



# NASA AMMOS Architecture Updates

*Brian Giovannoni*

*Jeff Estefan*

*Jet Propulsion Laboratory, California Institute of Technology*



# Message

- Transition Plan
  - Task Plans with Rec / Dels
  - Integrated schedule
- Plan for how the architecture will be instantiated
- 6 Year impl plan showing how tasks deliver key capabilities of the architecture
- SE schedule to address work to go
- Identified business opportunities



# Agenda

- AMMOS architecture progress to date
  - Context & Progress
  - Firming up architectural elements
    - System services
  - Major architecture decisions
- AMMOS architecture highlights
- Take away



## Take Away

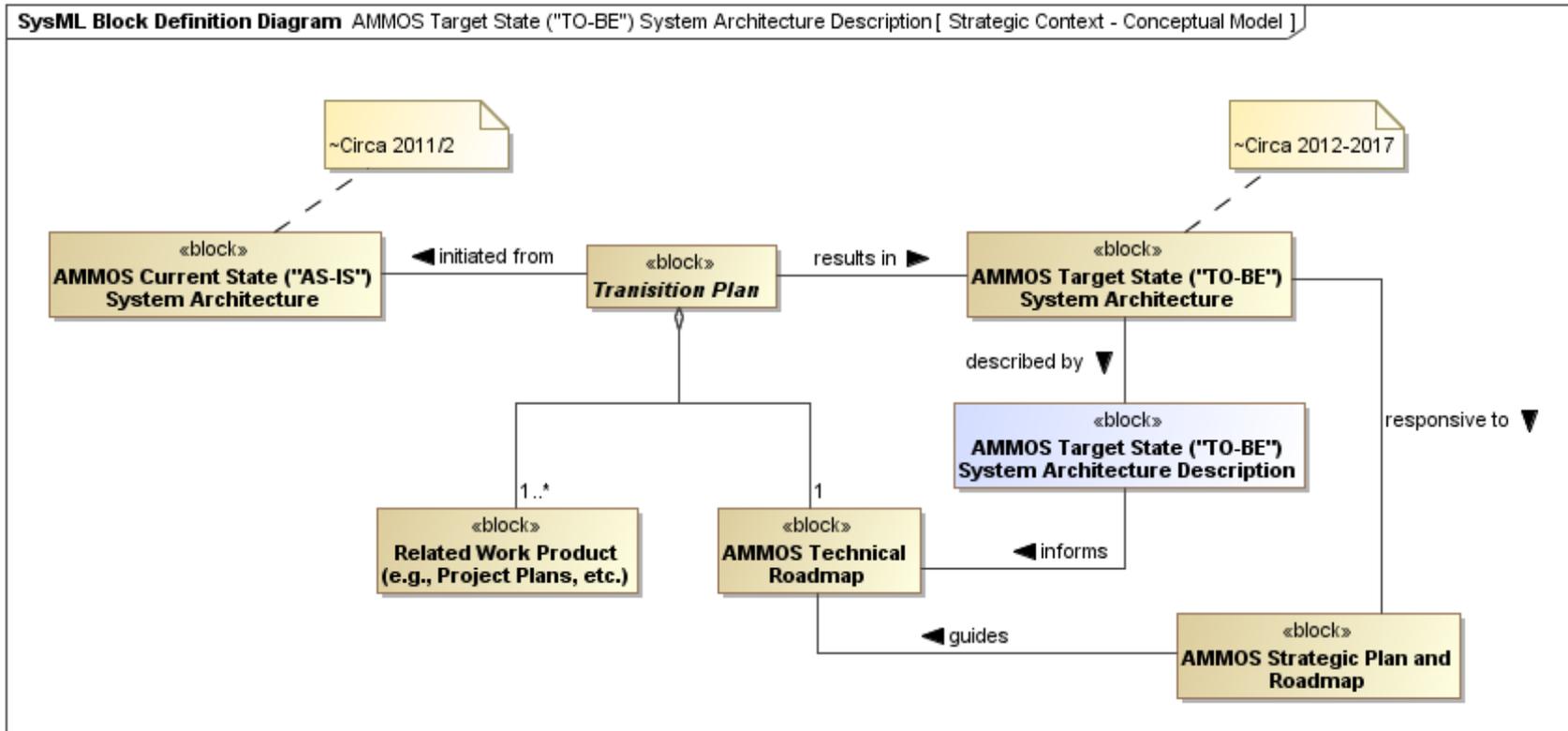
- Commonly managed application platforms independent of applications
  - Reduces cost to deploy and maintain
- Common and scalable compute platform
  - Reduces H/W cost and improves system flexibility
  - Reduces risk (test as you fly fly as you test )
- Standardizing on Common Information Exchange / Definitive source of shared information
  - Reduces system integration cost and complexity by standardizing on a canonical information model and associated shared services
  - Improves data and product accountability
- Adopting industry technology standards and support the development of CCSDS information standards
- Operations modernization allows missions to standardize planning execution & reconciliation operational processes



