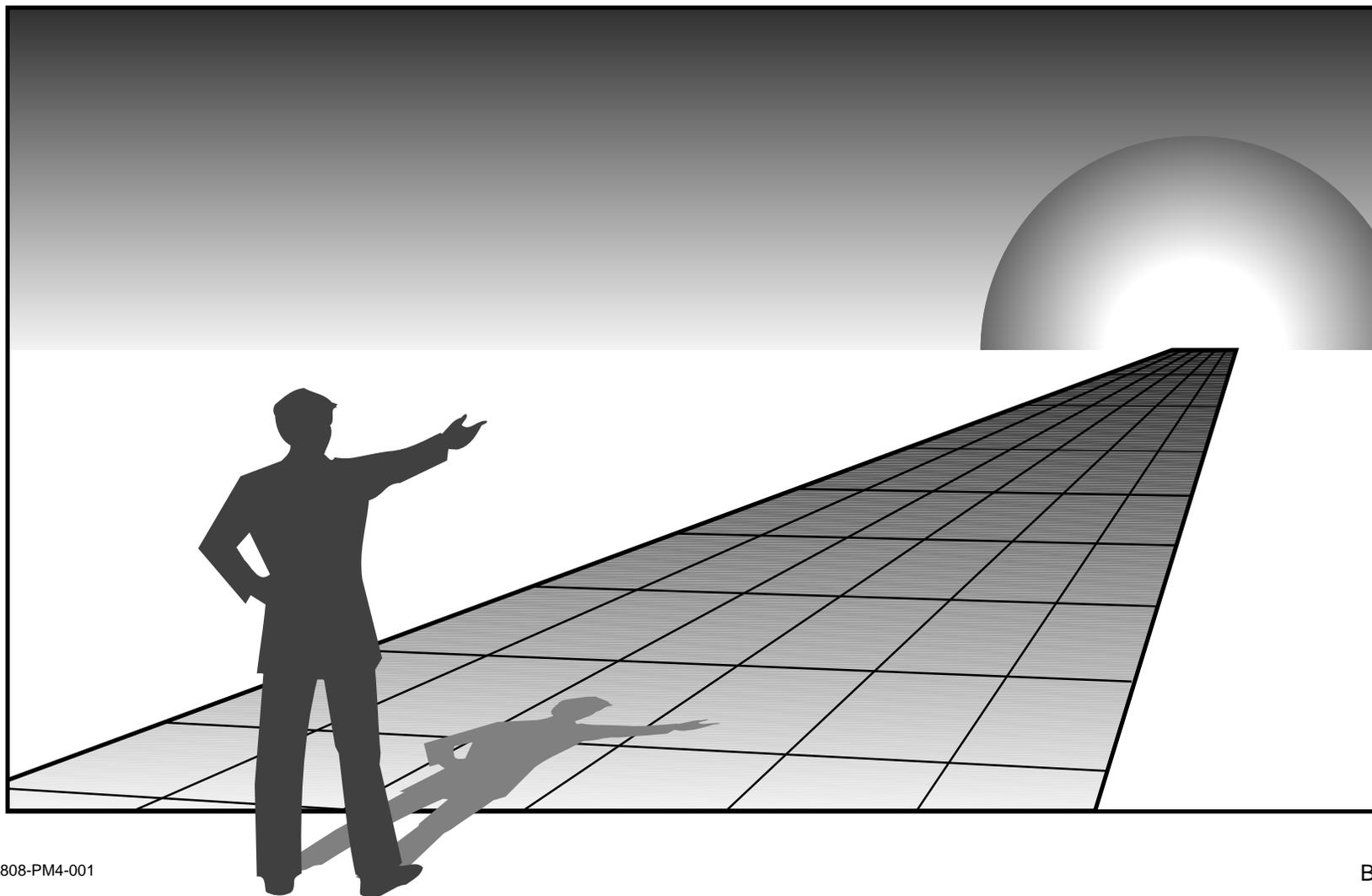
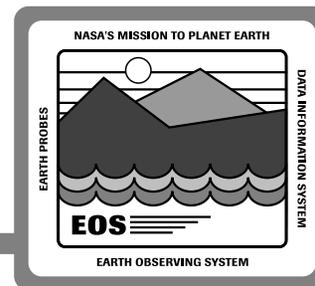


