

# Moving Towards a Common Ground and Flight Data Systems Architecture for NASA's Exploration Missions

Steve Rader<sup>1</sup>

*NASA Johnson Space Center, Houston, TX, 77058*

Mike Kearney<sup>2</sup>

*NASA Marshall Space Flight Center, Huntsville, AL, 35812*

Thom McVittie<sup>3</sup>

*NASA Jet Propulsion Laboratory, Pasadena, CA, 91109*

and

Dan Smith<sup>4</sup>

*NASA Goddard Space Flight Center, Greenbelt, MD, 20771*

## Abstract

The National Aeronautics and Space Administration has embarked on an ambitious effort to return man to the moon and then on to Mars. The Exploration Vision requires development of major new space and ground assets and poses challenges well beyond those faced by many of NASA's recent programs. New crewed vehicles must be developed. Compatible supply vehicles, surface mobility modules and robotic exploration capabilities will supplement the manned exploration vehicle. New launch systems will be developed as well as a new ground communications and control infrastructure. The development must take place in a cost-constrained environment and must advance along an aggressive schedule. Common solutions and system interoperability will be critical to the successful development of the Exploration data systems for this wide variety of flight and ground elements. To this end, NASA has assembled a team of engineers from across the agency to identify the key challenges for Exploration data systems and to establish the most beneficial strategic approach to be followed. Key challenges and the planned NASA approach for flight and ground systems will be discussed in the paper. The described approaches will capitalize on new technologies, and will result in cross-program interoperability between spacecraft and ground systems, from multiple suppliers and agencies.

## I. Introduction

This paper describes an approach for the Constellation program that is being proposed within NASA to insure interoperability of all Constellation program elements (vehicles, facilities, etc.). If accepted, this proposal would require the specification of a set of standard interfaces, and to some degree the underlying functionality beneath those interfaces. If the Constellation program accepts this proposal, it is hoped that the interoperability of elements in the Exploration era will achieve a level of compatibility that is unprecedented in major spaceflight programs.

All modern spaceflight programs will develop a Command, Control, Communications and Information (C3) architecture in some form. Some are developed piecemeal, and some are developed with a more-or-less integrated approach. This proposal that is being submitted to the Constellation program for evaluation is the most integrated approach to early program development of a C3I architecture that has ever been considered for a major NASA

---

<sup>1</sup> C3I SIG lead, Constellation Program Office, ZF, Johnson Space Center, Houston TX, 77058.

<sup>2</sup> Technical Assistant, Mission Operations Laboratory, EO-01, MSFC, AL 35812, AIAA Member.

<sup>3</sup> Manager, Goddard Mission Services Evolution Center, MS 581.0, Greenbelt Rd., Greenbelt, MD 20771.

<sup>4</sup> Chief Software Architect, Deep Space Systems, MS 126-255, 4800 Oak Grove Dr., Pasadena, CA 91109

spaceflight program. Over the next year, the Constellation program will be evaluating this approach, and if accepted, may start early prototype efforts to develop components of the Constellation C3I architecture.

Whereas the ISS program is the greatest assemblage of connected spacecraft modules ever attempted, the Constellation program is expected to be a challenge of a similar size, but for an assemblage of free-flying manned and unmanned spacecraft. Ultimately, the Constellation program may need to provide command and control connectivity between launch vehicles, manned transport vehicles, lunar/Mars surface ascent/descent vehicles, orbital relays, robotic rovers, surface habitats, logistics vehicles, EVA "away teams" on the surface, communications relays in orbit and on the surface, and the usual collection of terrestrial operations facilities. Because the coordinating spacecraft will be performing rendezvous and other complex operations many light-seconds away from earth, they will need direct proximity command and control links between them. These vehicles and systems will certainly be built by many different contractors, and some may be provided by other agencies and international partners. In order for the program to be successful in this environment, a more strongly integrated and standardized method of connecting these vehicles is required. Hence the proposal was developed for defining an integrated C3I architecture early in the Constellation program.

While it currently resides in the Constellation program within NASA, the approaches and specifications are being considered by other NASA exploration programs, such as the Robotic Lunar Exploration Program (RLEP). Additionally, while intergovernmental agreements are not yet established for international partnerships in the exploration programs, it is envisioned that these specifications would apply to exploration elements that are provided by other agencies that participate in the Constellation program.

