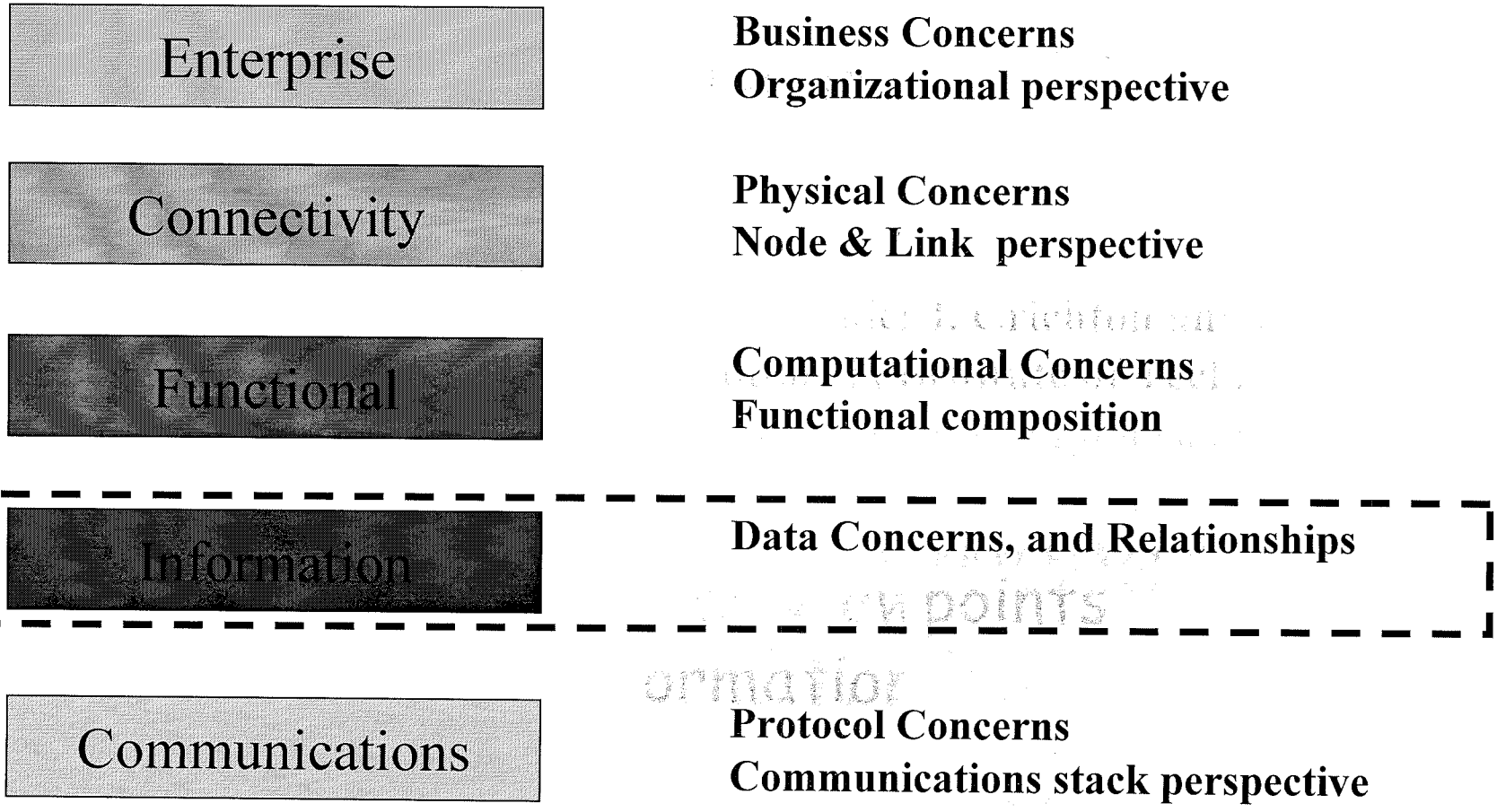
Packaging Data Products Using Data Grid Middleware for Deep Space Mission Systems

Chris A. Mattmann, Paul M. Ramirez, Daniel J. Crichton and J.S. Hughes
Jet Propulsion Laboratory, California Institute of Technology
{mattmann,pramirez,crichton,jshughes}@jpl.nasa.gov

May 2004
8th International Conference on Space Operations (Spaceops 2004)

Thanks to L. Reich, P. Shames and the rest of the CCSDS Architecture Working Group for some of these slides!

Architectural Viewpoints
Information



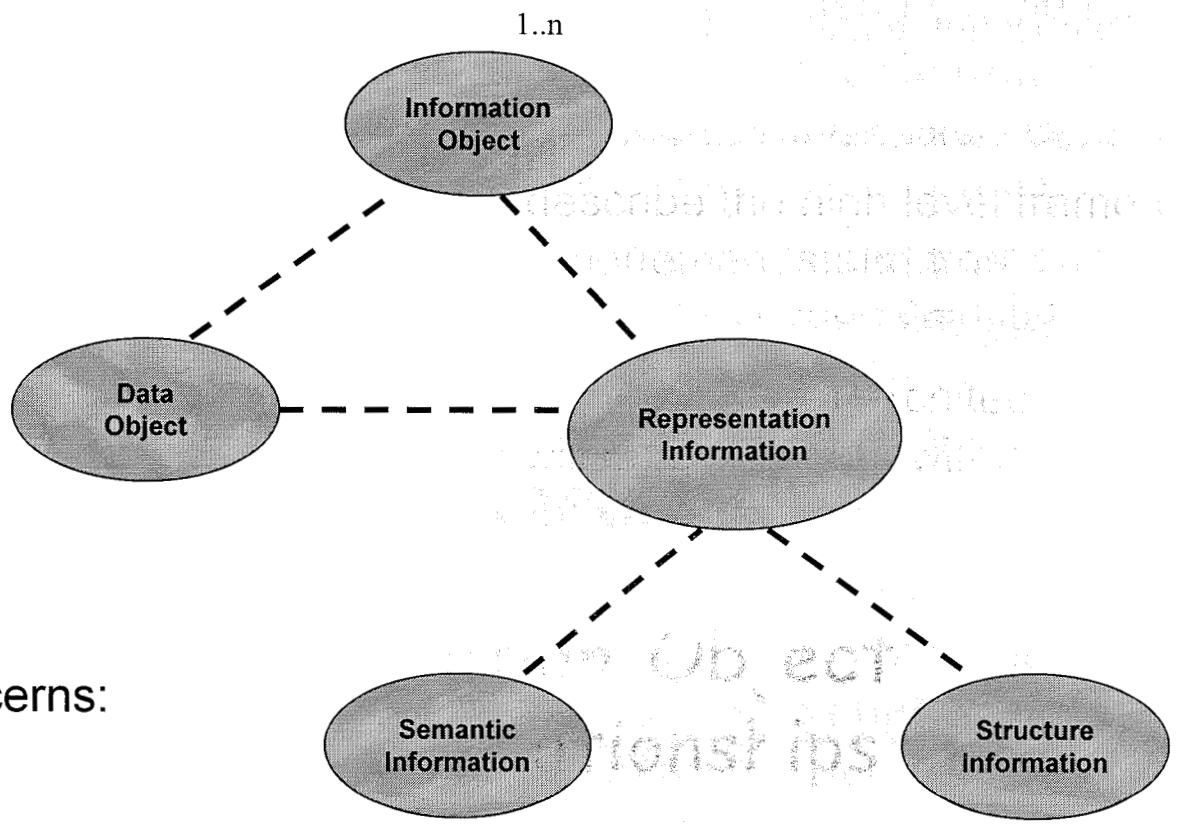
Background

- Information Architecture is one viewpoint of an overall system architecture, with two main elements …
  - Passive objects that include data description, structuring, and packaging elements (i.e. data architecture)
  - Active objects which include functional elements, with defined interfaces and protocols that manipulate information objects (i.e. functional architecture)
  - This presentation distinguishes these two sets of disparate elements
- The described set of functional elements are suitable for building a wide variety of flight and ground, operational and scientific systems
- A mapping of key elements & terms to the OAIS reference Model is provided
- Maps to CCSDS information management standards

Background, contd

- This general approach has been validated by actual use within both operational and science contexts, using two different implementation approaches
- The Information Architecture will describe the high level framework for connecting CCSDS information management standards
- The work to complete this reference Information Architecture and define all the required interfaces will be carried out within the CCSDS System Engineering Area, in the Information Architecture BOF
- The CCSDS Mission Operations and Information Management Area will be responsible for specification of standards and interfaces described by the high level information architecture

Information Object Basic Relationships

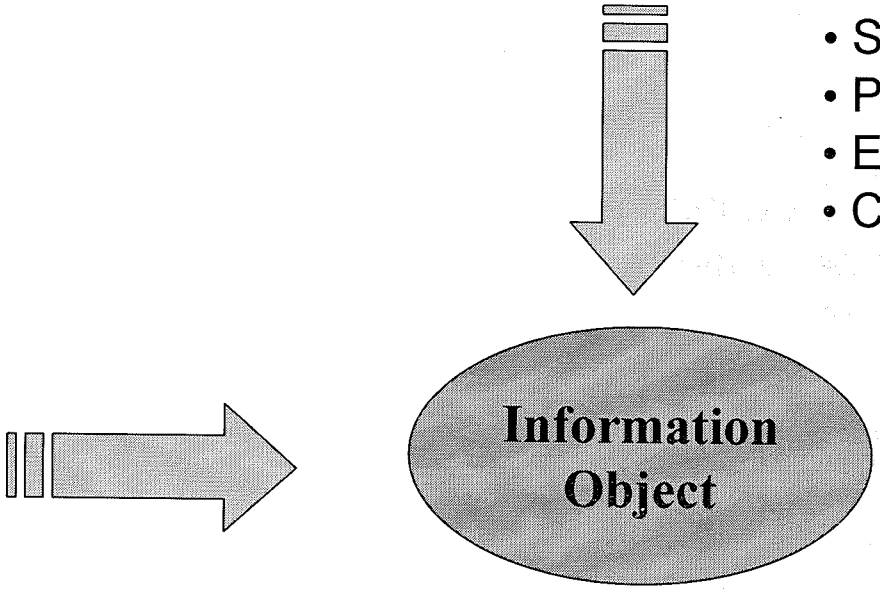


Information Concerns:

- Structure
- Semantics
- Relationships
- Permanence
- Rules

Information Object

Management Interface
- Rules
- Schema
- Permanence
- Element Types
- Constraints



Example: Viking Image Product

Input Data:
- Byte Streams
- Bit Streams

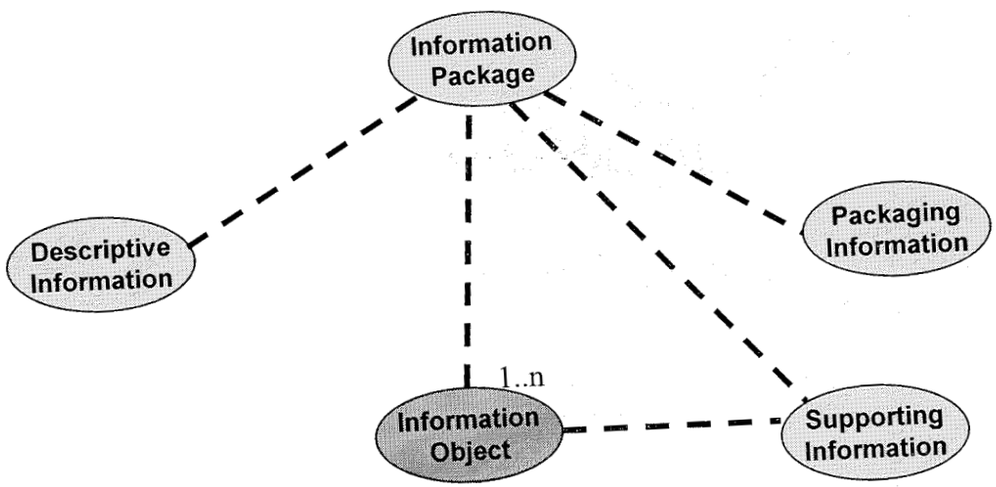
Core Method Capabilities
- Data
- Structure
- Semantics
- Relationships
- Type Conversion
- Constraint Checking
- get Data

