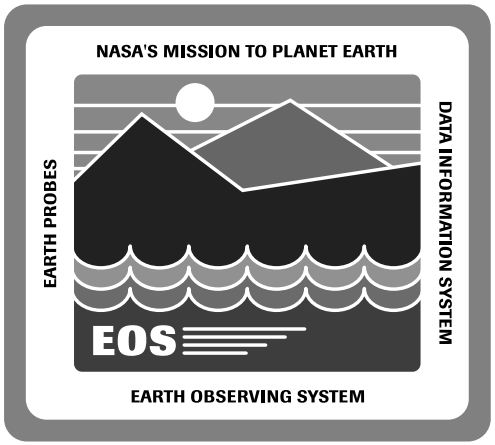
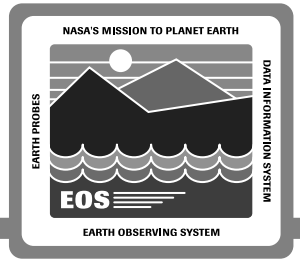



# Science Data Processing - Overview

## Mark Elkington

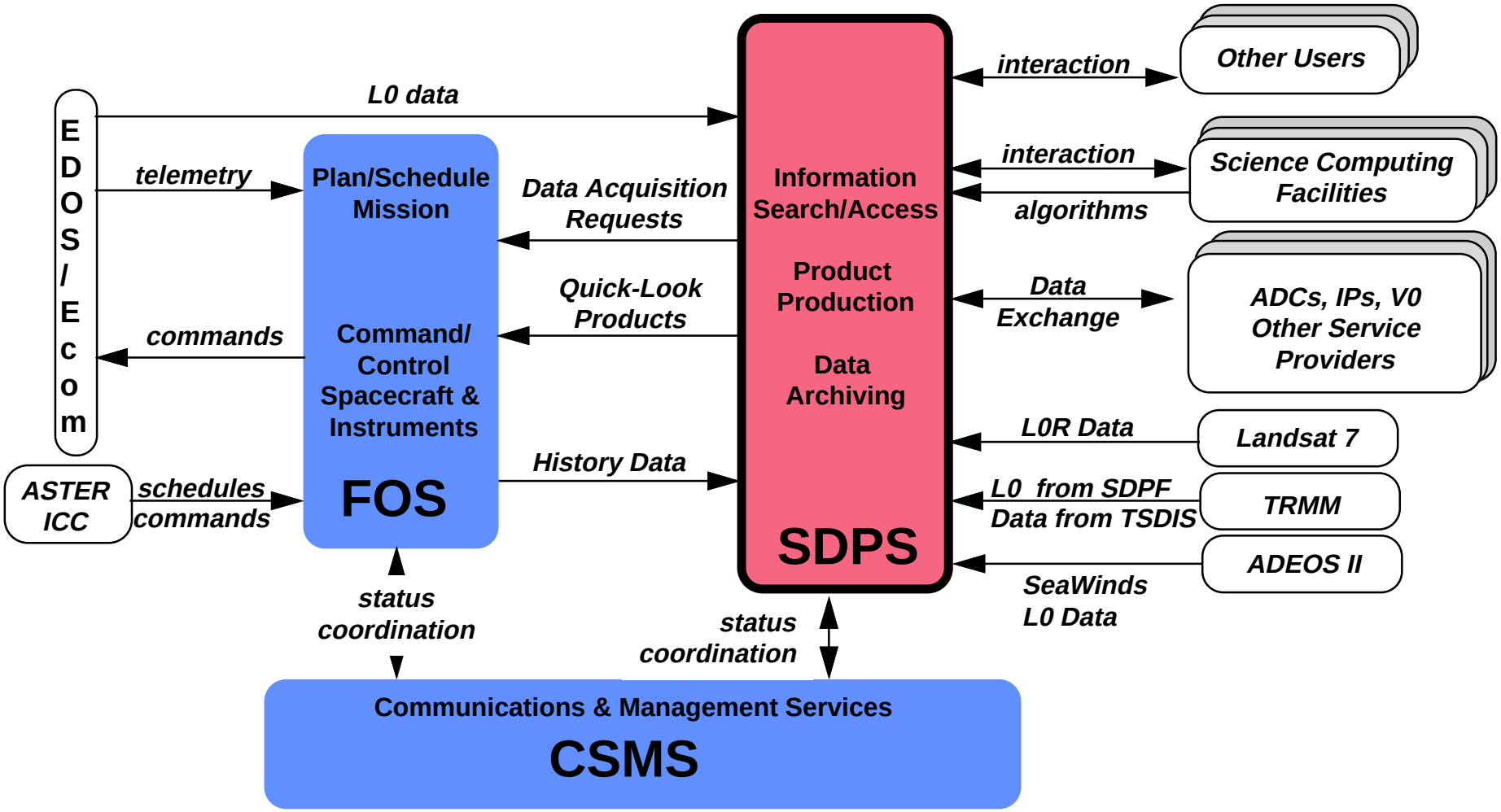
### System Design Review - 27 June 1994

# SDPS Outline



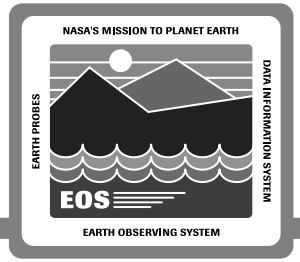

- **Context within ECS**
- **Mission & Key Requirements**
- **Requirements to Design Flowdown**
- **System Design Activities**
- **Changes Since SRR**
- **Position in ECS Reference Model**
- **High-Level Architecture**
- **Subsystem Introduction**
- **Splinter Session Road Map**

