The Engineering of Large-Scale Data-Intensive Information Systems

by

Larry Kerschberg

Center for Information Systems Integration and Evolution
Department of Information and Software Systems Engineering

School of Information Technology and Engineering
George Mason University

INFT 864/CSI 810 — Scientific Databases Course

Acknowledgments

- **Daniel Menascé, Hassan Gomaa, Jongpil Yoon, CISIE**
- **GMU-led Team for the Independent Architecture Study sponsored by Hughes Applied Information Systems**
  - Dr. Menas Kafatos, PI, Institute for Computational Sciences and Informatics, GMU
  - Drs. Ferris Webster and Jim Churgin, University of Delaware
  - Dr. James Kinter, Center for Ocean-Land-Atmospheric Studies (COLA)
  - Dr. Berrian Moore, University of New Hampshire
  - Mr. Hank Wolf, Project Administrator
- **GMU-led Team had 27 members: faculty, research associates, graduate students, and support staff**
- **Three Independent Architecture Studies**
  - GMU Team — GMU, UDel, UNH, COLA
  - UC—Berkeley, Sequoia 2000 team with Mike Stonebraker as PI.
  - University of North Dakota and U. of Nebraska

Outline of Presentation

- **Motivation: NASA's Mission to Planet Earth.**
- **Design Principles for Large-Scale *Distributed, Heterogeneous* Data-Intensive Information Systems (LSS).**
- **GMU Performance-Oriented Design Methodology for LSS.**
- **GMU Logical Node Architecture.**
- **Domain Model for EOSDIS Core System.**
- **Data and Information Architecture for EOSDIS.**
- **User Scenarios and the GMU Federated Client Server Architecture.**
- **Conclusions.**

Earth Observing System (EOS)
NASA's Mission to Planet Earth

- **<u>Series of satellites with instruments</u> to be launched in 1997 and 2002 which will monitor the Earth and its processes, and send data to earth stations.**
- **<u>Data-intensive</u> system which ingests more than *a terabyte of data per day*.**
- **<u>Resource-intensive</u> system with high performance computing requirements.**
- **<u>Highly distributed</u> with 8 Distributed Active Archive Centers, Science Computing Facilities, and Affiliated Data Centers.**
- **<u>Heterogeneous Data Sources</u> (e.g., NOAA, International Partners)**
- **<u>Multiple data types and formats</u>: Raw Radiances, Derived Products, Metadata, Images, in-situ measurements, Algorithms, Documents, etc.**
- **<u>Diverse User Community</u>: NASA 500 PIs, K-12 students, GCDIS, Policy-makers (EPA, DoD) and commercial interests (e.g., World Bank, Timber Industry, Oil Industry.)**

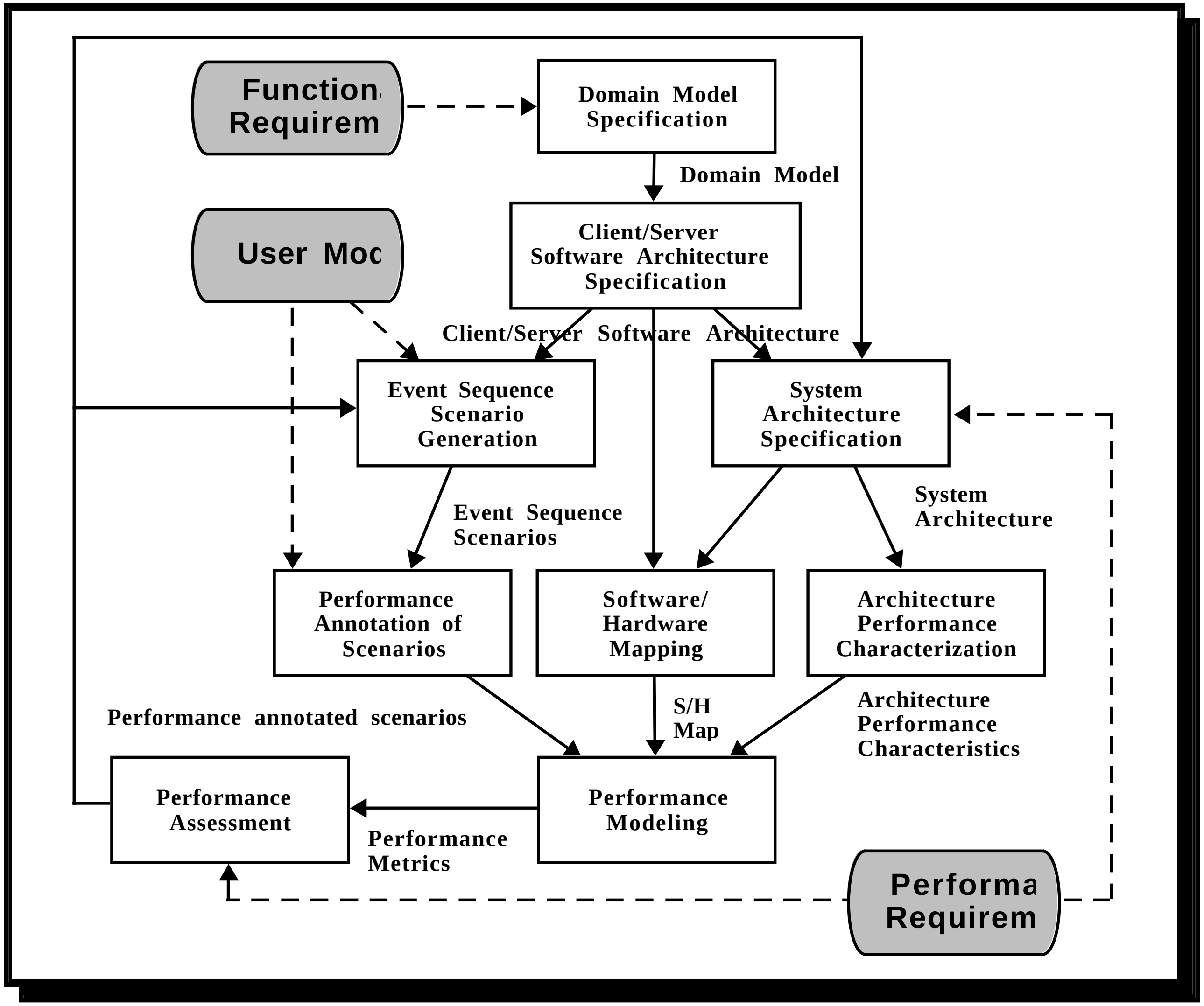
Design Principles for Large-Scale Distributed Data-Intensive Information System (LSS)

- **Location Transparency:** Users should be able to access any information object, e.g., data, metadata, browse images, images, without having to know its location in the system**.**
- **Modularity:** Nodes of the LSS should be configured for specific data processing and storage requirements based on a generic node architecture.
- **Minimization of User Connections:** Large user population implies that users should be able to perform many functions locally, thereby minimizing connections to LSS.
- **External Processing Capabilities:** LSS should be capable of scheduling and using spare cycles on networks of user workstations to augment its resource base.

