

DSMS Information Systems Architecture (DISA) Overview

Laura Needels

April, 2006

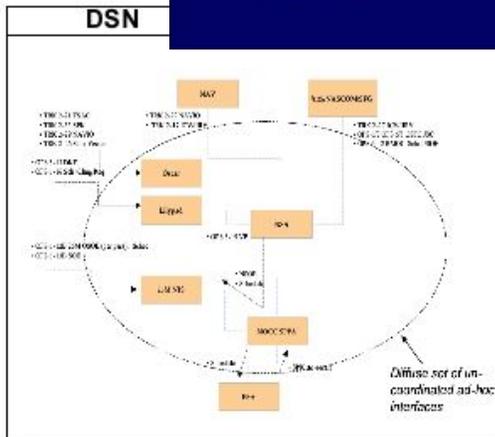
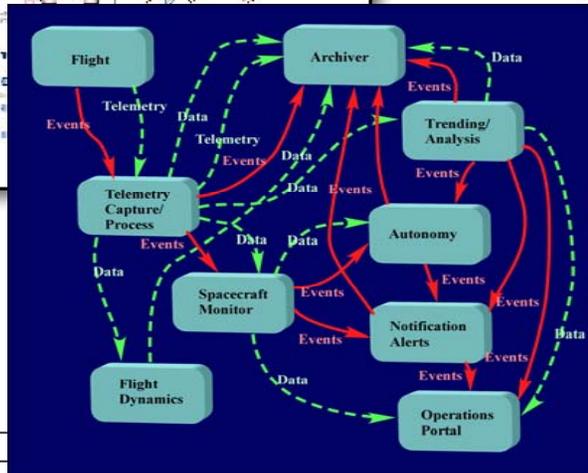
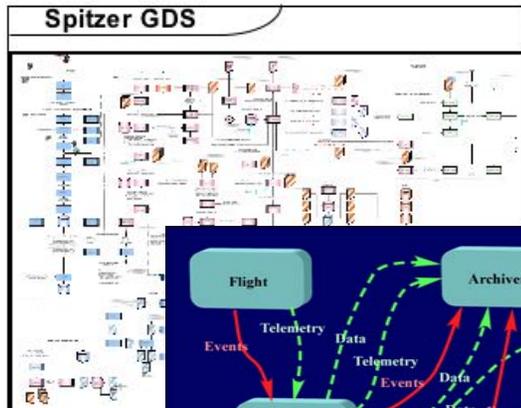
Overview

Why Are We Talking About Architecture? (1 of 5)

- AMMOS evolved in the 1980s – The age of the architecture is impeding our ability to add new capability and missions in a cost effective manner.
- The software architecture has a significant impact on
 - Ease of evolution
 - Adaptation/maintenance costs
 - Operability and deployability



Current Situation

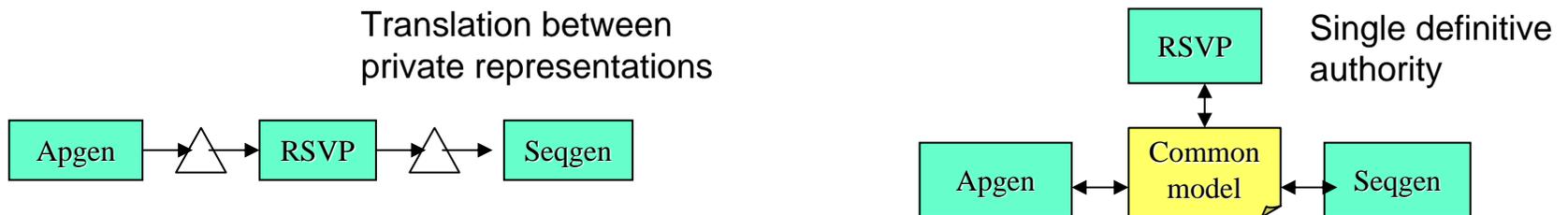


- **It Works!, but**
- High-cost to sustain/maintain capabilities, little to no budget for new functionality
- Redundant code-bases across instantiations, adaptation code is new
- Multiple systems/sub-systems affected by new/changing requirements (brittle environment)
- Lots of “glue ware” to bridge interfaces and fill functionality gaps
- Limited evolveability and scalability, multiple systems/sub-systems affected by change
- Limited integration, mostly point-to-point
- Point solutions due to a lack of an overarching “cohesive” architecture
- Limited “open” architecture
- Difficult to infuse mission changes back into baselines
- Platform and provider lock-in

Overview

Why Are We Talking About Architecture? (2 of 5)

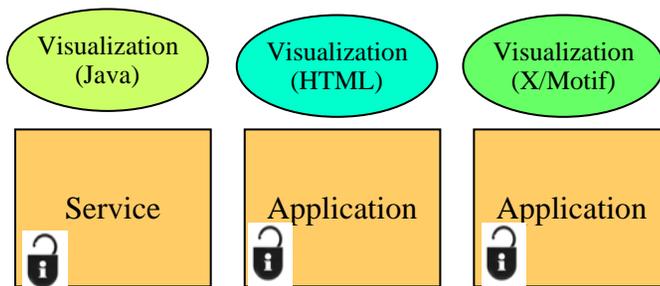
- Current system
 - Loosely federated applications
 - Domain knowledge, operational information, data, analysis, & visualization are embedded within the applications
 - Proprietary interfaces
 - Primarily via files w/application specific formats (some use standard headers)
 - Some application specific inter-process communications
- DSMS Architecture
 - Open up the architecture
 - Remove tight coupling between components to allow capabilities to evolve independently (SISs are the way we currently exchange info between components)
 - Make Information Exchange understandable
 - Move from private data representations to publicly understandable information



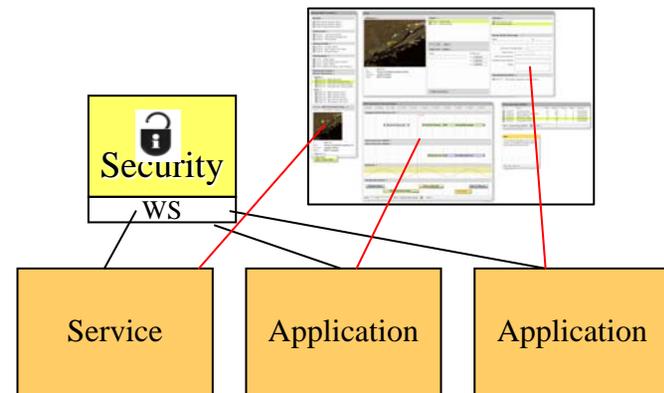
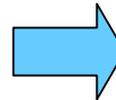
Overview

Why Are We Talking About Architecture? (3 of 5)

- DSMS Architecture
 - Use Common Solutions to common problems
 - Address common application needs with common infrastructure (security, visualization, etc.)