# AMMOS ARCHITECTURE – PROGRESS TO DATE



# Strategic Context (Conceptual Model)



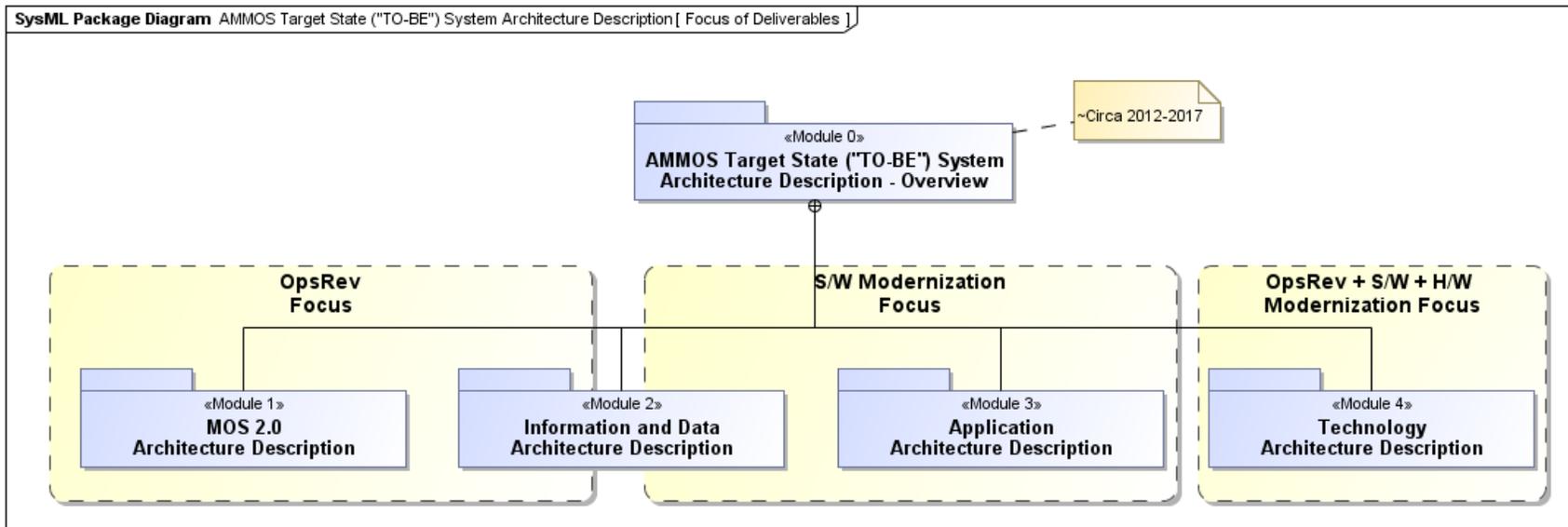
AsIs – Complete

ToBe – In draft / under review

Transition Plan – In development



# Progress: Architecture Deliverables



MOS 2.0 AD – In review

Information & Data AD – Planned

Application Architecture Description – In review

Technology Architecture Description – In review

Based on enterprise  
architecture approach ( TOGAF )



# Purpose & Scope of Deliverables (1/5)



«Module 0»

## AMMOS Target State ("TO-BE") System Architecture Description - Overview

- Purpose:
  - Describe overview of overall architectural approach, architecture deliverables, and strategic context
- Scope:
  - Top level document with links to other architecture description modules that collectively provide complete system architecture description of AMMOS Target State ("TO-BE") architecture



# Purpose & Scope of Deliverables (2/5)



«Module 1»  
**MOS 2.0**  
**Architecture Description**

- Purpose:
  - Describe system level overview of MOS 2.0 mission services and associated products
- Scope:
  - High level description of MOS 2.0 closed-loop control concept, mission services and their characteristics, product deliverables (e.g., interfaces, agreements, processes, training, and V&V), process layer security



# Purpose & Scope of Deliverables (3/5)



«Module 2»  
**Information and Data  
Architecture Description**

- Purpose:
  - Describe information and data models to be used in OpsRev and S/W modernization efforts
- Scope:
  - All aspects of information and data modeling (from ontology level to data schema to data layer security) centered on concept of Timeline, where Timeline is primary information object to be used in AMMOS-based Mission Ops of the future

