PDS4 Architecture Preliminary Recommendations for 2008 and Beyond

C. Acton, D. Crichton, S. LaVoie, M. Martin, T. Stein

November 28, 2007

I. Introduction

This white paper provides a summary of recommendations on the development of the PDS architecture captured from discussions of the PDS4 Architecture Working Group [Acton, Crichton, LaVoie, Martin, Stein]. In developing these recommendations, the Working Group developed a process for evaluating and defining an initial decomposition of the PDS4 system architecture. This process involved evaluation of the PDS Roadmap [1], the PDS Level 1,2,3 requirements [2], and standard architecture and archiving frameworks for representing the PDS system architecture [4, 5, 6, 7, 8]. The purpose was to identify the elements of the system, their drivers, requirements, and their individual roadmap. In addition, the Working Group identified "gaps" in the analysis to address missing drivers and requirements and to address core problems with PDS3.

II. Core Concepts and Background

The term "architecture" is used to describe the elements that make up the PDS System Architecture and their relationships. A widely used definition of "software architecture" is, as defined by the IEEE 1471-2000 standard, "the fundamental organization of a system embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution." [6]. To define the high level view of the PDS system architecture, the Working Group uses the concept of "architectural views" that is widely published in several architecture frameworks [6,7, 8] as a way to decompose the system into sub-architectures.[1] The sub-architectures that the Working Group uses to categorize elements of the system are the "process", "data" and "technology" architectures. This allows us to evaluate and evolve the elements of the sub-architecture independently. Given PDS funding, the working group feels that this agility is a critical principle. These three sub-architectures are discussed below.

PDS, as a system, is based on a set of processes for archiving. These processes form the "process architecture" for PDS and consist of such elements as ingestion, standards management, preservation planning, peer review and administration of the system. The processes describe steps and control flow and interactions that need to occur in order to manage and operate PDS as a whole. These processes are fundamental to the way in which PDS functions and the mechanisms by which the system is implemented. The system needs to be agile enough to allow for process changes as they occur.

Due to PDS' needs to support multiple missions, the system must have as one of its core architectural principles that it is "data driven". This means that PDS needs a robust and explicit "data architecture" that provides standards for the description of planetary science data [1.4.x] in a form that can be used by the system.

---

[1] These sub-architectures are often referred to as "models", but generally mean models as appropriate for the part of the system that they are describing.

The data architecture consists of a domain information model[2] along with specifications on how to organize the data. Within PDS, the information model largely consists of keywords, often called data elements, and their values organized into objects. The objects within the information model are related and used to describe science data results from the planetary science research domain. The objects, their keywords, and their values are captured into the Planetary Science Data Dictionary (PSDD) that has been used to manage the contents of the PDS information model.

In addition, the PDS data architecture also defines standards for the structuring and documenting science data using the Object Description Language (ODL). These syntactical standards are often called a "grammar" and serve as a formal data description language. PDS provides a number of core descriptive templates, implemented in ODL, that serve the purpose of documenting the archive and its contents including "label" and "catalog" files. The PDS Standards allow for quite a variation of structuring data products allowing for both attached and detached metadata labels to be associated with a data object. In addition, PDS allows for inclusion of non-PDS formatted data products (VICAR, FITS, etc.) which are described using PDS labels.

Lastly, the PDS data architecture provides standards for organizing the scientific results into volumes. These volumes serve the purpose of providing a mechanism for chunking, storing and distributing data by organizing it based on physical media limits. Traditionally, these limits have been CD-ROM and DVD-ROM storage capacity limitations, but other considerations such as a "screenful" of information or a file size that is convenient for a single download session have been used.

The PDS "technical architecture" for PDS is largely based on a set of functions that PDS performs. These include data ingestion that consists of receiving, validating and accepting data deliveries; data management which includes catalog management, storage, search, distribution, long term archiving; and providing a number of specific user services. In PDS3, PDS began the transition of moving services online and enabling sharing of data between nodes through a distributed infrastructure.

III. Drivers

Drivers for moving forward with PDS4 come from several sources: the PDS Roadmap [1], the PDS Level 1,2,3 Requirements, and input from the Management Council in the form of questions and problems that PDS4 should address.

The PDS Roadmap[1] provides a number of drivers for the next ten years that directly affect the architectural choices that PDS will need to make in all three architectural areas (new and updated processes, changes to the data architecture, and new functional capabilities for the system as part of the technical architecture).