Output Data
- Data values
- Metadata values
- Fault indications (semantic, type & constraint violations)

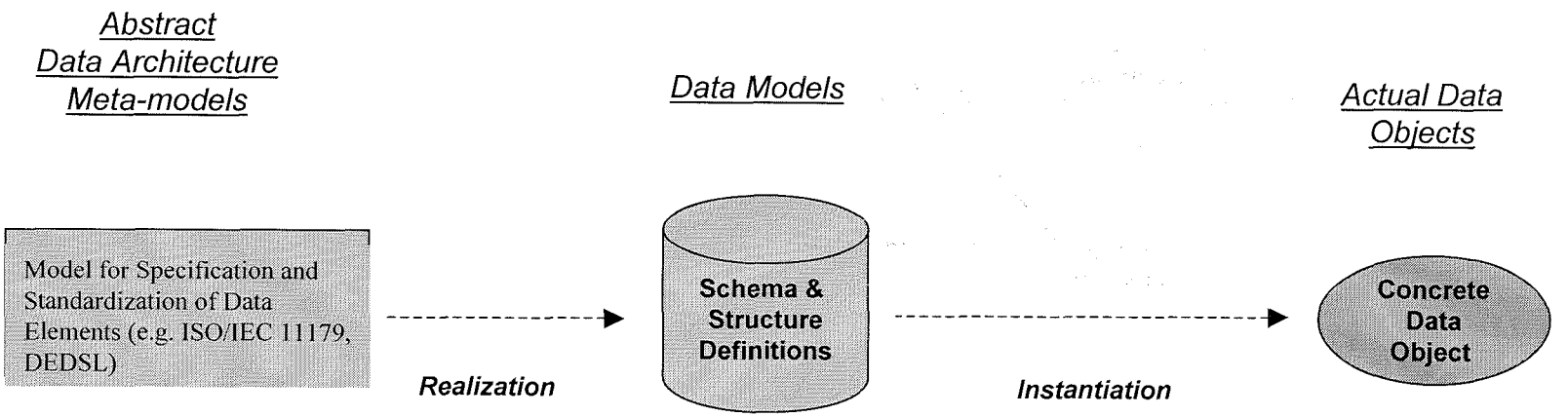
Concerns Associated with Passive Information Objects

- Metadata
  - Data description
    - Data elements & types
  - Structure
    - Arrangements of data elements
  - Semantics
    - Meaning of data
  - Relationships
    - Relationships among data elements
  - Security
    - User credentials, user permissions
- Packaging
  - Assemblies formed of one or more data structures
- Instantiation
  - N.B. All of these passive information objects must be instantiated on one or more real functional elements in order to participate in system activities

Information Object Extended Relationships



Relationships Among Abstract & Data Models



Instantiation of Meta-models as Data Models may be manual or automated

Data Models may be stored in a database, file, or as executable specifications within a program

Concrete Data Objects are the actual data as manipulated, used, exchanged or stored by programs

Information Object Specifications

- Information Object (which is a data object + a metadata object)
  - Data Object (the "bits")
  - Metadata Object (i.e. representation information)
    - Data Dictionary object (which is really just an information object with a data dictionary result)
  - Behavior
    - User Credentials (who I am)
    - User Permissions (what I can do)
- Examples
  - Archive Information Package (OAIS verbiage)
    - Object Reference (pointer to Object)
  - PDS Viking Image + Metadata
  - Submission & Dissemination Information Package (OAIS SIP, DIP)

Active Information Management Objects

- Active information management objects are functional elements which manipulate information objects
  - Used to build various information infrastructures
  - Support project, mission, and agency information management needs
  - Includes primitive objects and more complex assemblages

- Primitive Information Management Objects
  - Data Store Object (i.e. get/put/add/delete)
  - Query Object (i.e. find)

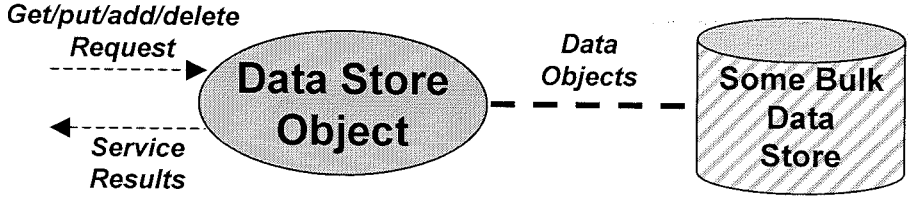
- Information Management Infrastructure Objects
  - Registry Service Objects
  - Repository Service Objects (archive, store, product service, …)
  - Location Service Objects
  - Domain-specific Objects

Primitive Information Management Objects

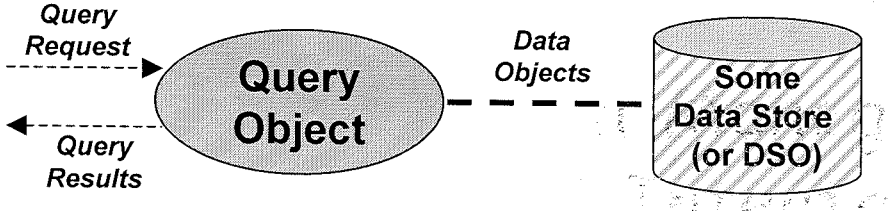
- **Data Store Object**
  - Store and provide access to data objects via a simple get / put / add / delete interface
  - DSO may provide short term or long term persistent storage of data objects
  - Different classes of DSO may be implemented by file system, SSR, DBMS, bulk storage server, computer memory or SAN

- **Query Object**
  - Accept queries via a simple query interface and return corresponding data elements
  - Returned data elements may be data about data, data about services, data about systems
  - Different classes of QO may be implemented by DBMS, simple file access, or table look-up