Non-interoperable approaches, difficult to change

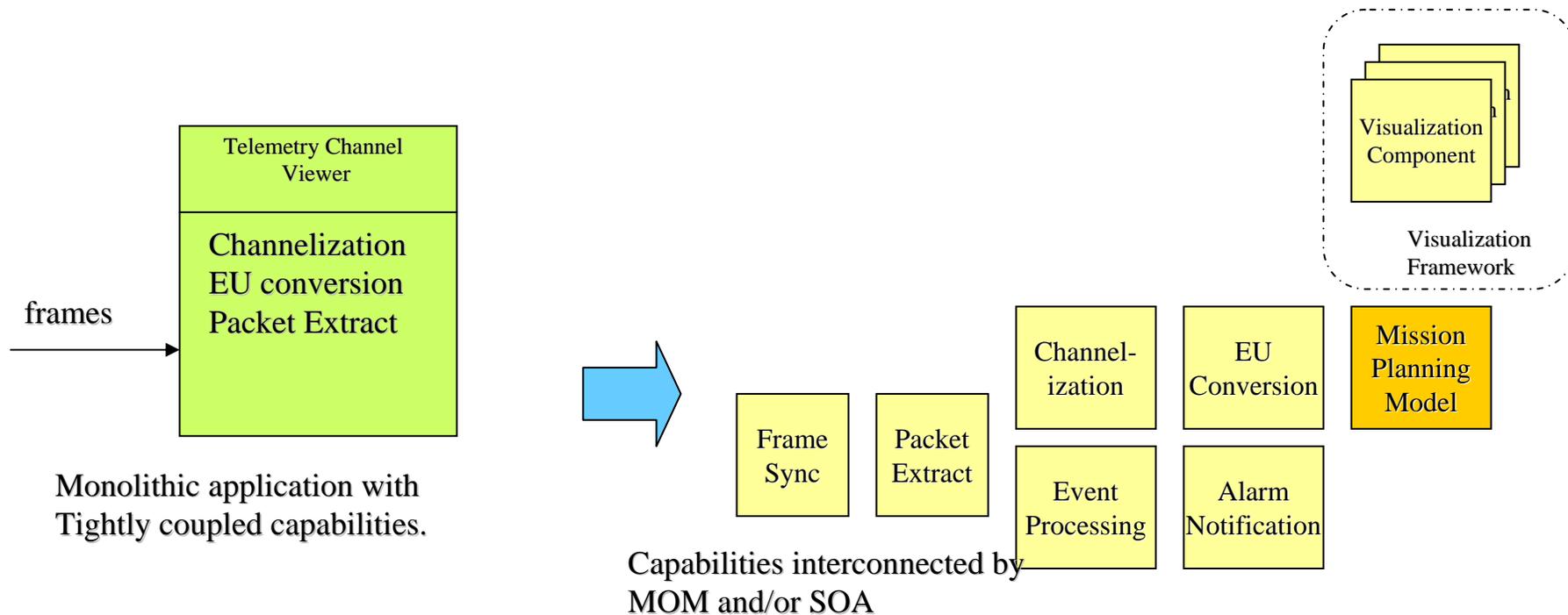


Interoperable by design. Better control & scope of change.

Overview

Why Are We Talking About Architecture? (4 of 5)

- DSMS Architecture
 - Composability
 - Provide sets of appropriately sized collections of interacting components and software services instead of monolithic applications

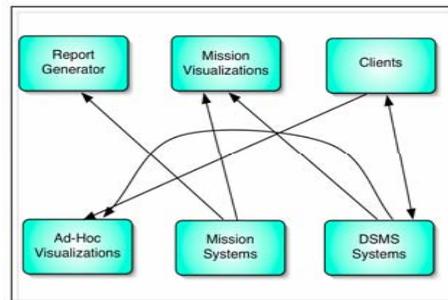


Overview

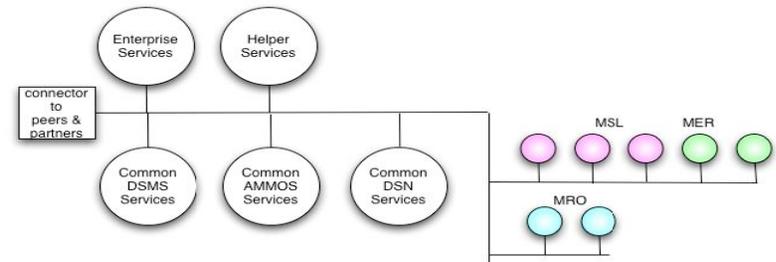
Why Are We Talking About Architecture? (5 of 5)

- AMMOS “New Architecture” does **NOT** mean that we throw out the current AMMOS
 - Some applications may require modifications
 - Some applications may have to be rewritten
 - Some new applications will be required
 - Leverages off DSMS Architecture work
 - Not a sudden change - Can be phased in as small steps which are not disruptive over time as funds become available

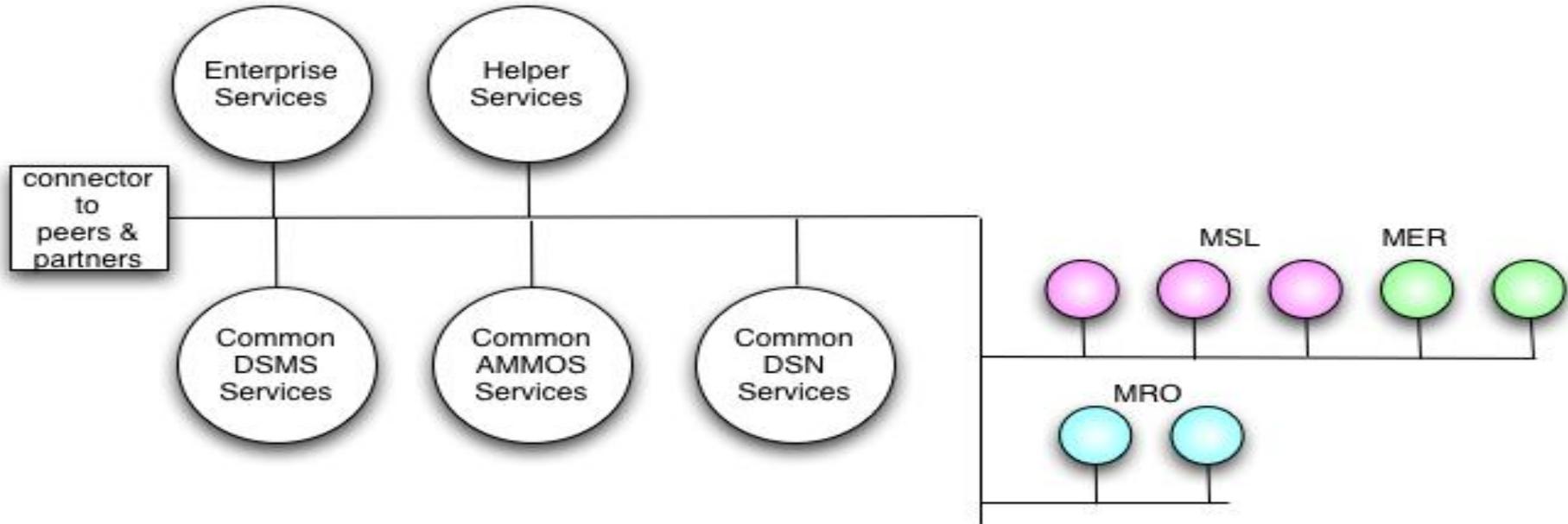
80s Architecture



Modern Architectures



“Our Vision”



- **Open up the architecture**
 - **Remove tight coupling between components to allow capabilities to evolve independently**
- **Make Information Exchange understandable**
 - **Move from private data representations to publicly understandable information**

Architecture **Vision** Statement

The AMMOS architecture is a collection of **inter-operable** network centric services and tools that **allows missions to choose a combination of common and mission specific tools and services**

Architectural Tenets: (abbreviated/translated)

1. Information is understandable and available to all authorized users

- Standardized definitions of data that all adhere to

2. All relevant capabilities available via Network Accessible Services

- Clear software capability definitions
- Industry-standard interfaces

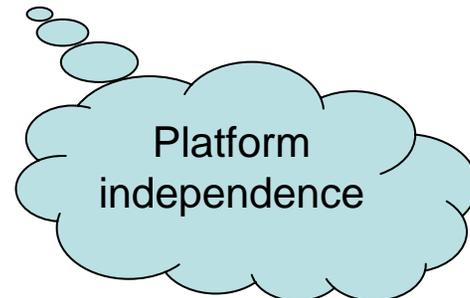
3. Composability

- Capabilities are built by interconnecting network accessible services (including COTS, GOTS).
- Client interfaces are decoupled from their underlying processing and data management functions. (Could choose to use the underlying processing capability but provide a different interface)

Architecture **Vision** Statement

Benefits To Your Project

- Easier to meet a spectrum of future evolving mission needs
 - reduced budgets
 - significant increases in U/L data rates & D/L data volumes
 - data accountability
 - ranges of automation
- Offers modular functionality that is easily adaptable
- Standardizes application interfaces and data
 - Missions can compose their MOS by choosing from a set of mission appropriate interoperable services and tools (including COTS and GOTS)
- Possible to have transparent evolution and use of the system
 - Isolate users from details of the underlying tools/services and how they are hosted and deployed
 - Possible to have multiple versions of service co-existing on the same network.



