PPI Node Server

PDS Technical Session
April 6, 2005

Basis

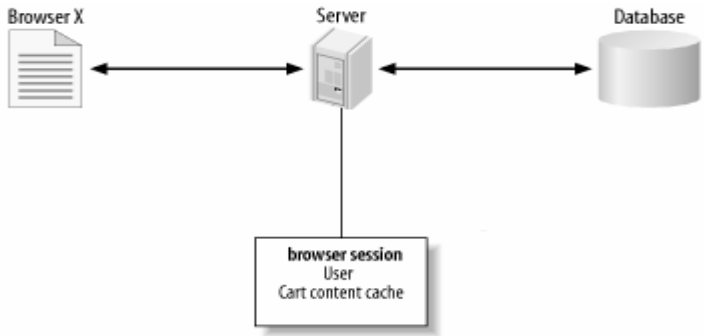
- Do the most needed in the most direct way.
  - Efficiency is important.
  - Don’t overburden a solution with technology.
    (Layered models help understand the problem, but are not the best solution)
- Metadata is the enabler
  - Metadata must be portable.
    - Never use the filesystem for metadata.
      (directory structure/names, file names, etc..)
    - Volume structure is for people.
  - Databases are transient.
    - Only the label is persistent
- Keep management close to the source.
  - Inventories should reside with the products.
  - Incremental build of datasets.

PPI Node Server

- Application server
  - Servlet
    - A servlet for each function (division of tasks).
    - Quick response (extension of server).
    - Easy to extend.
    - Portable – broad commercial support.
    - Well documented and supported interface.
  - Can be personalized
    - “portal” philosphy
    - “.pds” file type
  - Datacart
    - Shopping for data.

Data Cart

- A user can add a product to the cart while inspecting inventory.
- A user can take delivery of the cart contents as a single zip file.
- Contents of zip file are organized like a PDS volume.
  (i.e. data in a DATA directory, documents in a DOCUMENT directory)



Tools and Servlets

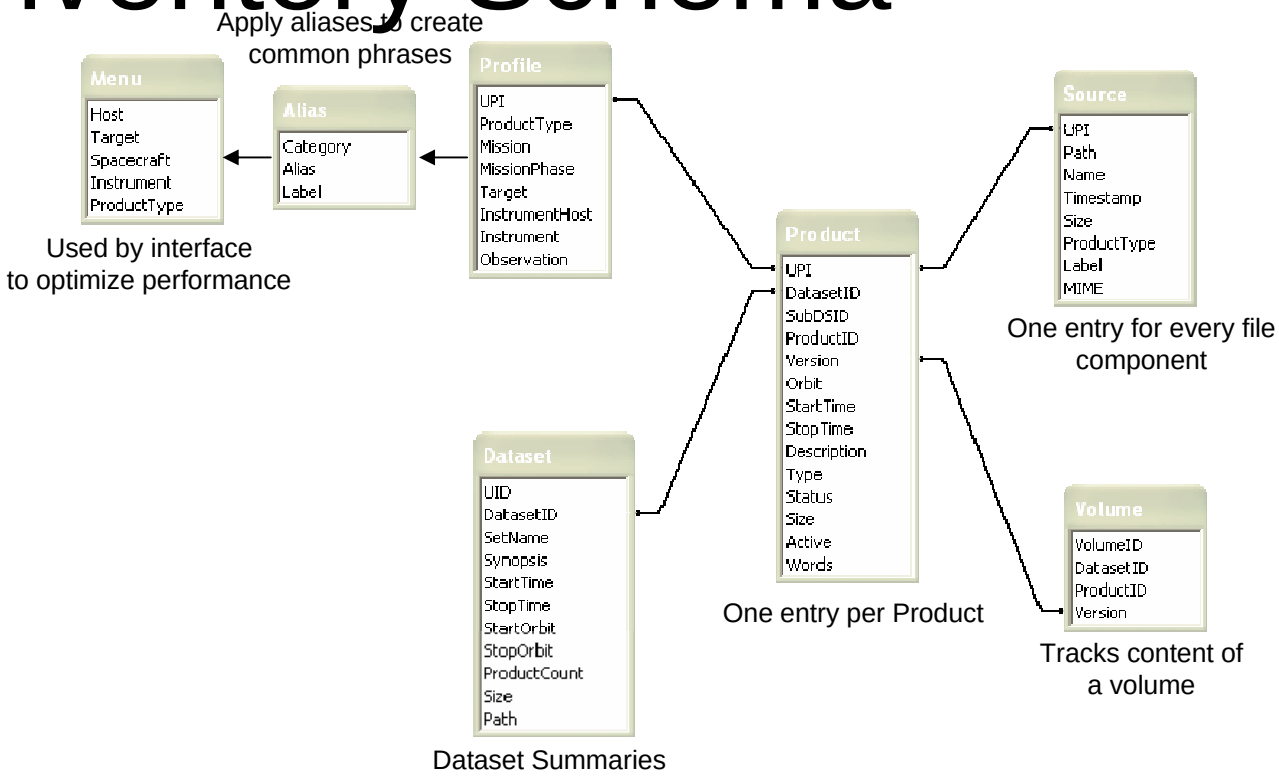
All Tools and servlets are written in Java.