Primitive Information Management Object Functional Interfaces

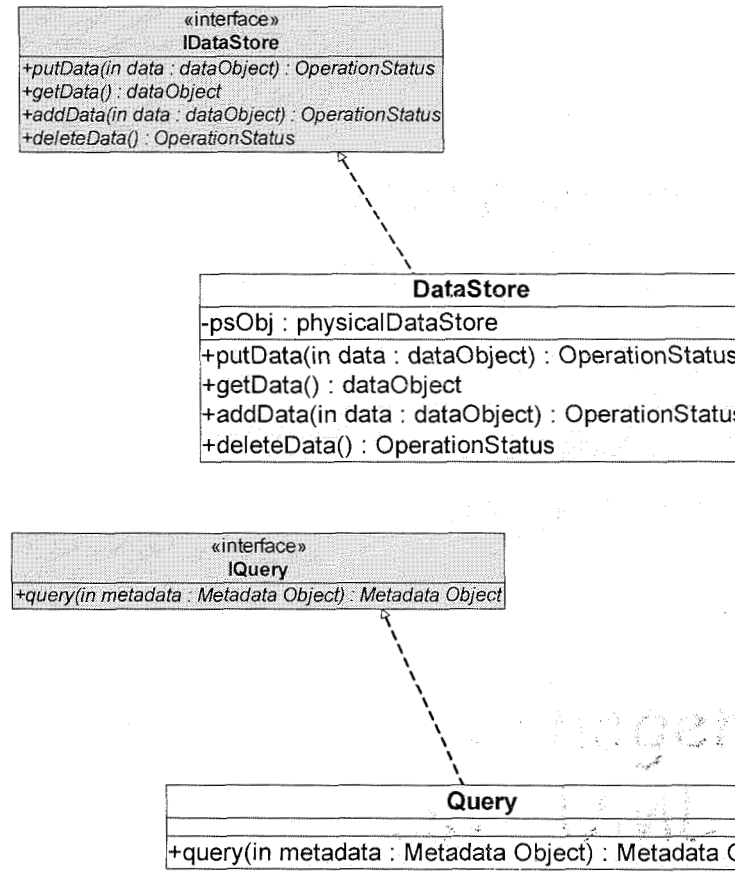


Interfaces:
- **Get / put**
- **Add / delete**

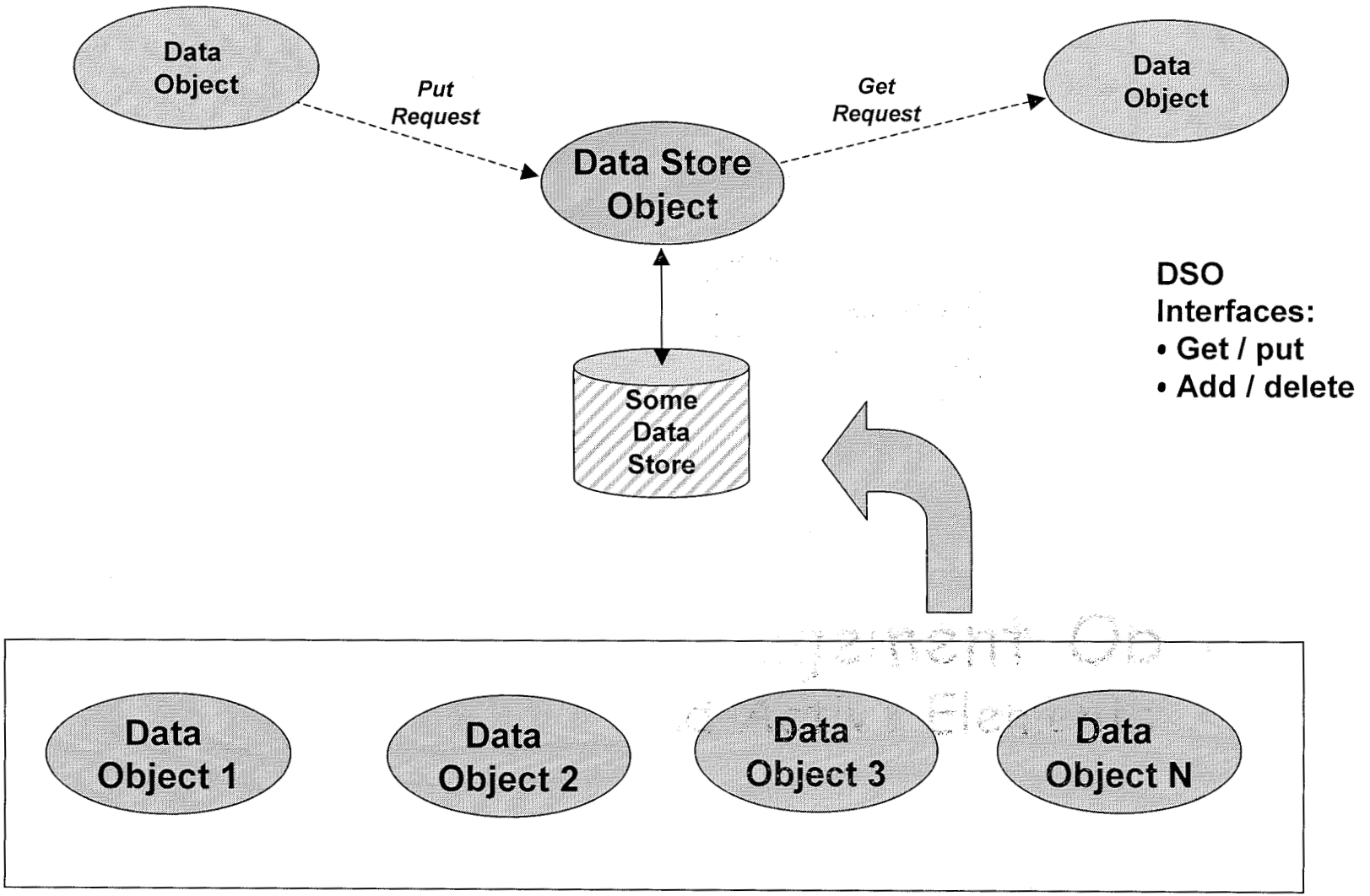


Interfaces:
- **Query**

Primitive Information Management Object Functional Interfaces: UML Class View



Primitive Data Management Object
Relate Passive and Active Elements

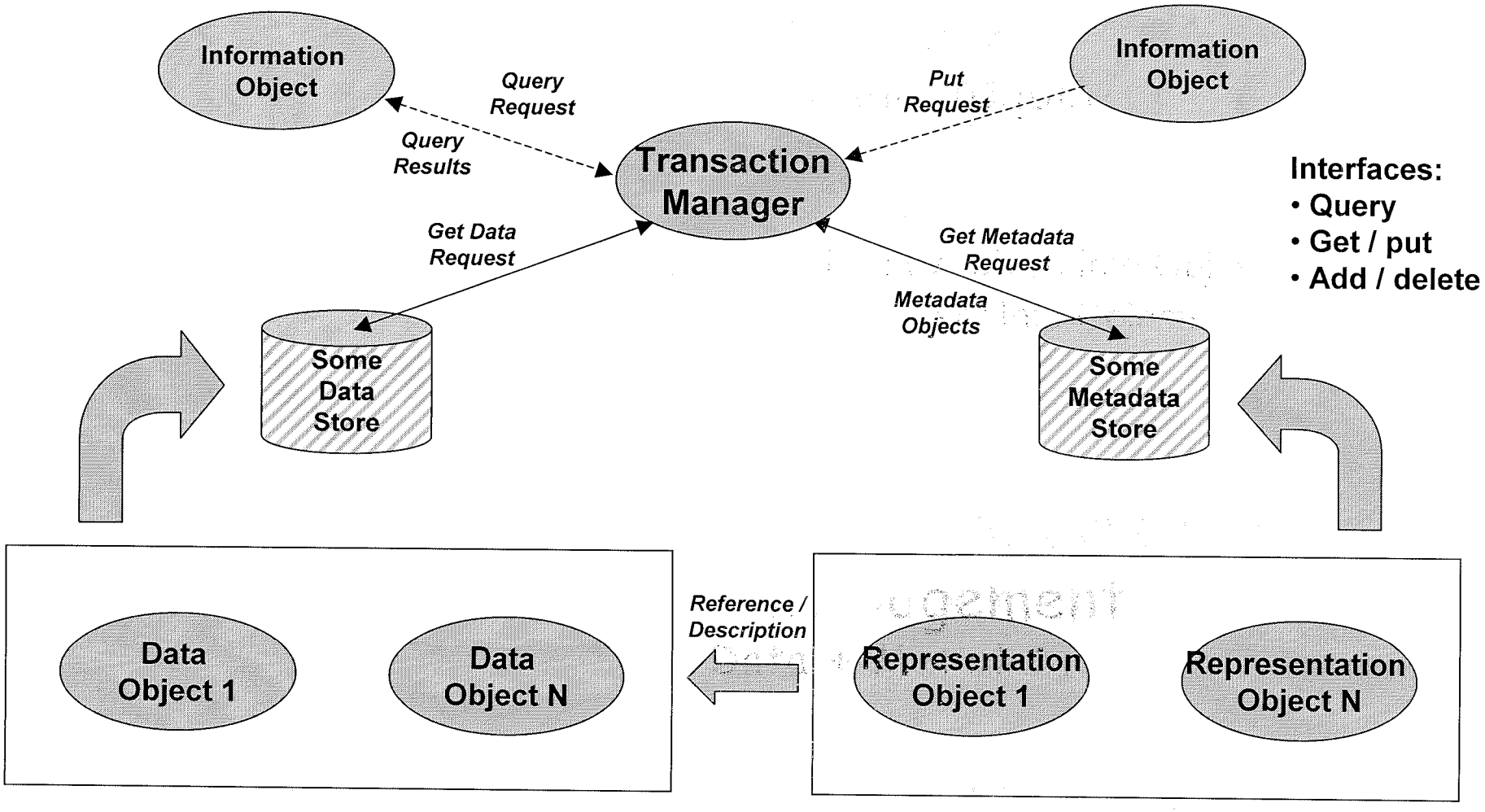


Primitive Information Management Objects: Transaction Manager Object

- Manage transactions across data and metadata sources that have constraints such as synchronization, failure points that require roll-backs, etc.
  - Interfaces
    - Get requests (implies we already know which Info Obj we want)
    - Quer y requests (query for a set {0…n} of Info Objects based on representational data)
    - Delete R equests
  - Packaging
    - *May* handle packaging of data object(s) with representational object(s)

Primitive Information Management Object

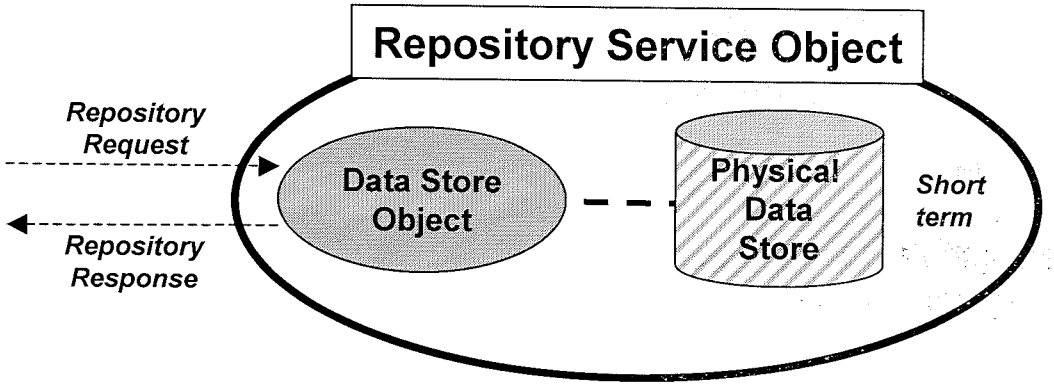
Information = Data + Metadata



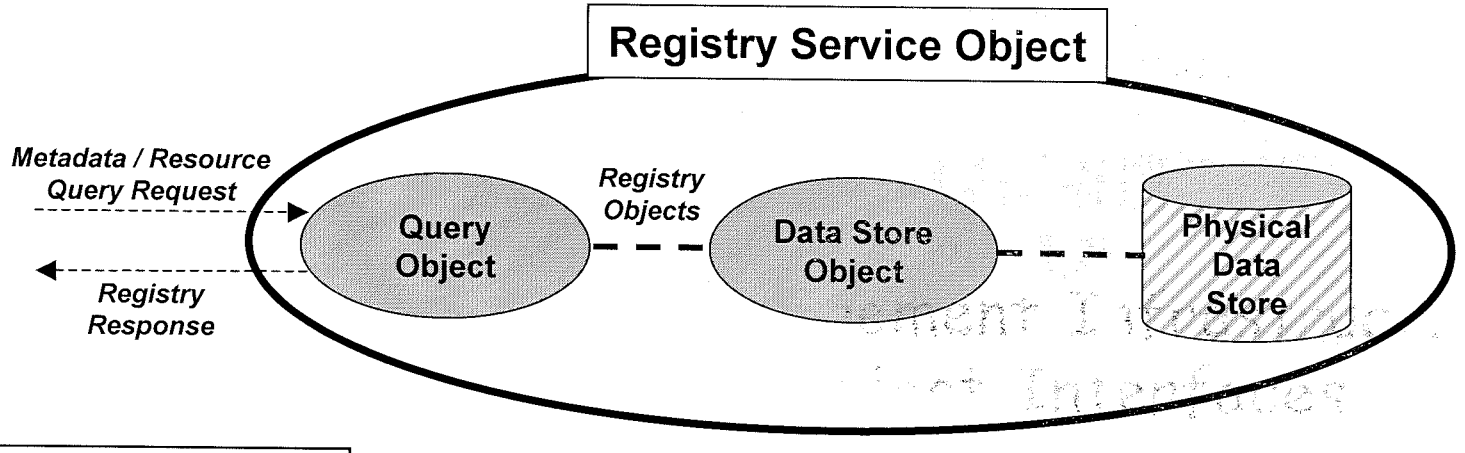
Information Management Infrastructure Objects

- **Repository Service Objects**
  - Manages data objects. Objects include
    - Repository Service Object (short term simple "puts" and "gets" into the repository)
    - Archive Service Objects manage "ingest" of products (long term, permanence)
    - Product Service Object manages domain/content "gets" which include
      - Packaging and delivery mechanisms for data objects (possibly physical)
      - Extraction and request processing, optional call-outs to …
  - Example data objects include: Products, documents, software, data dictionaries, schemas
- **Registry Service Objects**
  - Manages representational information associated with data objects
  - Example registries include: metadata registries, product catalogs, resource registries, name services, etc
- **Query Service Object**
  - Search individual and federated data sources
  - Use standard data query and special data fusion functions
  - Use various Registry and Repository Service objects as needed
- **Ingest Service Object**
  - Receives Info objects, manages which repositories they should be routed to, and communicates with Repository Service Objects through Repository Interface,
    - Repository service takes representational data about the incoming information object to issue one or more "puts" into its data source object which captures the info object.
- **Optional Domain Specific Service Objects**
  - Domain specific data correlation and fusion functions
  - Domain specific transform and mapping functions

Information Management Infrastructure Functional Object Interfaces



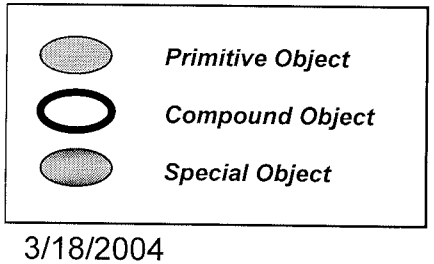
Physical Data Store
- *Disk / files*
- *Memory*
- *SSR*



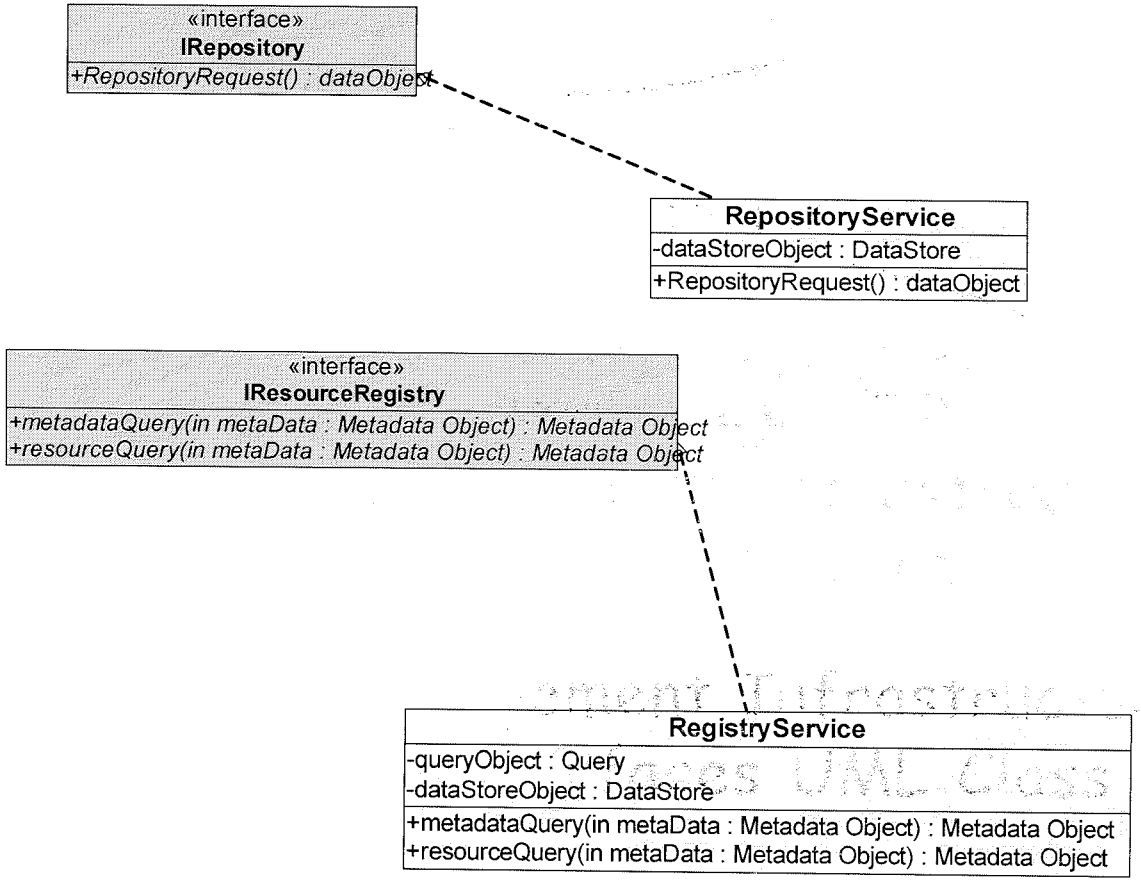
Metadata Objects
- *Descriptors*
- *Schema*
- *Models*