ECS Architecture

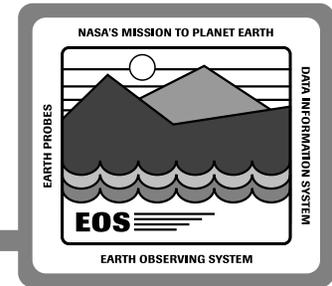
Bruce Moxon

13 - 14 December 1993

ECS Vision



ECS Architecture Roadmap



ECS Architecture Activities

Post-SRR Influences

Definition Process

Drivers and Principles

ECS Conceptual Architecture

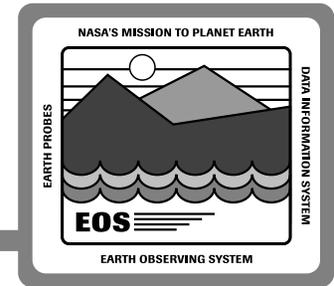
Interoperability

Interconnection

Service Interfaces

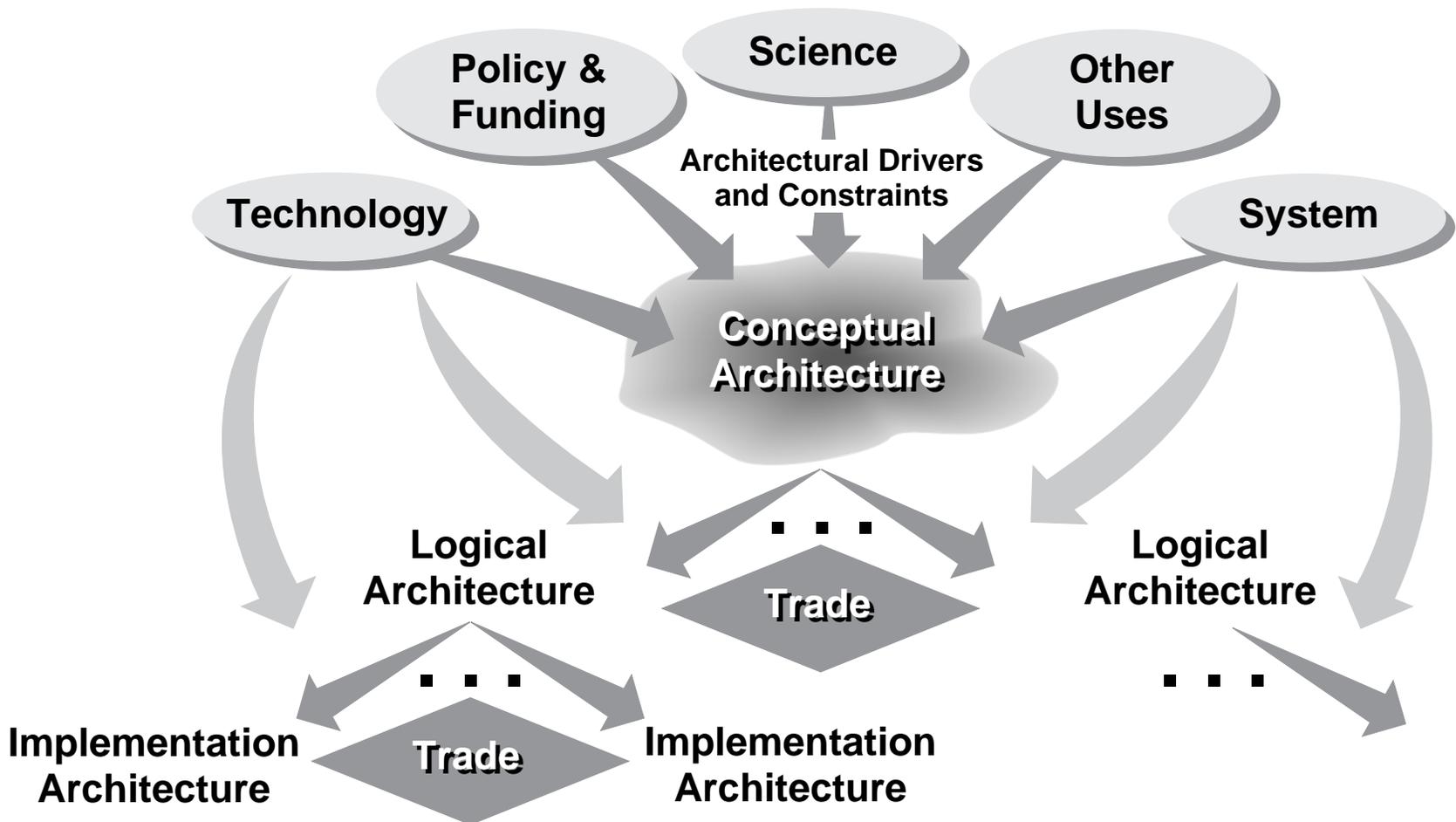
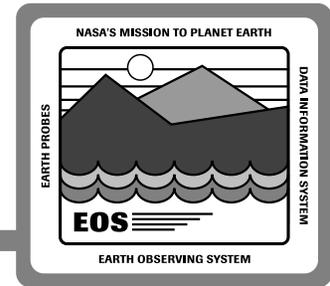
Data Management