Timeline is being introduced as as  
A potential CCSDS standard



# Purpose & Scope of Deliverables (4/5)



«Module 3»  
**Application  
Architecture Description**

- Purpose:
  - Describe architectural patterns for use in architecting individual AMMOS functional software applications as well as their integration across functional areas
- Scope:
  - Recommended application architecture tiers and layering, application integration strategy, common S/W services & characteristics, packaging & deployment, application layer security



# Purpose & Scope of Deliverables (5/5)

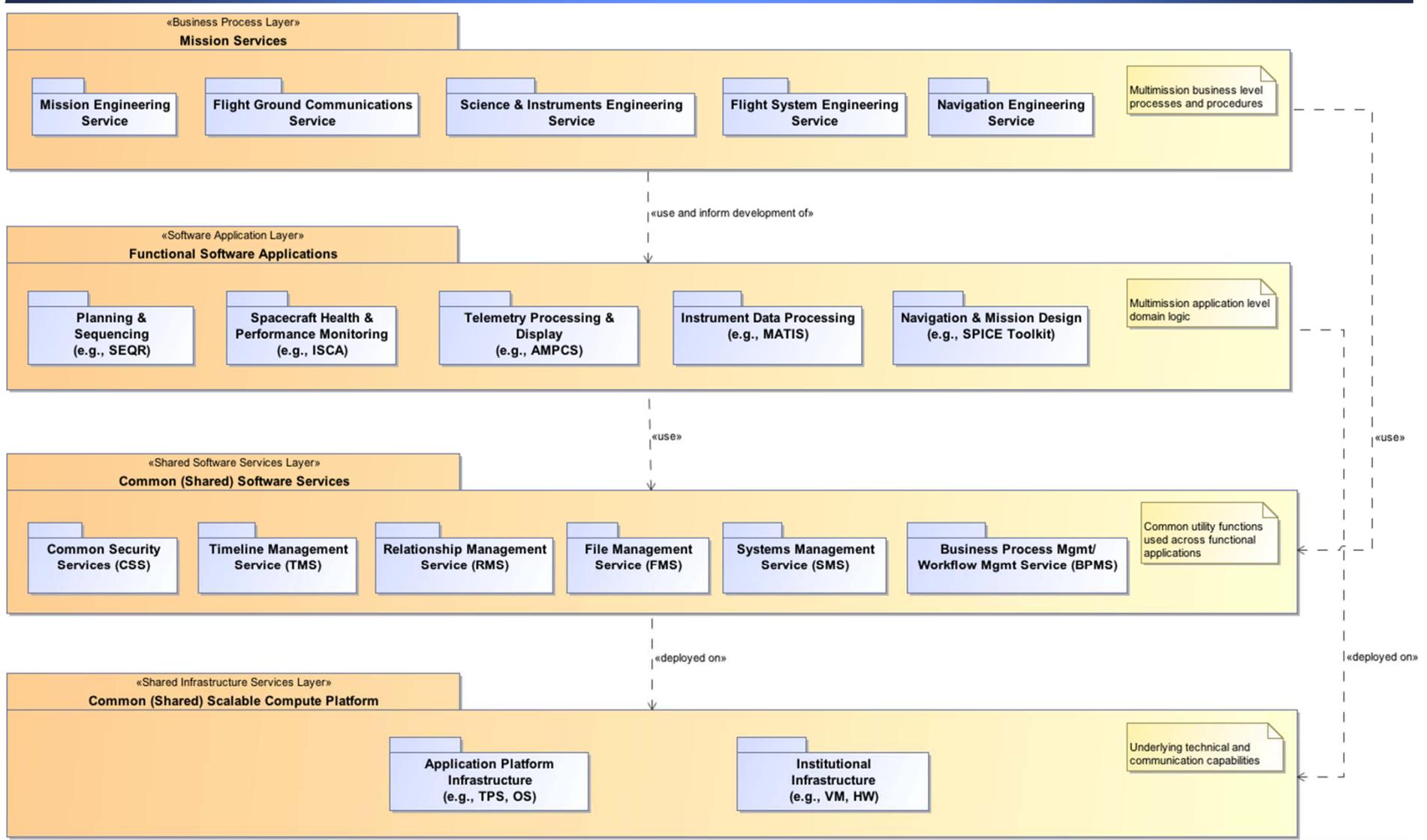


«Module 4»  
**Technology  
Architecture Description**

- Purpose:
  - Describe prescriptive set of technical standards (Tech Stds Profile) and Common Scalable Compute Platform (AMMOS Platform Description)
- Scope:
  - List of approved set of technical standards and supported technologies to be prescriptive for OpsRev and S/W & H/W modernization tasks, technology forecast, runtime compute platform for hosting AMMOS tools and services