Resource Objects
- *Descriptors*
- *Definitions*
- *Locators*

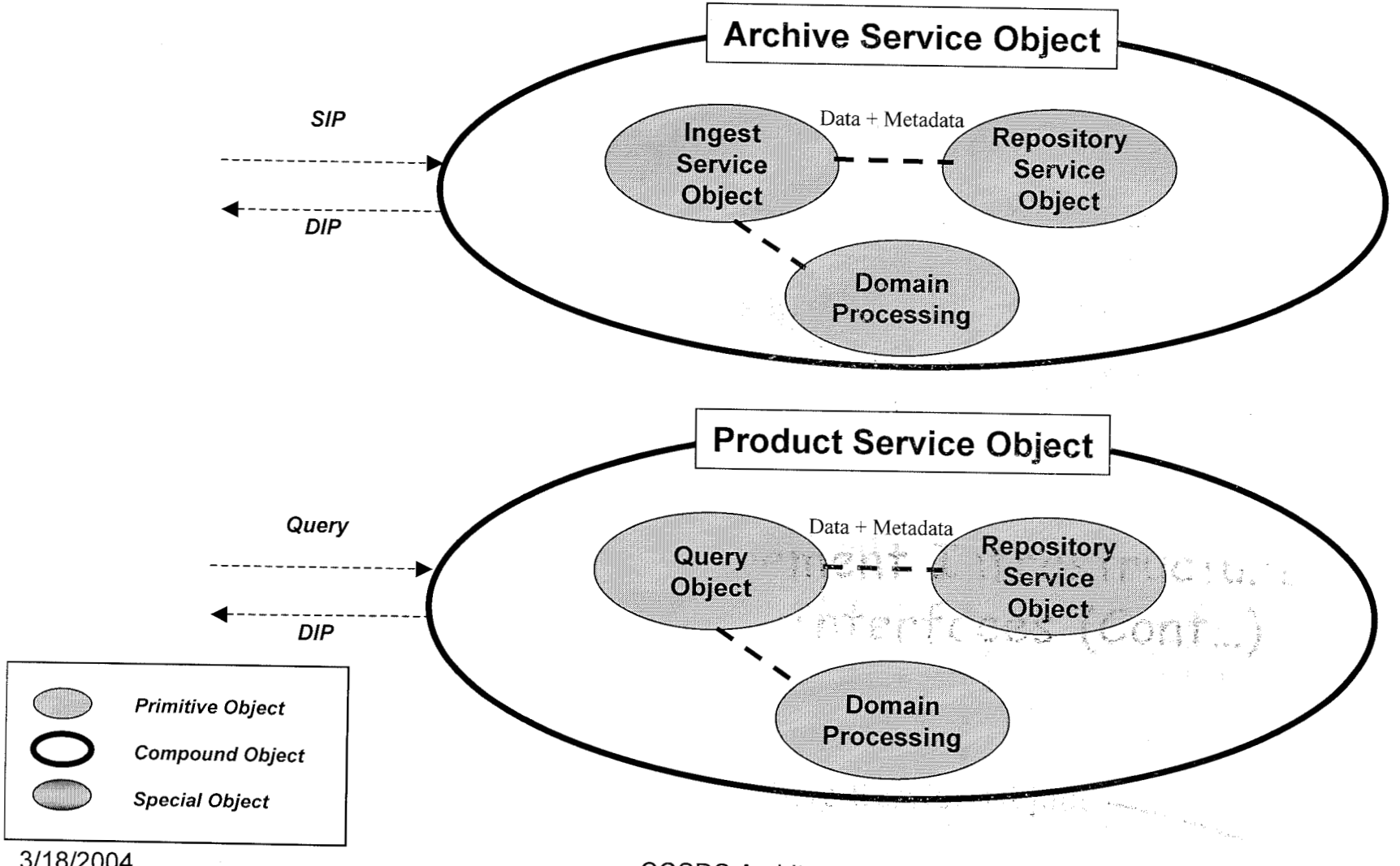
Name Svc Objects
- *Descriptors*
- *Names*
- *Addresses*
- *Attributes*



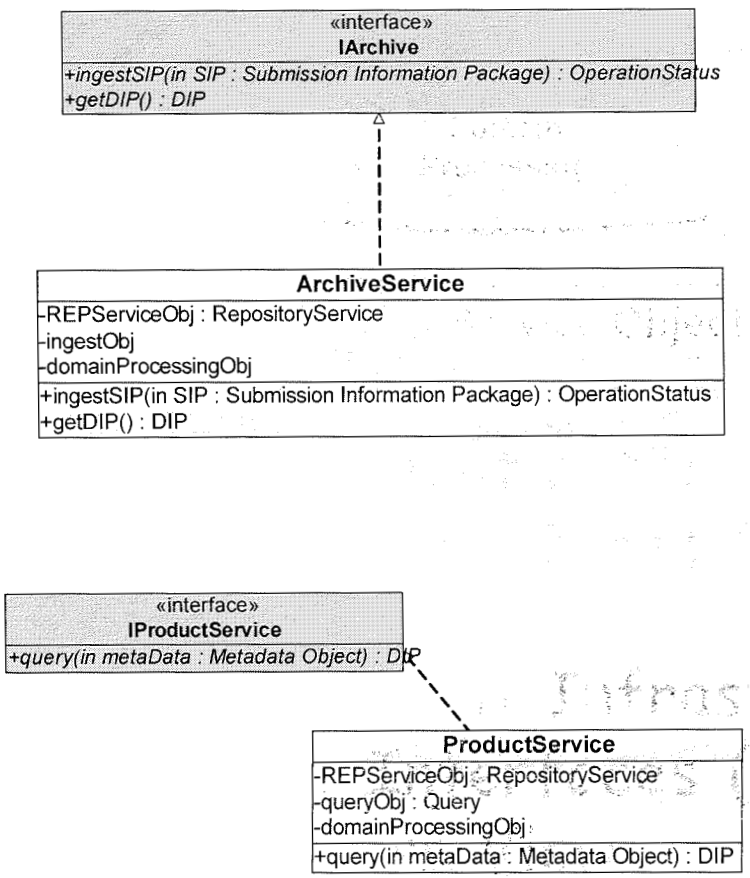
Information Management Infrastructure Functional Object Interfaces UML Class View



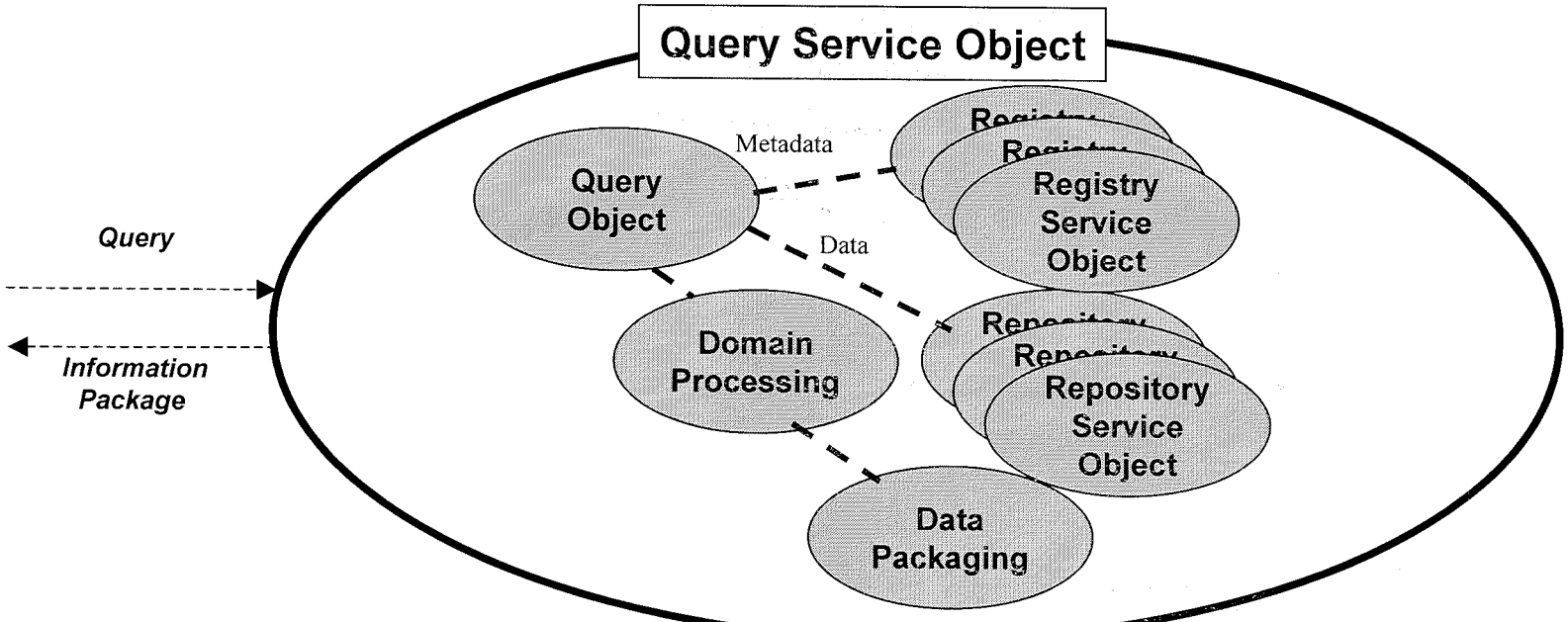
Information Management Infrastructure Functional Object Interfaces (Cont…)

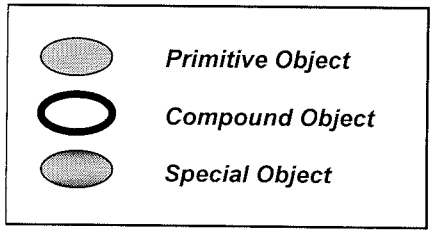


Information Management Infrastructure Functional Object Interfaces (Cont…) UML Class View