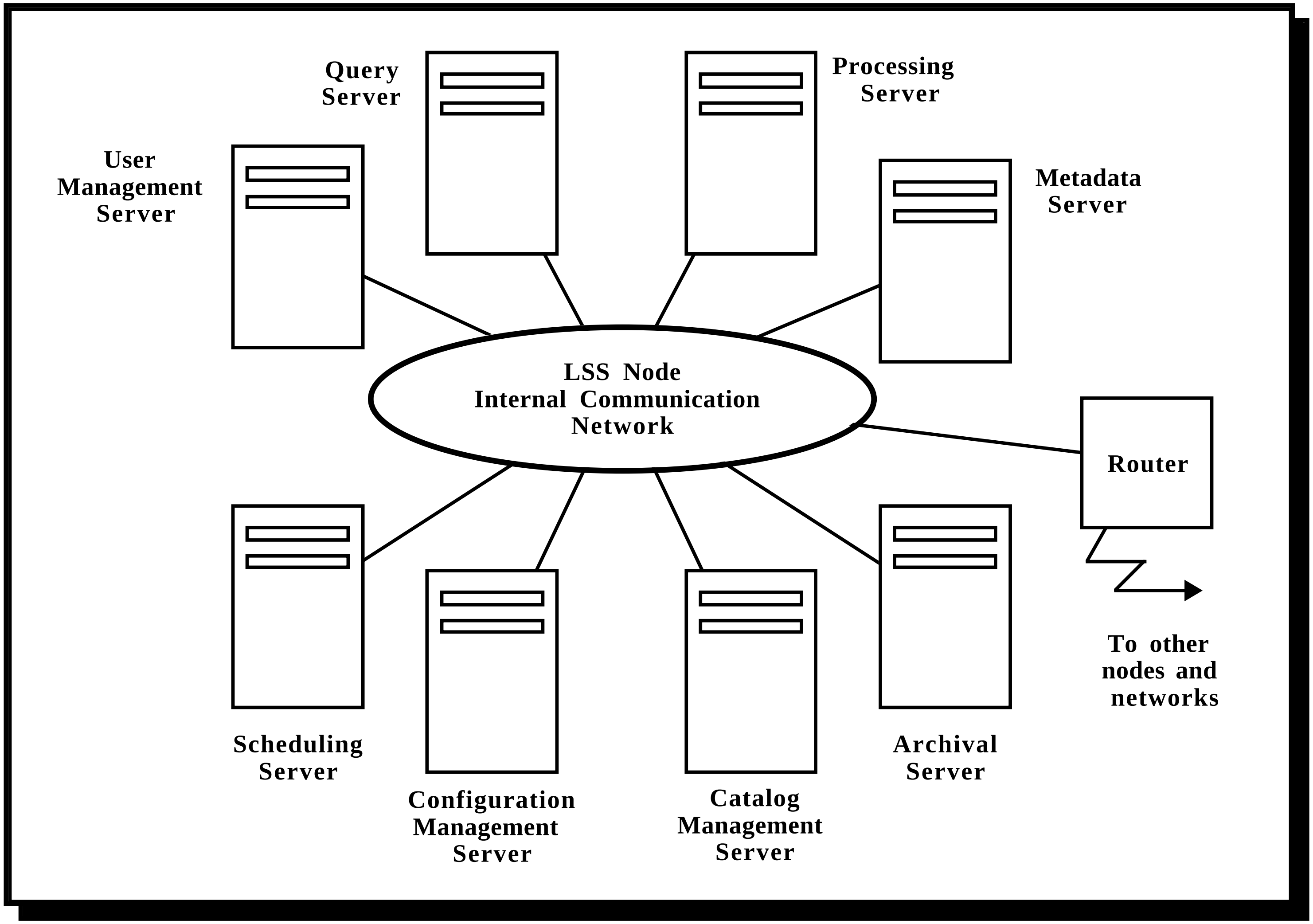
Design Principles for Large-Scale Distributed Data-Intensive Information System (LSS)

- **Separation of Functions:** Group and allocate functions to <u>specialized servers</u> that are optimized performance gains.
- **Scalability:** LSS should be able to <u>evolve</u> as new requirements are imposed on the system, e.g., a new I/O subsystem, additional computing resources added, or a major function allocated to a node.
- **Support for Heterogeneity:** Heterogeneous computing resources should be <u>scheduled to optimize processing</u>, heterogeneous <u>systems should interoperate</u>, heterogeneous <u>data should be integrated</u> into consistent representations.
- **Minimization of Data Transmission:** LSS will have high data transmission volumes. Data transmission should be <u>kept to a minimum</u>, and should be <u>routed through a minimum number of nodes</u>.