The PDS Level 1,2,3 Requirements were developed by the PDS Management Council. They provide a broad set of system level requirements for PDS and are intended to guide all subsystem development.

In the November 2007 Management Council meeting, a request was given to each node to identify a set of questions for PDS4 Working Groups to respond to. While we list an example set of questions from the Geosciences Node below, it is important to note that several nodes have added to this core set. Each of the questions has been considered in this report.

[2] A domain information model is used to richly identify data objects, attributes and relationships for a particular domain.

PDS4 Management Council Questions (source: Geosciences Node/Arvidson)

i) How will PDS-4 enable "one-stop shopping", i.e., seamless access to data that reside at multiple nodes?

ii) How will PDS-4 help users by delivering derived data products in the format, coordinate system, and map projection the user requests?

iii) How will PDS-4 help data providers by automating the design, production, and delivery of PDS data sets?

iv) How will PDS-4 ensure that PDS standards are simple, straightforward, and consistent so that data providers and users can easily understand and apply them?

v) How will PDS-4 ensure that data sets can be safely and efficiently archived in NSSDC and retrieved on demand?

vi) How will PDS-4 improve the data transfer, data integrity, and maintenance of PDS data sets?

Based on the available input from the roadmap, requirements, and MC, the working group extracted and created a categorized list of architectural drivers organized into thematic areas [3]. The summary of this list is as follows:

i) **More data.** PDS storage requirements are projected to increase from 40 TB to over 500 TB in just three years. This will require more automation, scalable high capacity storage systems and advanced data movement techniques.

ii) **More complexity.** Missions, instruments, and data are all becoming more complex. This will require an improved information model for archiving diverse data products (in situ, geographical, astronomical) as well as a modern online data dictionary with name space management and access control.

iii) **More producer interfaces.** PDS is facing an increasing number of missions, a greater number and diversity of data providers, and smaller, focused missions. This will require a streamlined standards architecture that is easy to learn and use, with more reliance on delivering data in standard data formats. Cross-platform archiving tools must be provided which can be used to design, generate, validate, and deliver archival data sets.

iv) **Greater user expectations.** The World Wide Web has led users to expect well-documented data to be readily available via text-based or graphical search systems with data delivery in a variety of formats compatible with their data processing systems. This includes access to tools for displaying or analyzing discipline specific data as well as special processing to produce higher order products.

v) **Limited funding.** The emphasis on smaller, faster, cheaper missions which often include international partners may limit the ability to provide products suitable for analysis by the broader science community. This puts a burden on NASA Data Analysis programs or on the PDS have to finish the job. As space exploration continues to become an international effort, PDS must expend increasing resources working with foreign agencies and international organizations to assure access to new mission data. The “internationalization” of space exploration will also necessitate additional

standards that promote data sharing and interoperability and an international core data model for archiving and for querying remote archives.

vi) **Creating a "system" from the federation.** The current PDS nodes operate autonomously and independently with limited distributed access via PDS-D to node repositories. This means that each site must do its own planning, design, review, procurement, code development, testing and operations. There is little sharing of technical expertise in this heterogeneous environment. A better approach would be to provide technology specifications to allow distributed and shared services across the federation, and to ensure that tools can plug into local environments. Common infrastructure services would be provided where it makes sense (physical media production, security, backup, mirroring, web site maintenance).

The results of this effort were captured and posted at the following website: http://pds-engineering.jpl.nasa.gov/index.cfm?pid=100&cid=119 . These will be expanded on in Section V, the PDS4 Architecture Concept.

IV. Core Architectural Principles

Architectural principles are often used to form a general basis for decision making of architectural choices for the system. Large-scale software systems typically have explicit principles that are used to guide the evolution of components and the systems for large enterprise teams [5, 6, 7, 8]. PDS inherently has principles by which it governs architectural choices. However, most of these have not been explicitly identified. In developing the architecture vision for PDS4 the working group identified a set of explicit principles based on the drivers and questions provided by the Management Council which is identified below:

i) **Model Driven.** The data standards and data dictionary serve as the source for the design of PDS data products. Software is designed in a manner as to evolve as the standards and dictionary evolve.

ii) **Archiving is the Priority.** While PDS has several functions, its principal function is in the capture and preservation of data.

iii) **Evolution of the system as a set of elements.** PDS, as a data system "facility" for planetary science results, must evolve. Given budgetary, mission, and user constraints, it is critical that PDS be able to evolve parts of the system over time. Separating the architecture into elements, and then components, facilitates the evolution of parts of the system while preserving others.

