	Data Formats	Grammar	Data Dictionary
Earth/EOSDIS	Little shared standards; some work in access info model	HDF, NetCDF	Replacing ODL/ PVL for XML	Little standards
IVOA	Simple standards	FITS	XML	Homegrown
SPASE	Access Info Model	Not Constrained	XML	Homegrown
NIH/NCI	Integrated Info Model	Not Constrained	XML	ISO 11179
DOE/Earth Sys Grid	Access Info Model	NetCDF	XML	Little standards
EPA	Access Info Model			ISO 11179
CASPAR	Integrated Info Model	Not Constrained	XML	DEDSL/ISO 11179
PDS3	Modeled at high level; ad hoc at product-level; archive model	Minimal Constraints	ODL	Homegrown

System/ Software	Distributed Architecture	Registry	Web Service I/F	Security
Earth/EOSDIS	Towards SOA	ECHO, but no distributed registry standard	No standard	No service
IVOA	SOA	IVOA Registry	REST	LDAP
SPASE	Towards SOA	Simple Registry	REST	No service
NIH/NCI	SOA/Grid	caGrid	SOAP	LDAP
DOE/Earth Sys Grid	SOA/Grid	Registry publishing mechanism	REST; Limited	LDAP
EPA	SOA (web services)			
CASPAR	SOA	ebXML Federated Registry		
PDS3	Limited network services	No distributed registry	REST (PDS-D); Limited	No service

Key Design Decisions & Recommendations Approved by MC (August 2009)

- Replace PDS3 ad hoc information model with a PDS4 information model that is now managed in modern tools (DDWG)
- Replace ad hoc PDS3 product definitions with PDS4 products that are defined in the model (DDWG)
- Require data product formats to be derivations from a core set; Support transformation from the core set (DDWG)
- Replace “homegrown” PDS data dictionary structure with an international standard (ISO 11179 RIM) (DDWG)
- Adopt a modern data language/grammar (XML) where possible for all tool implementations (SDWG)
- Adopt system of registries to support improved tracking and access (SDWG)
- Support remote access to data and services to bring the federation together both for ingestion and distribution (SDWG)

Backup

Guiding Information for Design

- Roadmap
  - http://pds-engineering.jpl.nasa.gov/projects/PDS4/Exchange/PDS_Roadmap.pdf
- PDS Level 1, 2, 3 (System-Level)
  - http://pds-engineering.jpl.nasa.gov/index.cfm?pid=5&cid=72
  - Really, not a PDS3 set of requirements
- PDS4 Concept Papers
  - http://pds-engineering.jpl.nasa.gov/index.cfm?pid=100&cid=119 (Architecture)
  - http://pds-engineering.jpl.nasa.gov/index.cfm?pid=100&cid=120 (Data Model)
  - http://pds-engineering.jpl.nasa.gov/index.cfm?pid=100&cid=121 (User Support)
- PDS Vision and Exec Summary
  - http://pds-engineering.jpl.nasa.gov/projects/PDS4/pds2010-execsummary20080701.pdf
- PDS 2010 Architecture
  - http://pds-engineering.jpl.nasa.gov/index.cfm?pid=100&cid=131 (System Architecture)
  - http://pds-engineering.jpl.nasa.gov/index.cfm?pid=5&cid=125 (Data Architecture)