



# OODT: Object Oriented Data Technology

## CSMISS IT Spotlight

### November 6, 2000

Dan Crichton/JPL
Steve Hughes/JPL

Science Data Management and Archiving Section (389)

*Jet Propulsion Laboratory, California Institute of Technology*
*National Aeronautics and Space Administration*

# *Object Oriented Data Technology Task*

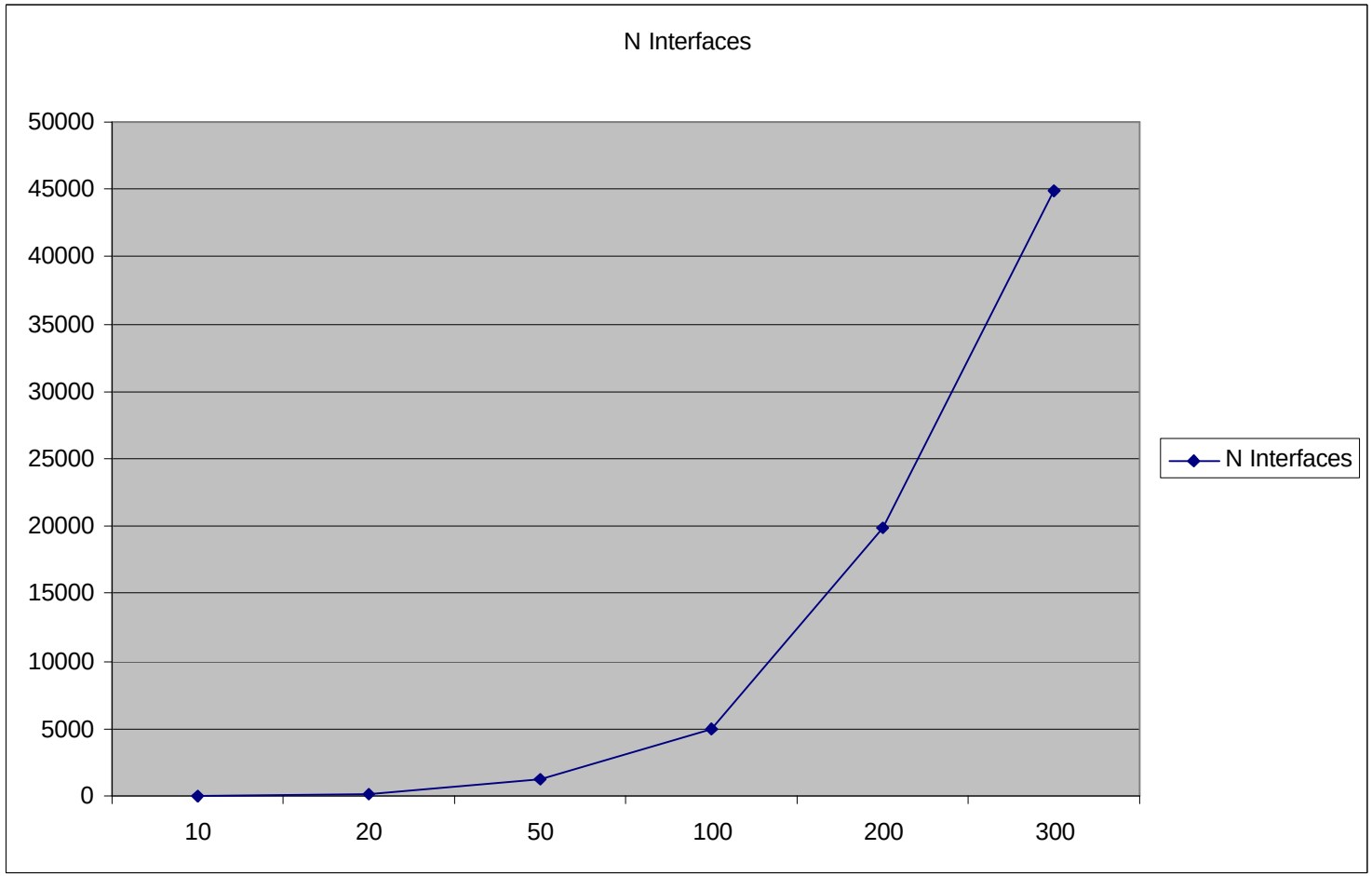
- Technology research task funded by the Office of Space Science (OSS) at NASA
- Part of the Space Science Applications of Information Technology (SAIT) program at JPL
- Started in 1998 and funded @ .5 FTE. 1999 & 2000 Funded @ 1.5 FTEs.
- Investigate new data system technologies for supporting data management, knowledge management and knowledge discovery
- Build data system solutions that are cross disciplinary and address interoperability between these systems

# *Problem Statement*

- Interoperability is an important *key* to unlock knowledge discovery
  - Allows scientists the ability to locate critical information
  - Enables *knowledge management and discovery* across the agency
  - Can be a key to scientific discovery

- But, interoperability is difficult. Data systems across the agency are
  - Difficult to access (no standard interfaces)
  - Geographically distributed
  - Have no standard language or protocol for data interchange
  - No common metadata model agency wide
  - Have no system for registration of data products agency wide
  - Have different internal representations for data products

# Managing Software Interfaces (# of Systems vs # of Interfaces)

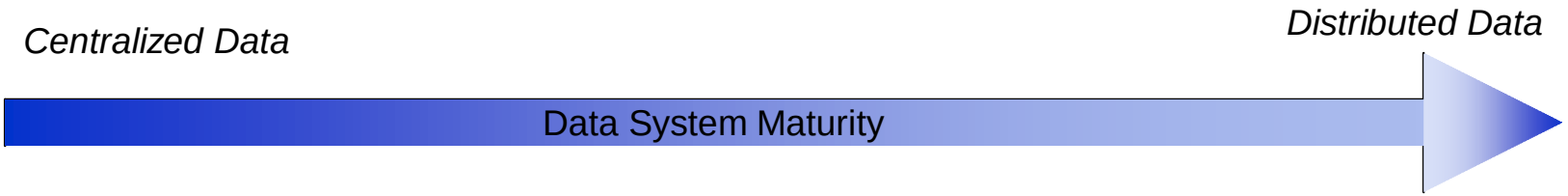



# of interfaces = $(n^2 - n) / 2$

# *OODT System Design Goals*

- Encapsulate individual data systems to hide uniqueness
- Provide data system location independence
- Require that communication between distributed systems use metadata
- Define a standard data dictionary structure and approach for describing systems and resources
- Provide a scalable and extensible solution
- Provide a mechanism for data product exchange
- Allow systems using different data dictionaries and metadata implementations to be integrated
- Define an architecture that can leverage off of open standard approaches