Post-SRR Architecture Influences

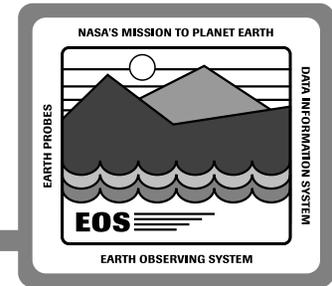


- **SRR Feedback**
- **Science Office Visits**
- **GCDIS/userDIS Study**
- **V0 Analysis**
- **Evolution and Evolvability Study**
- **User and Data Models**
- **External Systems Studies**
- **Architecture Teams and Trades**

Architecture Definition Process



Architecture Drivers



Science

- Meet science needs for Global Change research

Technology Advances

- Enable the system to take advantage of technological advances in software, processing capability, storage, and networking

System

- Scalability, evolvability, reliability, maintainability, availability

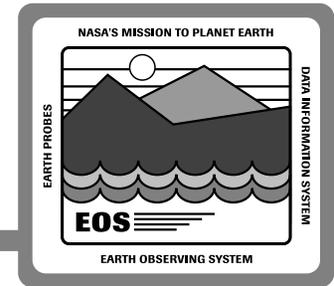
Other Uses

- Enable re-use of components outside of ECS (DAAC unique elements, GCDIS, userDIS, etc.)

Policy and Funding

- Produce an architecture that is policy neutral, allowing science, policy, and funding to guide implementation and distribution decisions

Mandate



Based on SRR feedback, and subsequent interaction with the science community, take a step back and re-evaluate the baseline system architecture.

Move from ...

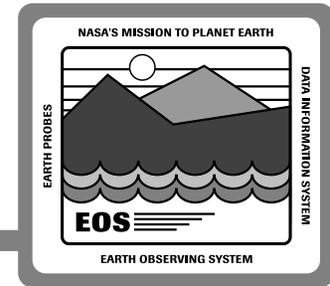
- ∅ Product approval and ordering**
- ∅ Metadata / data distinction**
- ∅ Limited provider implementation**
- ∅ Homogenous, centrally managed system components**

Towards ...

- √ Product “publishing” and “access”**
- √ Seamless view of all data**
- √ Extended provider implementation**
- √ Heterogeneous, autonomous system components**