GMU Performance-Oriented LSS Design Methodology

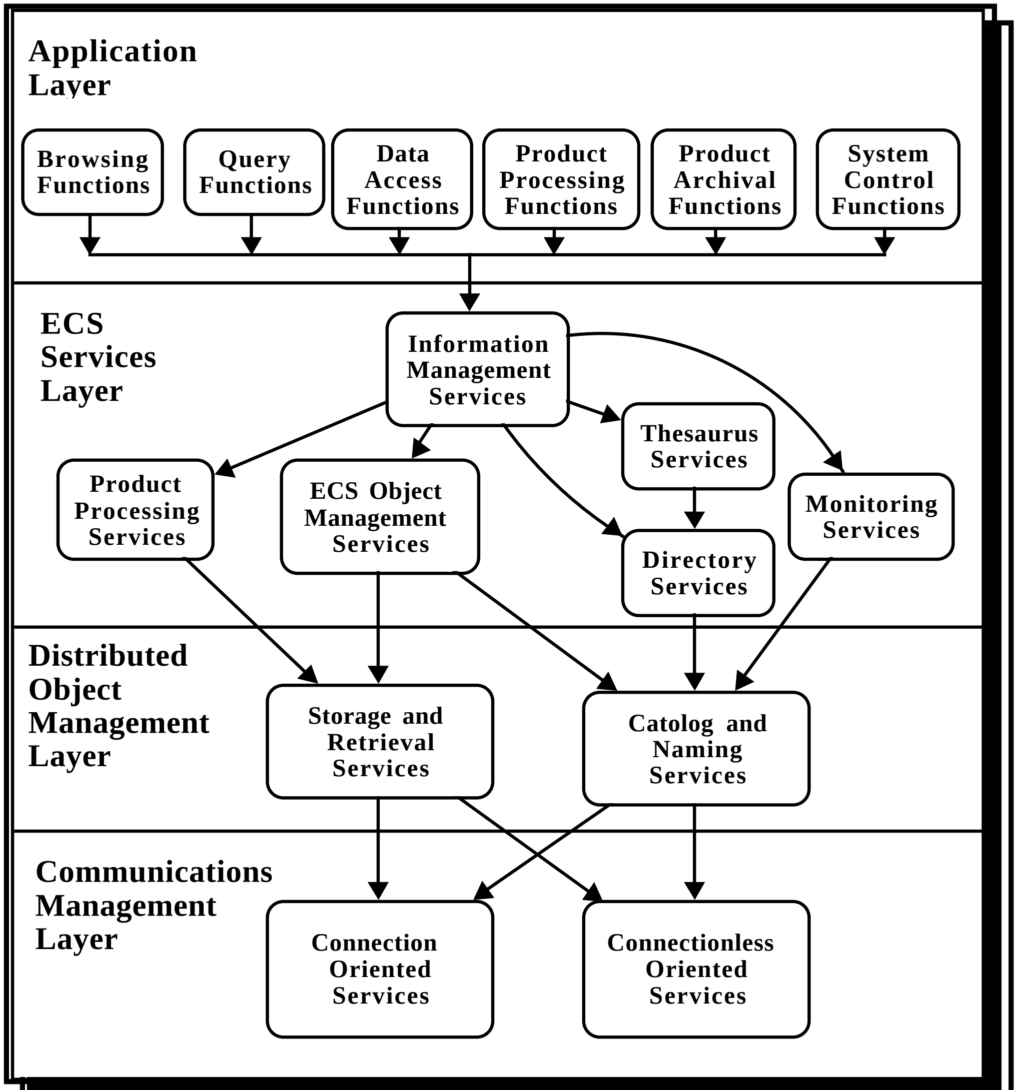


LSS Logical Node Architecture



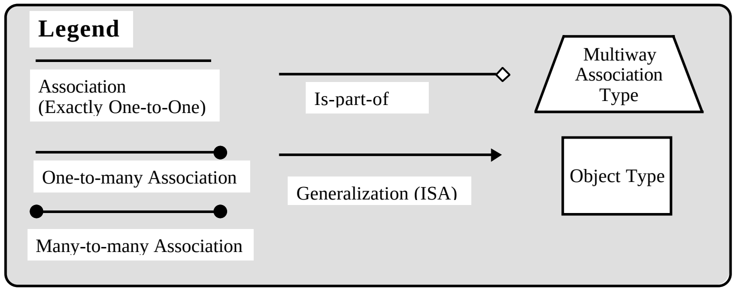
Domain Model for EOSDIS Core System

LSS Multi-Layer Service Architecture



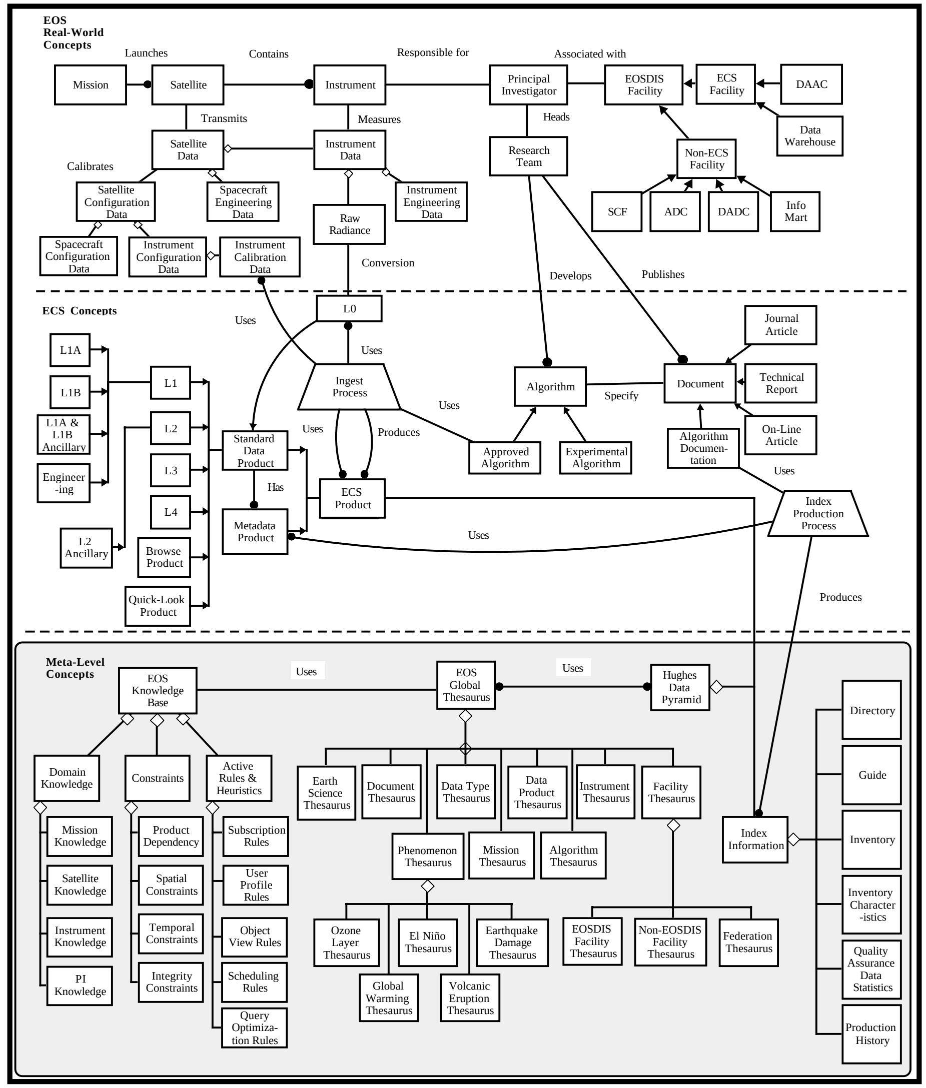
EOSDIS Data and Information Architecture