# OODT Components for Data System Maturity




| Basic Data Infrastructure | Data Archiving | Data Location | Data Product Exchange/ Interoperability |
|---|---|---|---|
| - Data Acquisition | - Catalog Systems | - Metadata | - Heterogeneous Servers |
| - Databases | - Data sets | - Distributed Data sets | - Data Interchange |
| - Data Analysis Tools | - Data Products | - Distributed Services | - Data Sharing |
| - Homogeneous Computing | - Metadata | | - Distributed Architectures |

# OODT Distributed Architecture

- Java based software middleware component architecture that provides a software framework for archiving, search and retrieval, and data product exchange
  - Archive Component
    - Provides centralized data archiving and cataloging of data products
    - Distributed
  - A Search and Retrieval Component
    - Manage metadata associated with resources
    - Locate resources across geographically distributed data systems
    - Distributed
  - Data Product Exchange Component
    - Support interchange (data sharing) of data products
    - Support heterogeneous implementations and systems
    - Distributed
  - Query Service Component
    - Ties search and product exchange services together
    - Distributed

# *OODT Technology Focus*

- Focus on building *middleware* components
- Focus on creating metadata “profiles” about data system resources
- Provide sufficient layers of abstraction in the architecture to isolate technology choices from architecture choices
  - XML (Extensible Markup Language) for the data content
  - CORBA (Common Object Request Broker Architecture) for the data transport
- Research technologies for implementing a distributed data architecture
  - Distributed Object Computing (CORBA, DCOM, etc)
  - Database Technology (RDBMS, ODBMS)
  - Data Access Technologies (O/JDBC, XML, etc)
  - Directory Implementations (LDAP)
  - Data Interchange (XML)
  - Communication Technologies (Web/HTTP, MOM, RPC, etc)

# OODT Prototype Environment

- XML parser: Apache/IBM Xerces 1.0.3

http://xml.apache.org

- XSLT: Apache Xalan 1.0.0

http://xml.apache.org

- CORBA: Orbacus 4.0.3

http://www.ooc.com

- Database: Oracle 8.1.5

http://www.oracle.com

- LDAP server: OpenLDAP 1.2.11

http://www.openldap.org

- Development language: Java 1.2

http://java.sun.com

- Web server: iPlanet Fasttrack 4.1

http://www.iplanet.com

- Server operating system: RedHat Linux 6.2

http://www.redhat.com

- Version control system: CVS 1.10.5

http://www.cvshome.org

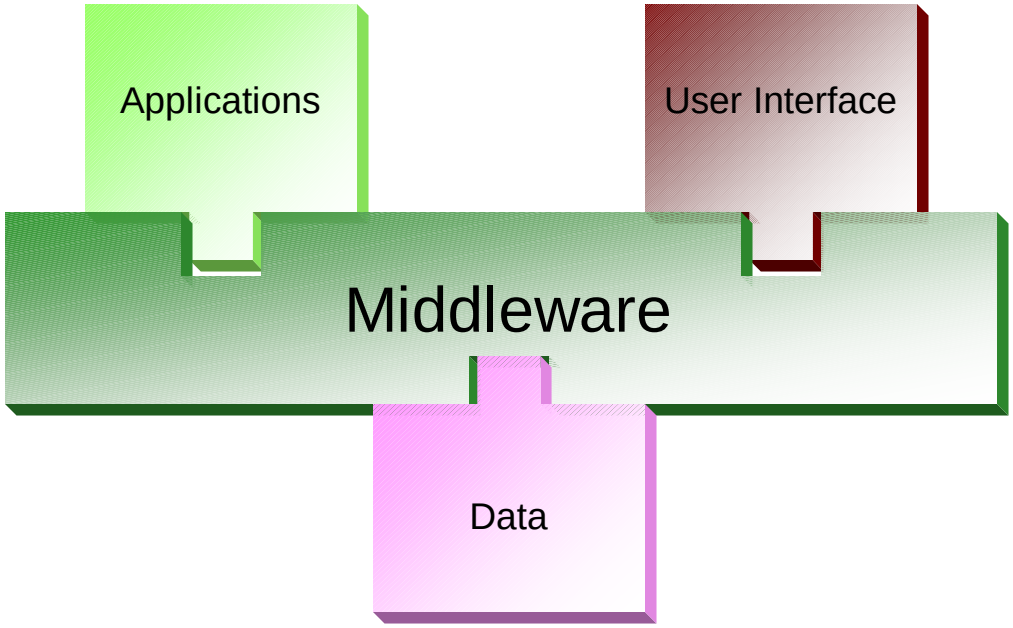
# *Focus on Middleware*

In the computer industry, middleware is a general term for any programming that serves to *"glue together"* or *mediate* between two separate and usually already existing programs. A common application of middleware is to allow programs written for access to a particular database to access other databases.

Messaging is a common service provided by middleware programs so that different applications can communicate. The systematic tying together of disparate applications is known as *enterprise application integration*.

