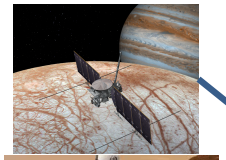
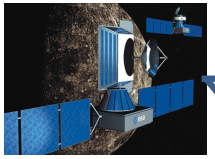
Enabling User / Community Services and Tools

Jordan Padams, Steve Hughes, Dan Crichton, Emily Law

Jet Propulsion Laboratory
California Institute of Technology

PDS Data Services Initiative





PDS Engineering

1. High Level Search; Content Products

PDS Node Archive

2. Product Level Search
3. Retrieve Product by LIDVID

PSA Archive

1. High Level Search
2. Product Level
3. Retrieve Product

JAXA Archive

1. High Level Search
2. Product Level
3. Retrieve Product

Search Services (1,2,9)

Access Services (3)

Transformation Services (4)

On-demand Algorithms (6,8)

Computational Services (6)

Data Access Portals

Visualization (7)

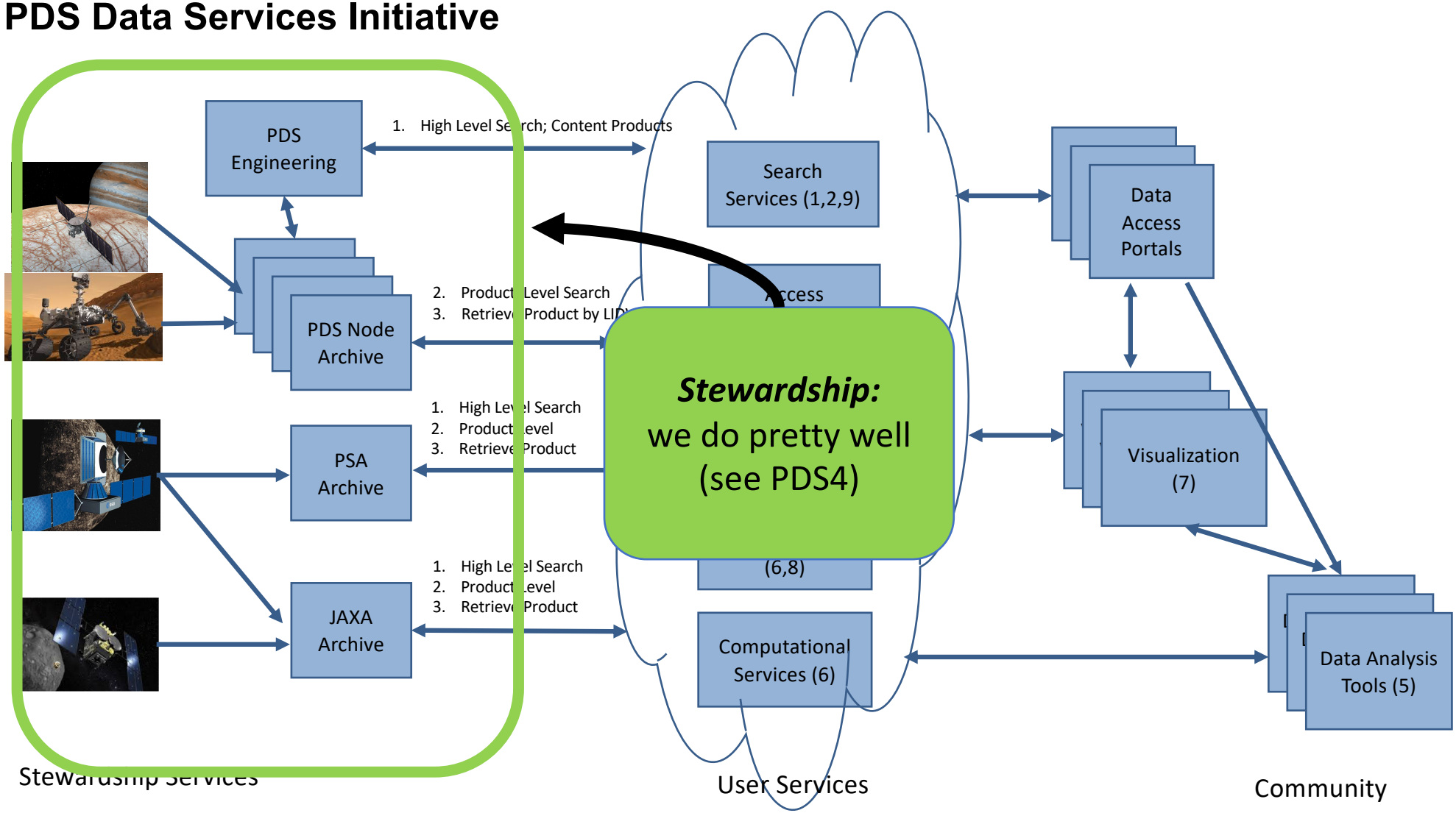
Data Analysis Tools (5)