- inventory: Scans labels and catalog files, extracts metadata and updates inventory database.
- labeler: A label generator with plug-ins.
- PDSLabel: A label parser class used by “inventory” and “labeler”.
- servlets – simple and task specific.

Servlets

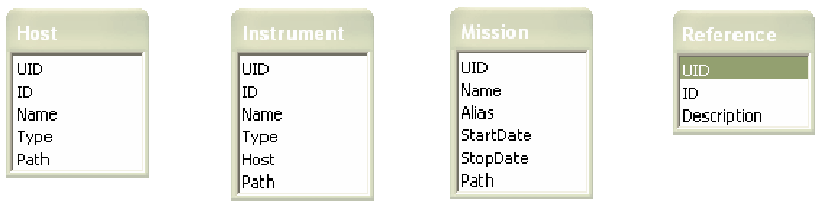
- AddCartServlet: Adds an item to the data cart.
- DatasetServlet: Queries for datasets.
- DeliverServlet: Package and stream products or data cart.
- DitdosServlet: Primary entry point, routes requests to specialized servlets.
- HostServlet: Queries host (spacecraft) table.
- InstrumentServlet: Queries instrument table.
- MissionServlet: Queries mission table.
- OptionsServlet: Access and management of session options.
- PageletServlet: Streams snippets of HTML.
- ProductServlet: Queries for products.
- ProfileServlet: Queries for profiles.
- ReferenceServlet: Queries the references table.
- SearchServlet: Presents search menus and routes requests.
- ShowCartServlet: View and manage the data cart.
- SourceServlet: Queries source file table.
- ViewServlet: Stream a file for viewing.