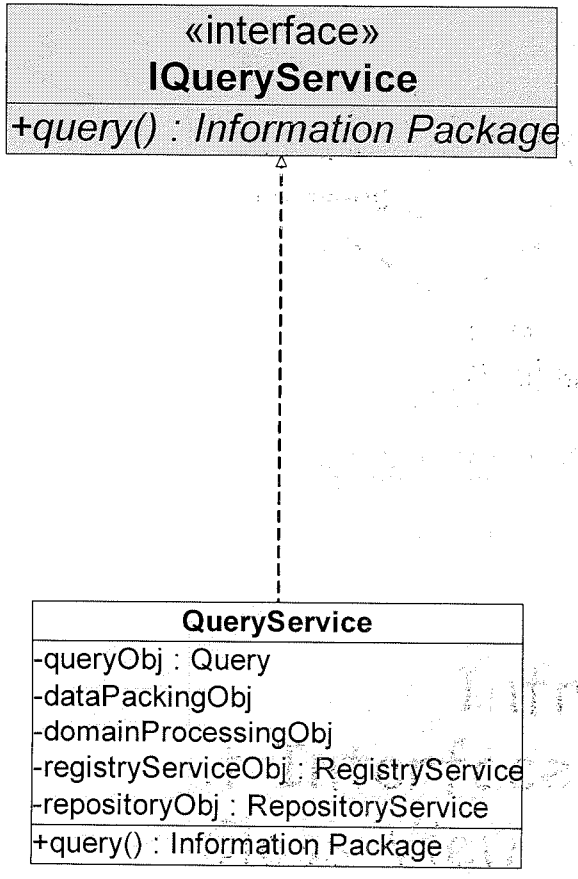


Information Management Infrastructure Functional Object Interfaces (Cont…)

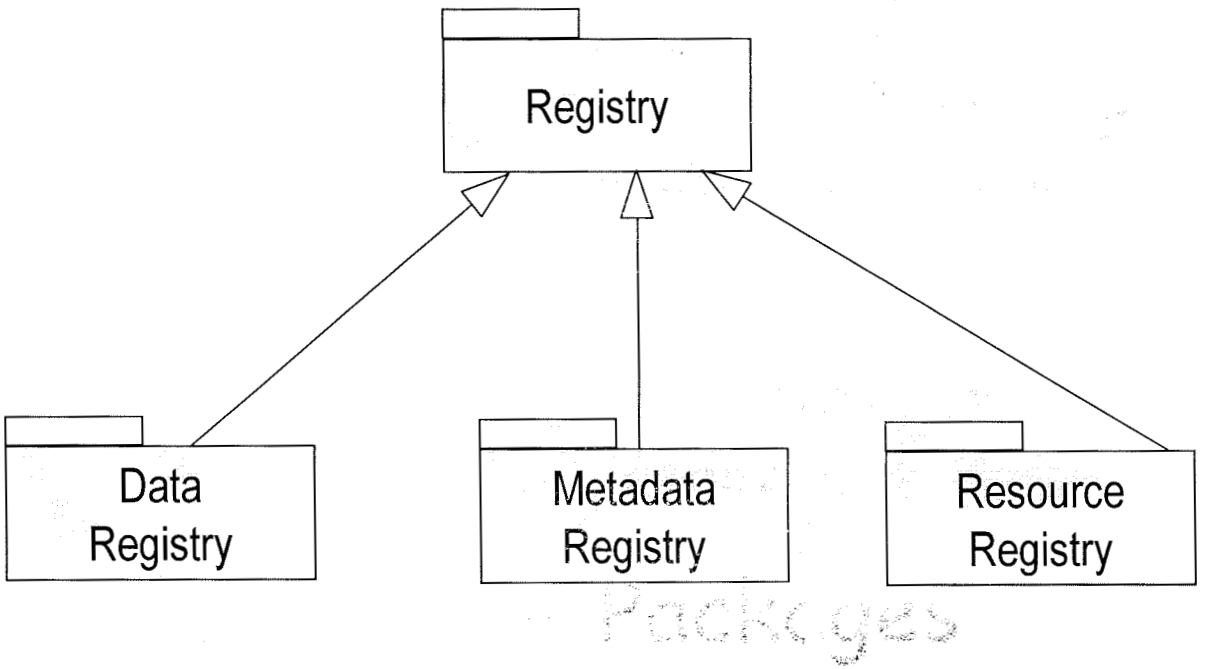




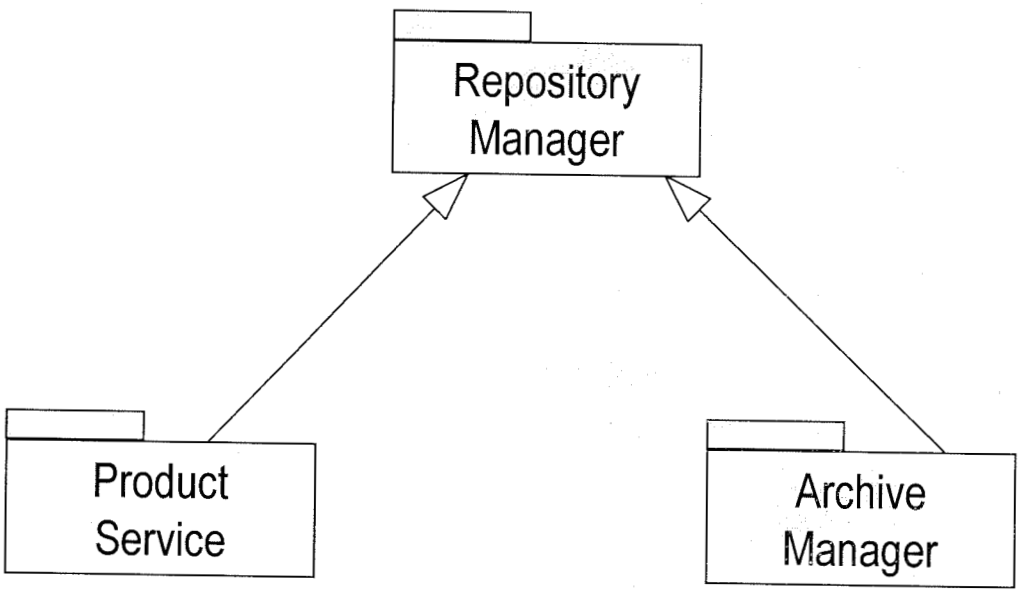
Information Management Infrastructure Functional Object Interfaces (Cont…) UML Class View