- **Client-IMS software supports access to the Server-IMS, the information web, and other ECS information resources and services.**
- **EOSDIS Information Architecture consists of:**
  - Data Architecture — serves as a road-map for navigation through information web.
  - Meta-Level Architecture — enhances data schema with knowledge.
    - » EOS Knowledge Base consists of domain knowledge, constraints and active rules.
    - » EOS Global Thesaurus consists of specialized thesauri with knowledge associated with the terms
  - Multi-layer framework — *Data Warehouse* and *Info Marts*.
- **Federated architecture (EOSFed) to support GCDIS.**

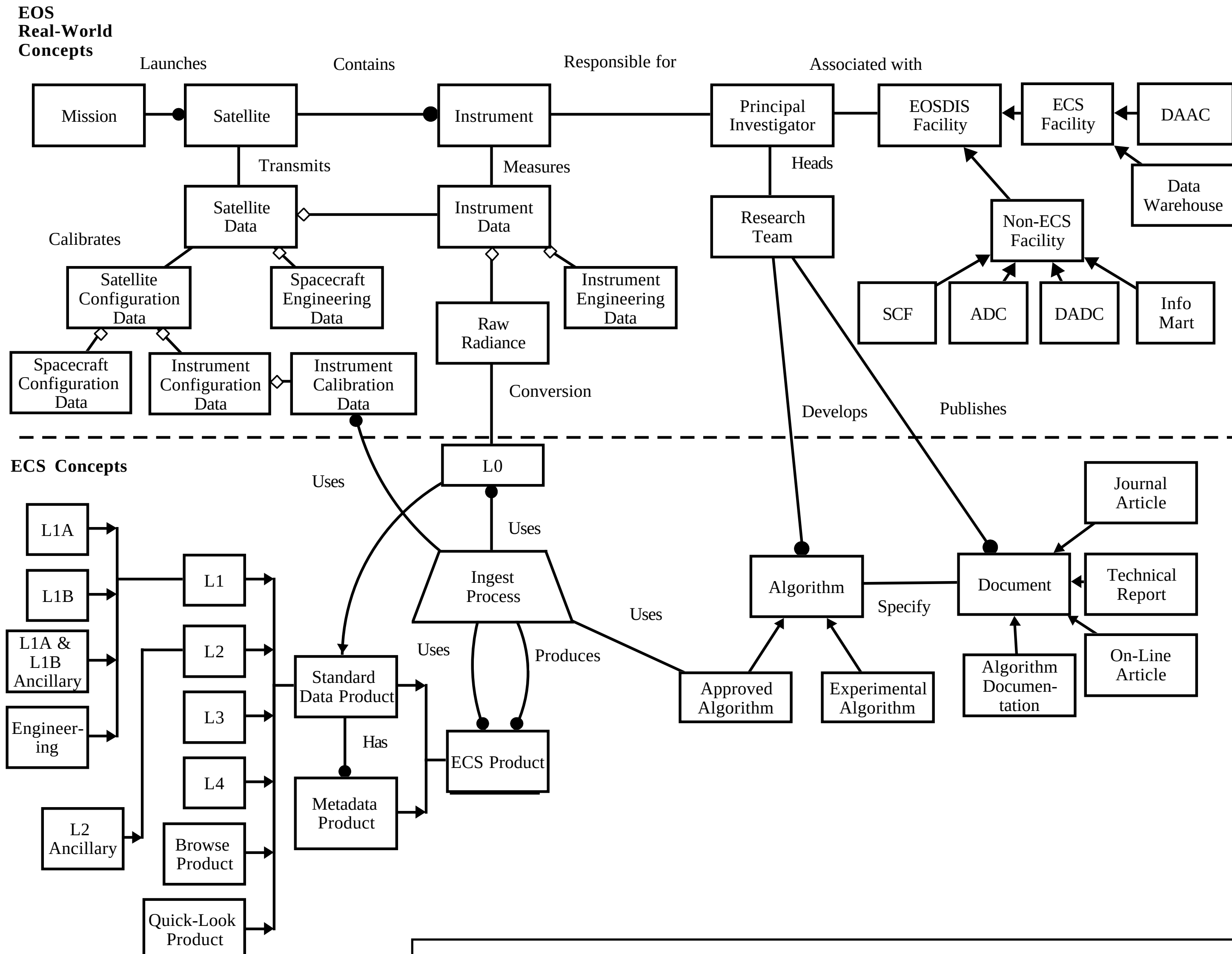


EOS Data/Knowledge Architecture

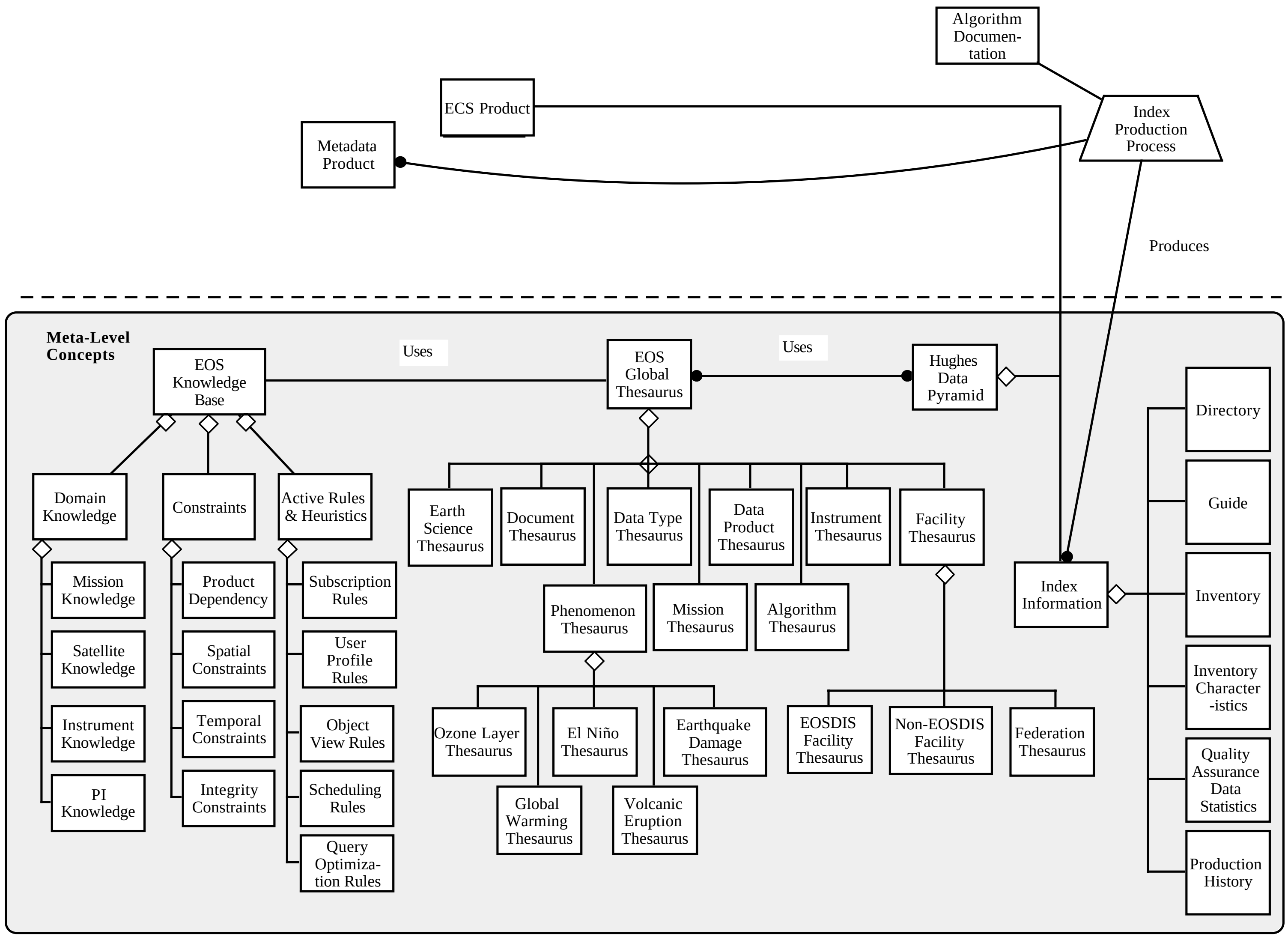
- Users access EOSDIS via the *Information Web.*
- *Information Web* is constructed using the Global Thesaurus, EOS Knowledge Base, Hughes Data Pyramid, and the ECS Data Architecture.
- Web allows users to specify the terms by which a query is formulated, and to link terms from multiple thesauri via the logical structures provided by the Data Architecture.
- The GT combined with the KB allows the thesaurus to be *active* and *intelligent*, thereby allowing user queries to be generalized, specialized and reformulated using domain knowledge and constraints.