*http://www.whatis.com*

# Role of Middleware




Middleware can tie application, data, and user interfaces together and hide the unique interfaces

# *Motivation for Middleware*

- Middleware allows for the encapsulation of individual data systems
  - Hide uniqueness by introducing the data architecture layer
- Allows for data abstraction
  - Provide common client interfaces to heterogeneous systems
  - Manage risk associated with technical decisions. Systems evolve independent of the clients.
- Enable reuse and promote standards
- Allow for incompatible systems to be tied together by introducing a middleware layer

# *Focus on Metadata*

- Metadata is data about data
- Provides descriptive information about the data
  - Classification, identification, etc
  - Metadata Example
    - Data Value: 55 (not descriptive)
    - Metadata Values:
      - Data Element Name:Vehicle_Speed
      - Unit: Miles per Hour
      - Description: The average velocity of a vehicle.
- Use standards where appropriate
  - ISO/IEC 11179 – A framework for the Specification and Standardization of Data Elements
  - Dublin Core – A metadata element set intended to facilitate discovery of electronic resources.

# *OODT Metadata Research*

- Develop methods for managing the semantics of data that are shared within and between domains
  - Terminology Base – Domain specific name space
  - Data Dictionary – Inventory of domain terms with definitions and other distinguishing attributes.
  - Ontology – A set of concepts, their relationships and constraints, all within the scope of a domain.
  - XML for metadata registry and communication
- Several I.T. efforts have shown the criticality of metadata in enabling data sharing and system interoperability

# *Why XML for OODT?*

- XML doesn't provide a "silver bullet," but it does allow us to refocus the problem on metadata
  - Metadata is a key to interoperability (http://www.cio.gov/docs/metadata.htm)
- XML is language neutral
- Allows the designer to separate the data and the transport (re: CORBA vs XML-over-CORBA)
  - Transport mechanism and data are not tied together
    - Could be XML/HTTP
    - Simpler deployments
    - Simpler interfaces
    - Allows technologies to grow and change independently
- Real value of XML is the process of describing the data

# CORBA vs XML over CORBA

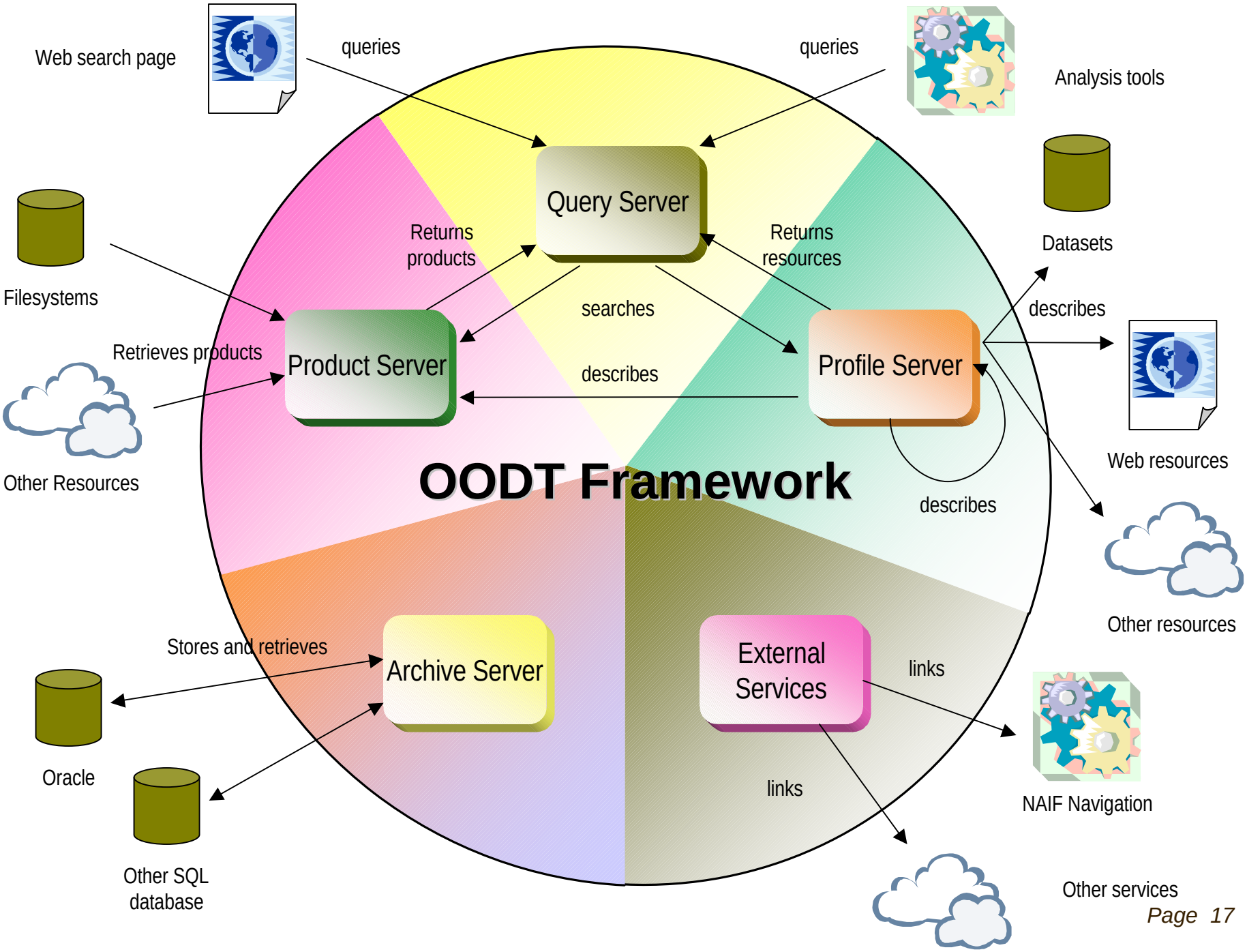
- CORBA method

```
module jpl {
   module user {
     interface UserManager {
        User findUser(string
                      name);
     };
     interface User {
       String getName();
     };
   };
};
```

- XML over CORBA/IIOP

```
module jpl {
   module user {
     interface UserManager {
       string do(string xml);
     };
   };
};
```

```
<transaction>
   <findUser>
     <user>
<surname>Doe</surname>
     </user>
   </findUser>
</transaction>
```