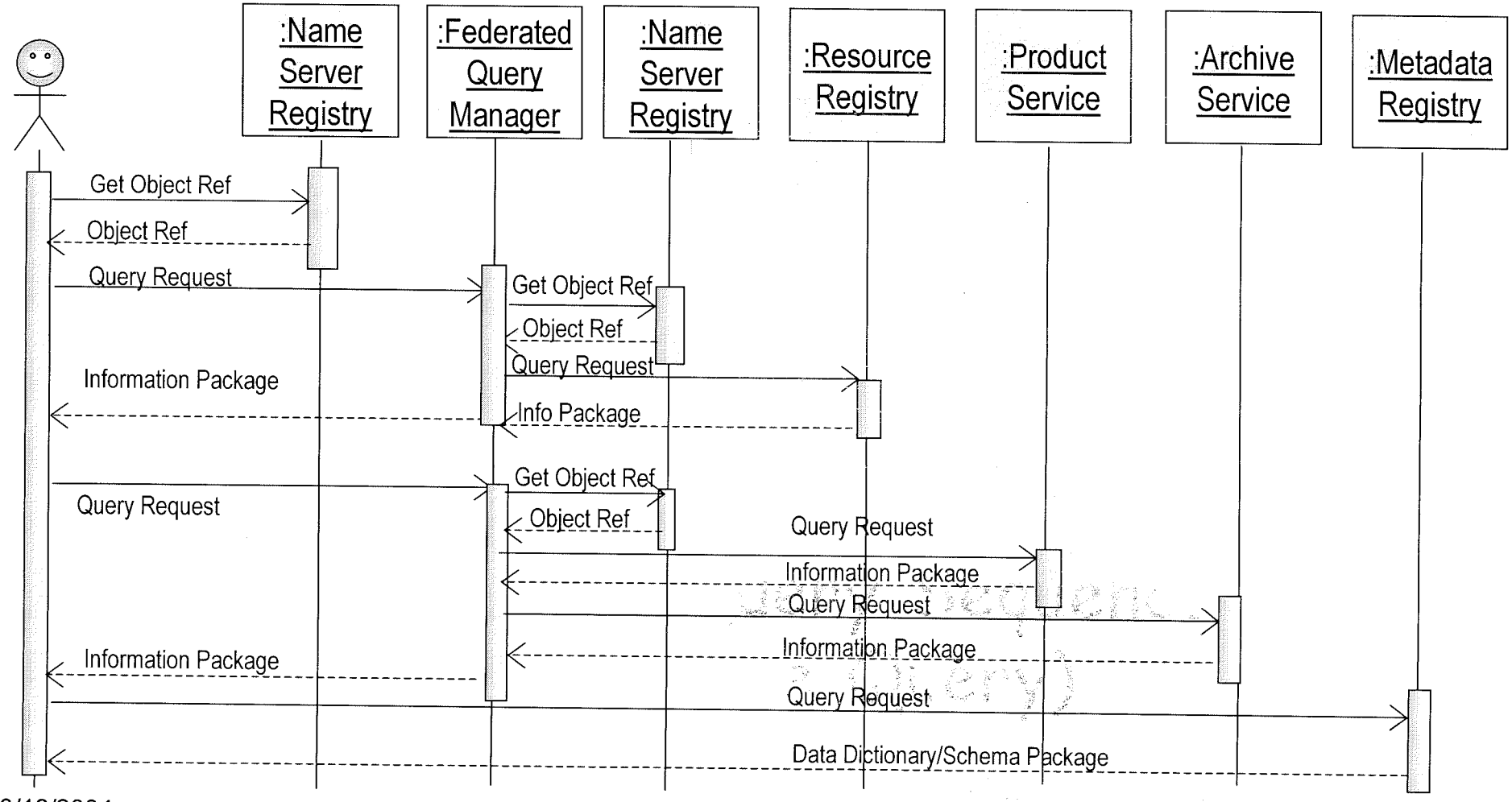
Registry Packages



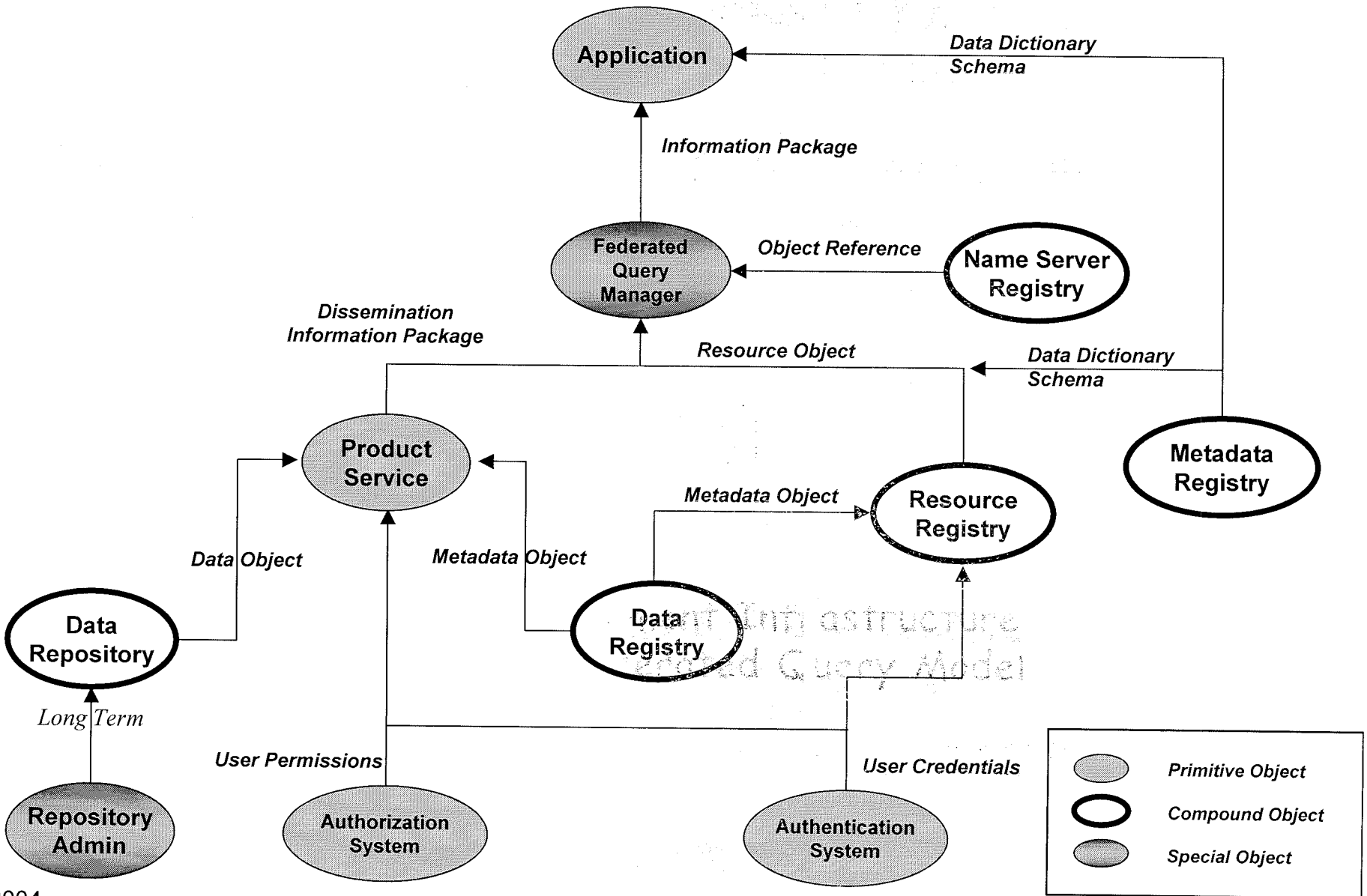
Repository Packages



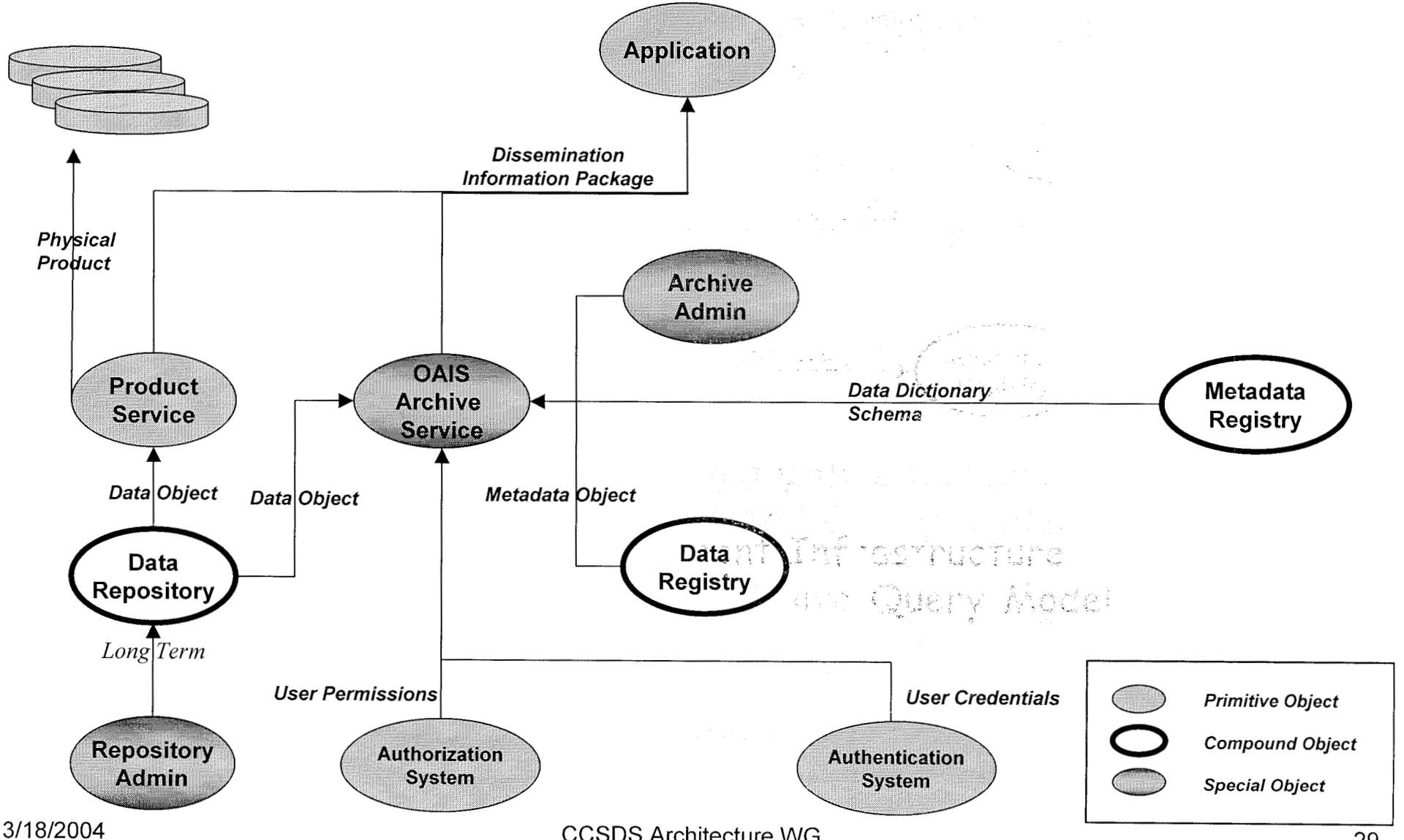
Federated Query Sequence (Two Phase Query)



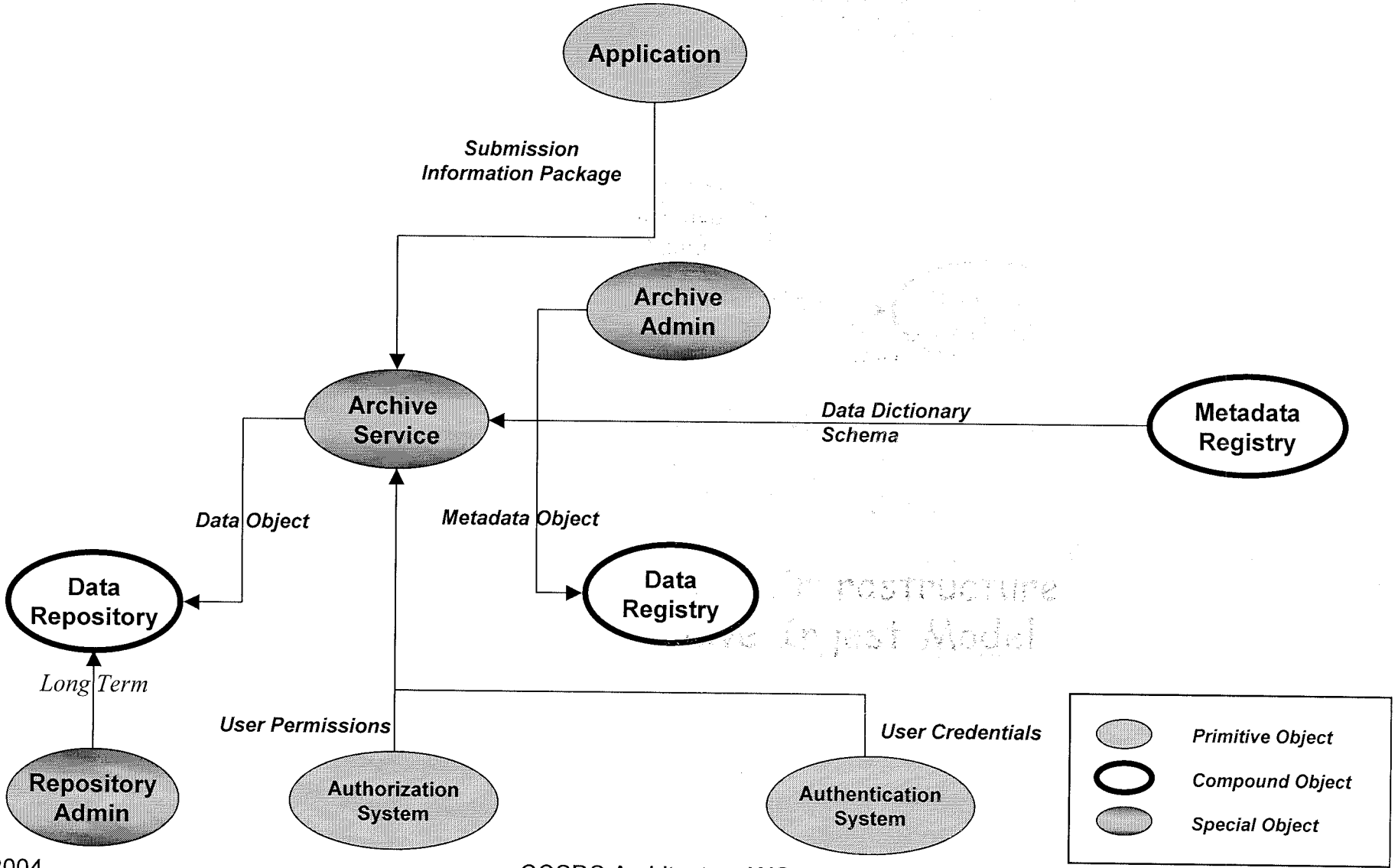
Information Management Infrastructure Example: Simple Federated Query Model



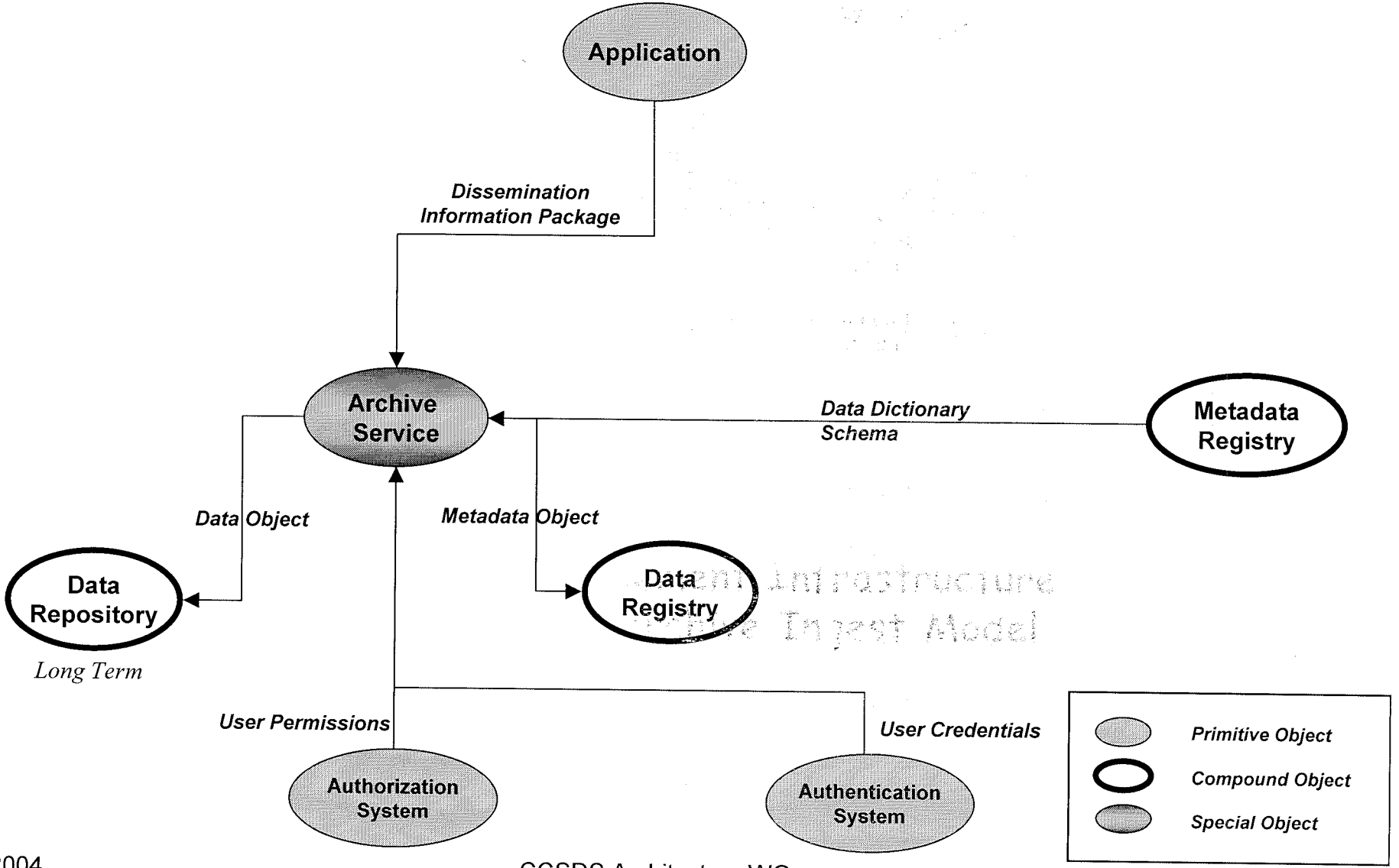
Information Management Infrastructure
Example: CCSDS OAIS Archive Query Model