iv) **Support for a distributed federation.** The system is built in such a way as to allow for changes to the federation including ownership of data, changes in the node structure, and repurposing of tools for use within their discipline.

v) **Use of Standards.** PDS will rigorously use standards by adopting, adapting and developing standard specifications, in that order.

vi) **Low cost of ownership.** In working with data providers, PDS should ensure that data providers can adopt and use its tools with minimal resource impacts and open source tools .

vii) **Diversity.** PDS is designed to support the diverse needs of data providers, missions, and the planetary science domain .

viii) **Scalability.** PDS is designed to scale the core functions of the system as the volume of data increases.

ix) **Explicit Design.** PDS elements are explicitly defined with unambiguous specifications that can be implemented by software developers.

x) **International Adoption.** The use of PDS elements, including its standards, is straightforward in order to facilitate use and adoption by international partners.

xi) **Integrity.** The PDS system adheres to a rigorous data integrity policy to ensure its system and data are reliable and available.

xii) **Timeliness.** It is important that the collaboration between PDS and a data provider begin as early as possible in the data creation process to ensure that PDS standards and tools can be inserted and effectively used. For missions, this interaction should start well before launch.

V. What is PDS4? The PDS4 Architecture Concept

In this section we will describe the architecture of PDS4, drawing on the PDS requirements, the drivers extracted from the roadmap, and the questions developed by the Management Council.

A major paradigm shift for PDS has been the movement to distribute data online. While much of this shift has already occurred, there has not been an explicit principle that all data will be online. As has been mentioned in the roadmap and other materials, users expect to be able to go online, search for data, and download it. We believe, given current storage capabilities and the outlook for the volume of data to be archived by PDS in the next five years, that **all data should be moved online**, replicated at another site and archived at the NSSDC as quickly as possible. PDS should consider itself an online system and should develop supporting technologies and standards to ensure that data is managed and exchanged online, including all data deliveries to the NSSDC. This is consistent with the Management Council discussions in August 2007.

Distributed services are another part of the evolution of making PDS an all online federation. Service-oriented architectures, grid computing and virtualization are all current concepts in the Information Technology (IT) community that enable organizations to offer online services to share data and computing resources. These services are tailored to support the needs of a specific domain, but for an enterprise such as PDS, can be used to deliver critical processing and data management functions as network accessible services on-demand. While we believe that many of these services should be driven by user needs, there is an inherent underlying architecture for building the distributed services that should be based on industry technology standards, where appropriate. It is important that PDS adopt and deploy predictable interfaces at all nodes, rather than having nodes build ad hoc services. Such services including the ability to query distributed product catalogs and to request and download data holdings are necessary and identified in the PDS requirement 2.8; however, they also can and should provide services for data transformation, subsetting, and other functions. One of the critical needs of having an online architecture is ensuring that all PDS data, including products, can be consistently queried across the PDS data holdings. As a federation, this requires that PDS nodes ensure that all data and services are online at their local node. As part of this architecture, we believe PDS would benefit from having technology standards and software to support construction of these services.

Data integrity is a critical part of providing an online system. The PDS Management Council, in its data integrity policy, has identified that data integrity consists of protecting against file corruption and data loss by

ensuring data is not corrupted, it can be tracked, and accessed. Part of ensuring data integrity is ensuring consistency of the data holdings across PDS. This is critical to ensuring that data is not lost. From the architecture view, we believe that PDS should adopt checksum standards for the data holdings delivered and archived by PDS. It should ensure that the movement of data online is architected in such a way as to prevent data loss to ensure data is tracked from delivery through to archiving at the NSSDC for both the dataset and product levels as specified in the PDS Requirements.

Data movement is also a critical part of providing an online system. Historically, PDS used the physical media and volume constructs as an implicit architecture for data movement. This included deliveries of data to PDS, between nodes and to the NSSDC. However, as PDS moves to an online system, it has no explicit standards for data movement. This includes both how data is *packaged* for transport as well as the underlying protocols for data transfer. Given data volume increases, we believe PDS, as a federation, must adopt technical standards and solutions for data movement (packaging and transfer). Current technology solutions for parallel data transfer have already identified that data sets of substantial sizes can be efficiently transferred. However, PDS needs to explicitly deploy such capabilities across its network to support the influx of data. In addition, PDS should also establish specific offline data transfer standards for instances where data is transferred to nodes and to the NSSDC.