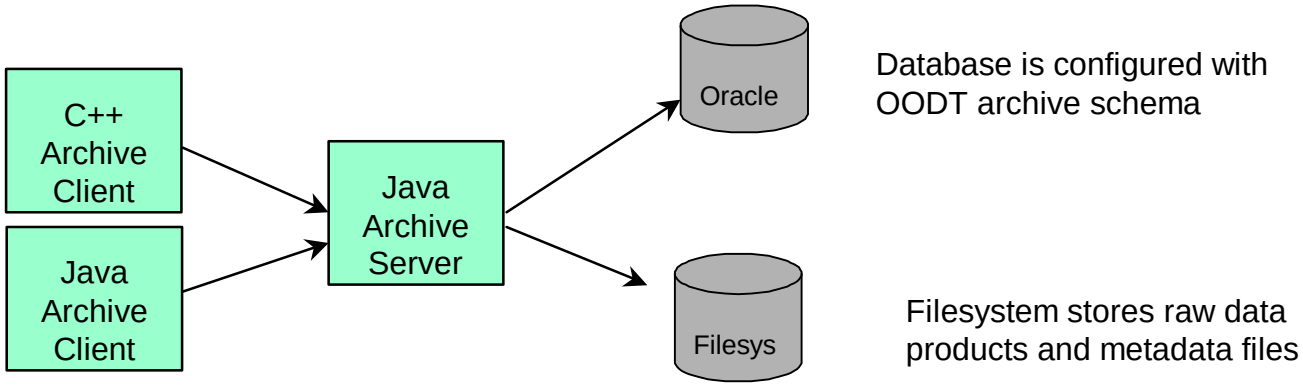

Web search page
queries
queries
Analysis tools
Query Server
Returns products
Returns resources
Datasets
Filesystems
searches
describes
Retrieves products
Product Server
Profile Server
describes
Other Resources
OODT Framework
describes
Web resources
Other resources
Stores and retrieves
Archive Server
External Services
links
Oracle
links
NAIF Navigation
Other SQL database
Other services

# *Data Archiving Goals*

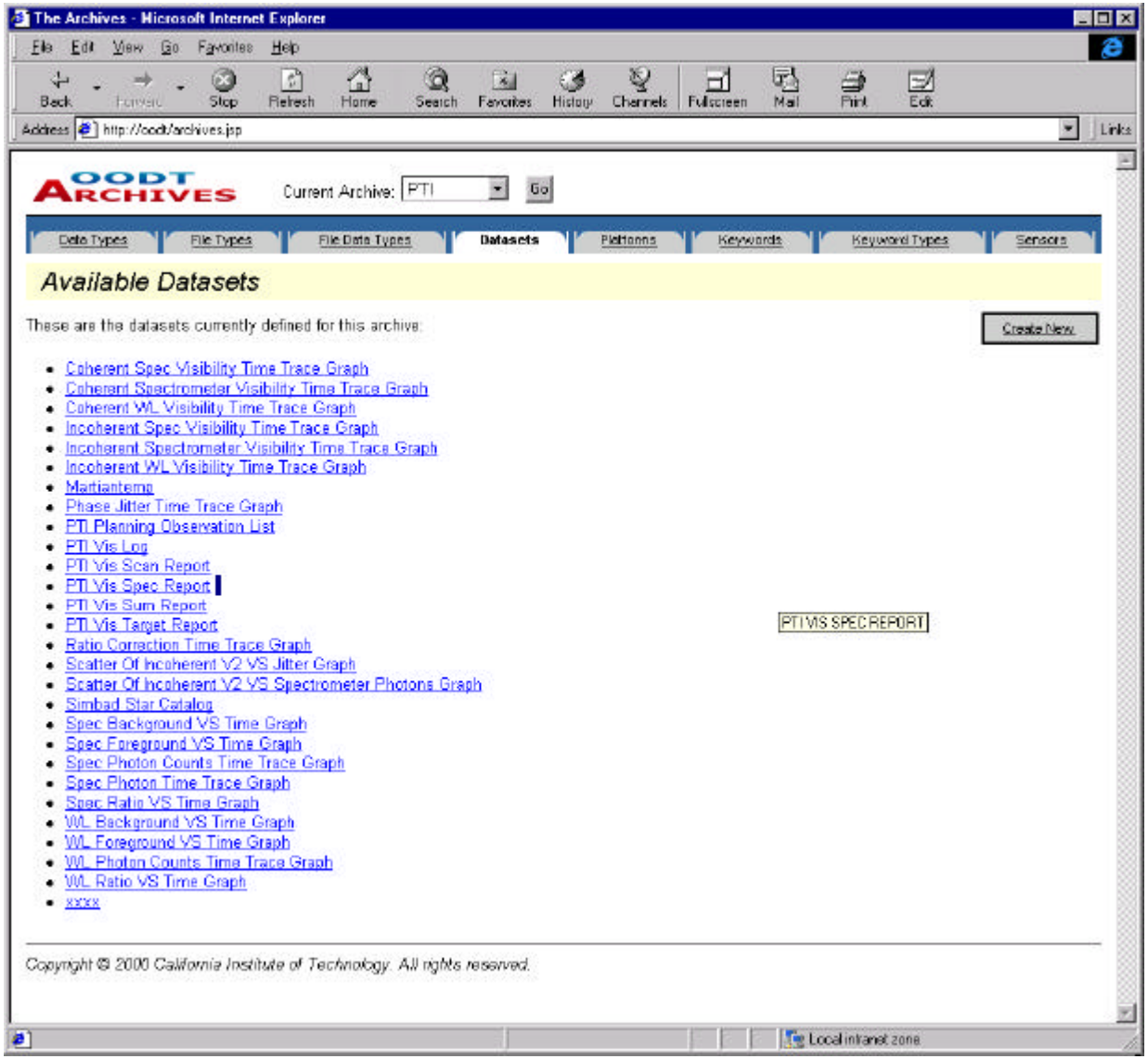
- Provide basic functions
  - Transport and management of data sets and products
  - Identification of products using metadata
  - Event driven processing associated with data sets
  - Ability to add, get and delete products from the archive
- Provide extensible data management approach
  - Database is dynamically generated and extended based on metadata content
- Build a service that is accessible via clients using common programming languages (Java, C++, etc)