Stewardship Services

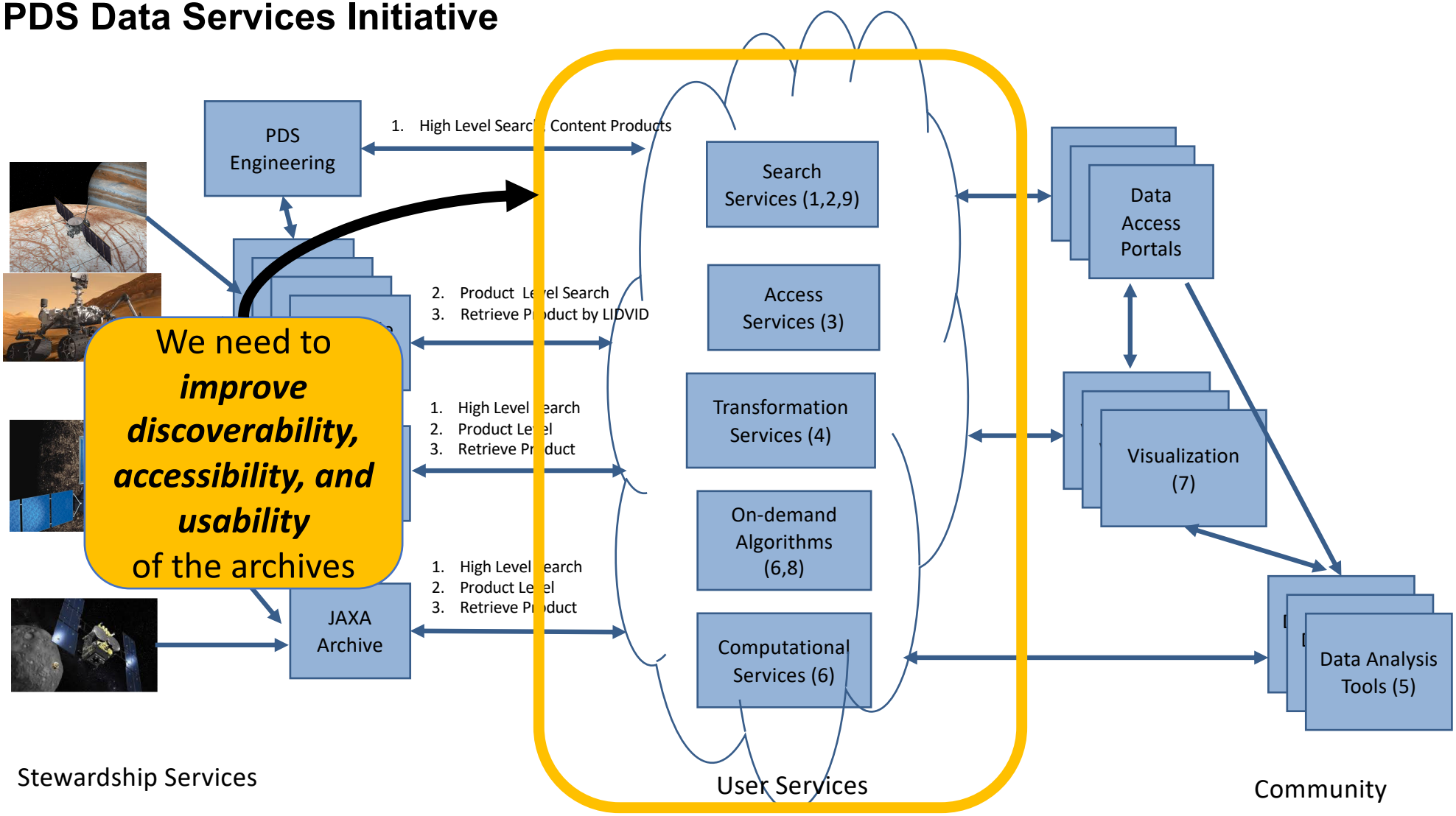
User Services

Community

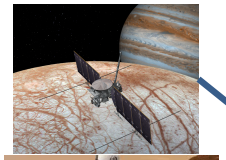
PDS Data Services Initiative

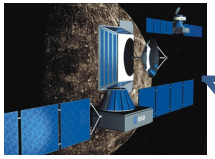


PDS Data Services Initiative



PDS Data Services Initiative





PDS Engineering

1. High Level Search; Content Products

PDS Node Archive

2. Product Level Search
3. Retrieve Product by LID

PSA Archive

1. High Level Search
2. Product Level
3. Retrieve Product

JAXA Archive

1. High Level Search
2. Product Level
3. Retrieve Product

Search Services (1,2,9)

Access

To ***enable*** the PDS and community to ***develop tools*** and ***use the data***

Computational Services (6)

Data Access Portals

Visualization (7)

Data Analysis Tools (5)