Achieving the Future State (1 of 2)

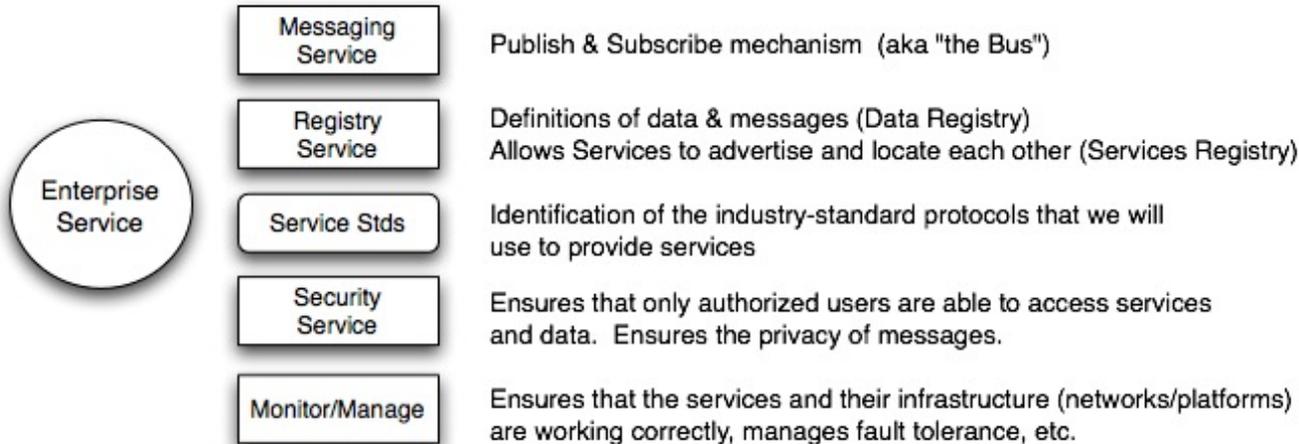
- Current DSMS Architecture has 3 Blocks of “core services”
- Block 1 (Accountability Core Service, Messaging, Registry)
 - Provides projects with the **ability to track an “idea” from planning through delivery and processing of data**
 - Messaging is a “utility” capability needed to support accountability and other aspects of the new architecture. Briefly, messaging is the ability to send and retrieve a message to an “information bus”.
 - Registry (data and service) is a “utility” capability needed to standardize data/services. Briefly, the registry is a “definition” of the data or service.

Achieving the Future State (2 of 2)

- Block 2/3 (Security, Common Visualization, Database, File/Collection, Workflow)
 - Security provides a “standard utility” to authenticate users
 - Applications **will not have to develop their own security infrastructure.**
 - Common visualization **standardizes user interfaces** by providing visualization utilities.
 - **Separate visualization from underlying processing capability**
 - Will **reduce duplicate storage of files.**
 - Defines standards for files and data processing, storage, access, and distribution

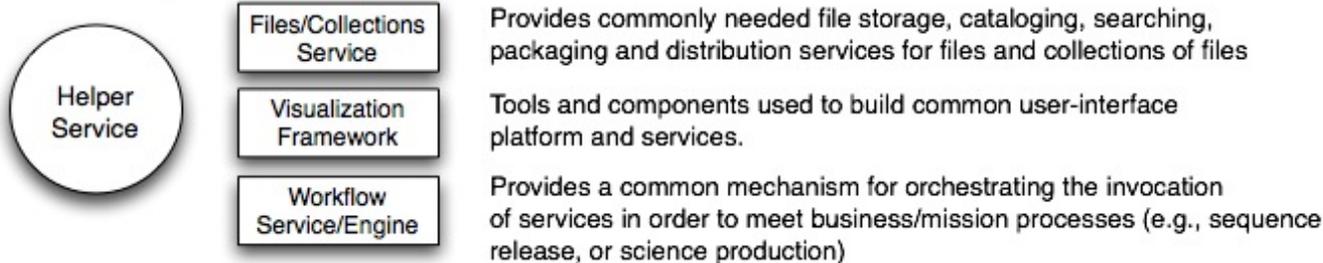
Service Types

"Enable" all other Services & their interactions



Provisioned by enterprise
Highly reliable
Primarily COTS
Industry standards & interfaces
DSMS configurations & patterns

"Augment the Capabilities" all other Services

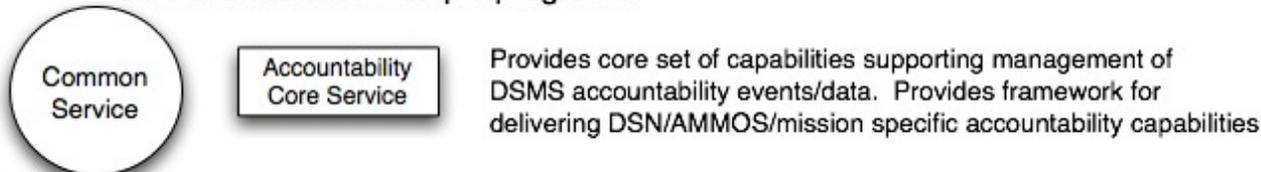


Primarily COTS but may have significantly tailoring.

Industry stds where possible

May be provisioned within a system or subsystem

Services useful to multiple programs



DSMS/AMMOS/DSN-specific

Space or community stds (where defined)

May be provisioned within a System or subsystem

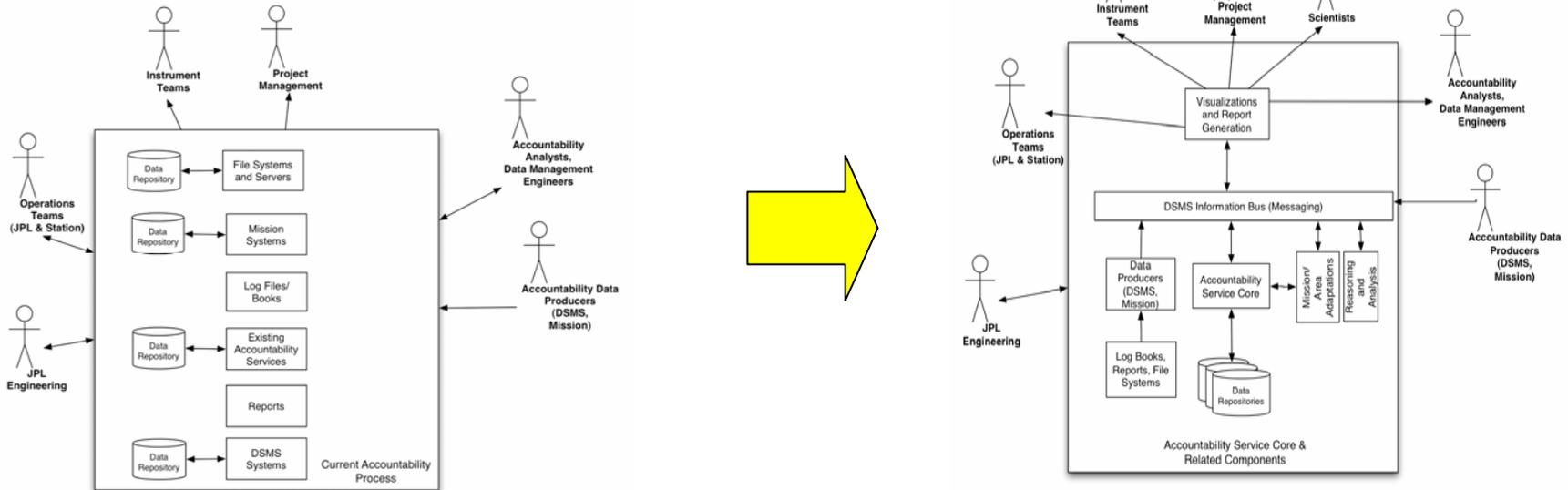
Block 1: Accountability, Messaging, Registry

Block 1: Enable Core Architecture

- Key enterprise services and information models enabling the architecture.
 - **Messaging Service (MOM)**
 - Provides an enterprise quality event architecture based on industry standard Message Oriented Middleware technique
 - Systems "publish" their information and data on a "bus" using a well defined set of messages and topics. Other systems (and/or users) are able to "subscribe" to received messages either reliably or best-effort delivery.
 - Systems do not need explicit knowledge of the other systems.
 - **Registry Service (data & service)**
 - **Data Registry** provides a DSMS-wide managed repository containing our key data definitions, message schemas, and information models which support the interoperability between our systems.
 - **Service Registry** provides a managed repository containing the sufficient information for DSMS systems to advertise service capabilities, discover and bind to appropriate service providers.
 - **DSMS Messaging Standards**
- Use these services and models to collaboratively build an **accountability core service**
 - Provides the core framework for supporting common accountability activities for DSN, AMMOS, and missions.
 - Provides the ability to manage accountability data (including storage, query/retrieval, and archival)
 - Provides a "plug-in" framework supporting system and mission-specific accounting needs.