The ECS Data Architecture

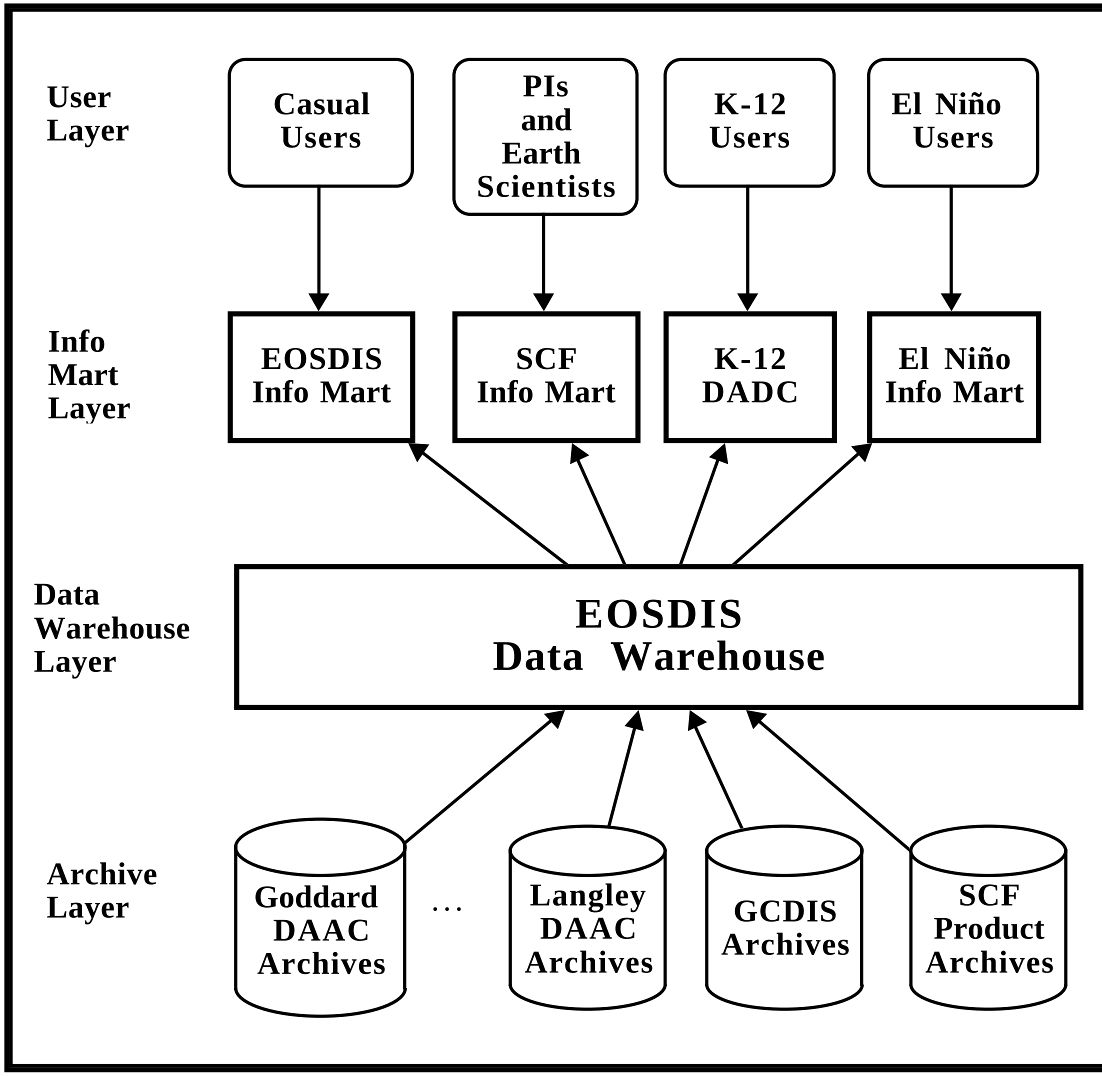


ECS Meta-Level Architecture



The GMU Multi-Layer Information Architecture

- **Users may belong to special communities, e.g., K-12, El Niño, Earth Scientists, etc.**
- ***Info Marts* provide *value-added* products and services to their constituents.**
- **Data Warehouse provides access to highly sought after products, stored in common federation format, specified by EOSFed.**
- **Archives from multiple sources can submit products to the Data Warehouse in EOSFed format.**



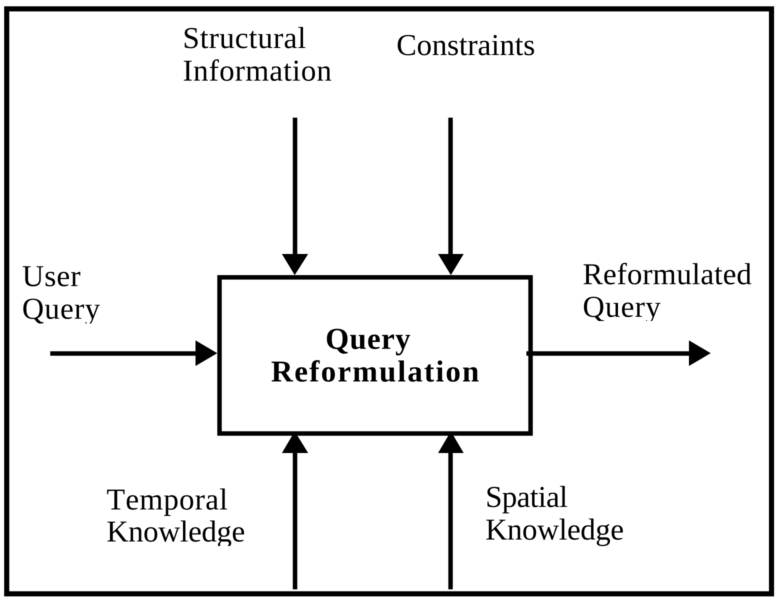
Information Management System Services

- **IMS provides information services to allow users and system components to cooperate:**
  - Access Services — User access, accounting, billing and budget services.
  - Information Request Services — browsing, cooperative query formulation, query optimization, persistent object-view specification, data lineage services.
  - Metadata Services — Thesaurus, Hughes data pyramid, knowledge base access, and knowledge translation.
  - Federation Services — Membership, client-server, language and data translation, temporal mediation, and spatial mediation.
  - Coordination and Scheduling — Product scheduling and request tracking services.

Cooperative Query Reformulation Service
Role of Constraints and EOS Knowledge

Modes of reformulation

Generalization, Specialization, and Analogy of terms, temporal and spatial concepts.



Constraints:

Spatial Constraint: *Equatorial Pacific* denotes the region between - 5˚N to 5˚S and 120˚W to 130˚W.

Temporal Constraint: Bucket temperatures were used for *in situ* temperature measurements until 1964. Injection temperatures have been used since 1965.

Fragment Constraint: The El Niño "relation" is fragmented into two relations, one for sea-surface bucket temperatures, and one for sea-surface injection temperatures .

Complex Sea Surface Temperature Query

Original Query:

SELECT Sea Surface Temperature FROM El Niño
WHERE Temperature < 25˚C and
REGION is *Equatorial Pacific* and
Years BETWEEN 1963 and 1969.

Budget Service predicts huge data result and Query Formulation Service suggests Temperature Range

Spatial specialization constraint applied:

SELECT Sea Surface Temperature FROM El Niño
WHERE Temperature BETWEEN 22˚C and 25˚C and
REGION BETWEEN 5˚N and 5˚S and 120˚W and 130˚W and
Years BETWEEN 1963 and 1969.

Equatorial Pacific Spatial Constraint

Temporal specialization constraint applied:

SELECT Sea Surface Temperature FROM El Niño
WHERE "Bucket" Temperature BETWEEN 22˚C and 25˚C and
REGION BETWEEN 5˚N and 5˚S and 120˚W and 130˚W and
Years BETWEEN 1963 and 1964.