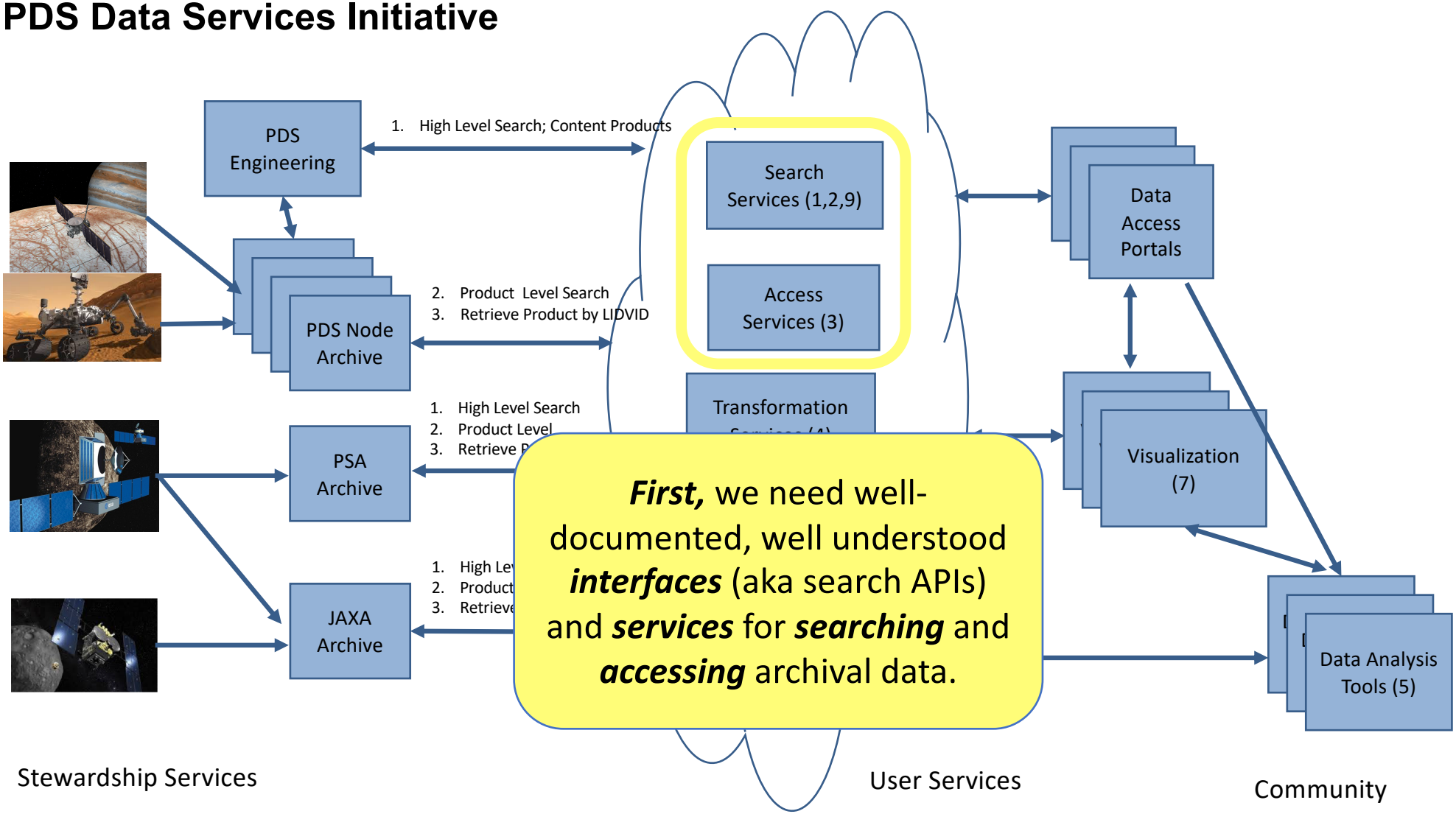
Stewardship Services

User Services

Community

Where do we start?

PDS Data Services Initiative



PDS Search and Access API Specification

*Developing a **REST-based API definition** and **set of guidelines** is to encourage application developers to have resources accessible to them via a RESTful HTTP interface based upon a consistent set of syntax and best practices. To allow for smooth, **seamless integration of PDS and IPDA search applications**, as well as provide the simplest possible experience for developers and scientist attempting to access these applications.*

PDS APIs SHOULD follow consistent design guidelines to make using them easy and intuitive.

High-Level Requirements / Guidelines

- PDS Search Specification should follow consistent design guidelines to make using the APIs easy and intuitive.
- PDS Search Specification should be defined using the OpenAPI Specification
- PDS Search Specification should leverage and be driven by the PDS4 Information Model, where possible. (e.g. query parameter naming)
- PDS Search Specification should include, but not limited to:
  - ***Defined query parameters***
  - Defined result formats (e.g. particular JSON definition)
  - Defined query syntax for complex queries (e.g. SQL, Solr API, etc.)
  - Defined RESTful URI endpoints
  - Details on how to perform a request
  - Details on how to specify the metadata returned
  - Details on how to specify the metadata returned
- PDS Search Specification should leverage ESDIS CMR, Google, and other industry lessons learned in defining this specification.
- PDS Search Specification should be designed and managed by new Search Integration/API Working Group

Initial Focus On Query Parameters

Goals of Query Parameter Specification

Due to the varying types of data available in the PDS, the applicability and use of certain query parameters will vary dramatically. However, the goals here are:

- To define a common syntax and best practice for defining query parameters based upon the PDS4 Information Model
- To define a *common set of query parameters* for top-level search criteria with the widest-ranging applicability across all PDS data
- To define a set of best practices and procedures for *extending the common set of query parameters* with node-specific query parameters