# OODT Generic Archive Service Component

- Catalogs data sets and products using a client/server architecture
- Archive server is written in CORBA and provides the mechanism to move products from the client to the server
- Data sets are configurable and use metadata for managing the product catalog
- Archive server provides transaction management for adding, updating and removing data products
- Prototype implemented using institutional Oracle 8i service



# Archive Management



# *Data Search and Retrieval*

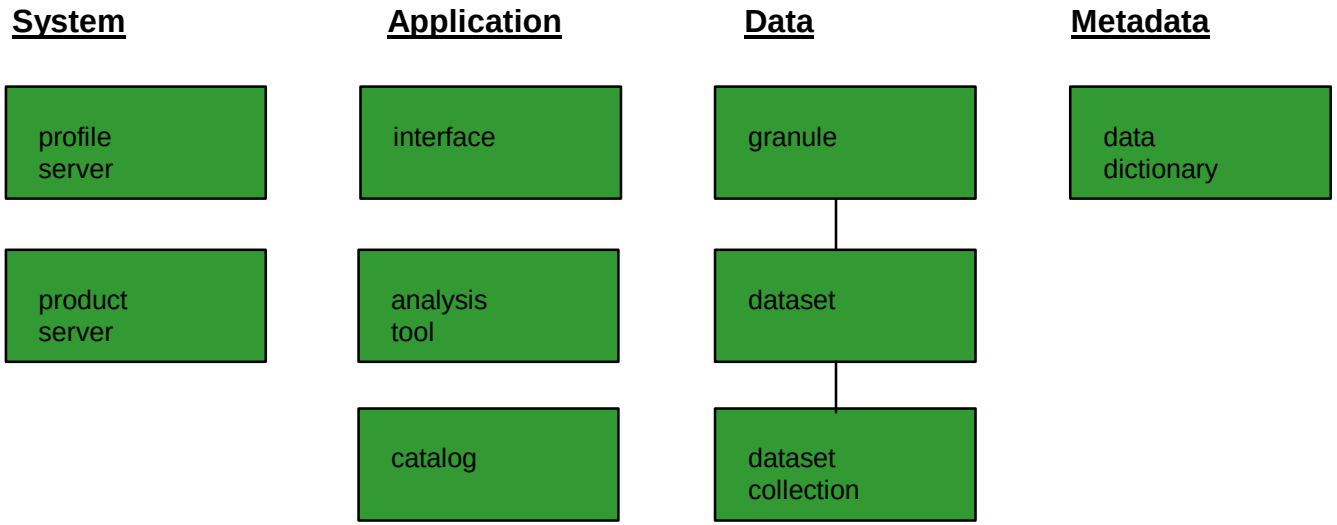
- Space scientists cannot easily locate or use data across the hundreds if not thousands of autonomous, heterogeneous, and distributed data systems currently in the Space Science community.
- Heterogeneous Systems
  - Data Management - RDBMS, ODBMS, HomeGrownDBMS, BinaryFiles
  - Platforms - UNIX, LINUX, WIN3.x/9x/NT, Mac, VMS, …
  - Interfaces - Web, Windows, Command Line
  - Data Formats - HDF, CDF, NetCDF, PDS, FITS, VICR, ASCII, ...
  - Data Volume - KiloBytes to TeraBytes
- Heterogeneous Disciplines
  - Moving targets and stationary targets
  - Multiple coordinate systems
  - Multiple data object types (images, cubes, time series, spectrum, tables, binary, document)
  - Multiple interpretations of single object types
  - Multiple software solutions to same problem
  - Incompatible and/or missing metadata

# *What is a profile?*

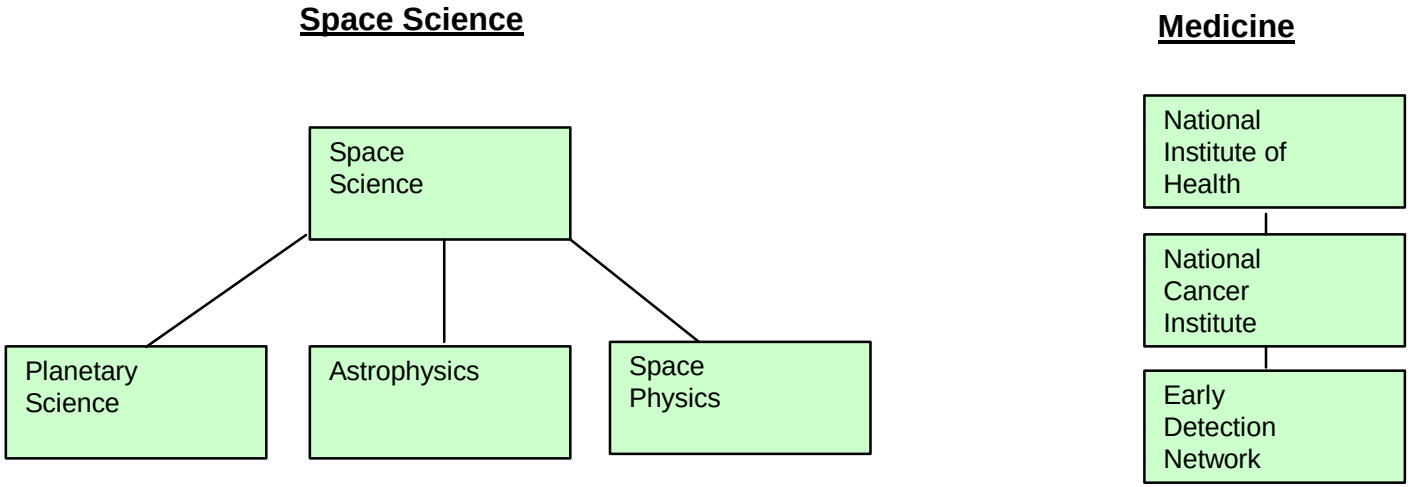
- Sets of resource definitions describing information about distributed data systems and their products
- Metadata descriptions of resources
- Examples:
  - Data Systems
  - Data Sets
  - Data Products
  - Interfaces
  - Other profiles

# Resource Profile Classifications

***Resource Classes***




***Resource Context (Discipline )***



# *Solutions to Data Search*

- Build metadata “profiles” that describe data system resources
  - Encapsulate individual data systems resources (Hide uniqueness)
  - Communicate using metadata (Provide metadata with data)
  - Enable interoperability based on metadata compatibility
  - Refocus problem on metadata development
  - Provide a core framework of software components to interconnect distributed data systems
- Define profiles using standard industry approaches
  - Use XML to describe profiles
  - ISO/IEC 11179 – A framework for the Specification and Standardization of Data Elements
  - Dublin Core – A metadata element set intended to facilitate discovery of electronic resources.