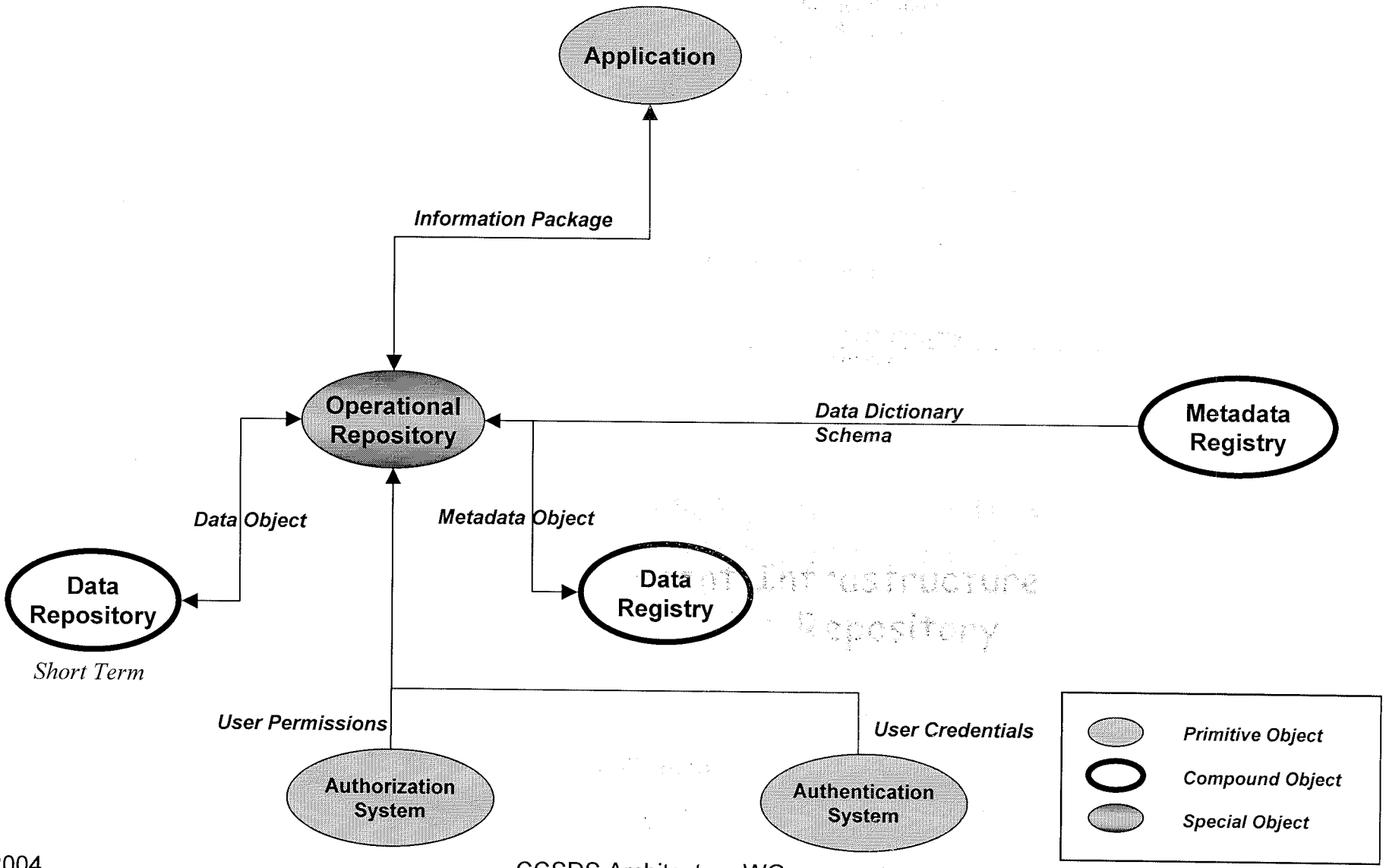
Information Management Infrastructure Example: OAIS Archive Ingest Model



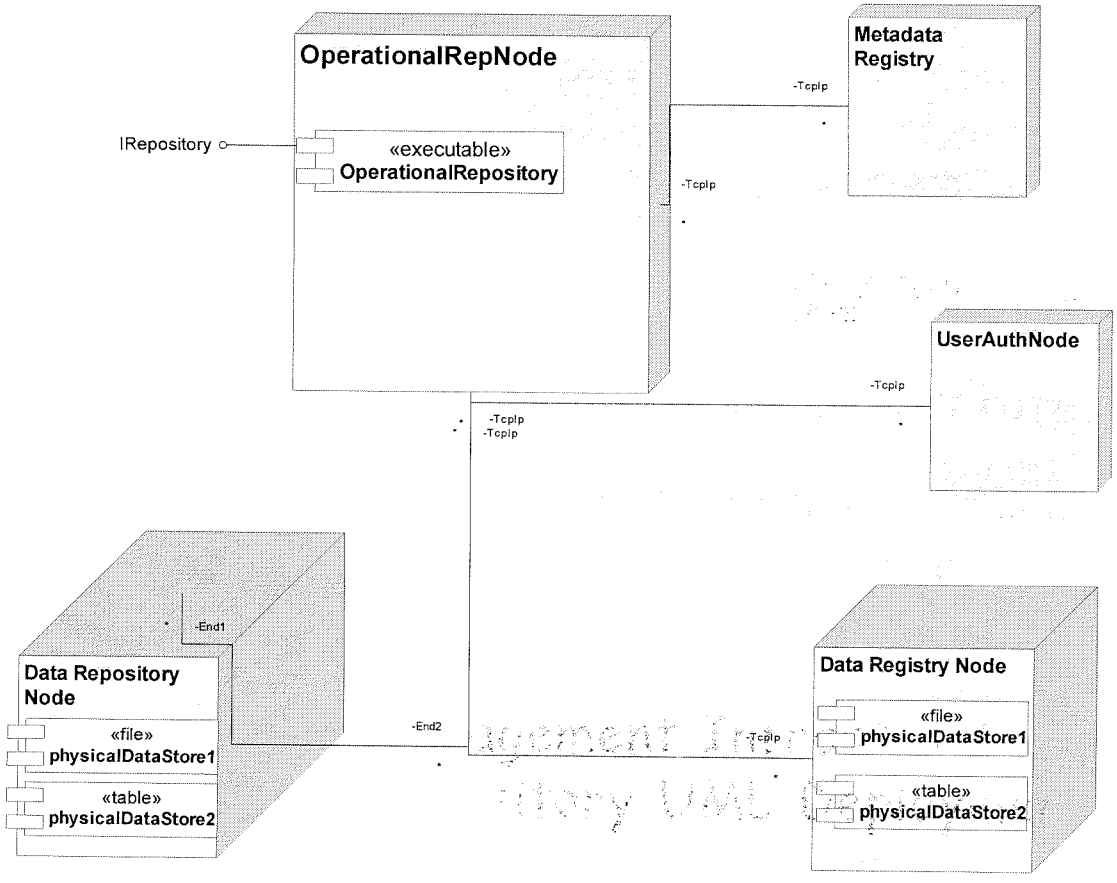
Information Management Infrastructure Example: Simple Archive Ingest Model



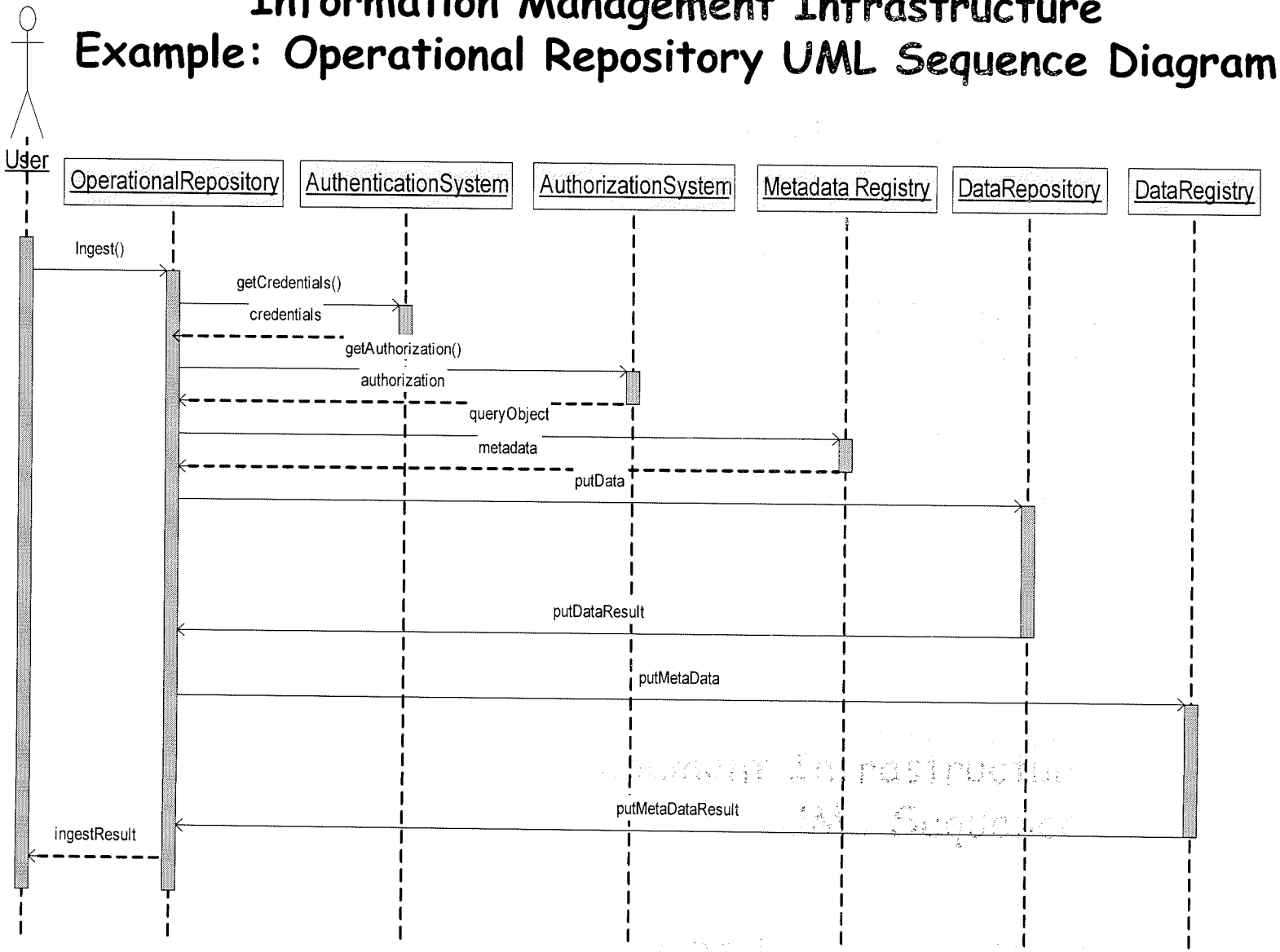
Information Management Infrastructure Example: Simple Get/Put Repository