# AMMOS Architecture Layering





# System Services

- Our target state system architecture will standardize on Timelines as the common information model
- Our target state system architecture we need a Common Information Exchange -> This will be composed of:
  - Timeline Management Service (TMS)
  - Relationship Management Service (RMS) – formally known as Information Management Service (IMS)
  - File Management Service (FMS)
- Our target state system architecture we need a common Business Process Management (BPM)/Workflow Service
- Our target state system architecture we need an Enterprise System Management system

BPM – Industry Standard  
Service based approach – Common industry approach to reuse  
ESM – COTS & GOTS product adoption



# Architectural Decisions (1/2)

## (Application Architecture)

- **AAD #1:** Adopt a *Resource Oriented* architectural style as the primary architectural style for AMMOS modernization applications across AMMOS functional areas.  
**Rationale:** Provides the most **flexible, adaptable, reliable, scalable, and multi-language/multi-platform-based** capability over other architectural styles such as *Distributed Object Oriented Computing* and *Service Oriented Architecture*.
- **AAD #2:** The Common Information Exchange (CIE) represents the primary strategy for application integration for managing shared data between AMMOS functional applications and represents a “logical” entity. This Implies that common software services that comprise the CIE are a set of discrete, modular software services each with a set of exposed interfaces for use by information producing and consuming applications.  
**Rationale:** Provides the most **flexible** and **adaptable** solution for **integrating functional applications across AMMOS** functional areas and providing time-varying MOS information in the form of Timeline representations, non-time-varying MOS information, and relationships among MOS information objects. Discrete services are exposed and managed as separate subsystems over a single ‘service-based’ interface to an opaque set of services, which is less manageable.

ReST – Adoption of common industry architectural style



# Architectural Decisions (2/2)

## (Application Architecture)



- **AAD #3:** Adopt Shared Database with REST as a Connector as the primary integration style for integrating functional applications across AMMOS functional areas.

**Rationale:** Provides the most **flexible, adaptable, reliable, scalable,** and **multi-language/multi-platform** and standards-based option for integrating functional applications across AMMOS functional areas. Enterprise messaging does not offer such favorable properties largely because the industry has not converged on a unified programming model and wire-level protocol that supports a multi-language/multi-platform environment with adequate flexibility and adaptability and that can scale to the level of the Web.

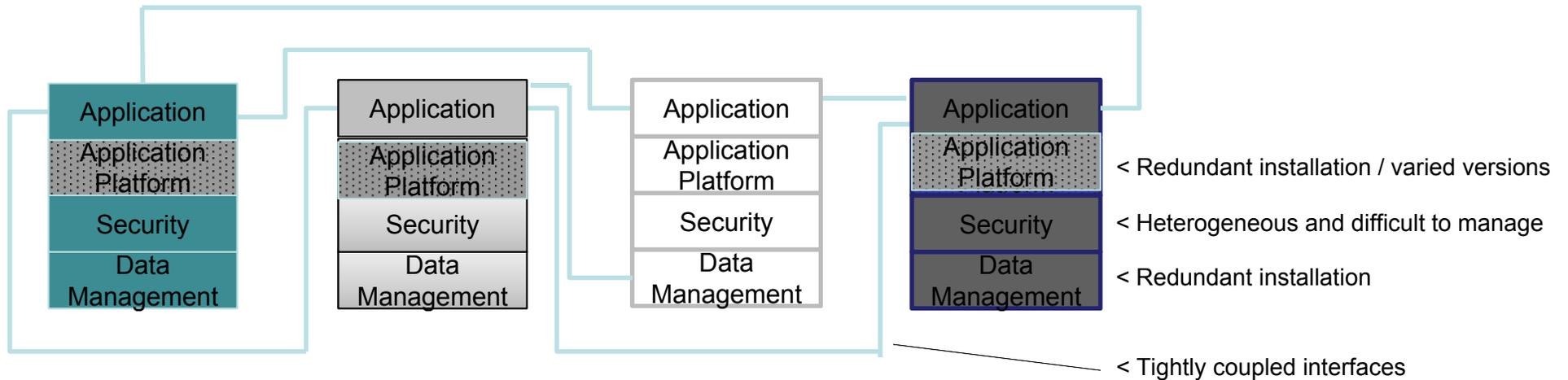
ReST – Adoption of common industry architectural style