This approach proposes that the program will specify for all Constellation program vehicles and facilities a set of C3I specifications which will define a range of communications interfaces from the physical layer to the application layer. The specifications will include "wire protocols" at the lower communications layers, upper layer communications middleware, some application layer interfaces, and interoperability-enabling architecture functionality outside of the communications domain (e.g. an information model). A key aspect of the technology which is specified will be a messaging system and framework which enables these many interfaces to be loosely coupled. This will allow the architecture to avoid operational flexibility and cost handicaps which have plagued prior programs because of non-standardized tightly coupled systems. The loose coupling in the communications architecture will allow easy reconfiguration and easy "plug in" of new systems, functions, components and new facility interfaces as the program operations concept evolves.

It is important to understand that at this point in the program, these approaches are only candidate approaches that are being considered by the Constellation program. They are currently proposed by the C3I Architecture Team, but adoption of these standards will require further evaluation, both technical and management. Technical evaluation will include software development, prototyping and testing. After the evaluation is completed, some or all of the approaches in this paper may or may not be adopted. Finally, any standards which are selected for adoption will eventually be incorporated into the requirements for the design and operation of Constellation vehicles and facilities.

It is also important to note that where there are existing standards that perform functions that meet the needs of the mission, this approach will adopt them rather than develop new competing standards.

The remainder of this paper explains the technical approach that is proposed by the C3I Architecture team. If the Constellation program accepts this approach, the program will issue a C3I Interoperability Specification as the mechanism to communicate the interoperability standard to the program participants. However, even if approved, it may be different from this approach, or it may include only some components of this approach. Since this paper is being published early in the Constellation program, all statements regarding the makeup or content of this specification or any other Constellation specifications should be considered to be proposals which are not yet approved or adopted by the Constellation program. They have not yet been accepted by the program, and do not constitute guidance from the program to any contractor, vendor or other entity outside the Constellation Program Office. The information in this paper is approved for release so that possible future program participants may be appraised of the potential advances that may be made in the Constellation program. If these proposals are accepted and implemented by the Constellation program, this will not be a "business as usual" program, and participants should be prepared to take a giant leap forward with us.



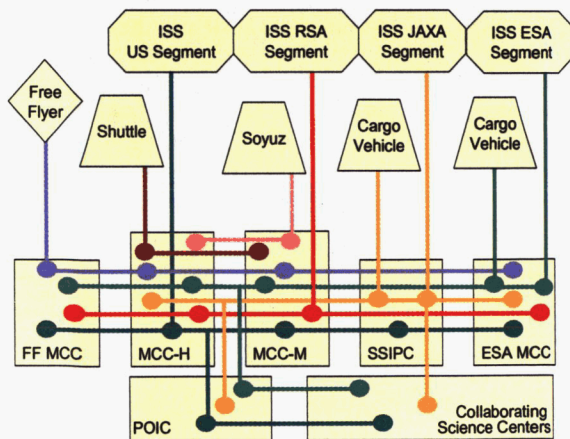
## II. Scope of C3I specifications

The scope of applicability of this proposed C3I Architecture effort spans the Constellation elements including flight elements (i.e. CEV, CLV, LSAM, EVA, etc.) and ground elements (i.e. MCC, LCC, POCC, test & checkout, remote science users, etc.). The set of interfaces defined are proposed to be applied as appropriate to each element to best meet each element's specific functional requirements while also supporting Exploration-wide objectives. Realistically, it is recognized that both these specifications and the exploration program will evolve over time. The program will start with a limited set of core interfaces, but additional specifications and capabilities will be phased in as the overall C3I architecture is adapted over time. Since "one size" will never fit all, the specifications are expected to be "tailored" and adapted for the unique capabilities and requirements of various elements. The proposed standards will apply to a given function (for example, voice signals) and any element that uses that function will be expected to comply with the C3I specification. The set of C3I interfaces defined will be applied as appropriate to each element to best meet each element's specific functional requirements while also supporting Exploration-wide interoperability objectives.