UNION

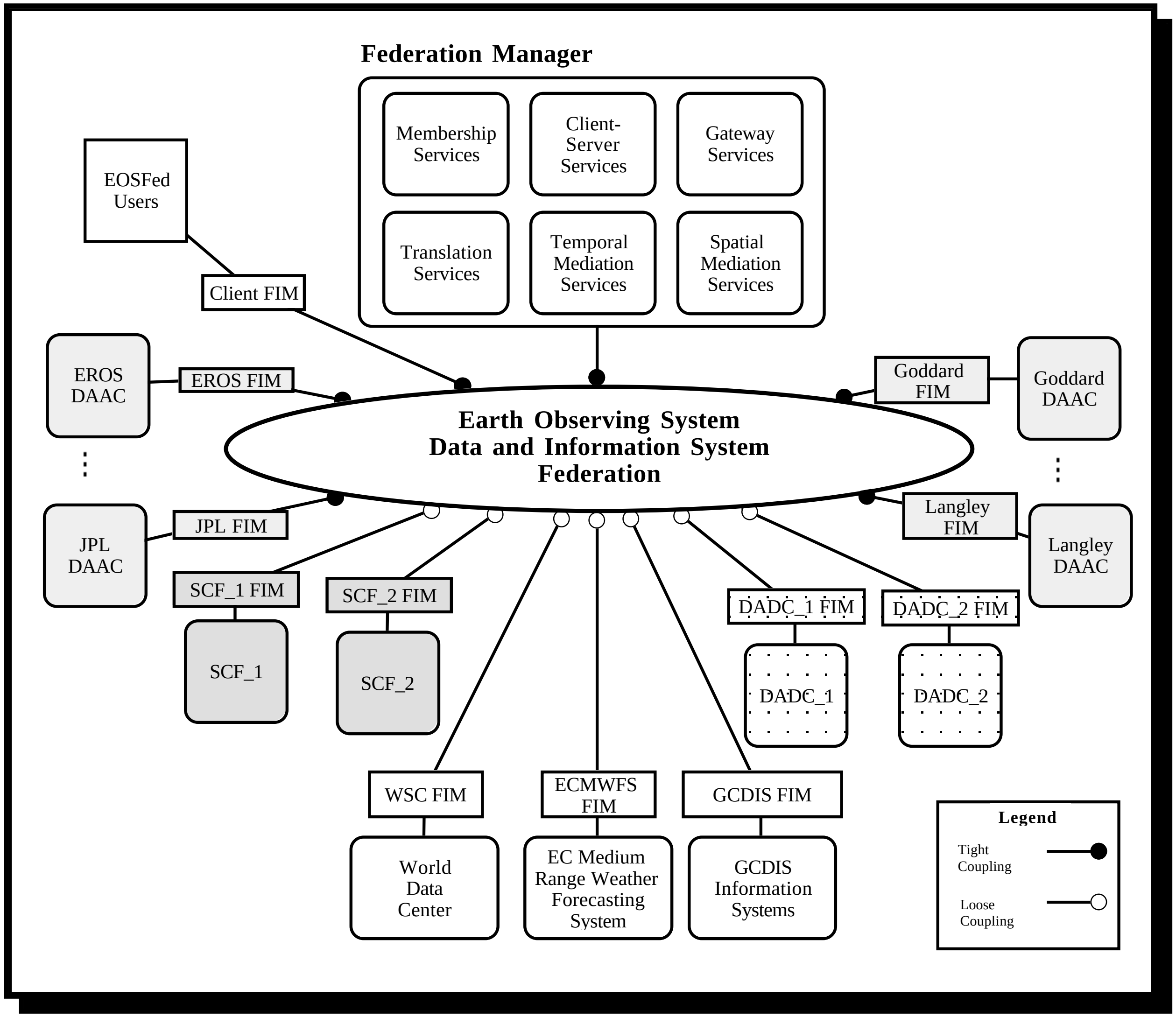
SELECT Sea Surface Temperature FROM El Niño
WHERE "Injection" Temperature BETWEEN 22˚C and 25˚C and
REGION BETWEEN 5˚N and 5˚S and 120˚W and 130˚W and
Years BETWEEN 1965 and 1969.

El Niño relation is fragmented by Bucket and Injection Temperatures

EOSFed — EOSDIS Federated Client-ServerArchitecture

- **Supports a large community of autonomous, heterogeneous information systems.**
- **Each system continues to conduct business as usual for local users, while sharing certain of its information resources with EOSFed members.**
- **Federation supports different types of coupling:**
  - Tight Coupling — in which other members depend on services from the member and are serviced through *contracts*.
    - Examples are DAACs and SCFs who cooperate in producing ECS data products.
    - Note GMU architecture allocates Level 0-3 to DAACs and Level 3-4 to SCFs, DADCs and Info Marts.
  - Loose Coupling — The member system is loosely federated in that it can access/ provide services through *agreements*.

EOSFed Architecture



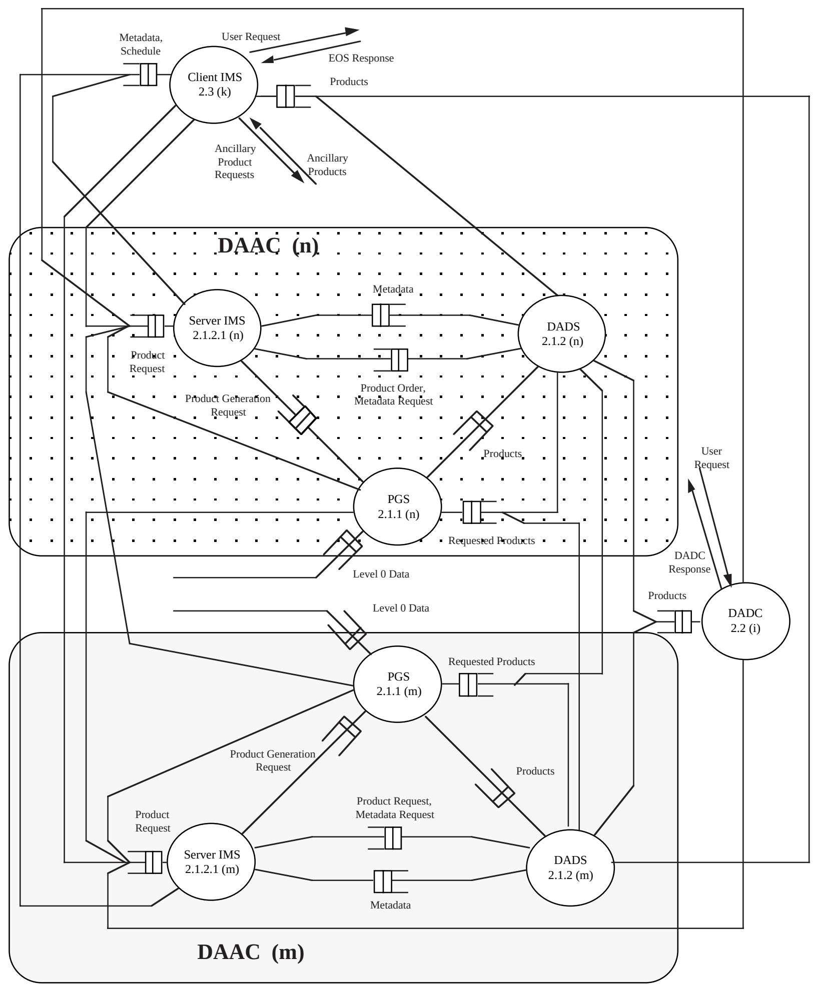
IMS Services for EOSFed

- **Federation Services to support EOSFed.**
  - Membership Services support organizations who wish to become members of EOSFed.
  - Client/Server Services are used to construct Federation Interface Managers (FIMs) for new members.
  - Translation Services are general services to support data units conversion, language translation, and Client/Server translations.
- **Temporal Mediation Services**
  - Support for multiple temporal granularities, translation from one time unit to another, and reasoning about different granularities.
- **Spatial Mediation Services**
  - Integration of information with differing spatial units, differing spatial resolution , differing formats, and from *multiple information sources*.