# SDPS Context




*All interfaces use communication services provided by CSMS*

# SDPS: Mission & Key Requirements



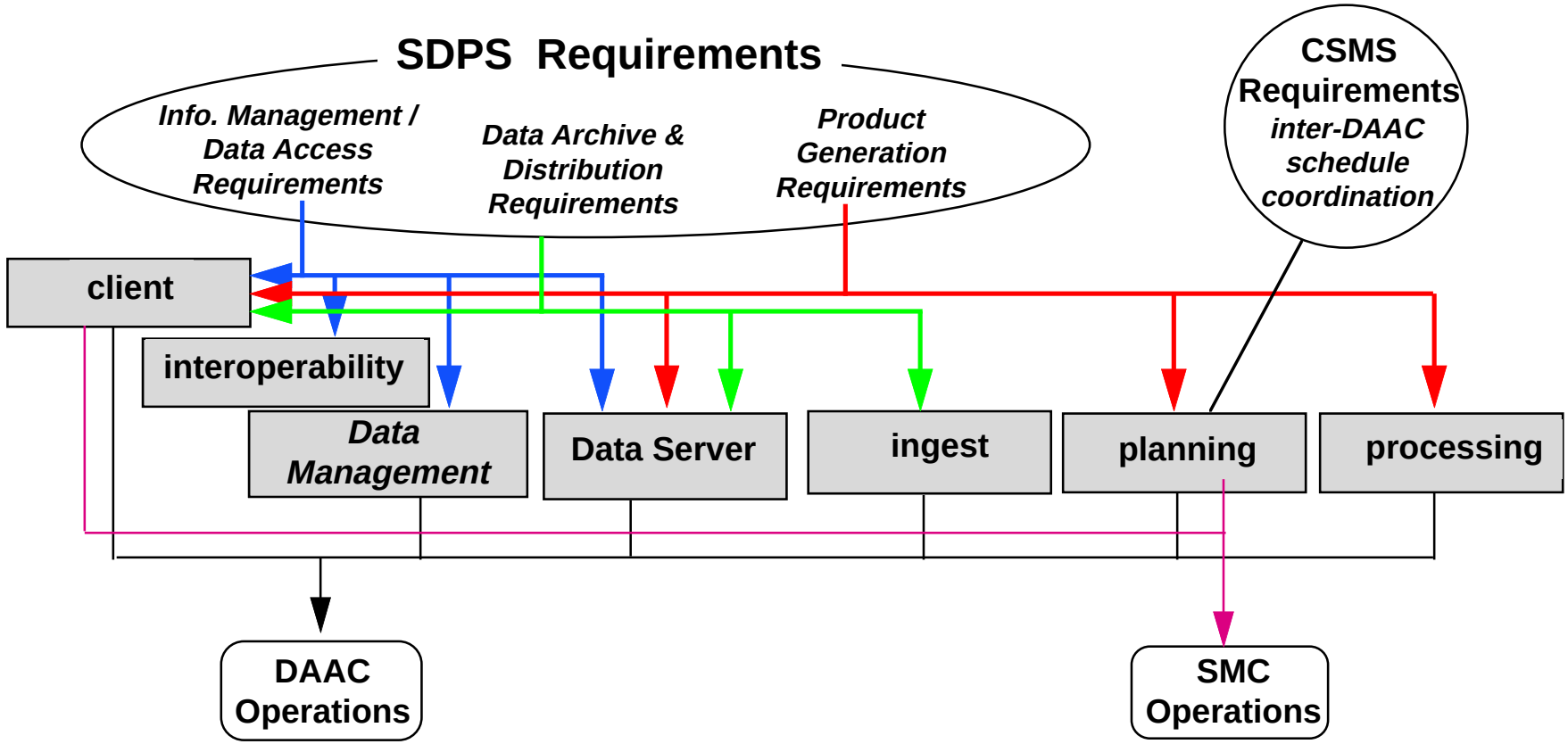

**SDPS is the segment that supports the long-term scientific utilization of the data**

**SDPS:**

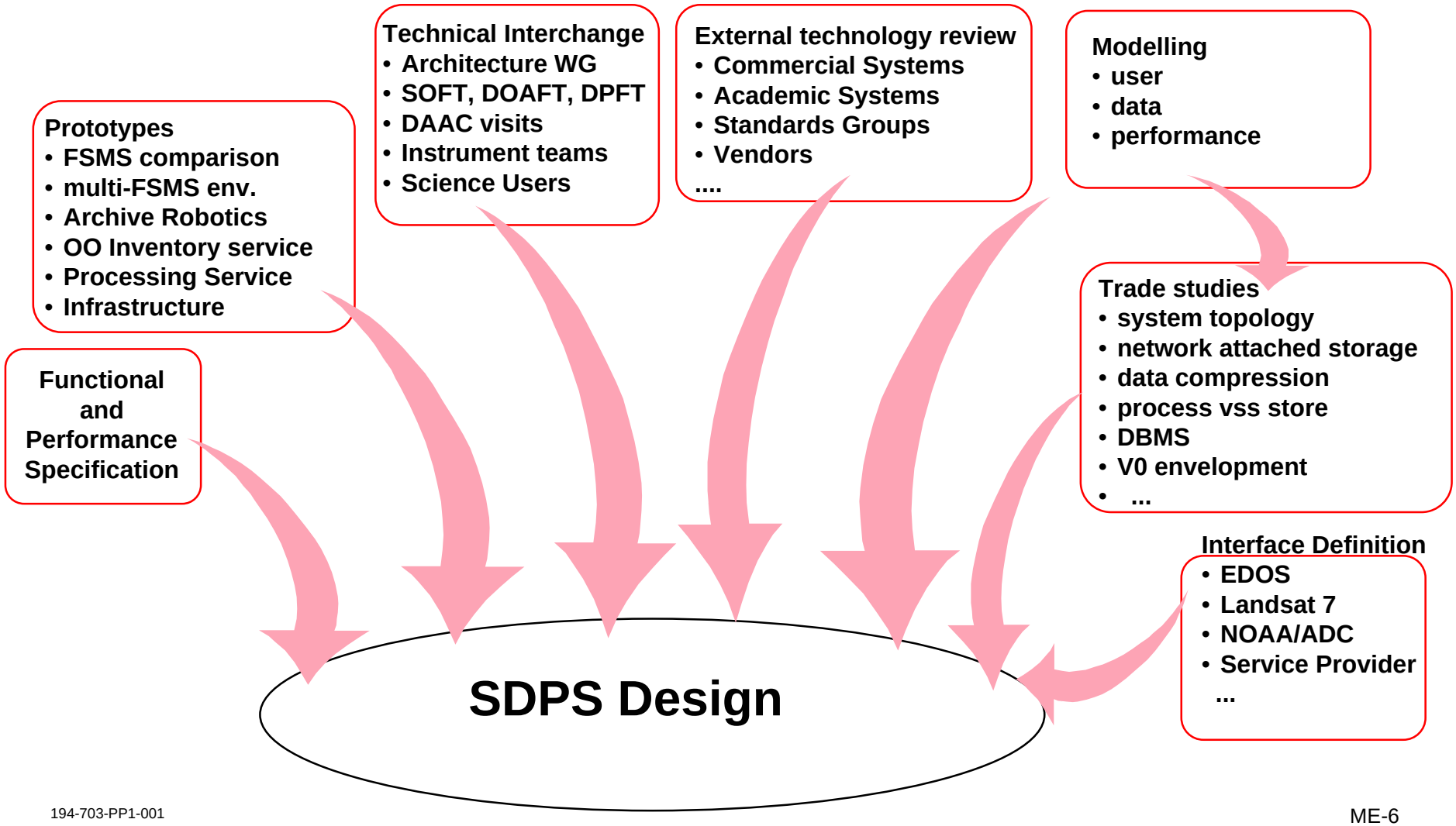
- **receives, processes, archives and manages DATA**
- **provides effective access to the DATA and RESULTS for the Earth science community in support of MTPE goals**
- **facilitates development and usage of new/improved algorithms**