## III. Architecture Drivers

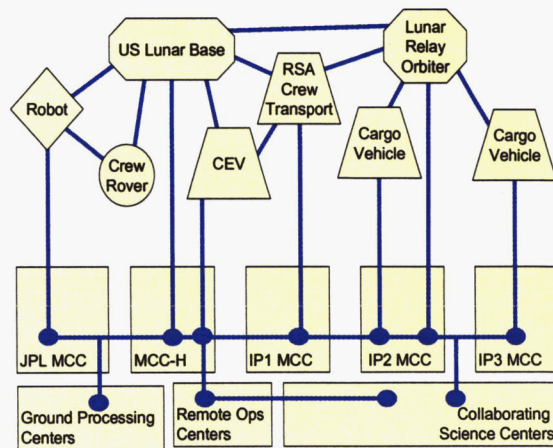
### • Unification (Standardization, Commonality, and Integration)

The proposed C3I architecture must support integrated operation among many different kinds of program elements. The mission concept for the Constellation Program includes the deployment of multiple space assets: launch vehicles, CEV, communications satellites, lunar/Mars transfer vehicles, lunar/Mars landers, lunar/Mars rovers, etc. The C3I task mandates interoperability among the unified population of these flight assets and various geographically dispersed ground control locations, including vendor sites, development and test sites, pre-launch processing sites, launch operations sites, and mission operations sites.



Current ISS telemetry flow (partial)

**Figure 2.1-1** The current International Space Station (ISS) has evolved to a wide variety of telemetry formats, expensive processing of multiple formats by multiple centers, and reduced operations capability.



Conceptual Constellation C3I architecture

**Figure 2.1-2** The C3I architecture proposes to create a unified structure, providing increased capability, and reduced facility costs.

Interoperability is based on communication. C3I must provide an efficient and powerful *common communication mechanism* to enable interoperation of Constellation elements.

Constellation vehicles and operations facilities must have *common data formats* (command, telemetry, voice, etc) such that critical vehicle functions can be processed by any designated operations facility with which the vehicle can communicate. This will provide increased operations flexibility, and therefore increased safety. For example, an unexpected need to use a different command path through a different facility will require a system reconfiguration, not a new software development effort.

Data structures and processing logic should be defined separately so that processing specifications can be based on common, well-understood interfaces. Again, *general purpose, data driven applications* can support multiple vehicles via data reconfiguration, rather than requiring new development for each vehicle. Note that, as technologies' interfaces become increasingly abstract, the number of interfaces tends to diminish and interoperability tends to become simpler. Figure 2.1-3 illustrates how technology is trending to higher levels of abstraction as more open standards are used to develop standard building blocks.

Note the C3I architecture is required to interoperate with Constellation elements either directly with C3I enabled avionics software or through other provisions. The C3I architecture is not currently required to interoperate with legacy ground systems; however, interoperation with some non-C3I-enabled Commercial Off-the-Shelf (COTS) products may be highly desirable.

### Technology Trends to Higher Levels of Abstraction

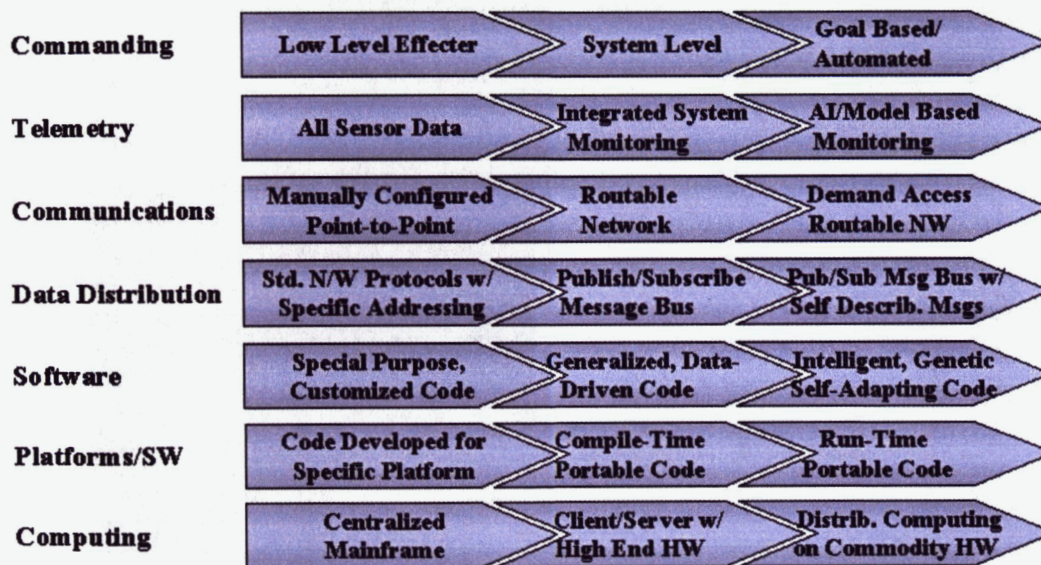


Figure 2.1-3 As technologies mature, interfaces tend to become more flexible and abstract and interoperability is simplified.