A more evolutionary system

Architecture Principles



Architect Around User Needs And Issues, Not Technical Solutions

⇒ *Architecture Provides Framework For Different Solutions (Evolvability)*

Use A Small Set of Good Concepts

⇒ *Reduces Number of Protocols and Interfaces*

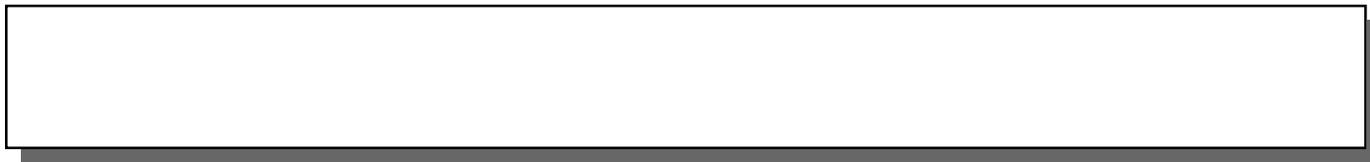
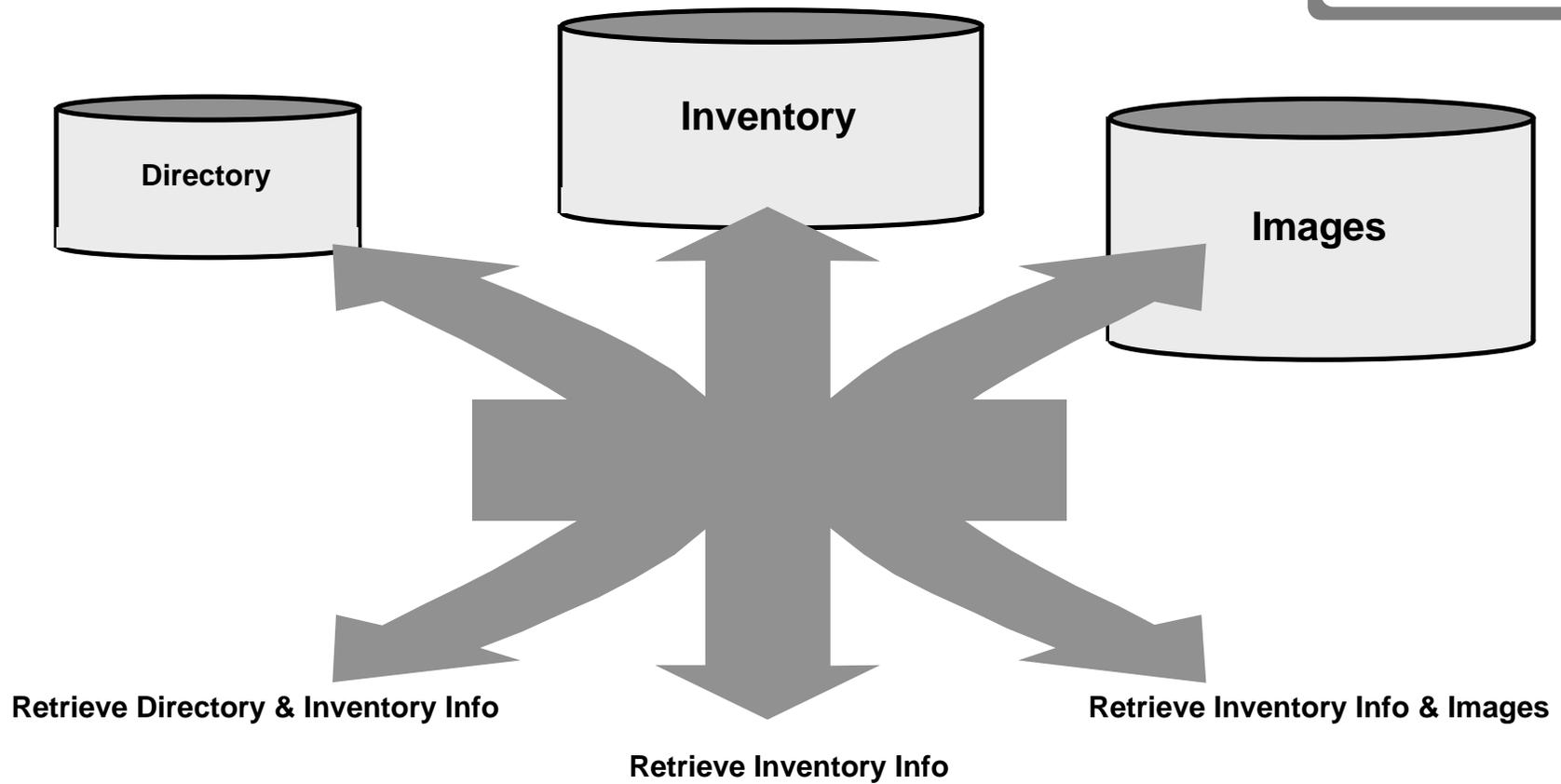
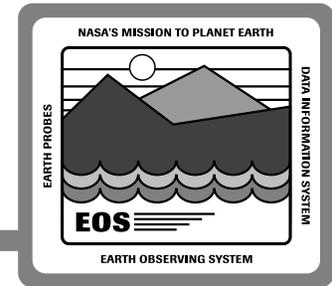
Apply Them As Universally As Possible