# Profile DTD

```
<!ELEMENT profiles
  (profile+)>

<!ELEMENT profile
  (profAttributes,
   resAttributes,
   profElement*)>

    <!ELEMENT profAttributes
      (profId, profVersion*, profTitle*, profDesc*, profType*,
       profStatusId*, profSecurityType*, profParentId*, profChildId*,
       profRegAuthority*, profRevisionNote*, profDataDictId*)>

    <!ELEMENT resAttributes
      (Identifier, Title*, Format*, Description*, Creator*, Subject*,
       Publisher*, Contributor*, Date*, Type*, Source*,
       Language*, Relation*, Coverage*, Rights*,
       resContext*, resAggregation*, resClass*, resLocation*)>

    <!ELEMENT profElement
      (elemId*, elemName, elemDesc*, elemType*, elemUnit*,
       elemEnumFlag*, (elemValue | (elemMinValue, elemMaxValue))*,
       elemSynonym*,
       elemObligation*, elemMaxOccurrence*, elemComment*)>
```

# XML Profile Example (1 of 2)

```
<profile>
  <profAttributes>
    <profId>OODT_PDS_DATA_SET_INV_82</profId>
<profDataDictId>OODT_PDS_DATA_SET_DD_V1.0</profDataDictId>
  </profAttributes>
  <resAttributes>
    <Identifier>VO1/VO2-M-VIS-5-DIM-V1.0</Identifier>
    <Title>VO1/VO2 MARS VISUAL IMAGING SUBSYSTEM DIGITAL …</Title>
    <Format>text/html</Format>
    <Language>en</Language>
    <resContext>PDS</resContext>
    <resAggregation>dataSet</resAggregation>
    <resClass>data.dataSet</resClass>
  <resLocation>http://pds.jpl.nasa.gov/cgi-bin/pdsserv.pl?…</resLocation>
  </resAttributes>
```

# XML Profile Example (2 of 2)

```
<profElement>
      <elemId>ARCHIVE_STATUS</elemId>
      <elemName>ARCHIVE_STATUS</elemName>
      <elemType>ENUMERATION</elemType>
      <elemEnumFlag>T</elemEnumFlag>
      <elemValue>ARCHIVED</elemValue>
    </profElement>
    <profElement>
      <elemId>TARGET_NAME</elemId>
      <elemName>TARGET_NAME</elemName>
      <elemType>ENUMERATION</elemType>
      <elemEnumFlag>T</elemEnumFlag>
      <elemValue>MARS</elemValue>
    </profElement>
</profile>
```

# *OODT Profile Service Component*

- Profiles are managed by profile servers
- Profile servers are written in Java
- OODT currently has three different registry methods for managing profiles which are configurable at run time
  - Flat File
  - RDBMS via JDBC (Oracle)
  - LDAP (OpenLDAP)

# *Data Product Exchange*

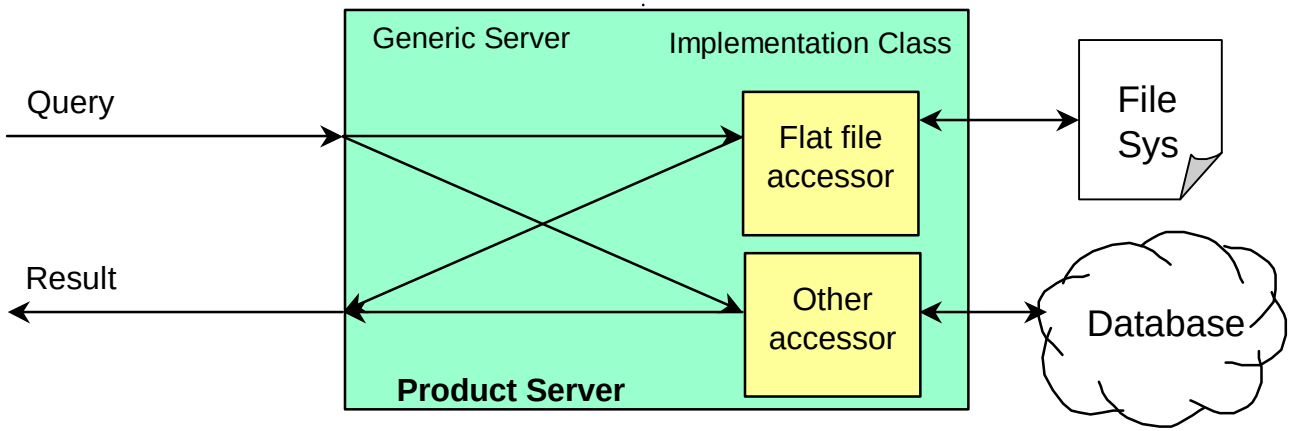
- Exchanging products requires access to each data system (RDBMS/OODBMS, Flat file, etc) which is difficult
  - Different vendor products
  - Non-standard interfaces
  - Different implementations (data model, home grown, COTS,etc)
  - Representations of data are different
  - Heterogeneous Platforms
  - Heterogeneous O/S
  - etc

# *Solutions to Data Product Exchange*

- Extend framework to support common access to distributed data systems by creating a “Product Service Component”
  - Product Servers - Middleware that negotiates the interfaces between the data system implementations
- Design the component to leverage off of
  - Consistent metadata and data dictionary
  - Consistent data interchange methods and protocols
- Provide data abstraction
  - Data and information hiding
  - Location hiding and independence
- Provide a standard language for communication
  - Use the OODT XML Query language for data interchange
    - Support “rich” query description including data elements and constraints
    - Support “rich” query results that include results in many different formats