**These core requirements map into seven SDPS subsystems**

# SDPS Requirements to Design Flowdown



# SDPS System Design Activities



# SDPS Changes Since SRR

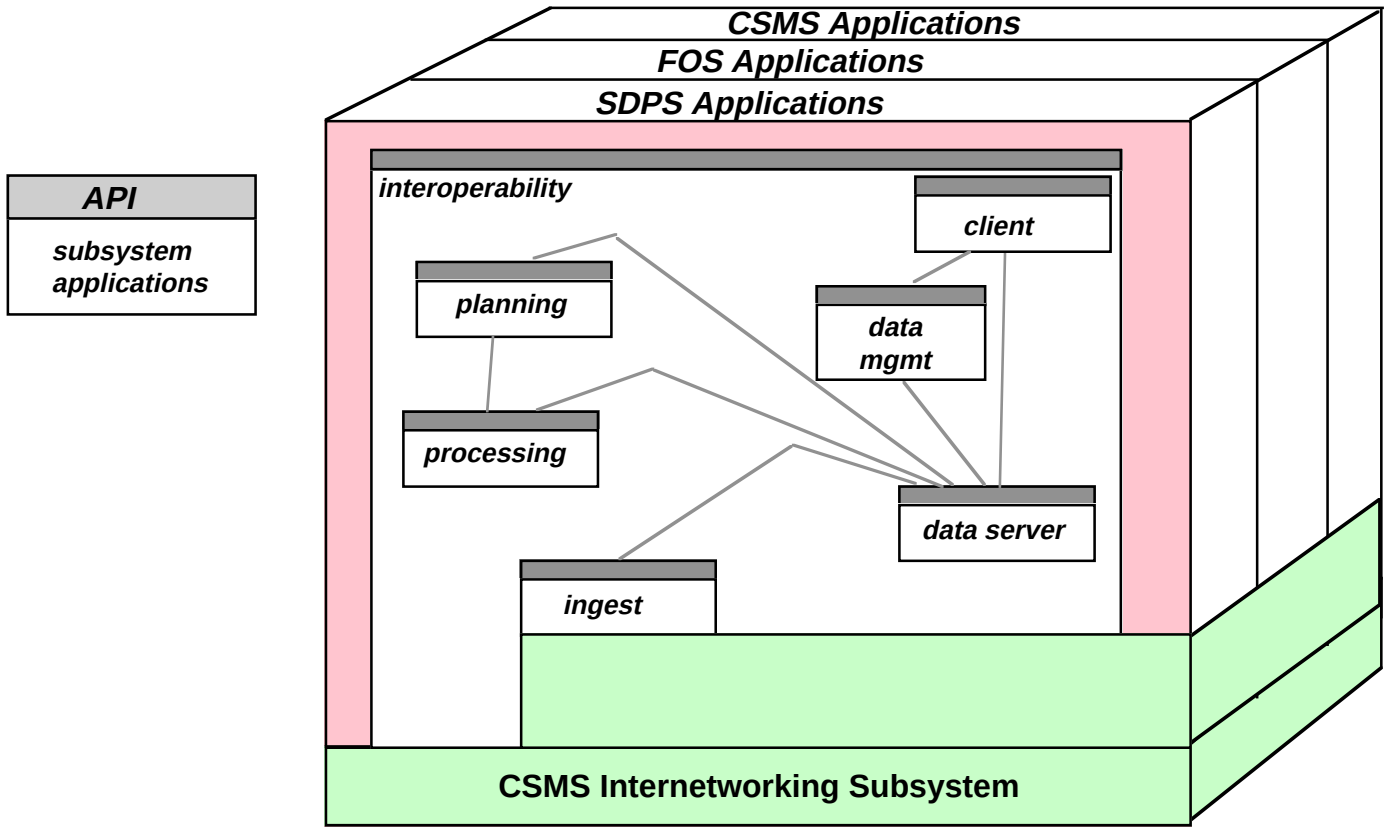
**Traditional Data Systems Concepts**

***Towards ...***

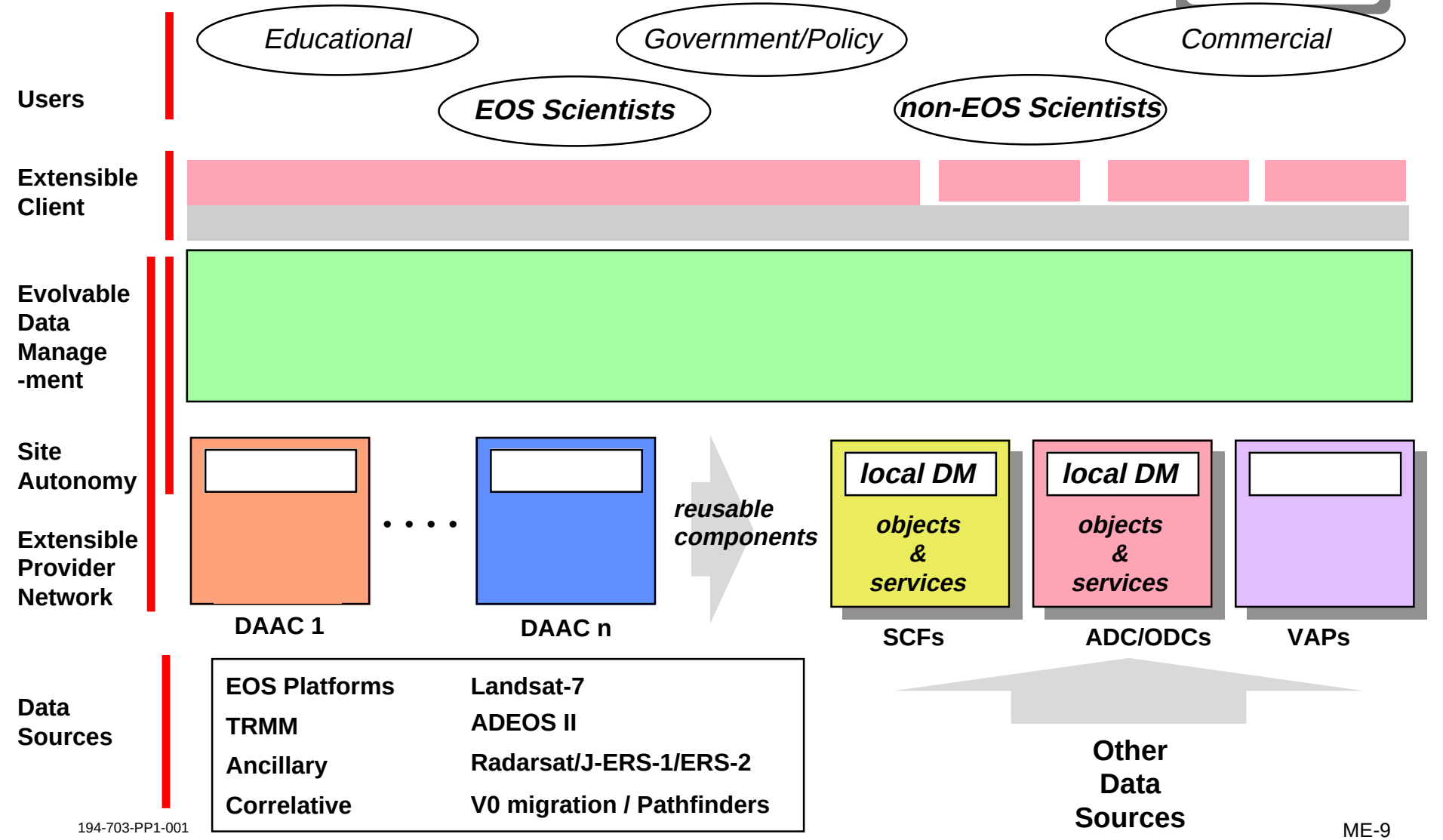
- **Product “publishing” and “access”**
- **Seamless view of all data**
- **Extended provider implementation**
- **Heterogeneous, autonomous system components**
- **Scalable, extendable components**
- **Policy neutral**
- **Component reuse outside of EOSDIS**