Accountability

- Enable greater visibility into the key processes and flows.
 - DSN - frame, packet, pass, etc.
 - AMMOS - observation request -> product delivered to scientists.

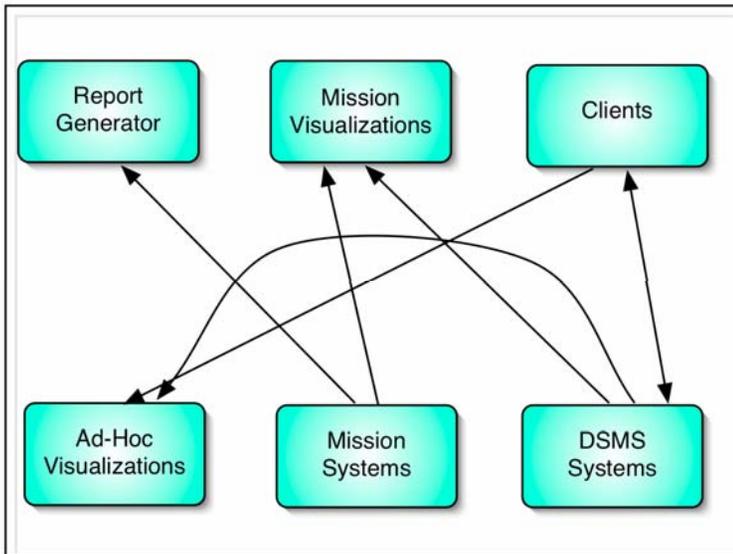


- **Data is widely scattered and cannot be easily accessed or understood.**
- **Correlation is localized and often done by humans (usually not made available to others)**
- **No concept of an end-to-end accountability that crosses system/subsystem boundaries.**
- **Accountability data is readily available on the messaging bus and defined by the data registry.**
- **Correlation is done by agents (mission or multi-mission) which publish their results on the bus (available)**
- **Supports the both local and end-to-end accountability.**

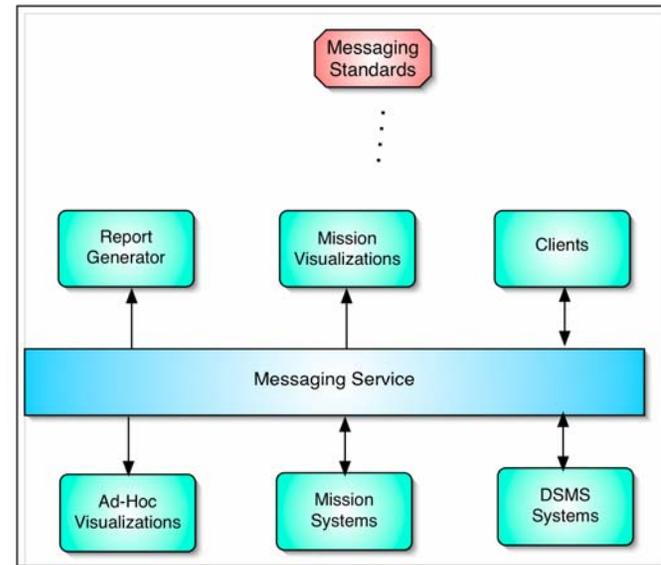
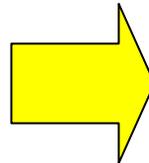
Status: Pilots for MRO and MER, FY06: producing key event messages, working with DSN and AMMOS to define accountability requirements and put in FY06 POP call. DSN SRDR in April.

Messaging Description

- Messaging will improve the current approaches to data exchange, providing an organized, flexible, and reliable mechanism



- *Information producers and consumers are tightly coupled. Changing one often changes the other.*
- *Difficult to intercept flow of information to “take advantage of it” for other uses, or to add capabilities.*
- *Very difficult to monitor and test systems (different interfaces, data standards, no inherent test points)*



- *Information producers and consumers are decoupled.*
- *Information is easily accessible to any authorized user or system*
- *Messaging provides a natural point for monitoring or testing the system (e.g., publish/subscribe to msgs)*

Status: Messaging is being fielded within the DSCCs as part of SPS in Sept. Initial fielding in NOCC and AMMOS test string this summer.

Registry Description

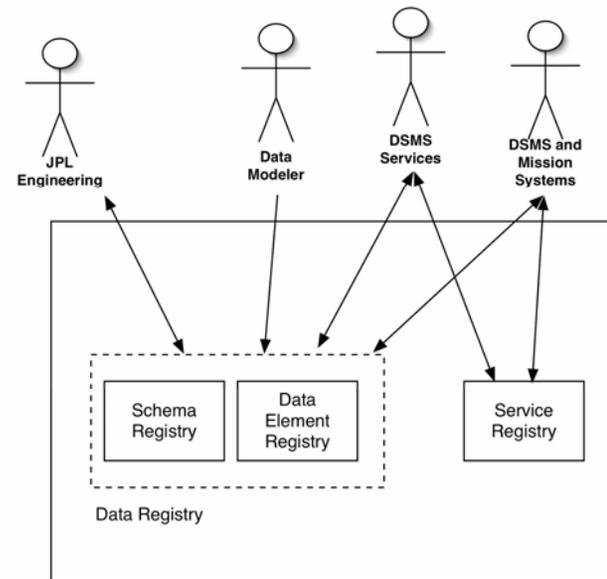
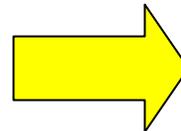
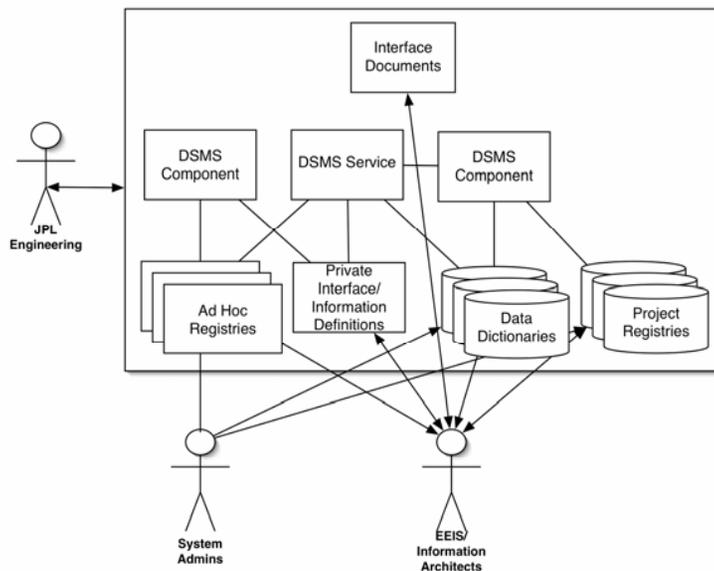
- Two Registries – Data Registry, Service Registry.
- The DSMS Registry Service provides a common approach to registration, storage, and retrieval of the metadata associated with different types of data entities or services on the network.
- The DSMS Data Registry will encourage the use of external, public information definitions and distribution of information in standard formats.
- The DSMS Service Registry will provide dynamic location of other services by service users/clients, increasing flexibility in service deployment and network topology.

Data

Service

Registry Description

- The DSMS Registry Service provides a common approach to registration, storage, and retrieval of the metadata associated with different types of data entities or services on the network.
- The *Data Registry Service* will encourage the use of external, public information definitions and distribution of information in standard formats.
- The *Service Registry Service* will provide dynamic location of other services by service users/clients, increasing flexibility in service deployment and network topology.



Status: Currently working with the JPL CIO on fielding approaches & with DSMS on how to make these the official repository of our interfaces, data definitions, messages, schema, etc.

Compliance

- Key aspects of the architecture are represented as “Compliance Areas”.
- Each has an associated compliance criteria which provides metrics needed for assessment.