Due to the fact that international participants may be collaborating in all phases of the programs, the technical characteristics of these specifications and systems may need to be exportable and extensible to international teams.

- **Flight-Worthiness**

An implication of the unification challenge is that the architecture must support operations of both flight and ground program elements. It must moreover support the direct interoperation of flight elements without ground system involvement.

The C3I task mandates a common architecture that can be applied to control locations of all types. It is likely that Constellation operations in Mars and lunar mission scenarios will entail ad-hoc communications and ad-hoc command and control with limited ground support.

The design of components of the C3I architecture must therefore be qualifiable for flight systems.. In particular:



- Over-reliance on COTS software that may not be easily flight-qualifiable will not be acceptable; flight-qualifiable alternatives with comparable functionality must be available.
- NASA-owned C3I core software must be flight qualifiable: ports to flight-worthy operating systems (e.g., VxWorks) must be provided, memory footprint must be minimized, etc.

#### • **Adaptability**

The architecture must support operations in many different kinds of environments. Constellation exploration missions will range from near-Earth operations to Mars operations. The C3I architecture will therefore need to support communications in environments ranging from high-speed continuously connected terrestrial and near-earth networks to the low-rate, intermittently connected Earth/Mars links. The physics and geometry of these operations significantly affect the communications requirements. Round trip delays in interplanetary communications can be as high as 40 minutes with communications link coverage ranging from continuous to schedule to opportunistic.

Common C3I functions must be able to tolerate these dramatic variations in communication characteristics, so as to prevent unmanageable proliferation of special-purpose software variants. The communications system must include applicable, *specialized protocols* and support both store-and-forward and bent-pipe relay models.

#### • **Flexibility**

The architecture must support the introduction of new elements and technology advances over many years. Technology is advancing at an ever-increasing rate, enabling more capability for less cost. Historically, NASA has attempted to minimize risk by minimizing change, freezing mission configurations long in advance of launch. Even when the technology in use is the state of the art at the time of the freeze, by the time the system becomes operational it is obsolete. Moreover it is often too expensive to upgrade, because the prospect of change was not seriously considered during its design. As a result, NASA spends heavily to sustain obsolete systems that limit ability to evolve operationally.

The C3I architecture must enable incremental improvement in all C3I systems, to leverage technological advancements and to accommodate the changing needs and requirements of the Constellation over its lifetime. To this end, the architecture is based on a *layered, modular approach* that allows components of the architecture to be upgraded individually without requiring modifications to the rest of the system. This acknowledges the likelihood that new “best” technologies are yet to come. Examples include the use of commodity computers and trends toward reconfigurable software radios.

Additionally the C3I architecture development process (described later in this report) actively promotes advanced development efforts. This work may leverage technology advancements that can make Constellation operations more efficient or more reliable.

#### • **Scalability**

The architecture must support concurrent control of widely varying numbers of elements in any single system.

Even modest Constellation operations concepts result in many varied communications configurations entailing command and control of multiple elements. As the Constellation builds up, the potential number of elements requiring simultaneous command and control increases. Additionally, the architecture must support operations throughout the full life-cycle of each element.

The C3I architecture must therefore be scalable to support configurations ranging from a single operator supporting a simple bench-level avionics test at a vendor site, to hundreds of operators at multiple control centers supporting complex space operations. Scaling also applies to the number and complexity of



Figure 2.5-1 The C3I architecture must scale from large control facilities down to individual science users.

targets a given instance will control at a given time. This could be a single system component (as in the bench-level avionics test) or four orbiting elements that must be docked and assembled in orbit.

Increasing the scope of control center operations to account for this increases the importance of optimizing the operability and usability of the C3I architecture. There must be active stakeholder participation, including the end users, system managers, vehicle contractors, and interface users, in all phases of development. These stakeholders also will come from different operations cultures and it will be important to remember that personal preferences do matter and that a “one size fits all” approach should be avoided.