***A more evolutionary system***

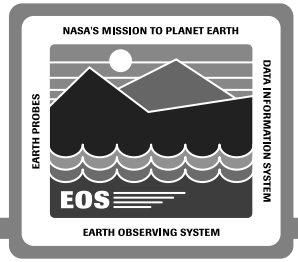
# Position in ECS Reference Model



# SDPS Conceptual Architecture

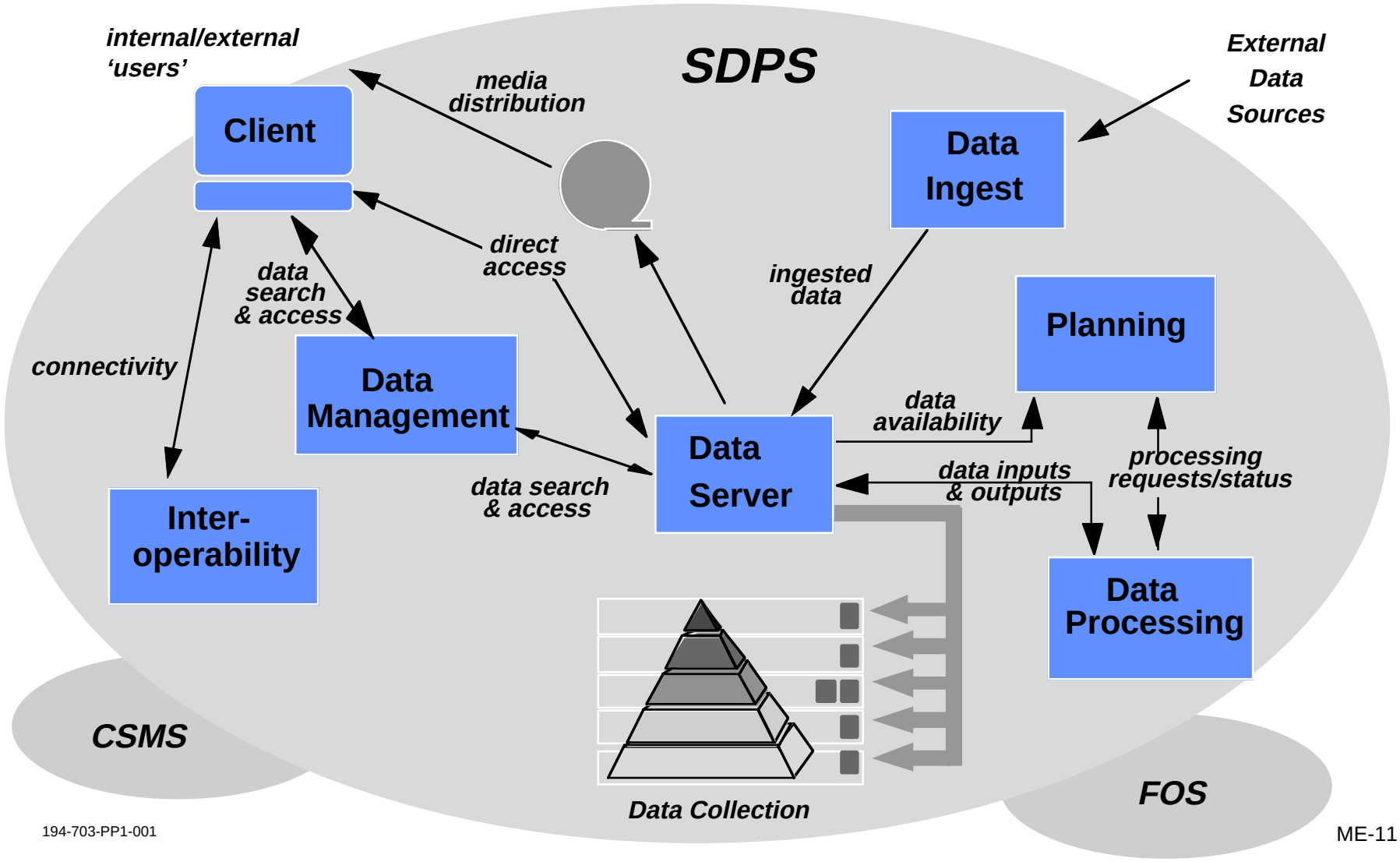


# SDPS Architecture Features

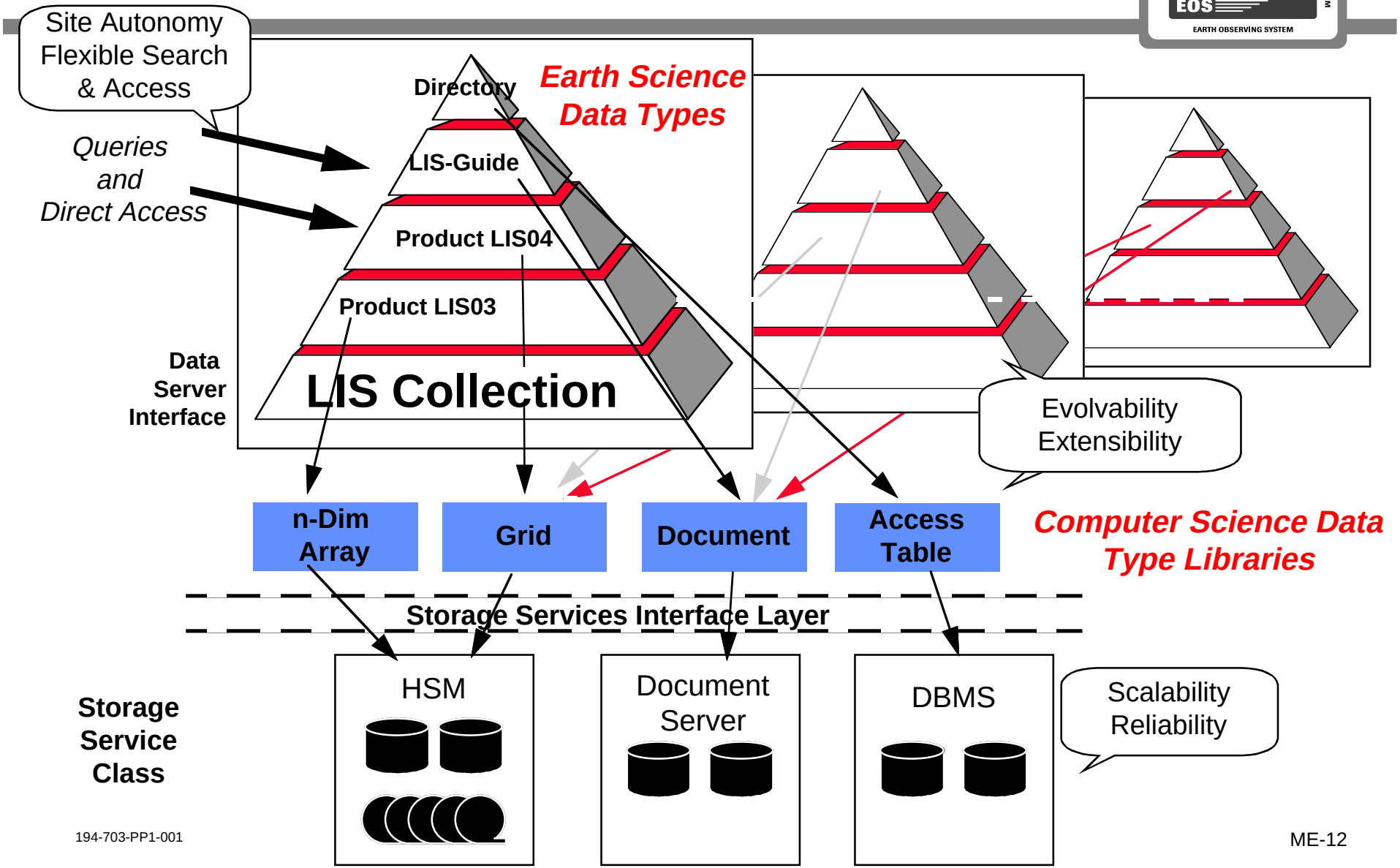



- Object- and service-oriented architecture
- Extensible client environment
- Evolutionary approach to Data Management
- Logical Distribution of components
  - Supports transparent physical distribution
  - Supports service migration
- Site autonomy and heterogeneity
- Extensible provider network
  - Domain specific representations
  - Development of new types and services
  - Incorporation of heritage systems
- Reusable components for GCDIS / UserDIS

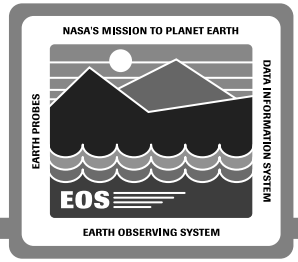
# SDPS High Level Design



# Data Server


Site Autonomy
Flexible Search
& Access
Queries
and
Direct Access
Directory
LIS-Guide
Product LIS04
Product LIS03
LIS Collection
Earth Science
Data Types
Data
Server
Interface
Evolvability
Extensibility
n-Dim
Array
Grid
Document
Access
Table
Computer Science Data
Type Libraries
Storage Services Interface Layer
Storage
Service
Class
HSM
Document
Server
DBMS
Scalability
Reliability

# Data Server




**Each *Data Collection* is managed by a *Data Server*:**

- **consists of many types of logical data objects (ESDT)**
- **uses several types of physical data structures (CSDT)**
- **Collections can be DAAC and Discipline unique**