- To provide a *managed central location* for describing and documenting all query parameters across the PDS.

Query Parameter Options

Note: These are ***NOTIONAL*** ideas only.

More work needs to be done.

Custom Query Parameters

- Manually curated names for query parameters
- For example:
  - `target`
  - `mission`
  - `instrument`
  - `instrument_host`
  - `observation_time`
  - `filter_name`
  - `wavelength`
  - `center_longitude`
  - `center_latitude`
- In many cases, these are PDS3 Keywords that would map to a PDS4 Xpath
- These parameters often tie directly to form-based search interfaces
- Other possible use cases for custom query parameters:
  - Grouping several fields together
  - Enhanced metadata not necessarily in PDS4
    - e.g. image features, tagging products with improved metadata
  - Derived metadata

Custom Query Parameters

Pros

- Allows for grouping of parameters together
- Heritage with many current search engines
- Heritage with PDS3 nomenclature
- Human-readable
- Similar to PDAP and EPN-TAP protocols
- Allows for searching for non-PDS metadata

Cons

- Requires manual curation of all parameter names used in the API
- Parameter names can be vague
- Does not utilize PDS4 IM
- API design should consider but not be driven by search user interface design

PDS4 Shortened XPath Parameters

- Use a combination of the PDS4 Information Model and ***dot notation*** representations of an XML XPaths
- Query parameters will use a combination of an attribute with its parent class in *all lowercase*:

```
{parent_class}.{attribute}
```

- For example:
  - `time_coordinates.start_date_time`
  - `time_coordinates.solar_longitude`
  - `geom.SPICE_Kernel_Identification.geom.spice_kernel_file_name`
  - `msn.surface_mission.msn.start_sol_number`
  - `orex.mission_information.orex.mission_phase_name`
- Exceptions to this rule will exist, in which case additional ancestors can be added to dot notation

PDS4 Shortened XPath Parameters

Pros

- Ties directly to information model
- Leverages hierarchical, semantic naming provided by PDS4
- Less ambiguous that custom parameters
- Not reinventing the wheel again

Cons

- Does not allow for grouping of parameters beyond parent class*
- Does not allow for searching non-archival attributes*
- Not human-readable (only con for documentation sake)

* Recommendations from Search API Working Group can drive enhancements to IM to support this.