In addition, PDS should develop core, multi-mission **archiving tools** that are agile enough to be inserted into discipline-specific pipelines and processes. These include data design and preparation, validation, submission and display. Much of this effort is already been initiated with VTOOL and LTDTOOL, however, we believe it is important to provide submission tools, an in particular, provide improved support for delivering data to PDS, particularly for small data providers such as NASA Data Analysis Programs. Submission tools would allow for online submission of data products to PDS to facilitate the ingestion process[3]. In addition, PDS should have a basic capability to display any data product. While much of this is obvious based on the current PDS requirements and PDS tools, it is important to state that a fundamental principle should be that these tools should run on popular operating system configurations, be easy to install and operate, and should be extensible. In addition, the architecture should promote standard application interfaces (APIs) so nodes can adopt client libraries and use them in generation, validation, submission and display functions within their own environments.

Improved **automation** of PDS is another part of the architectural vision and was represented in the drivers extracted from the PDS roadmap. The implication of increasing diversity, product types and volumes is ensuring that PDS automates core, human-intensive functions to allow for scalability of PDS. It is critical that automation be understood from data delivery through to the NSSDC. Construction of the PDS Catalog, which is used for tracking and high-level searching of PDS data holdings, has largely been a human-intensive effort. We believe that building automated mechanisms with critical quality assurance checks for interacting with the PDS Catalog will improve the efficiency and effectiveness of PDS overall by reducing the time to ingest new data sets, reducing the interactions between data engineers, and improving the data integrity of PDS. Again, constructing the architecture to have application interfaces for interacting with the PDS Catalog through application program interfaces (APIs) will allow for extensibility of the architecture as well as specialization by nodes who have their own tools and pipelines.

[3] PSI has demonstrated this with OLAF

PDS should also have an explicit **search** architecture and strategy. One of the critical needs of PDS4 is ensuring that users can locate all data holdings within PDS. While there are varying search implementations across PDS, there is no explicit overall search architecture. The search architecture should consist of defining the search metadata, the necessary technical components for search, and the methodologies that include multiple search layers. It is important that the search architecture be validated using user scenarios, as specified by the PDS4 User Services WG. Without an explicit search architecture, PDS will have a difficult time meeting its search objectives. And, while some of these objectives are well known now, we can expect that our needs will evolve over time. We, therefore, need to ensure that the search architecture is designed in such a way as to support extensibility and allow for reconfiguration as new search methods are deployed. In some cases, it will be necessary to relabel or reform certain older datasets to address incompatibilities that would preclude searching. Examples of such incompatibilities are coordinate systems, map projections, and physical units.

The above search architecture should be integrated with online, web-based tools such as **portals**. The word "portal" is somewhat of a buzz-word; however, the concept is important to PDS. A portal provides a "one-stop" shop for access to information. The PDS Homepage, in many respects, is a portal. Many of the nodes have websites which resemble a portal for their discipline. Portals synthesize information and allow for management of content including news and announcements. Implementing portals can be an effective way of working with communities since the content can be oriented for that community. From a technical standpoint, there are varying degrees of portal software from "home-grown", to open source, to commercial off-the-shelf with extensive vendor support. Portal software provides publishing pipelines for updating the content of a website including a workflow for its approval. It also implements standards such as Web 2.0[4] which provide specifications for web tools which promote collaboration and sharing between users. The PDS4 WG envisions that Web 2.0 technology, for example, would be useful for making PDS more dynamic. RSS (Really Simple Syndication), for example, has been part of the Web 2.0 initiative and provides capabilities to distribute news to any portal subscribing to the RSS-feed across the Internet. This type of capability could allow for certain news feeds to be distributed to all PDS portals as well as non-PDS portals to announce major events such as the release of data. Other examples could be targeted blogs and web-casts allowing one-on-one interactions with node scientists related to certain data sets or other items of interest to a Discipline Node's community. Multimedia tutorials could be provided to offer step-by-step recipes for creating reduced data products.

The PDS4 Architecture needs to provide the infrastructure to enable both client tools to plug into PDS and the development of specialized **user tools and services**. While NASA has historically funded many efforts to build data display and analysis tools, these have generally not been built to integrate with PDS. In addition, many of the processing capabilities are built into the standalone tool, rather than as a service within the PDS infrastructure that could be reused. The PDS4 Architecture needs to allow for integration of processing capabilities enabling server-side processing of data prior to delivery to a client tool. In addition, PDS should promote standards that specify how these tools plug into the infrastructure. PDS should also focus on providing a set of core tools for working with the data as has been done with NAIF/SPICE and ISIS.