# AMMOS ARCHITECTURE HIGHLIGHTS

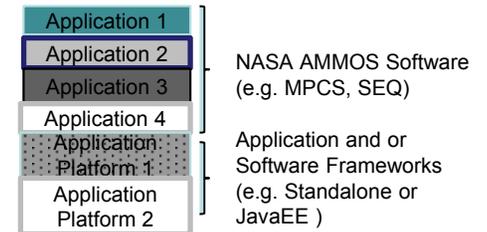
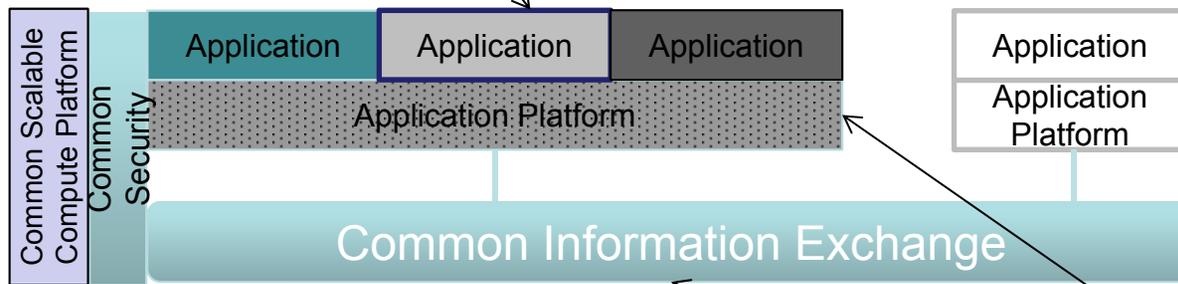


# AMMOS Application Modernization



Modernization Activities

Application refactoring to simplify deployment



Compute platform – industry standard hypervisor  
 Industry standard security approach

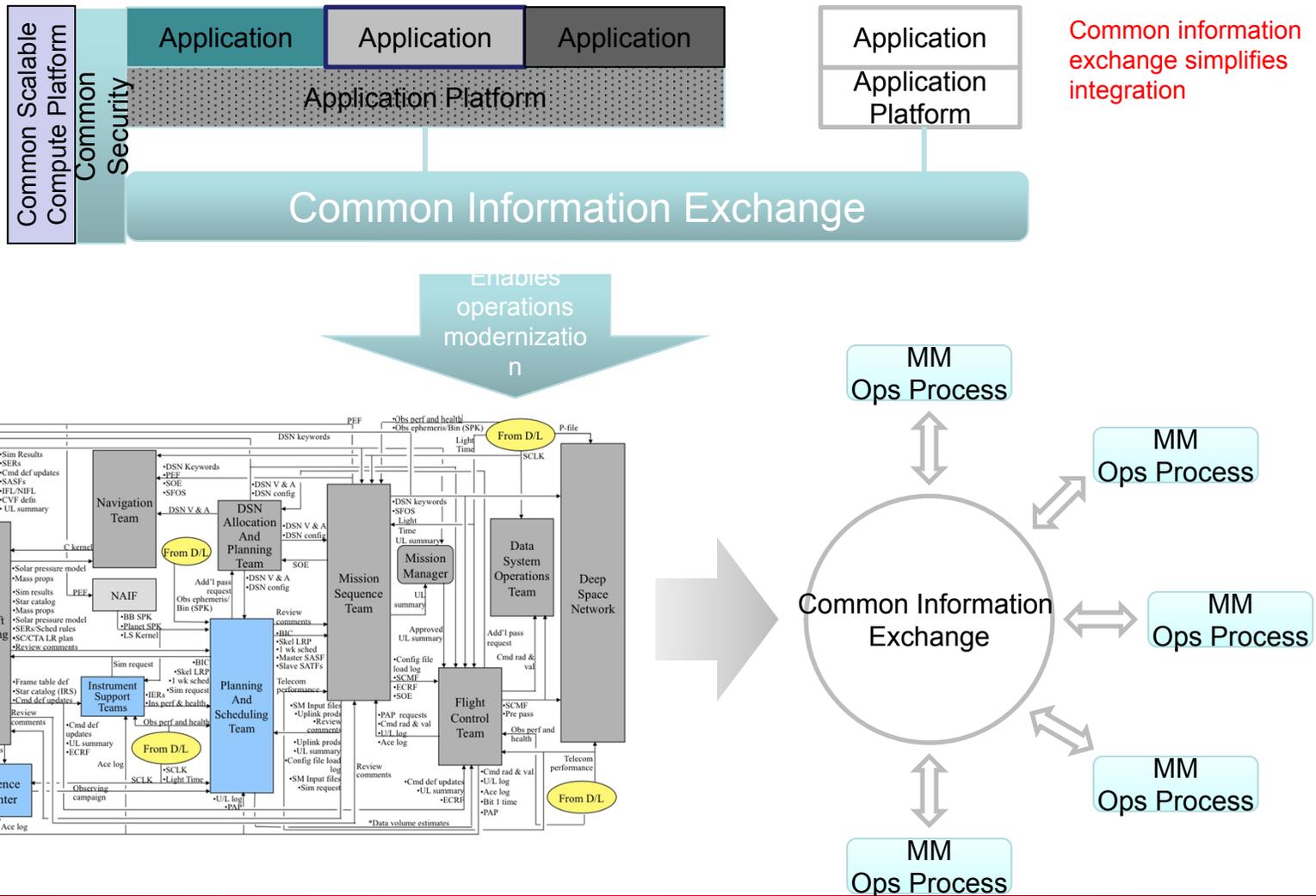
Common software and infrastructure services