#	Compliance Area	Status
1.0	Information & Data Arch.	In Review
2.0	Common Access Mechanisms	In Review
2.1	Service Interface	In Review
2.2	Messaging	In Review
3.0	User Interface	In Review
3.1	Common Look/Feel (style)	Draft
3.2	Rich Client	Draft
3.3	Zero Footprint Client	Draft
4.0	Security	
4.1	Authentication	In Review
4.2	Authorization	In Review
4.3	Confidentiality	Draft
5.0	System Management	In Review
6.0	Service Quality Reporting	TBD
7.0	Support Service	
7.1	File Storage	Draft
7.2	Data Storage	Draft
8.0	Cross Platform Support	Draft
9.0	Modularity	TBD
10.0	Reuse	TBD
11.0	Deployment	TBD

current

FY06 goal

Information/Data Compliance Criteria

Level 0 (private)

- Format and meaning of data exchanged between this application and external entities is not publicly defined.

Level 1 (documented)

- Format and meaning of data exchanged between this application and external entities is defined in the relevant DSMS document (e.g., 820-013, 820-014, 820-019, etc).

Level 2 (publicly described)

- Format and meaning of data exchanged between this application and external entities is registered in the DSMS registry.
- Including: messaging formats, service interface definitions, & data element definitions

Level 3 (mediateable)

- Application provides a mechanism to mediate/translate between the application's unique data and the appropriate community of interest information/data models.

Level 4 (transition)

- Application exchanges information with external entities using both the appropriate community of interest information/data and legacy (publicly described) models.

Level 5 (fully compliant)

- Application exchanges information with external entities using only the appropriate community of interest information/data models.

Draft Messaging Compliance

- **Level 0 (private)**
 - Application delivers and uses its own messaging approaches.
 - Including: IP messaging (UDP, MC, etc.), application/system specific MOM, etc.
- **Level 1 (common use)**
 - Application uses DSMS provided messaging service on a not-to-interfere basis and registers/uses a private namespace. Application does NOT comply with or use DISA namespace, topic or messaging standards and definitions
 - E.g., application's name spaces and definitions do not collide with DISA's.
- **Level 2 (initial integration)**
 - Application uses DSMS provided messaging service, adheres to DISA namespace standards, and registers its topics within the DISA namespace in the DISA registry, but does not comply with or use DISA messaging standards.
- **Level 3 () Reserved**
- **Level 4 (fully compliant)**
 - Application uses DSMS provided messaging service, and adheres to all DISA namespace, topic and message definition and usage standards.

More Info?

- DSMS Software Architecture docushare site:
 - Overall architecture & reviews
 - <https://ind-lib.jpl.nasa.gov/docushare/dsweb/View/Collection-4129>
 - Architecture Governance Process (proposed)
 - <https://ind-lib.jpl.nasa.gov/docushare/dsweb/View/Collection-4223>
 - Capabilities:
 - Messaging
 - <https://ind-lib.jpl.nasa.gov/docushare/dsweb/View/Collection-2316>
 - Registry
 - <https://ind-lib.jpl.nasa.gov/docushare/dsweb/View/Collection-2091>
 - Accountability
 - <https://ind-lib.jpl.nasa.gov/docushare/dsweb/View/Collection-1568>
- DSMS Software Architecture Working group (open to all)
 - Meetings: Tuesdays 1000-1115 in 301-271
 - Lyris email list: join-dswag@list.jpl.nasa.gov

Backup

Architecture Infusion Tasks

- DSN
 - Infusion of messaging into SPS task (FY06) will be exposed in FY07 as a DSMS service hosted on enterprise hardware.
 - Working with DSMS SE on new starts: [McVittie]
 - **Telemetry string upgrade/replacement (Data Capture and Delivery)**
 - **Service Management upgrades.**
 - **Service Scheduling System (RFP)**
 - Generating requirements on DSN Accountability (SRDR in April) [DeMore]
 - **Identifying accountable artifacts and sources.**
 - FY06 POP initiative for DSN Accountability, DC&D, etc.
- AMMOS
 - Study task funded by MGSS to look at how to infuse the architecture into AMMOS. [Needels]
 - **Identifies how architecture will support AMMOS strategic initiatives**
 - **Establish some part of accountability in each of the MGSS Program elements (MDAS, MPS, IOS, NMD)**
 - Working with MGSS SE on new starts: [McVittie/DeMore]
 - **MCS (aka AGDS, Chill, etc)**
 - Generating requirements on End-to-End Accountability [F. Hammer]
 - FY06 POP initiative for End-to-End Accountability

GDS System Engineering

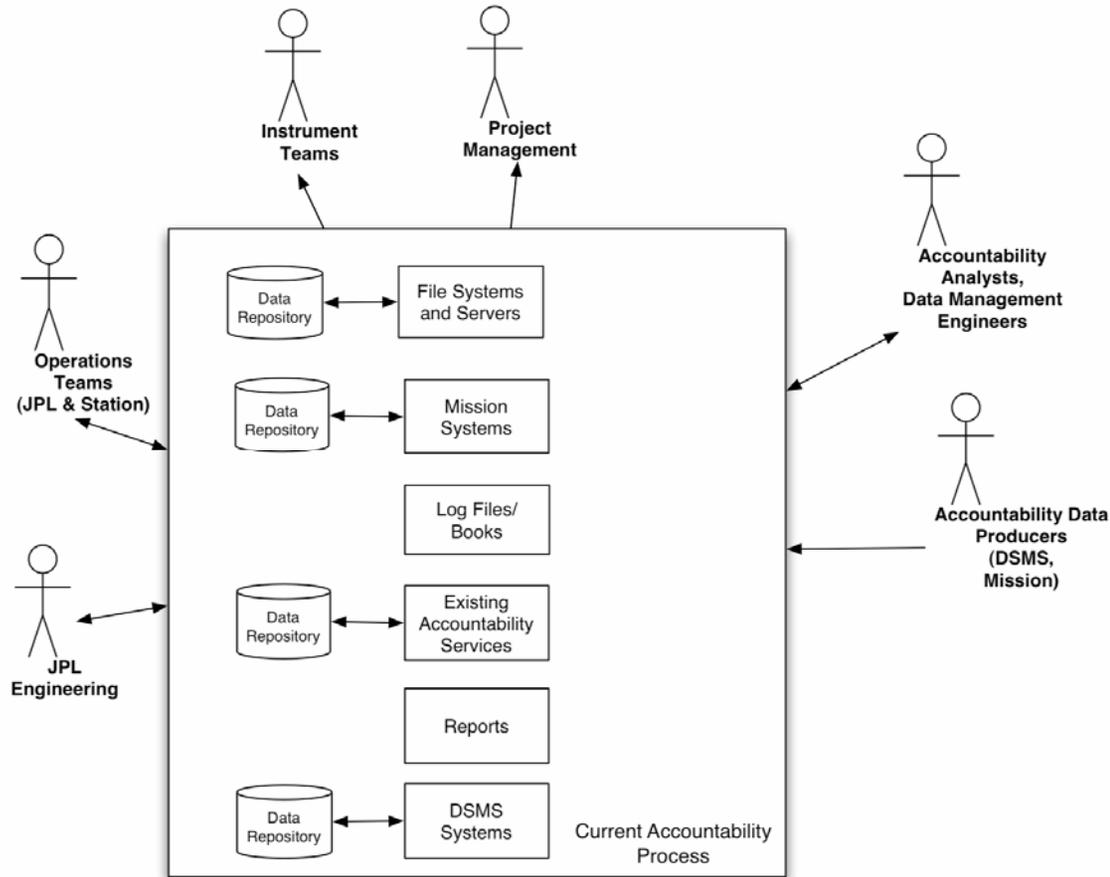
Short Term & Long Term Changes

- Mission choices in how we compose a GDS
 - Plugging-in alternate components, support for product lines/families.
 - Investigate COTS tools & alternatives to GDS capabilities -- what's out there?
 - Alternate approaches to charging missions for capabilities.
- Process of integrating and testing a GDS will be different
 - Message bus as a key integration point.
 - Definition of the messages that need to be produced/consumed to exchange information between systems as well as provide basic control.
 - How interfaces will be defined and discovered
 - Some SISs and ICDs will become registry items.
 - New approaches to testing and validating the behavior of the GDS.
- Long term changes
 - Will be able to upgrade different pieces of the GDS at different times.
 - Easier to have different versions of developing software on the system at a time. (Think ATLO).
 - Software should be more consistent.
 - Have potential to have greater choices in software components for pieces in the system.