Information Management Infrastructure
Example: Operational Repository UML Deployment Diagram



Information Management Infrastructure
Example: Operational Repository UML Sequence Diagram



Information Management Active Objects In Summary

- Are Functional Objects, but …
  - Perform basic Information Management Services
  - Provide services to many different Functional Objects and to other Information Management Objects
  - May be implemented using simple files and tables or using various commercial interface or DBMS solutions
  - May be assembled to create higher order Information Services (operational repositories, registries, archives, …)
  - Some Nodes may consist of only Information Management Objects, some buffers, and related Service Management functions
  - Local implementations may use any suitable technology within an operational domain
  - Interfaces that are exposed for cross support and interoperability must be standardized

Information Management Passive Objects In Summary

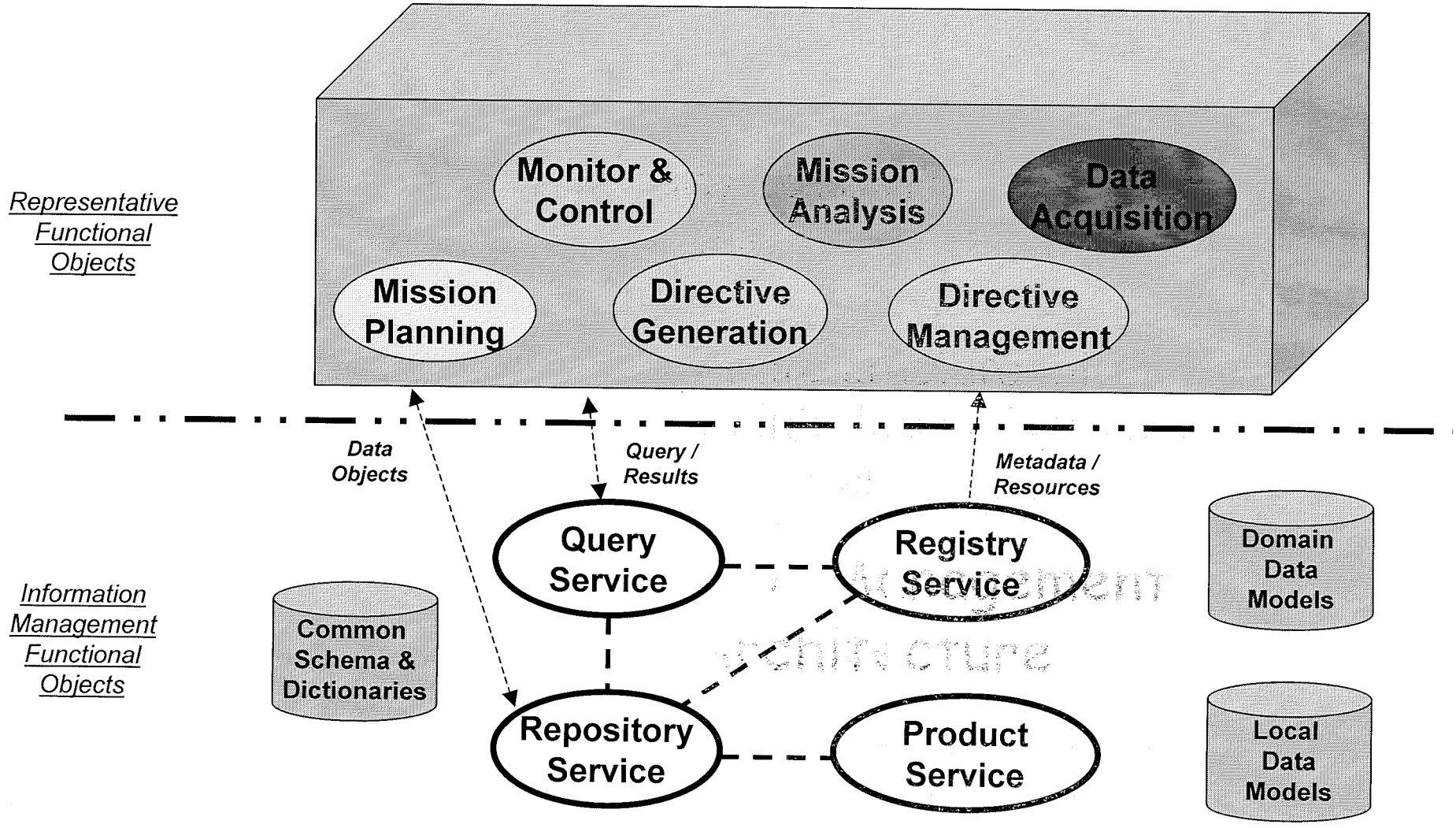
- Submission Information Package
- Dissemination Information Package
- Archive Information Package
- Data Objects
- Resource Objects
  - Data about data
  - Data about schema
  - Data about resources
- Information Objects
- Query Objects

Information View
Service Interfaces

- Information Object Specifications (passive)
  - Data description, structure, packaging, semantics relationships
  - Policies, permanence, rules, constraints
  - Schema / DTD specifications

- Information Management Infrastructure Interfaces (active and require service interfaces & protocols)
  - Repository Services (Get, Put, and Ingest)
  - Registry Services (Query, Get, Put)
  - Query Services (Query)
  - Domain Services (Abstract Interfaces)

Space Information Management Functional Architecture

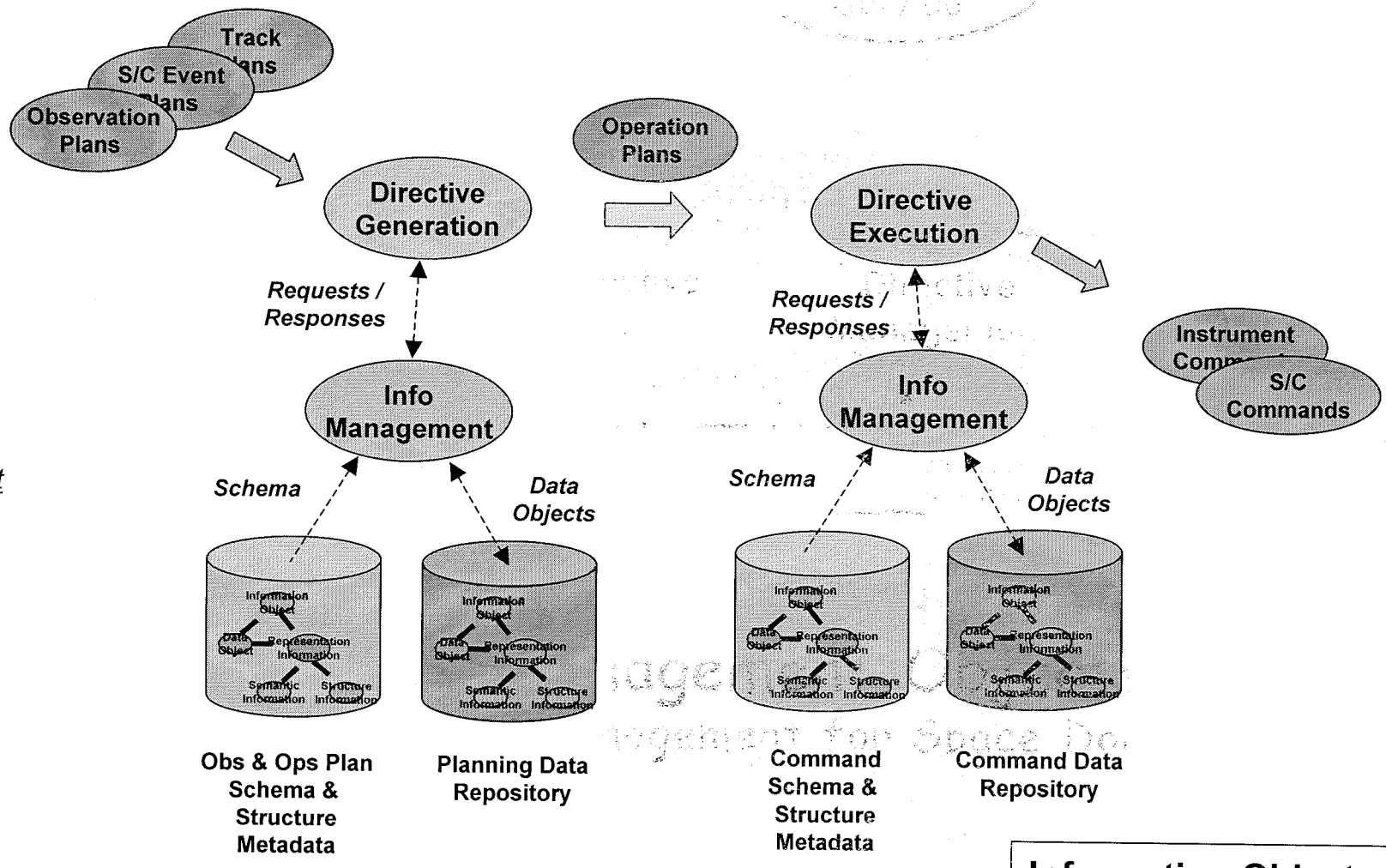


Information Management Objects

Example: Information Management for Space Domain

Example Functional Objects

Information Management Service Objects



Schema & structure may be embedded in the function itself

Actual data objects that are exchanged between Applications at run time

Core Schema
Common Schema
Extension Schema

Information Objects are exchanged by Functional Objects

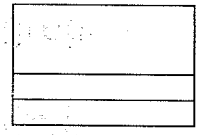
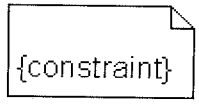
Information View Definitions

- The Information View is a view on a space data system that focuses on the information used by that system. This includes structural(syntactic) and semantic views of the information, the relationships among information elements and rules for their management and transformation.
- Structure - The information that imparts meaning about how other information is organized. For example, it maps bit streams to common computer types such as characters, numbers, and pixels and aggregations of those types such as character strings and arrays.
- Semantics -rules by which syntactic expressions are assigned meanings
- Relationships
- Data element
- Information Package: The primary Information Object and associated Supporting Information that is needed to aid in the use or the preservation of the primary Information Object. The Information Package has associated Packaging Information used to delimit and identify the primary Information Object and Supporting Information. It may also have associated Descriptive Information used to aid in finding the primary Information Object.
- Information Management Objects: The specific primitive Functional Objects used to manage, locate, store, describe, and distribute data objects within a system

Information View Definitions, contd

- Information Object: A Data Object together with its Representation Information.
- Data Object: Either a Physical Object or a Digital Object.
- Representation Information: The information that maps a Data Object into more meaningful concepts. An simple example is the ASCII definition that describes how a sequence of bits (i.e., a Data Object) is mapped into a symbol.
- Structure Information: The information that imparts meaning about how other information is organized. For example, it maps bit streams to common computer types such as characters, numbers, and pixels and aggregations of those types such as character strings and arrays
- Semantic Information: The information that provides additional meaning, beyond that provided by the Structure Information, to a Data Object.
- Descriptive Information: The set of information, consisting primarily of Package Descriptions, which is provided to support the finding, ordering, and retrieving of an Information Object by users or clients.
- Packaging Information: The information that is used to bind and identify the components of an Information Package. For example, it may be the ISO 9660 volume and directory information used on a CD-ROM to provide the content of several files containing an Information Object and Supporting Information.

Information Concepts and Notations (UML)

Construct	Description	Syntax
class	a description of a set of objects that share the same attributes, operations, methods, relationships and semantics.	
association	a relationship between two or more classifiers that involves connections among their instances.	
aggregation	A special form of association that specifies a whole-part relationship between the aggregate (whole) and the component part.	
generalization	a taxonomic relationship between a more general and a more specific element.	
constraint	a semantic condition or restriction	

Top Level Object & Enterprise Model