# OODT Product Server Component

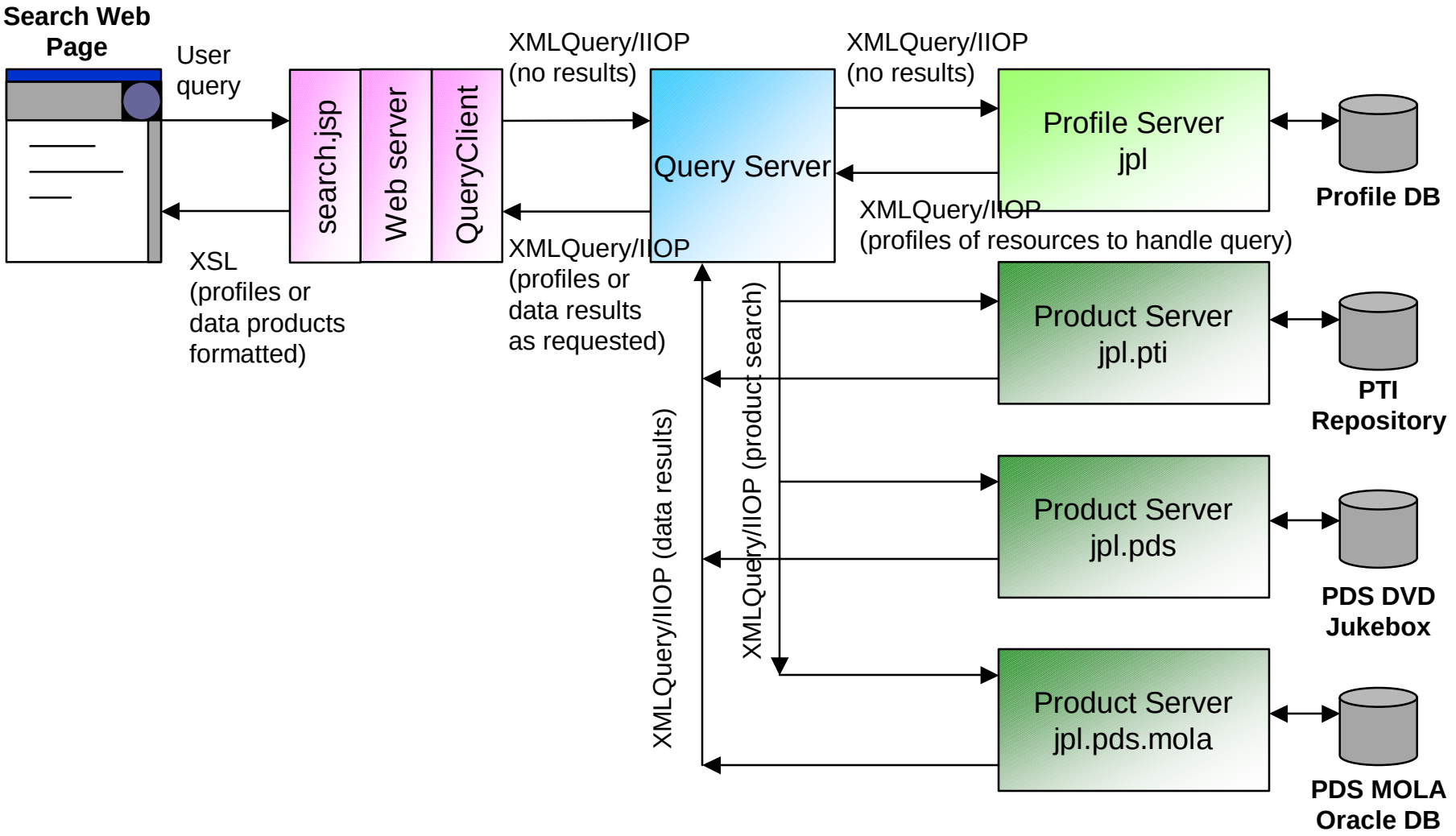
- The Product Server plugs into the OODT framework and manages the “handshake” between the data system and the OODT system.
- Extensible by dynamically loading objects at runtime which are specific to the data system model
- Queries and results are passed using an OODT XML Query structure
- Encapsulates one or more data sources for standardized access



# *OODT Query Service Component*

- Manages all queries for the identification and retrieval of data products
- All components are identified by a unique name and managed in a CORBA name server
- Queries to multiple profile or product servers occur concurrently
- Queries are described using the OODT XML Query structure
- Ties together the profile and product server components for the OODT framework