Current Accountability Capabilities

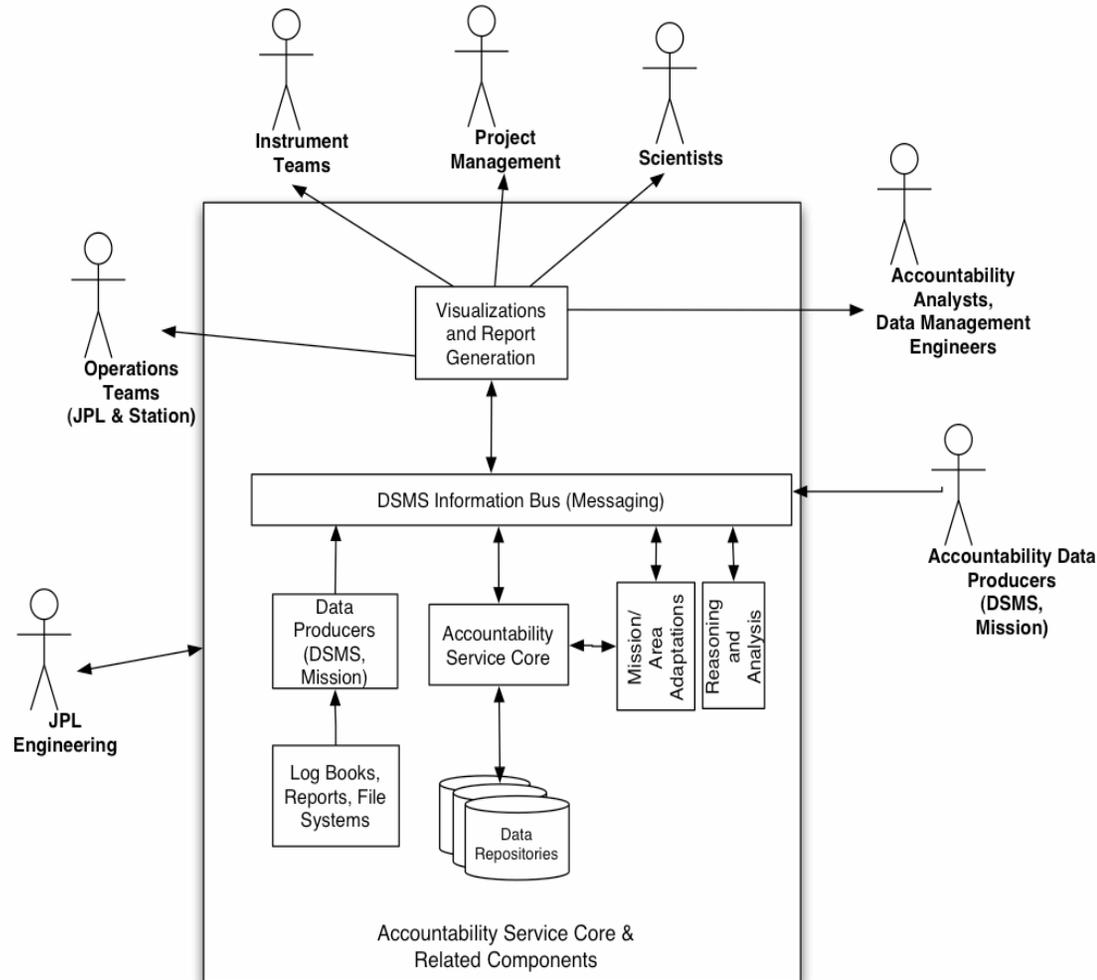
- Current approaches
 - Log & report analysis (DMT, instrument teams, accountability analysts...)
 - Files & spreadsheets (All missions!)
 - DSN/AMMOS subsystems (DC&D, TTC&DM, MIPL..).
 - Existing AS
 - Multi-mission (e.g., Frame DAS)
 - Project developed (e.g., MRO eeDAT)
 - Proprietary interfaces.

- Issues:
 - Complex interactions; can't be diagrammed.
 - Must learn and integrate with many mechanisms.
 - No consistency in information definition.
 - Not enough automation; human analysis in the loop.
 - Not end-to-end.
 - Hidden & private information.

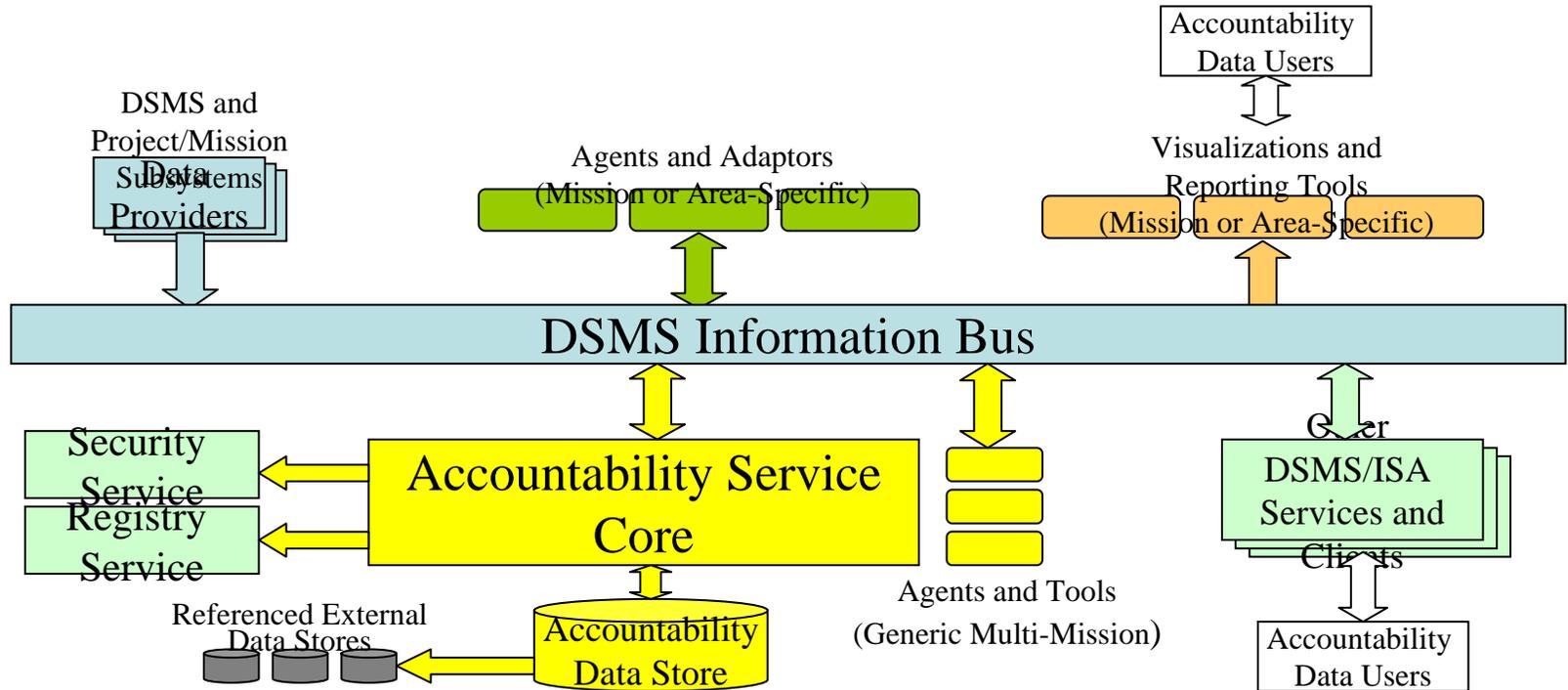


Proposed Accountability Capabilities

- Typical Operations:
 - Information producers: publish to Info Bus,
 - Information consumers: subscribe, correlate, publish back to Bus,
 - Core service stores data,
 - Information consumers: provide visualizations and reports,
 - Information consumers: generate events to trigger mail alerts etc.
- Based upon a common approach:
 - Data externally defined and publicly available.
 - Interfaces based on messages.
 - Fewer information sources for operators, analysts, management.
- Customization layered on top of basic capabilities.
- Interactions and data flows simplified.