Operability also includes designing with operations work flow in mind. This is key to providing effective tools that are crucial to reducing operations cost and maximizing automation capabilities.

Scalability is sometimes difficult to provide for all functions. The benefits of including a given component in the common scalable core system must be balanced against cost, complexity, and reliability.

- **Cost Effectiveness**

The architecture must succeed despite limited budget resources.

The Constellation program contemplates a significant increase in flight activity, and available funds to support that activity may not increase at the same rate. This implies that C3I should be designed to operate much more efficiently than current control center systems: support reconfiguration must be much more rapid and much less labor-intensive; system status visibility must be much improved, and so on. The C3I architecture must in short be affordable to build, operate, and maintain. Cost reduction must be a key driver in the formulation of all development, operations, and integration processes.

The architecture must therefore provide features that allow separately developed components to be integrated easily. The integration of tightly coupled systems has proved expensive and it often precludes upgrade of system components due to the possible “ripple” effects. The architecture should provide a *layer of isolation* between system components that reduces specific inter-component dependencies. A good example of this concept is the “Plug-n-Play” concept used with many computer components. Generalized, data driven application software, built upon well known specifications and/or sets of building blocks, can easily be integrated together with little or no prior knowledge of the other components in the system.

Simplified system integration will also streamline the reconfiguration of operations facilities to support new missions, a further cost reduction.

The C3I architecture must promote the use of *common equipment and software* at multiple ground facilities, reducing costs arising from redundant development efforts. Additionally, continuous focus should be placed on *work flow automation* and *enterprise information access* with goals placed regularly to reduce staffing levels for all processes.

Reliance on proprietary techniques and protocols can limit the effectiveness of the C3I architecture in a number of ways: the opaqueness of proprietary technology may be perceived as a source of mission risk, reliance on vendor support for extensions may retard evolution toward more capable modes of operation, and the cost of licensing proprietary technology may inhibit the distribution of useful software. All of these problems can be mitigated by adopting *open standards* where possible.

The C3I architecture is being developed for human space flight. As such, safety and reliability are vital. To verify and certify the systems in a complete and thorough way requires *extensive testing*. The ability to test the system, efficiently and thoroughly, must be designed into the architecture and processes from the beginning. It should leverage automation to continually improve test coverage, reduce regression test time, and collect meaningful quality metrics. Given the increased number of system configurations, new test methods and innovations will be required.



#### IV. Architecture Overview

The C3I architecture provides a robust approach to providing a common, powerful, integrated C3I solution. The architecture defines a set of elements, discussed in this section, that make up a given C3I system or “instance.” An instance might be a full control center, a test and checkout facility, or a space vehicle (such as CEV). From one instance of the C3I, authorized users can control one or more “targets” where a target is a space vehicle, Ground Support Equipment (GSE) test hardware, or even another control center. Generalized, any C3I “instance” has the capability to communicate and control any number of “targets”. For a ground based control center, this might be monitor and control of orbiting a CEV, a Lunar Lander, and a Lunar Injection Stage as they rendezvous and dock. For an EVA crewmember on the Moon, this might entail controlling the suit systems (“instance”) as well as controlling nearby lunar scientific experiments and robotic rovers (“targets”). This basic concept is shown in Figure 3-1.

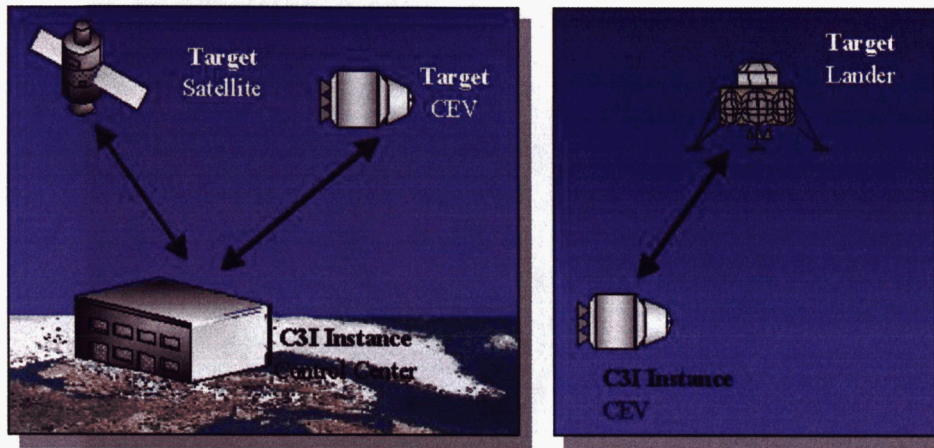


Figure 3-1 C3I Command & Control Concept

Targets built on the C3I architecture will allow full interoperability with C3I controlling instance while legacy targets may require more specialized adaptation.