**Access to collection is via the *Data Server Interface* for**

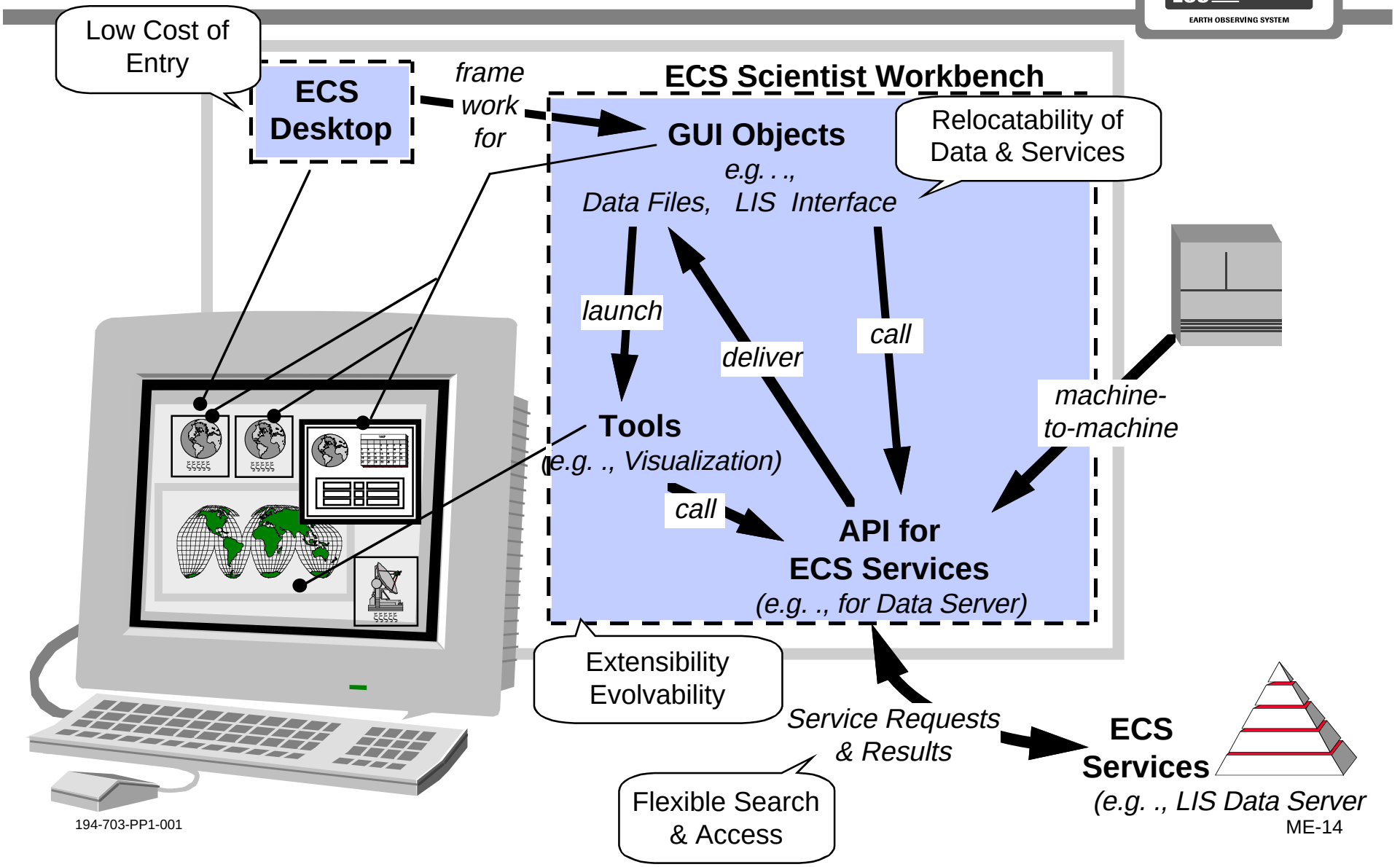
- **searching, retrieval, special services**

**Contents and services of collection are**

- **defined in a *Data Server Schema***
- **described in a *Data Server Data Dictionary***

**Uses *Storage Service Class Interface* to employ different types of data management and storage technologies**

# Client


Low Cost of Entry
ECS Desktop
frame work for
ECS Scientist Workbench
GUI Objects
e.g. . .,
Data Files, LIS Interface
Relocatability of Data & Services
launch
deliver
call
Tools
(e.g. ., Visualization)
call
API for ECS Services
(e.g. ., for Data Server)
machine-to-machine
Extensibility Evolvability
Service Requests & Results
ECS Services
(e.g. ., LIS Data Server
Flexible Search & Access
194-703-PP1-001

# Client

**ECS Client provides search and access capabilities to ECS services:**

- **User access via GUI Objects**
- **Machine-to-Machine access via API Libraries**

**Basic *ECS Desktop* provides GUI framework for:**

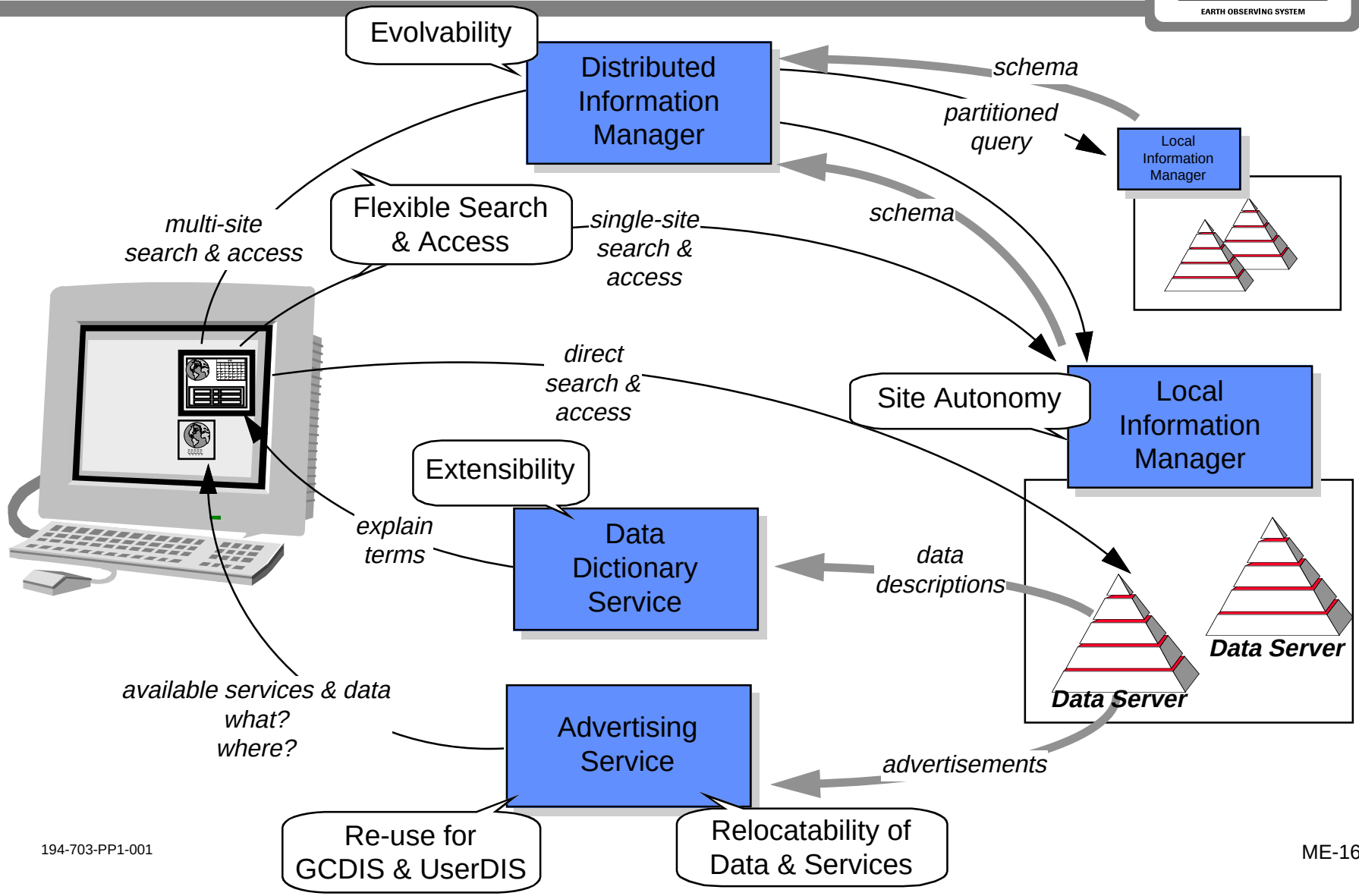
- **installation of GUI Objects and tools**
- **tool launching, object embedding**
- **data format translation**