⇒ *Makes Things More Reusable and Interoperable*

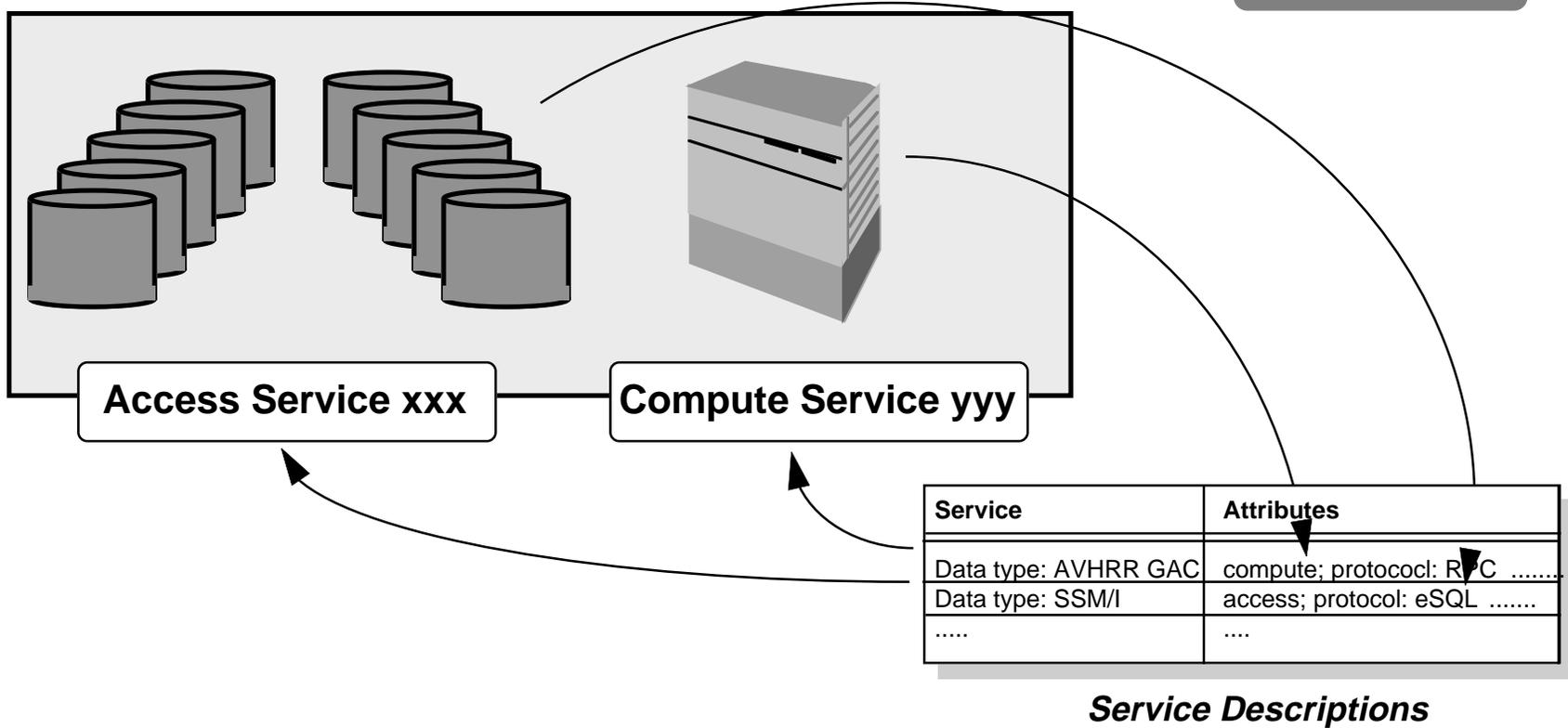
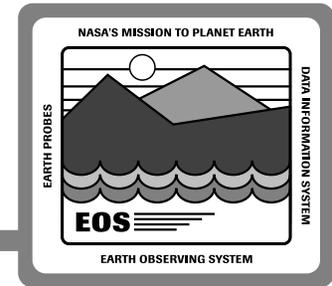
Employ, Adopt, and Contribute to Standards