Inventory Schema

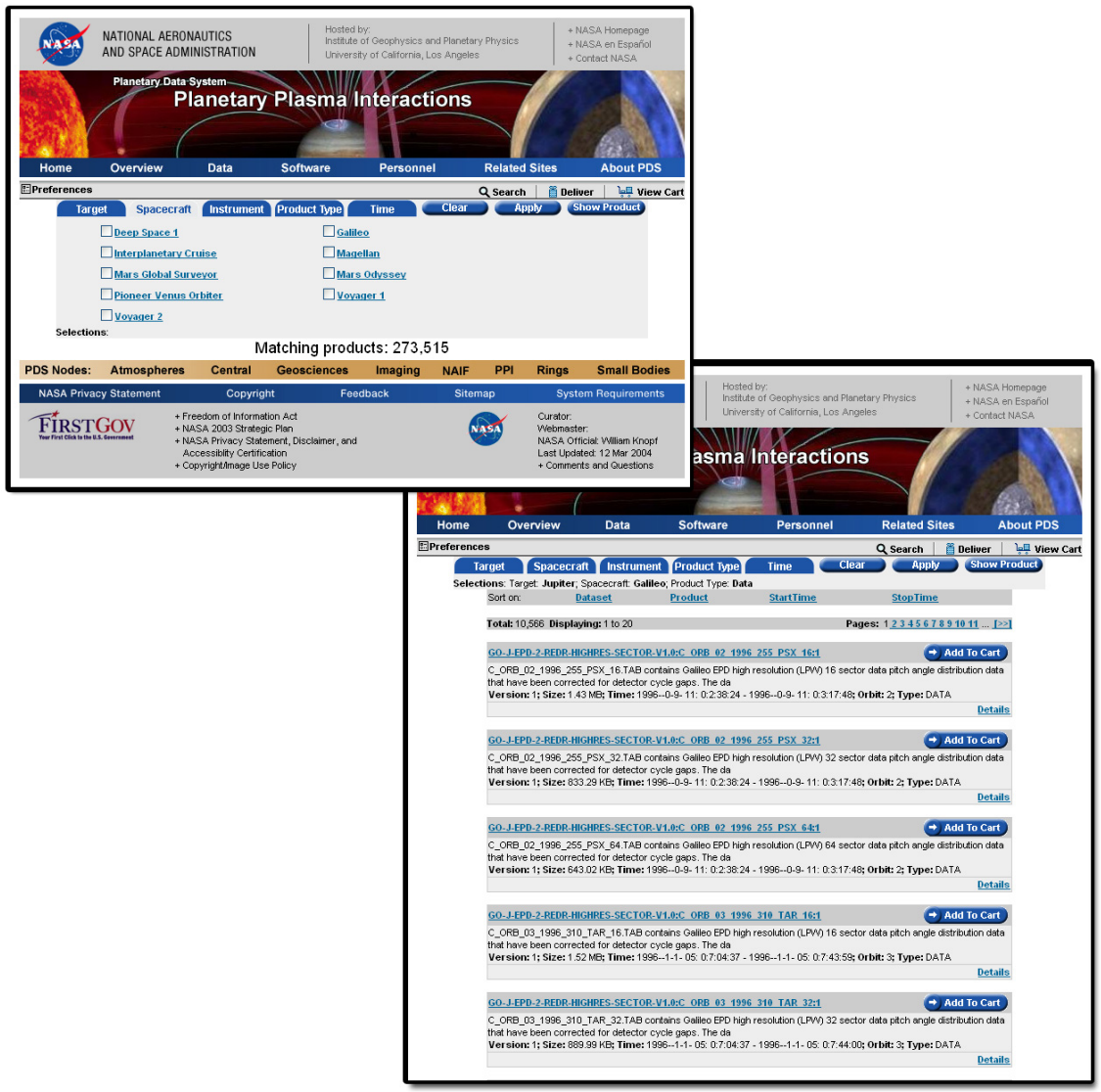


Catalog Tables

Used to provide depth for ID lookup



The Interface



What's Been Learned

- Connections to databases must be established at initialization.
  - Establishing a connection can take many seconds, so resident services (like servlets) are better than non-resident (cgi).
- Using a different connection for each type of query improves performance because of results caching.
  - Initial query (loading the cache) may take several seconds.
  - Subsequent queries can take 10th of a second.
- Scalability is a real issue.
  - The larger the inventory the longer it takes.
  - It appears that dividing inventory into many tables (i.e., one per dataset?) will improve performance.
- Choose the right time for locking in structure.
  - Incremental building of datasets means a volume isn't a volume until sometime in the future.
- Stability of Sun's Application Server is questionable.