Note that while the C3I architecture is designed to apply to ground **and** flight assets, the degree to which it is applied to flight assets may vary depending on program decisions. Regardless of these decisions, the architecture should remain valid. However, in this report, there are many examples that illustrate the advantages of the architecture in the context of a flight element.

The C3I architecture is very broad in its application and yet very basic in its implementation. At the core of the C3I architecture are five main architectural elements. These elements are the basis of the architecture and provide fundamental functionality that can be applied across the Constellation. A summary of these elements is listed in Table 3-1 with more detailed information provided for each in section 3.2.

Table 3-1 Main Architectural Elements

Element	Description	Benefit
Layered Partitioning of Functionality	Isolates changes and simplifies implementation and control of common services.	Increased capability over time due to technology advancements that can be more easily incorporated into the systems.
C3I Messaging Middleware based Framework	Standardizes application's interfaces with the system and with each other while providing efficient network data distribution.	Reduces development integration costs, enables more efficient small development teams, and provides more efficient information distribution.

Element	Description	Benefit
Integrated Information Model	Provides enterprise information access and multiple vehicle support with the use of data driven applications.	Reduced ops. costs from workflow automation and more efficient information management.
Standard Command & Control Method	Provides a standard interface and logical protocol for implementing command and control functions.	Reduces training and simplifies operations (increasing safety and ops. reliability) across the increased number of vehicles and systems.
Integrated Space and Ground Network	Allows Constellation elements to exchange information efficiently and effectively.	Reduced cost of operating the network with smarter, more efficient utilization of the communications links.

The following sections describe the new approaches and concepts used with the C3I architecture and each of the architectural elements.

- **New Approaches and Concepts**

Several new approaches and concepts that are proving highly effective in the commercial sector are woven into the C3I architectural elements introduced in the previous section. These include **loosely coupled, interoperable system of systems; consistent command and control; open/standards based systems; and switched/routed networks**. The following section discusses each of these and how the C3I architecture uses them as the basis for its fundamental elements.

1) *Loosely Coupled, Interoperable System of Systems*

Loosely coupled interoperable system of systems have been demonstrated to be effective in reducing integration costs and enabling small, manageable development teams. For the C3I architecture, this concept is implemented using a **layered partitioning of functionality** that isolates changes and simplifies implementation and control of common services and especially the **framework** that standardizes the interactions with the system and each other. Additionally, the implementation of generalized, “data driven” applications with an **integrated information model/system** to manage that data provides yet another level of interoperability. These architectural elements are discussed further in the sections that follow.

2) *Open, Standards Based Systems*

The commercial world has widely benefited from the creation and use of open, standards-based systems and protocols as driving force toward interoperable systems. The C3I architecture makes heavy use of these standards and protocols and builds upon them to create an **integrated space and ground network**. Additionally, the C3I architecture will attempt to drive the creation of appropriate standards for messaging and information interchange. Open source and standards based system has also allowed for the migration away from expensive, specialized hardware and toward commodity solutions.

3) *Consistent Command and Control*

The Constellation Program will include multiple vehicles and systems which must be managed with a relatively small team of astronauts and ground controllers. A **standard command and control model** must be applied across all program elements such that operators use common methods to interact with the vehicle systems. This method must address traditional commanding issues (authorization, execution, verification, etc...), but also integrated procedure execution, mixed manual and automated execution, command authority handover, and manual override. Command and control interfaces must have a simple, consistent implementation. While this concept is not necessarily new, the C3I architecture introduces a new approach to this problem using virtual system representations that exist as data applied to generalized data-driven applications.