⇒ *Leverage Significant External Development Efforts*

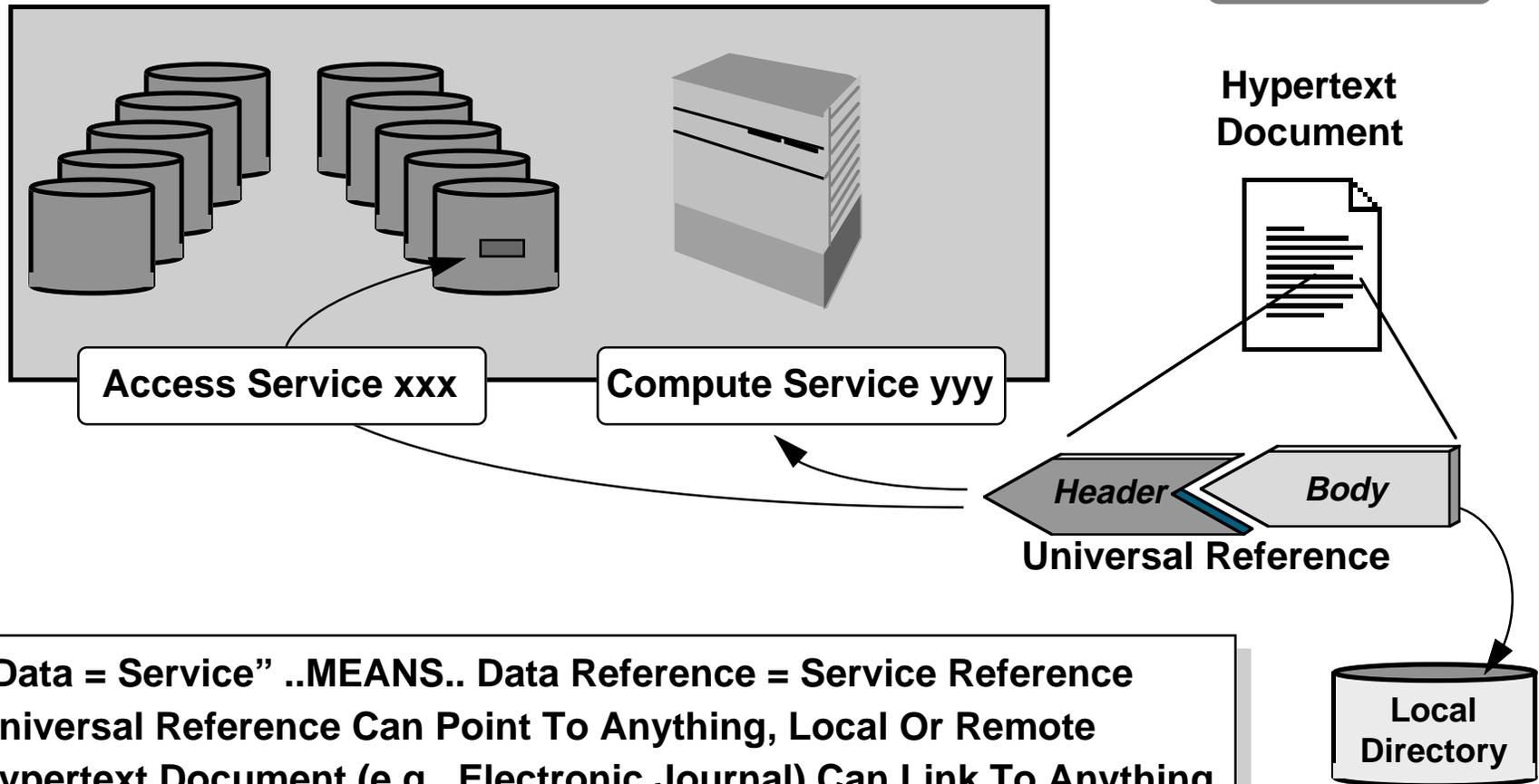
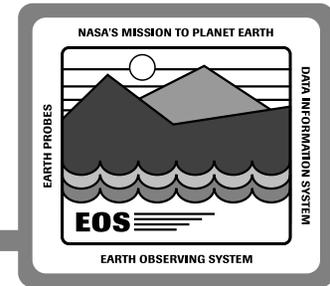
Meta-Data = Data