Standards are another critical area of the architecture that has been addressed several times already. However, it is important that PDS have a standards initiative that addresses both data and technology standards. While data standards are being addressed in the PDS4 Data Model WG, the technology standards is not something that has ever been consistently addressed by PDS. As we declare ourselves to be an online system, PDS needs a set of technical standards that guides how system components "plug" together. This will enable the evolution

[4] http://en.wikipedia.org/wiki/Web_2

of PDS towards a virtual federation. These technical standards should address how PDS can "link" with non-PDS systems, particularly international partners. It is critical that both data and technology standards have simple, explicit and rigorous computer science specifications so non-PDS computer science personnel can implement them. In addition, they need to be constructed in such a way as to allow for PDS to grow both domestically and internationally.

In essence, the PDS4 architecture is one that explicitly embraces being an online system with well thought out interfaces and architecture choices that allow for it to grow and evolve over the next ten years. One of the critical questions that need to be asked when considering the PDS4 architecture is whether it comprehensively implements the PDS Requirements and how well it responds to the principles identified above.

VI. Decomposition of the PDS System Architecture

As mentioned earlier, the PDS Architecture WG reviewed the requirements, assessed the drivers and constructed an overall architecture for PDS that decomposes the system into elements classified by process, data and technology sub-architectures. This allows each of these elements to be considered either individually or as a set of elements as part of a future PDS4 project. As part of this process, the team identified "gaps" where we felt there should be a new element but no supporting PDS Requirement exists currently. This decomposition is identified below:

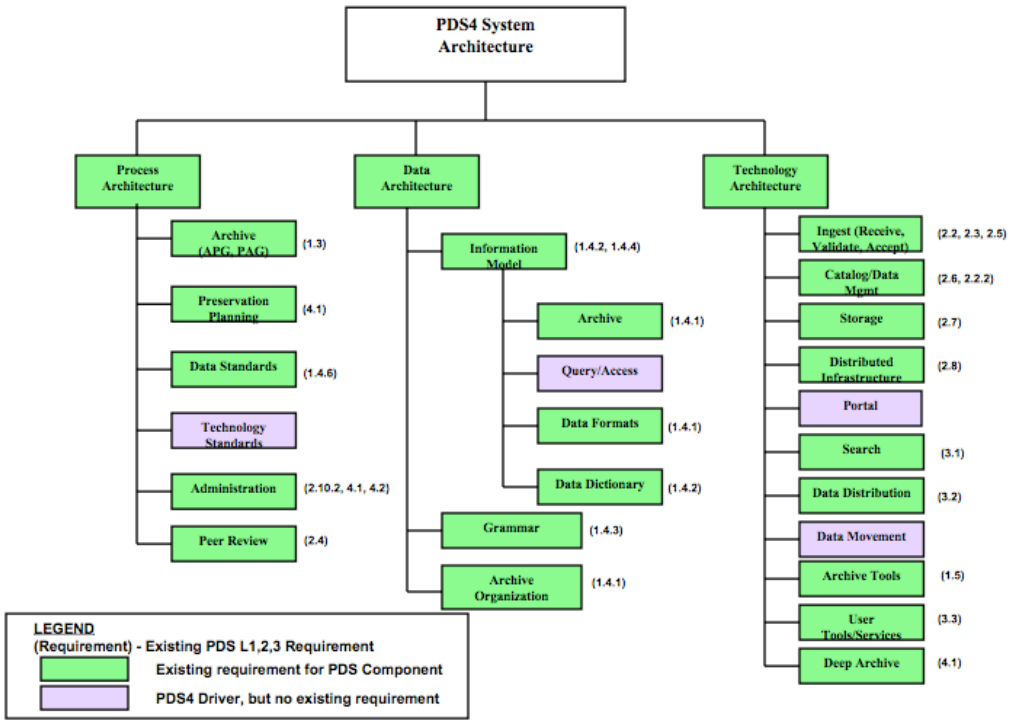

Figure 1: PDS System Architecture Decomposition

VII. Management Council Implementation Recommendations

The PDS4 Architecture Working Group has several suggestions that are introduced below. Many of these can be inserted directly into the steps adopted by a PDS4 implementation plan. These include:

- *PDS should update its PDS Requirements for PDS4*

The PDS Level 1/2/3 Requirements should be comprehensive to cover PDS4. Augmentations or changes to the requirements should be identified before PDS moves forward. This should include addressing drivers, for which there is no specific PDS Requirement. The PDS4 Driver Matrix that the WG prepared has identified some new requirements.