**Framework provides extensibility to add:**

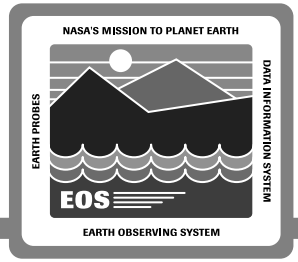
- **science user tools & data types**
- **(GUI interfaces to) new services**

**Character-based Interface for Basic Access Functions**

# Interoperability & Data Management



# Interoperability & Data Management




**Find services and data - retrieve service representation**

- ***Advertising Service***

**Get explanations of terms**

- ***Data Dictionary Service***

**[Get Direct Access to a Collection**

- ***Data Server Service*** **]**

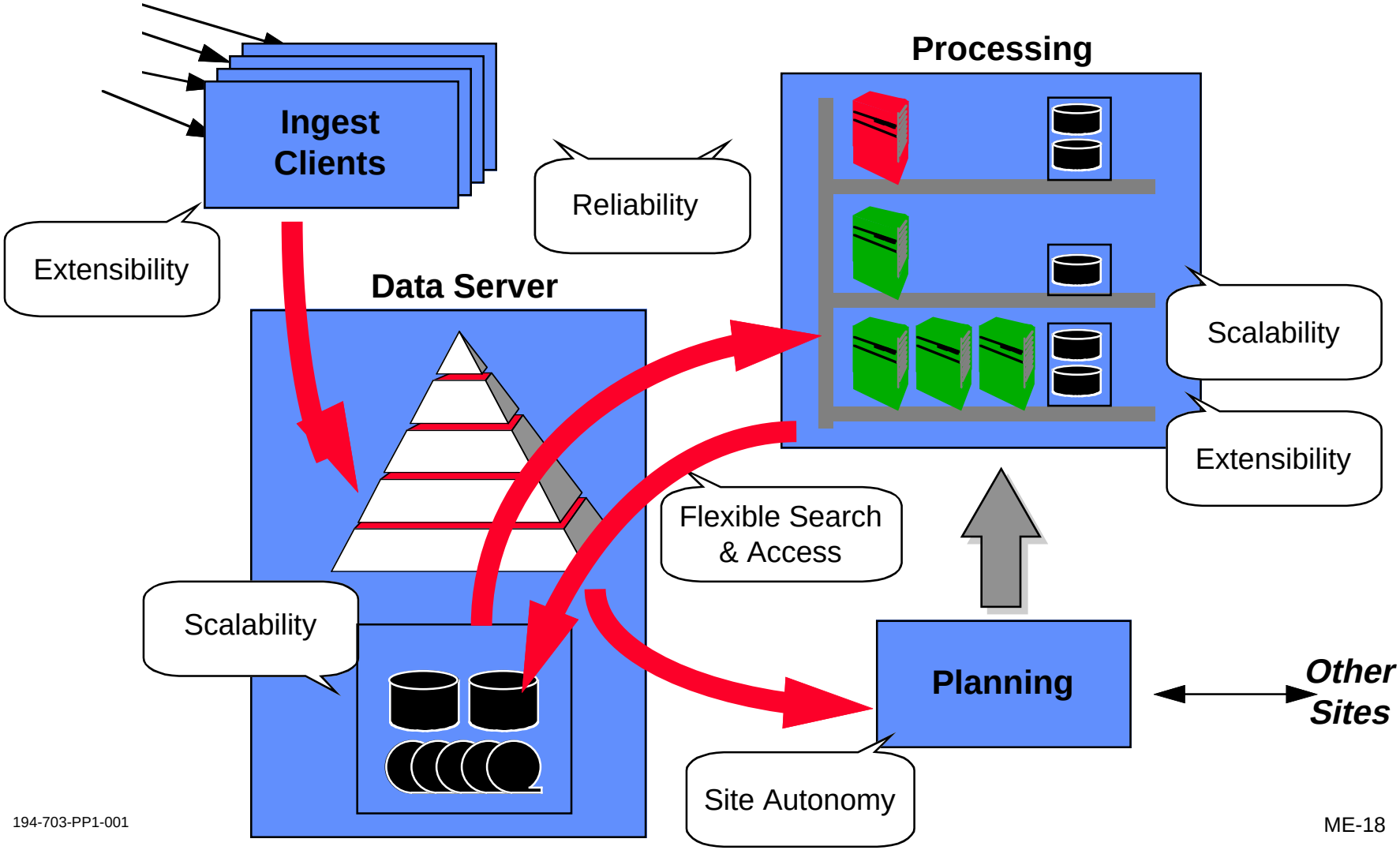
**Search several collections at a site + provide site autonomy**