Data = Services

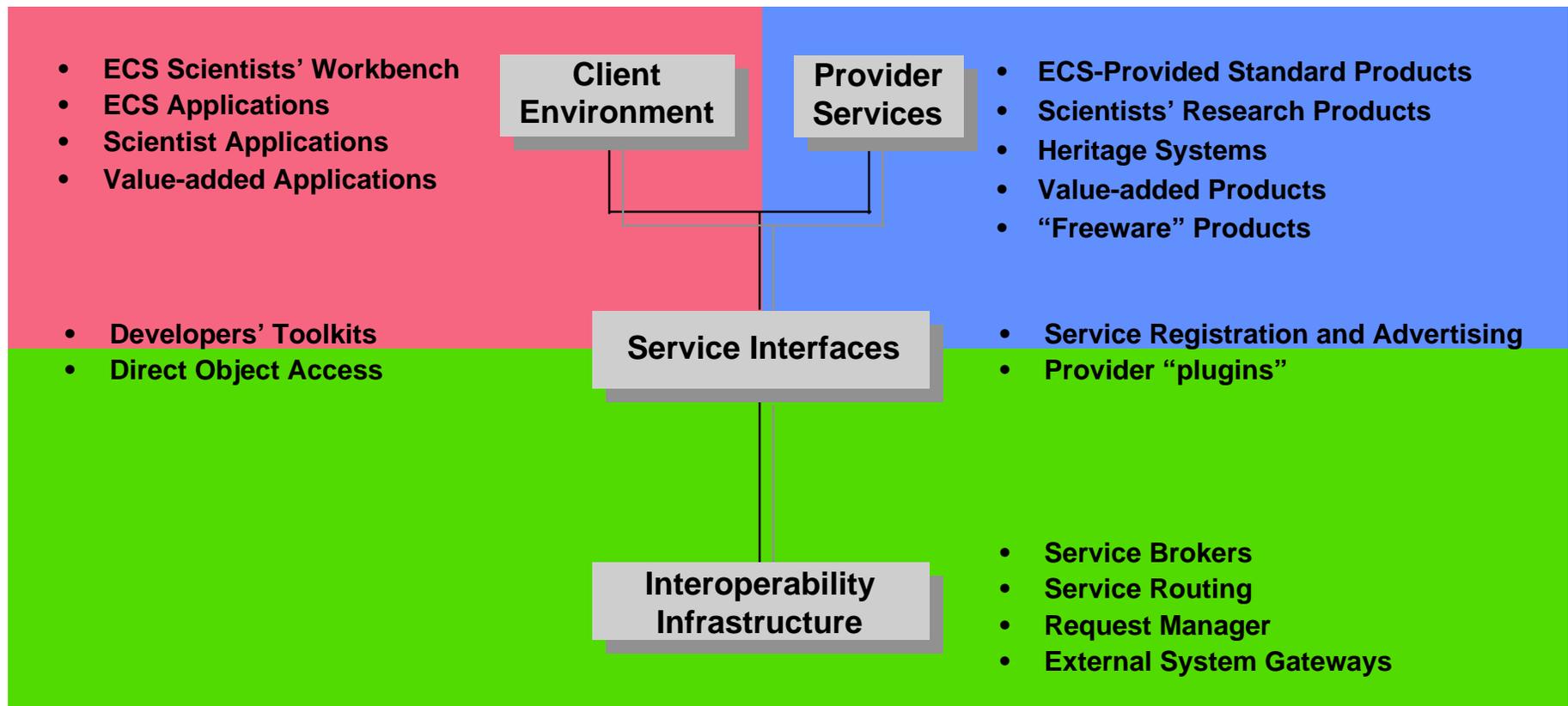
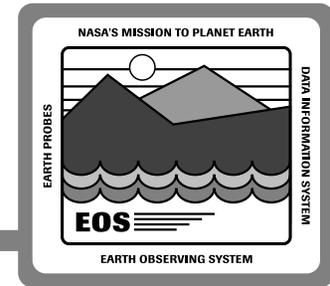