Common loosely-coupled information integration / Definitive source of shared information

Commonly managed application platforms independent of applications

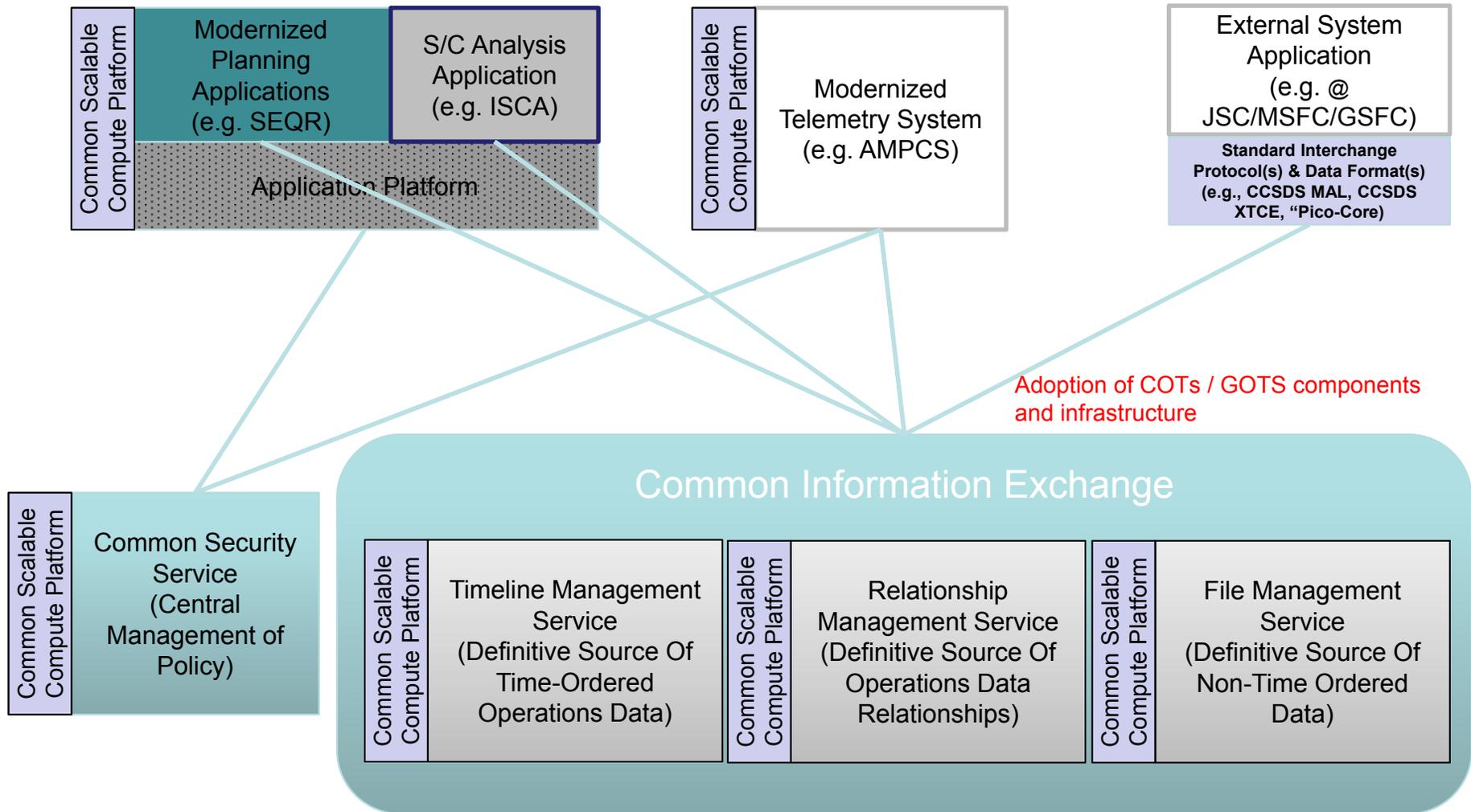


# Operations Modernization





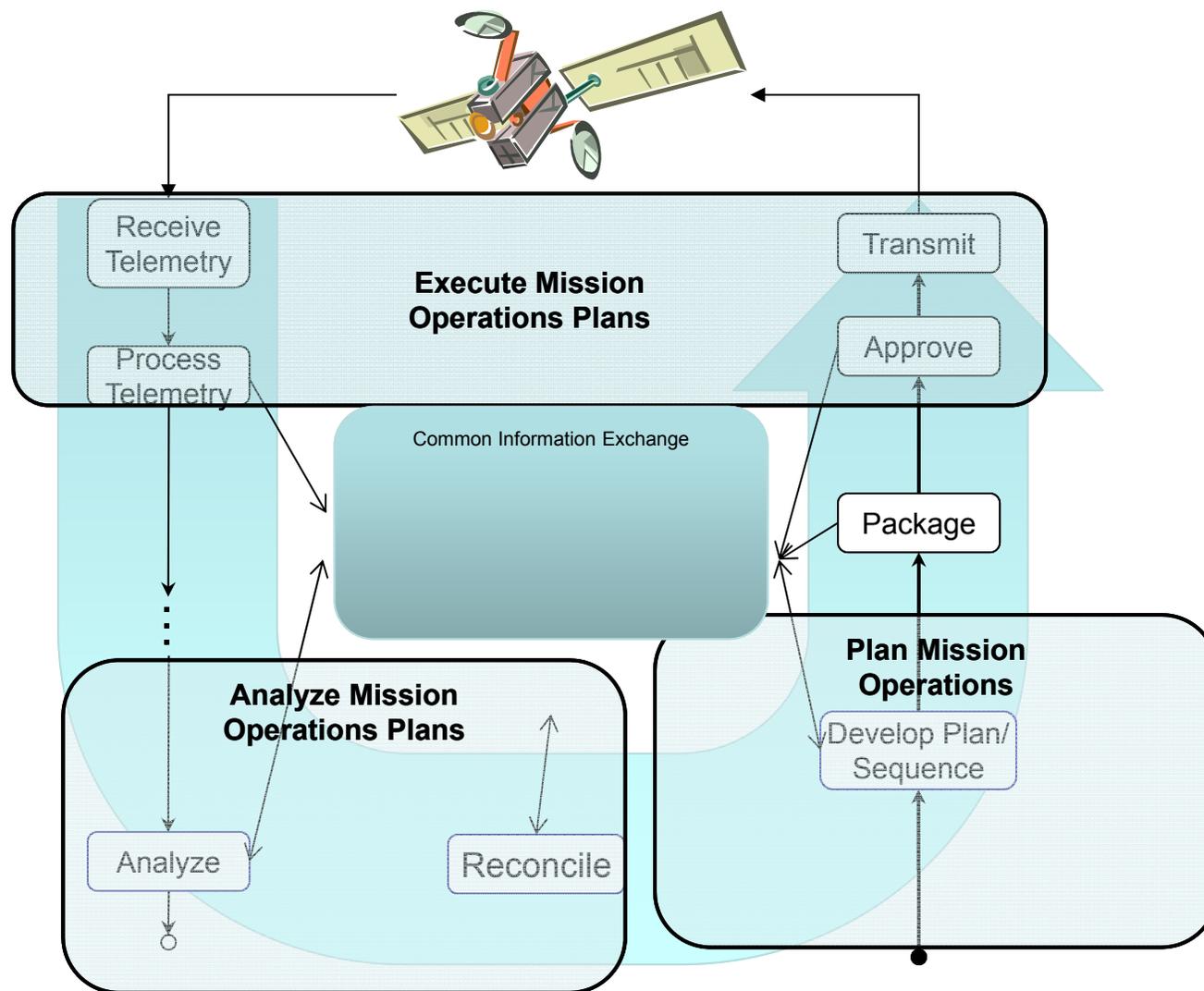
# Application Modernization In Context Of Architectural Elements