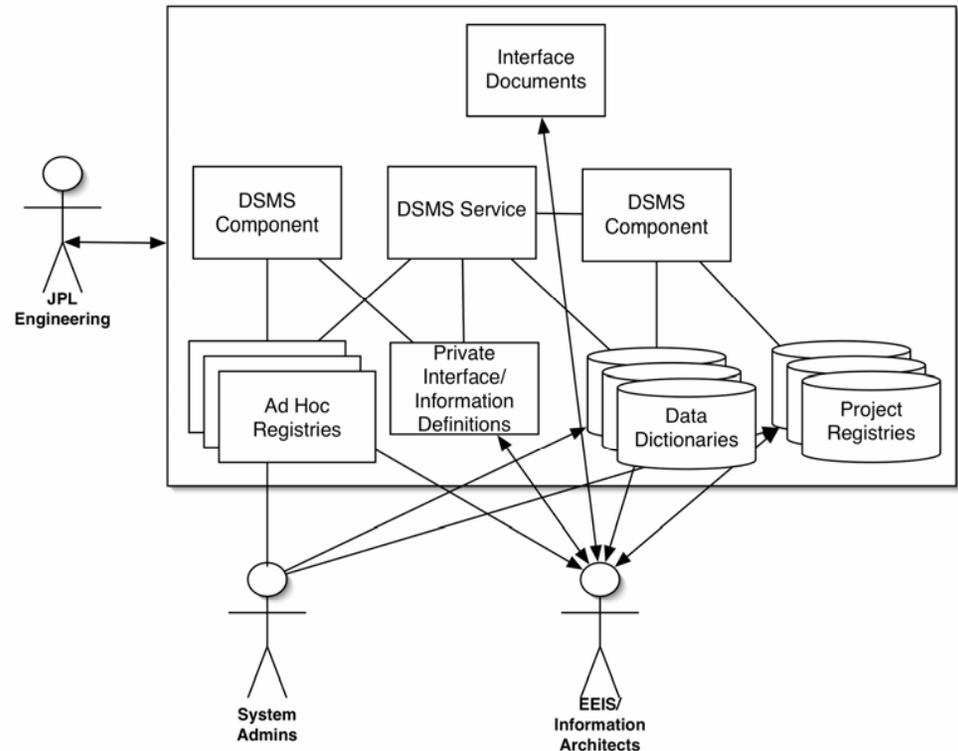
Accountability Description



- Infrastructure will support tracking of processes and artifacts such as:
 - **Planning, production, and delivery of science data products and telemetry files.**
 - **Production and delivery of engineering data products.**
 - **Reports of transmitted frames and analysis of data gaps.**
 - **Uplink of spacecraft command/sequence files.**
 - **Progress of planned spacecraft activities.**
 - **Logging of spacecraft or ground events.**
 - **Provision and performance of DSMS service instances.**

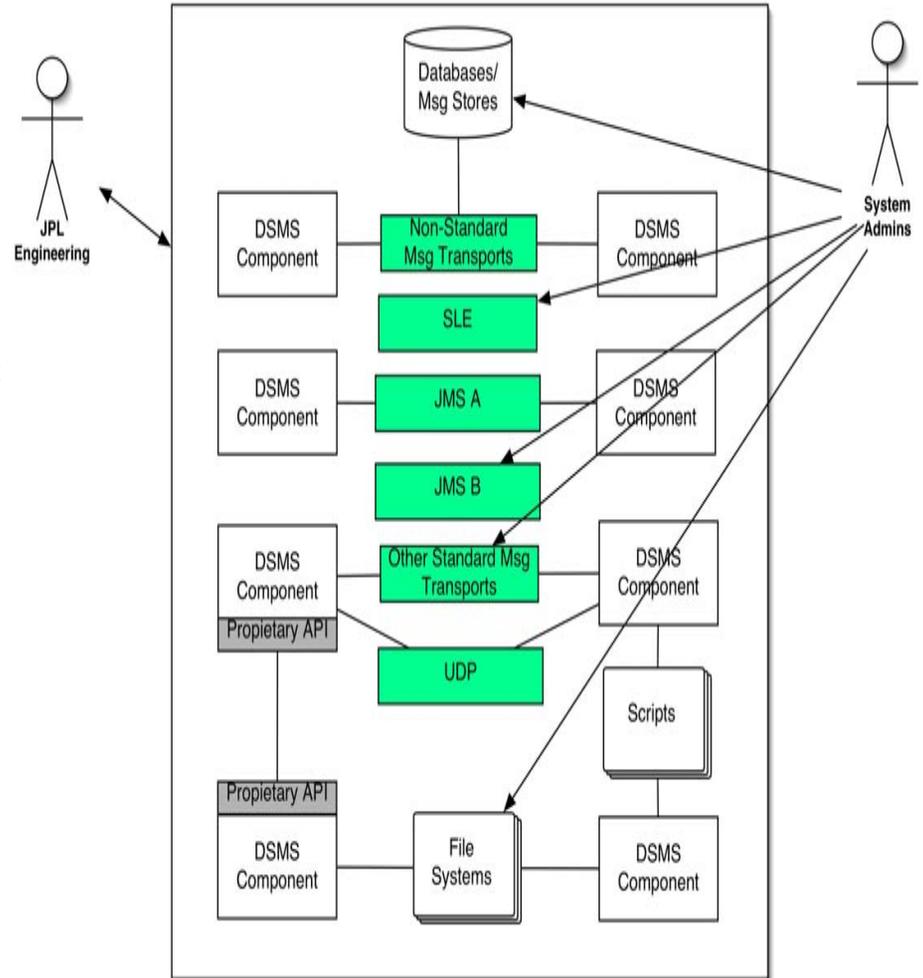
Current Registry Capabilities

- Current approaches:
 - **API specifications defined in static interface documents**
 - **Services bound by configuration files**
 - **Repositories duplicated and are project specific (silo'd)**
 - **There are institutional and isolated project data dictionaries available (the Planetary Data System dictionary, for instance) but their use by DSMS applications is limited**
- Issues:
 - **Manual configuration of services (relocation, balancing difficult)**
 - **No consistency in information definition.**
 - Information not available in usable and understandable electronic form
 - Often information cannot be verified at runtime against data definitions
 - **Different data referred to with similar terms, no data definition. federation**
 - **No clear source for authoritative definitions of data, structures, interfaces etc.**
 - **No support for versioning of multiple iterations**