Universal Reference

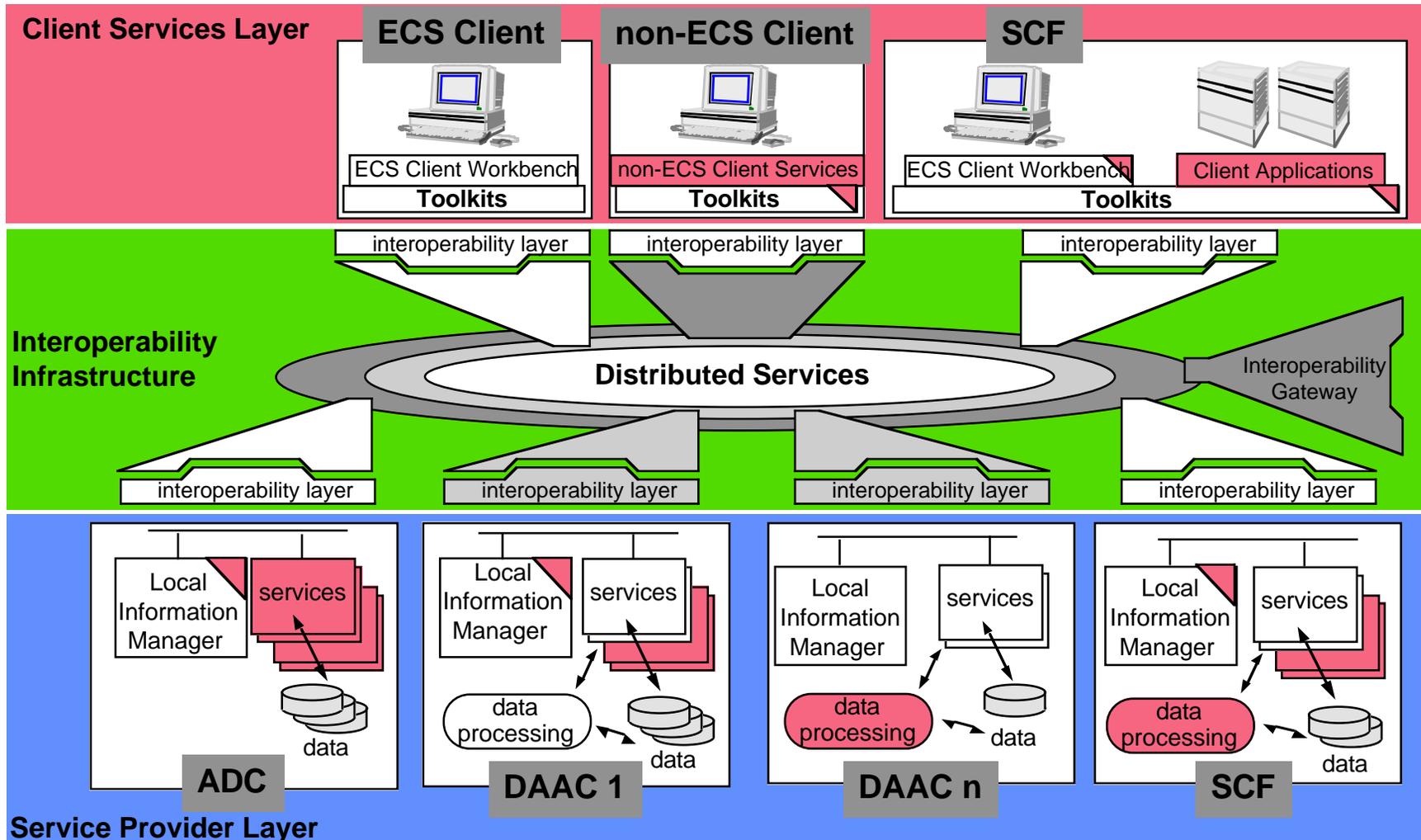
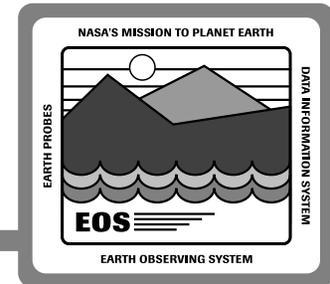


- **“Data = Service” ..MEANS.. Data Reference = Service Reference**
- **Universal Reference Can Point To Anything, Local Or Remote**
- **Hypertext Document (e.g., Electronic Journal) Can Link To Anything**

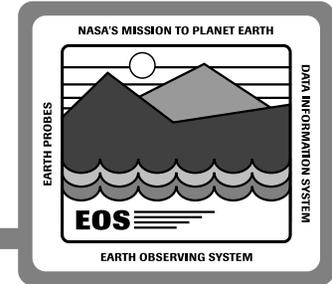
ECS User-Provider Model



ECS Conceptual Architecture



Conceptual Architecture Features



- **Layered Service architecture**
 - **Client, Interoperability, and Provider services**
 - **Services are location independent**
- **Autonomous, heterogeneous sites**
 - **Data service "providers"**
 - **Client "environments"**
- **Extensibility into heritage systems and value-added services**
- **ECS context**
 - **Interoperability infrastructure**
 - **Baseline provider components (DAACs)**
 - **Provider service support (SCFs)**
 - **Client environment support (ADCs, ODCs, IPs, SEDAC)**
 - **ECS interoperability gateway (heritage and value-added systems)**