- ***Local Information Management Service***

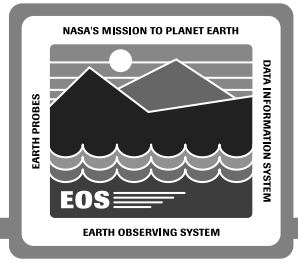
**Multi-site searching**

- ***Distributed Information Management Service***

# Ingest, Planning & Processing

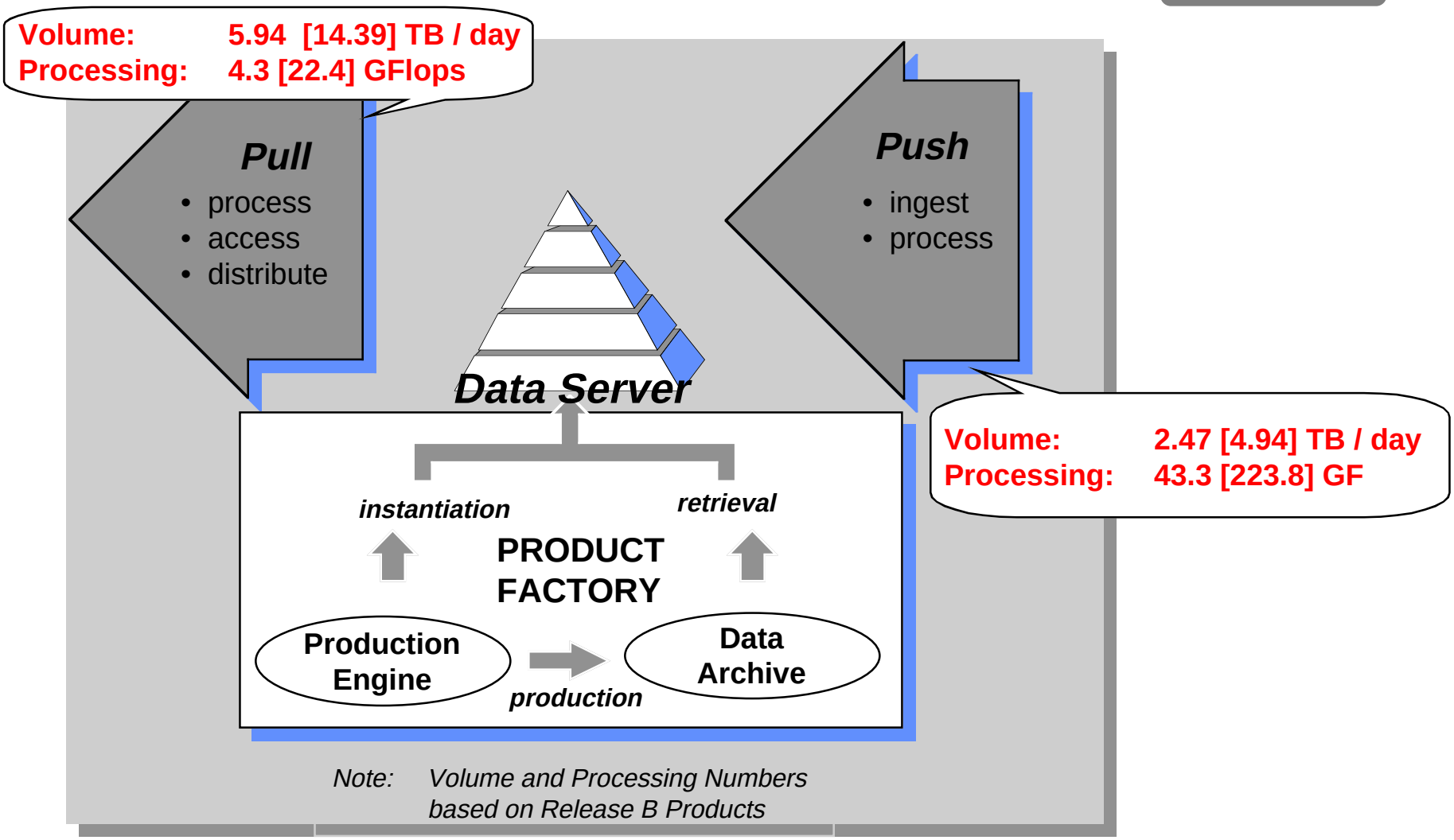


# Ingest, Planning & Processing

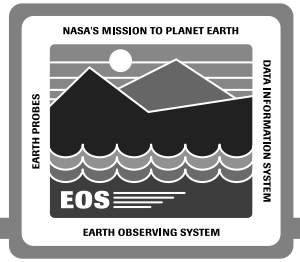



- **Populate pyramid and produce data needed to support effective search and access**
- **Cost-effective scaling to processing requirements - ability to distribute resources**
- **Hardware and software measures to support reliable data production**
- **Site autonomy in planning - support for coordination of distributed production**
- **Flexible quality assurance -- on-site and off-site**
- **Support routine and *ad-hoc* data production**
- **Simplify extension of ingest to new data sources**

# Push - Pull Balance



# Push - Pull Characteristics



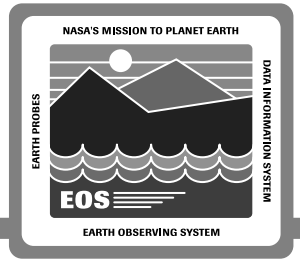

## Data Server is Common Foundation

- shared resource pool for effective utilization and reduced cost
  - policies and pricing used to guide resource management
  - reallocation of resources to meet changing demands
- Product Factory abstraction supports flexible implementation
- reuse of common services
  - across provider sites
  - across data types

## Scalability, Evolvability

- Push: configuration and distribution of processing strings
- Pull: configuration of data servers at base providers
  addition of value-added provider services
  SCFs as providers
  new service providers (value-added / fee-for-service)

# SDPS: Splinter Session Roadmap




- **Design Approach**
- **Data Server & Client Subsystems**
- **Interoperability & Data Management Subsystems**
- **Ingest, Planning and Data Processing Subsystems**
- **Scenarios**
  - **#2 - Experienced Scientist Interaction**
  - **#3 - Machine-to-Machine interaction**
- **Software Implementation Design**
- **Hardware Implementation Design**
- **Evolvability Tests**
- **Release Plans**