- *PDS should understand the problems with PDS3; PDS should understand what is working well in PDS3*

In assessing a future design for PDS4, it is critical to address how the design responds to these issues. While there are clear drivers for PDS4, we also want to ensure that we clean up plaguing problems from PDS3 and earlier. At the same time, we want to make sure we leverage the core elements of PDS that are working well. The Working Group recommends that PDS capture and manage such a list through the Engineering Node.

- *PDS should identify a prioritization of PDS4 projects*

Given resource and other operational constraints, PDS needs to be realistic about the scope, commitment and priority of building PDS4. The Working Group recommends that PDS identify a series of projects for implementing PDS4. These projects should build on existing projects which are already in place following the current engineering approach where each project has a timeline, requirements and deliverables, and is reported monthly to the PDS Management Council. It is important that the priority for these projects be identified by the Management Council so as to ensure the implementation plan is consistent with the needs of PDS.

In developing the prioritization, the PDS4 Architecture WG recommends the following new projects be initiated, in priority order, to address the elements in the PDS4 architecture:

i) **PDS Data Standards.** The PDS Data Standards serve as the underpinning of the system. They should be addressed first with an effort to explicitly define the PDS4 data standards for implementation.

ii) **PDS Technical Standards.** The PDS Technical Standards are critical to defining how the system "plugs" together. These standards should be identified prior to building any future system.

iii) **Online Data Integrity.** PDS should review and adopt the Data Integrity Working Group requirements and plan for their implementation including staging all data online, implementing checksum protections, and deploying an end-to-end tracking capability.

iv) **Portals, Search and Distribution.** PDS should define an integrated search architecture which allows users to "seamlessly" search, download, and transform data products from a single-point of entry. This should be considered phase II of the work that has already been performed by the User Interface Working Group started in 2006 (Law, Rose, Beebe, King, Gordon). Input for this WG should come directly from the PDS4 User Service Working Group.

v) **Distributed Services.** PDS should ensure that services are built using the technical and data standards identified earlier and are consistently deployed to PDS nodes in order to provide data services that

work with all data holdings. These services should extend beyond just basic access and should consist of functions prioritized by the Management Council, particularly what services would be of value from the Discipline Node (DNs) advisory groups.

vi) **Data Movement and Delivery.** PDS should define an architecture and infrastructure to support data movement across PDS including standards for data packaging and transport (structure and protocol). Such a capability should be used to efficiently move data across the PDS network.

In summary, the implementation plan for PDS4 should address the schedule and plan for implementing each project. The implementation plan should have a five-year horizon serving, in theory, as the implementation plan for the PDS Requirements ensuring that each requirement is implemented along with a minimal assessment of its implementation.

VIII. Summary

This white paper is intended to serve as a preliminary report to the Management Council regarding a concept for the overall architecture of PDS4 and a plan for identifying how to move from PDS3 to PDS4. The WG considers this an opportunity to "fix" some of the blemishes that have historically plagued PDS as well as improve upon other qualities of PDS that are quite good. The WG also recognizes that emerging technology capabilities will also allow PDS to move forward and offer a greater set of services through a virtually integrated environment. At the same time, it is clear that user and resource constraints will not allow PDS to simply build and cut-over to PDS4. It needs to be phased. In many cases, PDS has already begun the transition; it just hasn't been explicitly called "PDS4". We believe it is time to call these efforts PDS4 and to define the set of targets as recommended in this white paper.

IX. References

[1] PDS Roadmap, February 2006

[2] PDS Level 1,2,3 Requirements, August 2006

[3] PDS4 System Architecture Driver Matrix

[4] Reference Model for Open Archive Information System, CCSDS 650.0-B-1, January 2002.

[5] Federal Enterprise Architecture Framework (FEAF), Version 1.1, 1999.

[6] Recommended Practice for Architectural Description of Software-Intensive Systems, IEEE 1471-2000, 2000.

[7] Reference Model on Open Distributed Processing (RM-ODP), ISO 10746, 1998.

[8] The Open Group Architecture Framework, TOGAF 8.1.1, 2006.