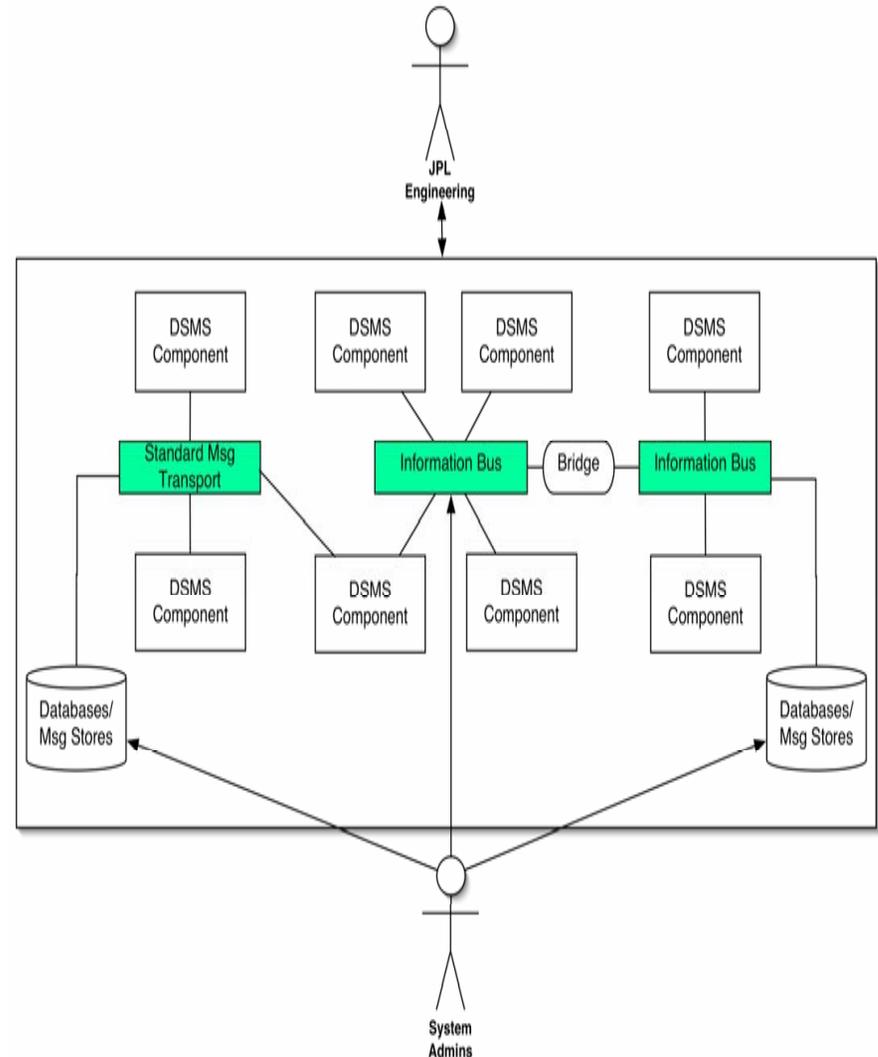
Current Messaging Capabilities

- Messaging
 - Space Link Extension (SLE)
 - CCSDS Asynchronous Messaging Service (AMS)
 - MON2
 - RNS protocol
 - Custom Encode UDP
- Use of Java Messaging Services (JMS) is on the rise at JPL
 - **Multiple implementations with independent content /organization (non-interoperable)**
 - No accepted approach for sharing between JMS instances (stovepipes)
 - **Significant differences in how systems are configured (security, topics, etc.)**
 - Developers and integrators need to be aware of different providers
- Issues:
 - **Multiple mechanisms - must learn and integrate with**
 - **Definition of information is the responsibility of system, no standards established**
 - **Admins must configure and manage multiple diverse systems**
 - **Information available only within slightly extended stovepipes**
 - **Integrators/developers must know about multiple messaging technologies**



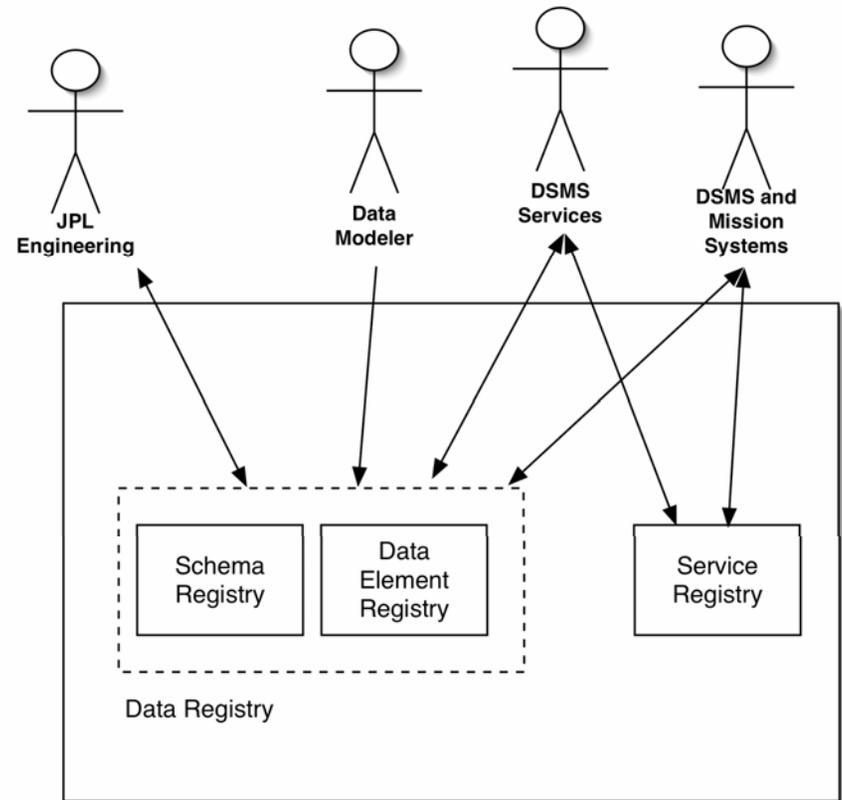
Proposed Messaging Capabilities

- Standardized Information Flows
 - Messages are externally defined around common messaging standard and registered in publicly accessible repository
- Decoupled
 - Components not required to know about each other
 - Easy non-intrusive integration of new components, for capability augmentation, test, monitoring, workflow etc.
- Common Approach
 - System administration unified
 - Messages transmitted over common open mechanism
 - Knowledge of fewer interfaces required by developers, integrators
 - Replaces multiple implementations with common functionality
 - Common security approach for information flows
- Delivery of messages is guaranteed
- Bridging between messaging instances is addressed



Proposed Registry Capabilities

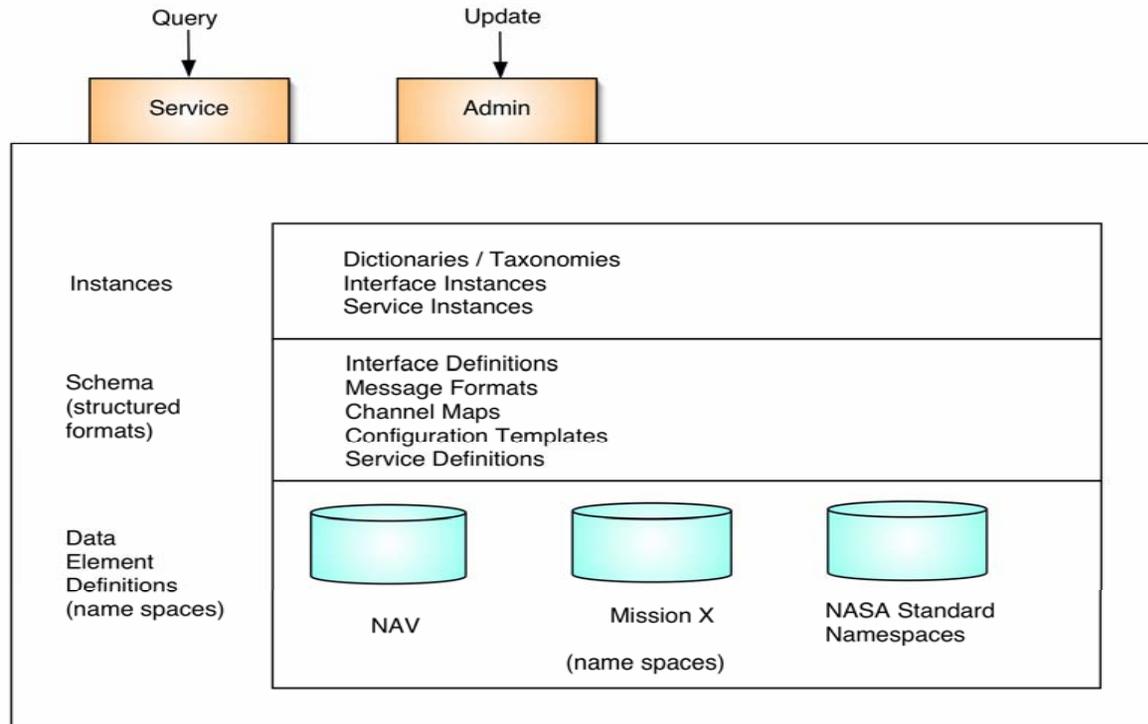
- Typical Operations:
 - **Data Modelers define data element definitions, and schemas and store in Registry**
 - **Definitions and schemas available at build-time for Engineers**
 - **Definitions and schemas available at run-time for DSMS and Mission systems (dynamic)**
 - **Services are discovered by DSMS and Mission systems**
- Use of a common Registry:
 - **Data externally defined once and publicly available to all.**
 - **Fewer data repositories for operators, modelers and engineers**
- Software systems can dynamically validate data against schemas
- Public data definitions well understood, accessible and reused.
- Namespaces defined in a federated model avoids collisions and enables richer data relationships.



Registry Description

- The DSMS Registry Service provides a common approach to registration, storage, and retrieval of the metadata associated with different types of data entities or services on the network.
 - **Data element definitions, dictionaries, message/channel formats, file schema, etc.**
- The DSMS Registry Service will encourage the use of external, public information definitions and distribution of information in standard formats.
- The DSMS Registry Service will provide dynamic location of other services by service users/clients, increasing flexibility in service deployment and network topology.

Registry Description



- The DSMS Registry Service supports (for example):
 - Definition of data elements (type, use, units, validation criteria etc.)
 - Definition of messages schemas to describe messages on the Information Bus
 - Location/binding information for services
 - Interfaces for accessing the service