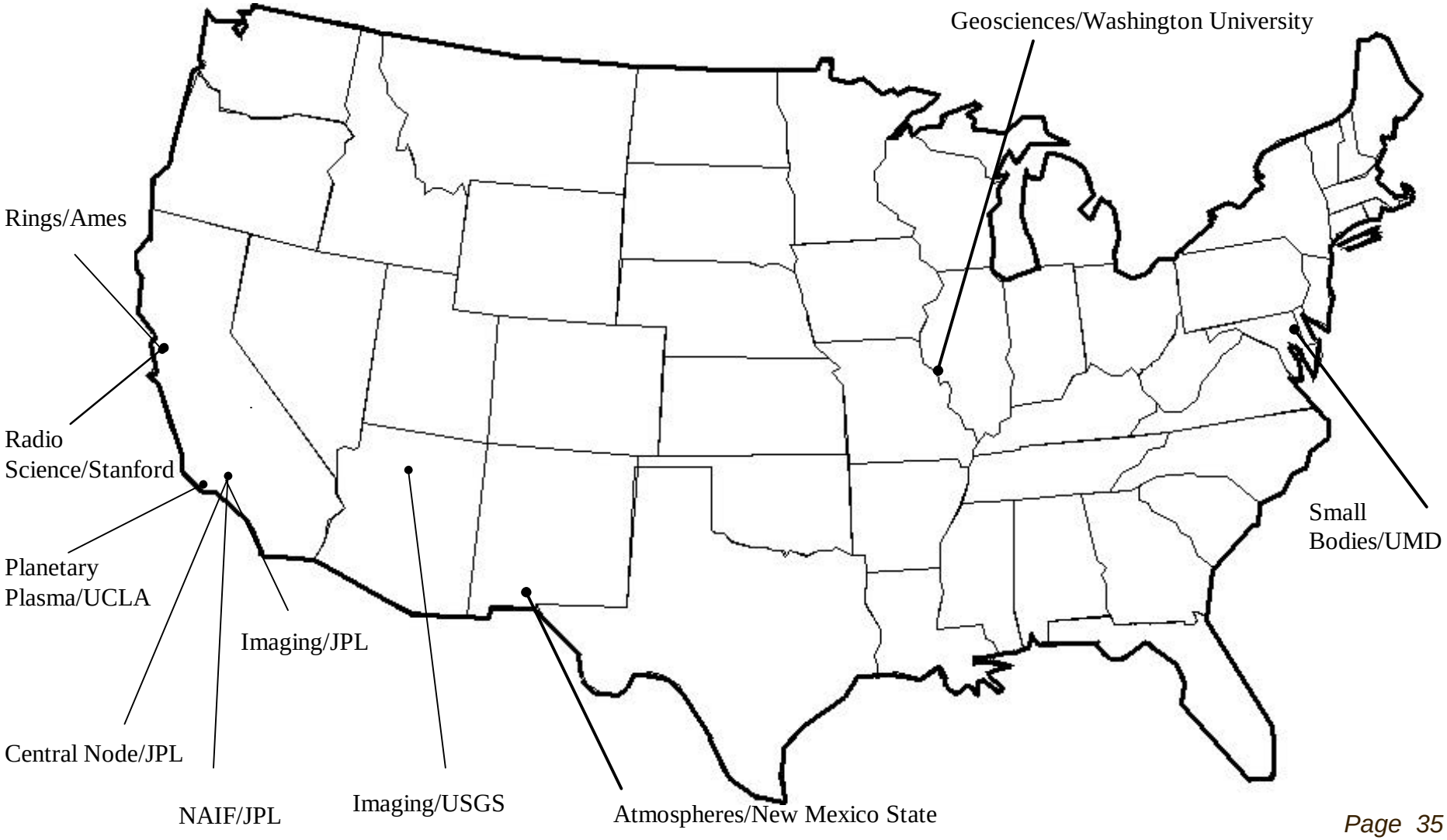
# OODT Query Flow Example



# *OODT Insertion in the PDS*

- PDS is the official planetary science data archive for NASA.
- PDS is a distributed system designed to optimize scientific oversight in the archiving process.
- OODT is focusing on insertion of technology into PDS
  - Providing a long term architecture to improve the ability for scientists to retrieve data within the PDS
    - Refocus the problem away from technology solutions
    - Provide and leverage the existing metadata infrastructure
  - Providing solutions to access and correlate heterogeneous data products and systems
  - Supporting the PDS distributed node architecture

# PDS Nodes and Institutions (Silos)



# *Other OODT Efforts*

- Early Detection Research Network from the National Cancer Institute (NCI)
  - Initiating a prototyping effort to link two centers together to demonstrate interoperability
- Children's Hospital, Los Angeles and Johns Hopkins Medical Institute
  - Interested in using JPL OODT technology to link pediatric physiological data between the hospitals
- ICIS Funded Enterprise Data Architecture (EDA) effort to build core components as part of JPL's infrastructure

# More Information

- OODT Papers (http://oodt/doc/papers)
  - "Science Search and Retrieval using XML" by OODT Team. Presented at Second National Conference on Scientific and Technical Data, National Academy of Sciences, Washington D.C. March 2000.
  - "A Distributed Component Framework for Science Data Product Interoperability" by OODT Team. Presented at the 17th Annual International CODATA conference. Baveno, Italy. October 2000.
- Planetary Data System

  http://pds.jpl.nasa.gov
- Dublin Core

  http://purl.oclc.org/dc
- Extensible Markup Language

  http://www.w3c.org/XML
- ISO/IEC 11179: Specification and Standardization of Data Elements
- Federal CIO Statement on Metadata

  http://www.cio.gov/docs/metadata.htm

# *Backup Slides*

# *XML Query Example (1 of 2)*

```
<query>
 <queryAttributes>
  <queryId>OODT_XML_QUERY_V0.1</queryId>
  <queryTitle>OODT_XML_QUERY - PDS DIS Query Example</queryTitle>
  <queryDesc>PDS DIS Query for TARGET_NAME = MARS</queryDesc>
  <queryType>QUERY</queryType>
  <queryStatusId>ACTIVE</queryStatusId>
  <querySecurityType>UNKNOWN</querySecurityType>
  <queryRevisionNote>2000-05-12 JSH V1.2 Updated for new
                                                    prof.dtd</queryRevisionNote>
  <queryDataDictId>OODT_PDS_DATA_SET_DD_V1.0</queryDataDictId>
 </queryAttributes>
 <queryResultModeId>ATTRIBUTE</queryResultModeId>
 <queryPropogationType>BROADCAST</queryPropogationType>
 <queryPropogationLevels>N/A</queryPropogationLevels>
 <queryMaxResults>100</queryMaxResults><queryResults>0</queryResults>
 <queryKWQString>TARGET_NAME = MARS</queryKWQString>
```

# XML Query Example (2 of 2)

```
<querySelectSet></querySelectSet>
 <queryFromSet></queryFromSet>
 <queryWhereSet>
  <queryElement>
   <tokenRole>elemName</tokenRole>
   <tokenValue>TARGET_NAME</tokenValue>
  </queryElement>
  <queryElement>
   <tokenRole>LITERAL</tokenRole>
   <tokenValue>MARS</tokenValue>
  </queryElement>
  <queryElement>
   <tokenRole>RELOP</tokenRole>
   <tokenValue>EQ</tokenValue>
  </queryElement>
 </queryWhereSet>
 <queryResultSet></queryResultSet>
</query>
```