Other Considerations

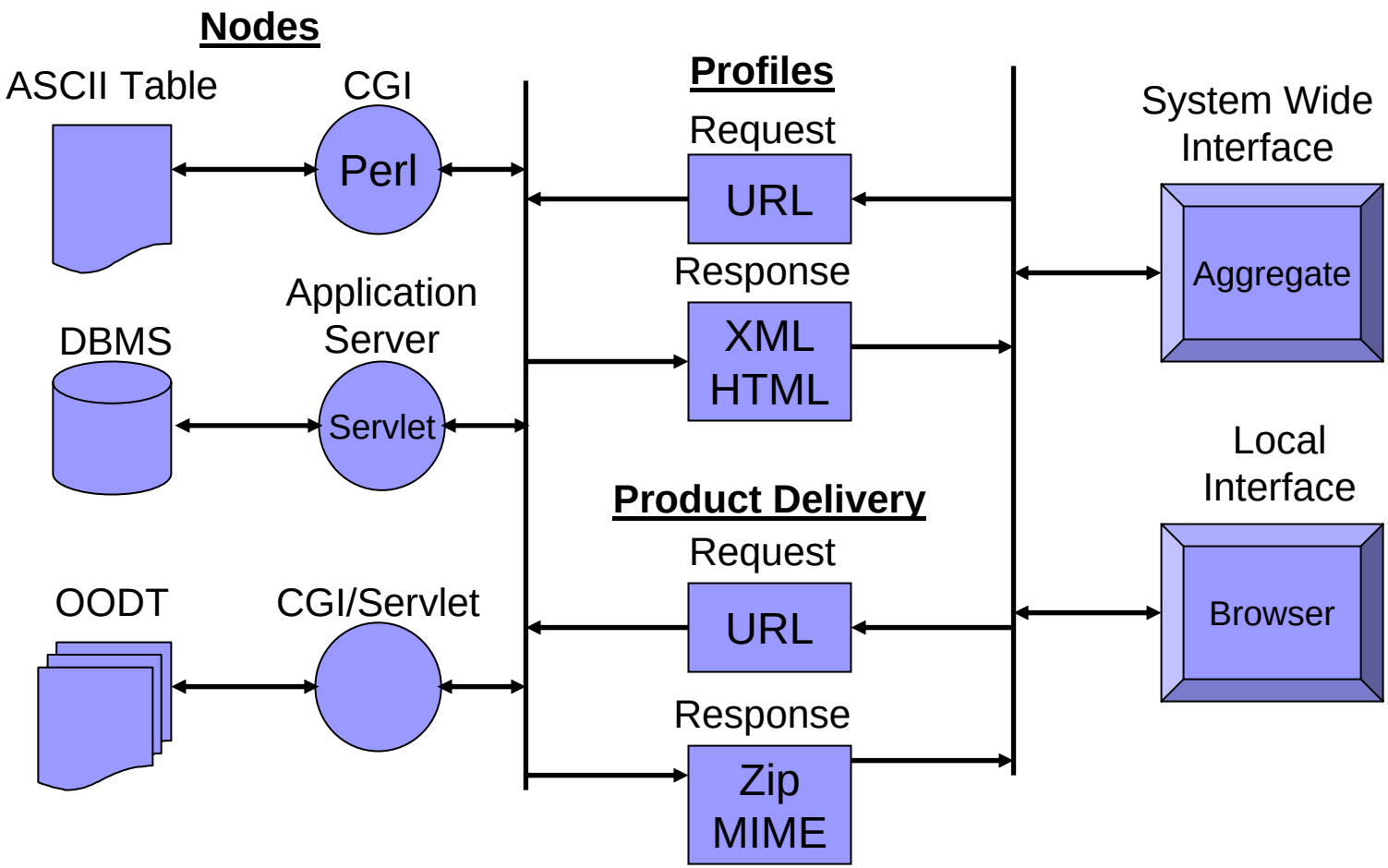
- Could OODT be used?
  - The Node server is a local service so multiple layers of abstraction is unnecessary.
  - Currently queries are to a MySQL database which resides on the same server. Any database system could be used.
- System Design
  - Define messages not implementations

    because
    - Success of the web is URLs and HTML.
    - PDS labels are effective because they are independent of implementation.
    - All components of an implementation should be replaceable. Message content (metadata) is persistant.
    - Nodes and projects have different environments and need flexibility in implementation.

System Wide Services

- System Design
  - Define messages not implementations

    because

    - Success of the web is URL and HTML.
    - PDS labels are effective because they are independent of implementation.
    - All components of an implementation should be replaceable. Message content (metadata) is persistant.
    - Nodes and projects have different environments and need flexibility in implementation.
    - Single interface (multiple implementations)

System Wide Search Service