PDS4 Full XPath Parameters

- Use entire XPath in dot notation for all attributes to define the parameters

- For example:
  - `Product_Observational.Observation_Area. Time_Coordinates.start_date_time`
  - `Product_Observational.Observation_Area.Time_Coordinates.solar_longitude`
  - `Product_Observational.Observation_Area.Discipline_Area.geom.Geometry.geom.SPICE_Kernel_Files.geom.SPICE_Kernel_Identification.geom.spice_kernel_file_name`

PDS4 Full XPath Parameters

Pros

- Ties directly to information model
- Leverages hierarchical, semantic naming provided by PDS4
- As explicit as it gets with tying the value directly to the IM
- Not reinventing the wheel again

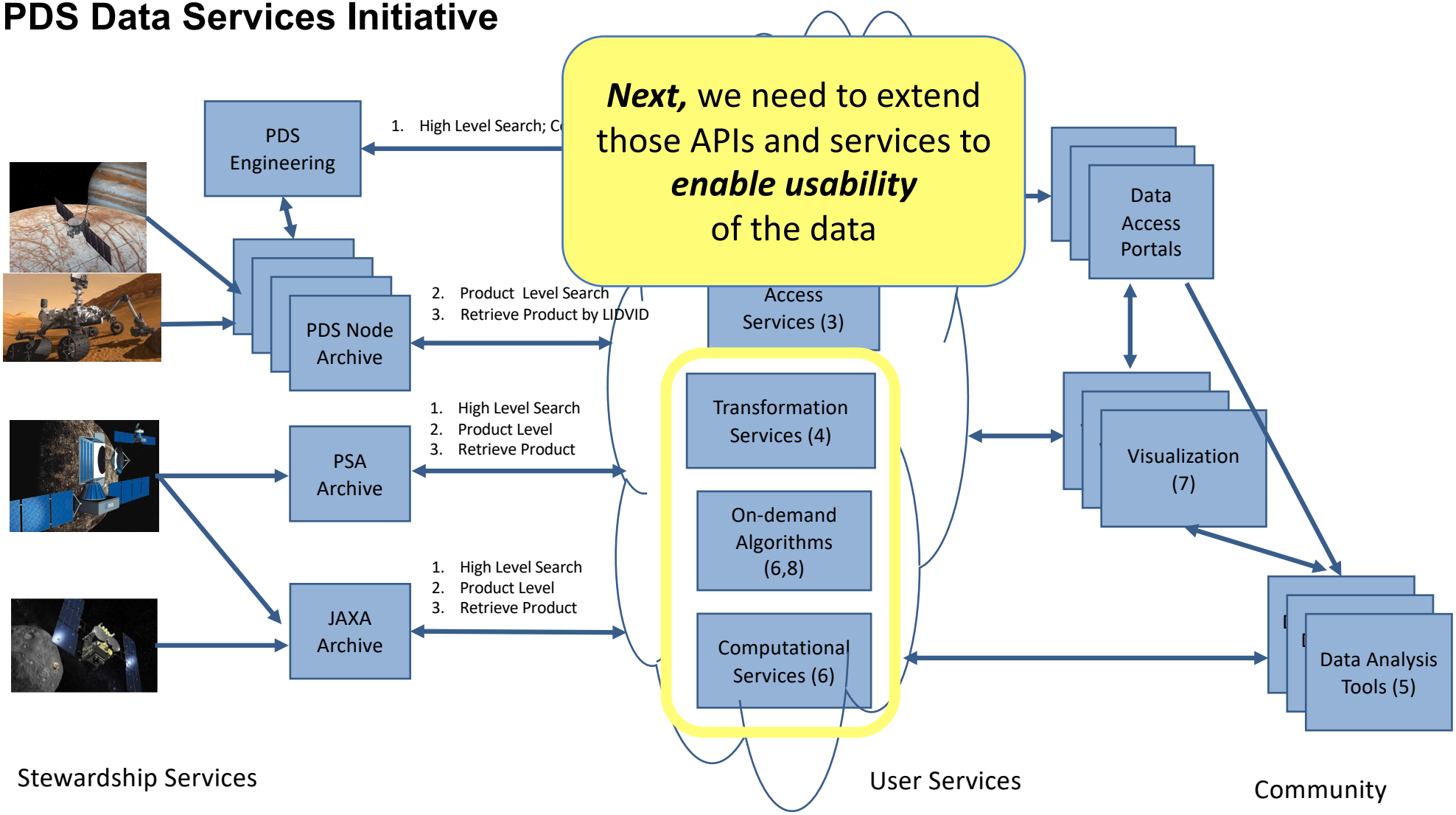
Cons

- Does not allow for *any* grouping of parameters (e.g. target identification across product classes)*
- Does not allow for searching non-archival attributes
- Adds additional complexity to request definitions
  - Will require POST request for large queries
- Not human-readable (only con for documentation sake)

Other API Specification Work

- Result / return formats (e.g. particular JSON definition)
- Query syntax for complex queries (e.g. SQL, Solr API, etc.)
- RESTful URI endpoints – similar REST endpoints at every PDS node
- HTTP request details – basic information on accessing the API
- How to specify return formats
- How to specify paging
- How to specify metadata you would like to return

PDS Data Services Initiative

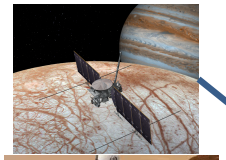


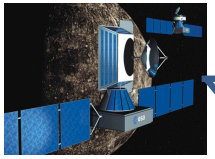
Enabling Usability of Data

Now that we have a baseline set of APIs for search, we can extend those to build out microservices for improving usability of data

- Extend APIs to transformations of data using either Transform Service or other transformation tools across the PDS
- Define programmatic APIs for accessing and manipulating data alongside the archive
- Development of transformation and other on-demand services or algorithms to manipulate data without requiring download
  - E.g. Jupyter notebooks, cloud-based transformation services

PDS Data Services Initiative





PDS Engineering

1. High Level Search; Content Products

PDS Node Archive

2. Product Leve
3. Retrieve Proc

PSA Archive