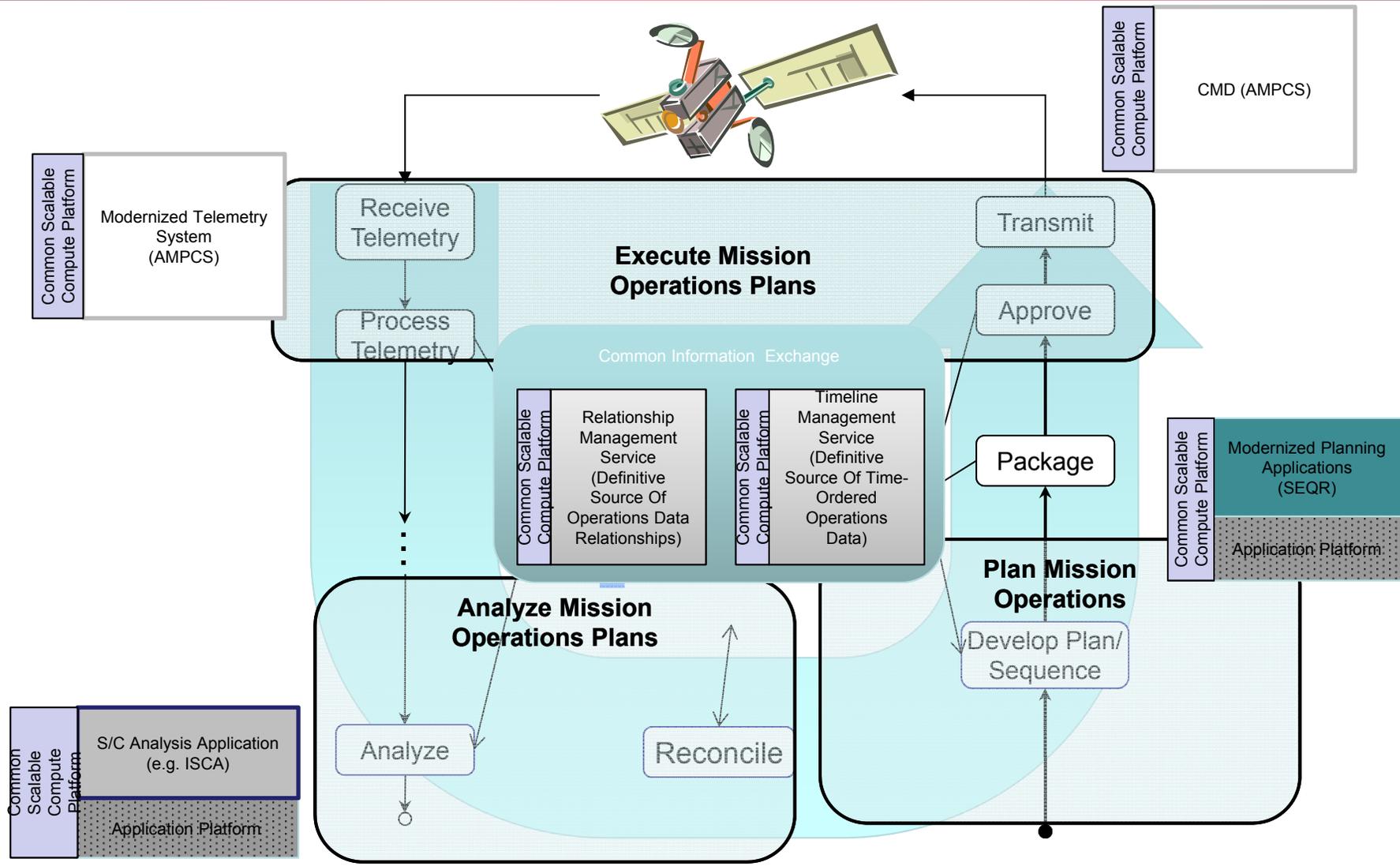
# Closing the U : How Our Task Support

Allows missions to  
standardize planning  
execution &  
reconciliation  
operational processes





# Closing the U : How Our Task Support





## Take Away

- Commonly managed application platforms independent of applications
  - Reduces cost to deploy and maintain
- Common and scalable compute platform
  - Reduces H/W cost and improves system flexibility
  - Reduces risk (test as you fly fly as you test )
- Standardizing on Common Information Exchange / Definitive source of shared information
  - Reduces system integration cost and complexity by standardizing on a canonical information model and associated shared services
  - Improves data and product accountability
- Adopting industry technology standards and support the development of CCSDS information standards
- Operations modernization allows missions to standardize planning execution & reconciliation operational processes