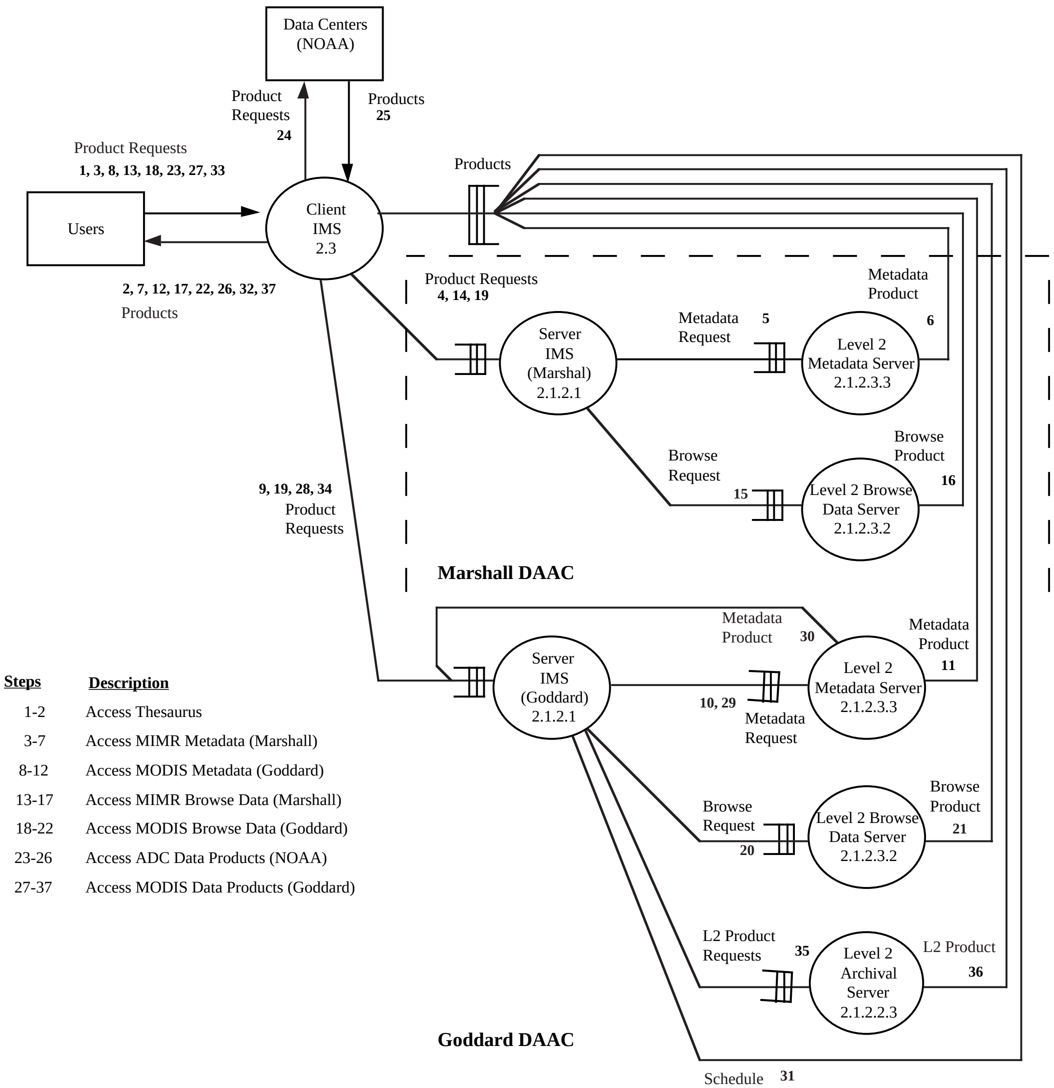
User Scenarios and the GMU Federated Client Server Architecture

- **Earth Scientists on the GMU team formulated several user scenarios:**
  - Prototypical Push Scenario
  - Terrestrial Science Scenario
  - Oceanographic Scenario
  - El Niño Scenario
  - El Niño Reprocessing Scenario
- **Scenarios were used to validate the client-server software architecture and to provide performance-annotated scenarios for the performance model.**
- **El Niño scenario incorporated into the Hughes set of scenarios.**

EOS Concurrent Object Architecture Diagram



Oceanographic Scenario Event Sequence Diagram



Steps	Description
1-2	Access Thesaurus
3-7	Access MIMR Metadata (Marshall)
8-12	Access MODIS Metadata (Goddard)
13-17	Access MIMR Browse Data (Marshall)
18-22	Access MODIS Browse Data (Goddard)
23-26	Access ADC Data Products (NOAA)
27-37	Access MODIS Data Products (Goddard)

Conclusions

- **Large-scale data-intensive system architectures must be modeled from multiple viewpoints :**
  - User Models — Characterization of workload for *push* and *pull* scenarios.
  - Client-Server Software Architecture
  - Software and Hardware Architecture
  - Data and Information Architecture
  - Telecommunications Architecture
  - Performance Model
- **Data Intensive Systems are also *knowledge-intensive*:**
  - Rich metadata needed to index, characterize and access data and information holdings,
  - Domain knowledge needed to navigate the information web of concepts that relate information holdings.
- **Large-scale systems must be *engineered* to meet a complex set of performance, functional, architectural, economic and political requirements.**

Recommendations for DBMS and File Systems

- **Database Management Systems**
  - Multidatabase support for EOSFed concepts.
  - Support for rules as *first class* objects to support active databases and the intelligent thesaurus.
  - SQL-based object-relational systems that support both structured and unstructured data, as well as scientific data types.
- **Storage Systems**
  - System architecture:
    - » Network-attached storage.
    - » Separation of *data* and *control* paths (e.g., HIPPI for data and FDDI for control).
  - Hierarchical file and storage management system:
    - » Should be based on IEEE Mass Storage Reference model.
    - » Next-generation systems, e.g., HPSS from National Storage Laboratory should be evaluated and evolved.