1. High Level Se
2. Product Leve
3. Retrieve Proc

JAXA Archive

1. High Level Search
2. Product Level
3. Retrieve Product

Search Services (1,2,9)

Then, we can move onto developing, improving, and integrating tools that ***access***, ***visualize*** and ***analyze*** the data

Algorithms (6,8)

Computational Services (6)

Data Access Portals

Visualization (7)

Data Analysis Tools (5)

Stewardship Services

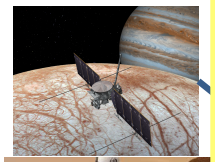
User Services

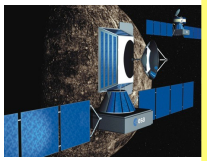
Community

Improving Discovery, Distribution and Usability

- New search, access, and usability services and APIs can then enable the development of new ways of interacting with the data including
  - Improving the provenance of our data
  - Using machine learning to better understand our data
  - Providing analysis platforms for minimum effort to perform scientific research including user defined algorithms
  - Providing novel data visualization technique for representing the data and research results

PDS Data Services Initiative





PDS Engineering

1. High Level earch; Content Products

PDS Node Archive

2. Prod ct Level
3. Retri ve Prod

PSA Archive

1. High evel Sea
2. Prod ct Level
3. Retri ve Produ

JAXA Archive

1. High evel Search
2. Prod ct Level
3. Retri ve Product

Search Services (1,2,9)

All while ***continuously improving***, ***curating***, and ***augmenting*** the metadata provided to and supplied by the archives

On-demand Algorithms (6,8)

Computational Services (6)

Data Access Portals

Visualization (7)

Data Analysis Tools (5)

Stewardship Services

User Services

Community

“Search is only good as the metadata you provide it.”

- Every person who has ever tried to develop a search engine

Improving Planetary Science Metadata

- ***Continuous curation, improvement, and augmentation of PDS archival data is critical to the success of PDS search***

- Develop best practices and guidelines for context / collection / bundle metadata to enable search
- Improve context product metadata
- Improve collection/bundle metadata
- Update PDS4 Information Model and PDS policies to allow non-archival analysis data to improve search
  - E.g. augmented metadata provided by science community