4) *Switched/Routed Networks*

Ever since telephones moved from switchboard operators to circuit switching hardware, the commercial world has built entire infrastructures upon the basic switched, routed IP network. It is time that NASA moved its space network from a pre-planned, switchboard operator configuration to a full end-to-end automated network. By



leveraging standard protocols and techniques, the C3I architecture provides an end-to-end, **integrated space and ground network** that will provide the efficiency and flexibility required by Constellation operations.

- **Architecture Layers**

The first element of the C3I architecture is the partitioning of functional layers as shown in Figure 3.2-1. The main tenet of this approach is to **keep it simple, decouple functions, and leverage modern technologies**. The separation and isolation of these layers allows for the internals of one layer to change, without affecting any of the other layers. This reduces the amount of reengineering required and thus allows the system to evolve more affordably. Additionally, the layers provide a means to implement and manage common or shared services. For instance, the architecture could more easily implement standard mechanisms for how all applications report status. The following sections briefly outline each of these layers.

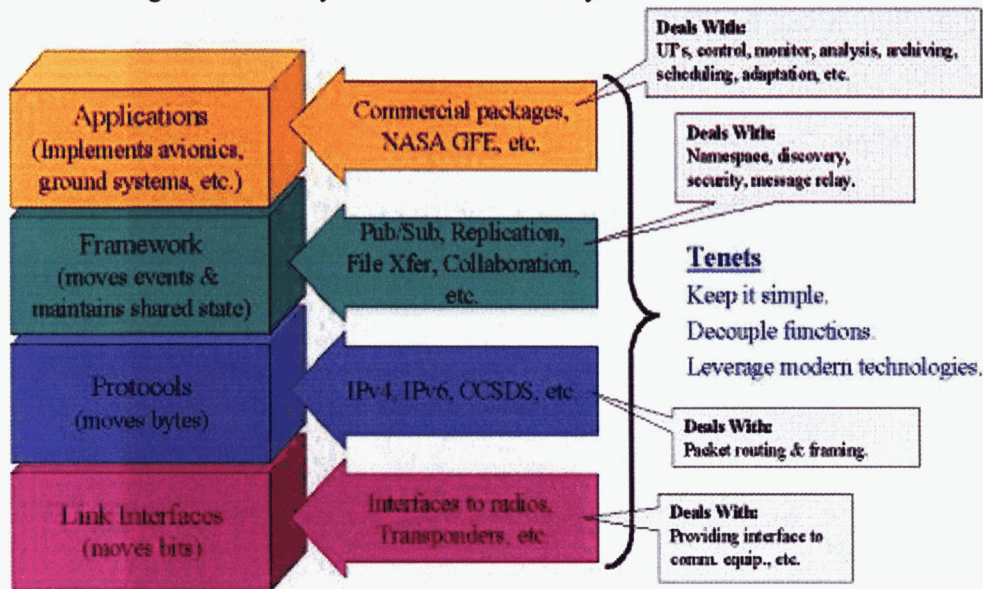


Figure 3.2-1 C3I Architecture Layers

### 5) Applications Layer

The applications layer provides the collection of software applications that provide all of the real functionality required for command and control. These applications provide the tools for **monitor and control, planning and analysis, information computation/ management**, and even **voice and video** (see Table 3.2.1-1). Applications developed for the C3I architecture follow the “plug-n-play” concept using the C3I framework (discussed further in section 3.3) and “data driven” design principles. Applications can be plugged into the network and be configured in real-time to support desired operations. The applications layer can support both “native” applications, those built using the framework Application Programming Interface (API), and legacy or COTS applications using framework adapters.

Initially, the basic functionality of C3I applications may be similar to existing command and control capabilities. However, integration with the C3I framework will enable advanced features, such as task and workflow automation to emerge. This results from the seamless information exchange mechanisms that are provided by the messaging framework and the common automation features embedded in the common framework services and building blocks. Note that these results have been experienced on message-bus implementations at several NASA centers.



While there will be a common set of applications that provide core C3I functions, the architecture also supports tailoring specific instances as required. A control center instance may require a large set of applications including planning and analysis tools, while a test facility may require a small set of applications, but also require a custom application to control GSE. The C3I architecture is expected to provide a “shopping list” that can be used to select certified applications to assemble and deploy a given instance. Over time, this list is expected to grow as more applications are certified. However, duplicate or similar functions may also be combined or removed from the system over time to reduce sustaining costs.

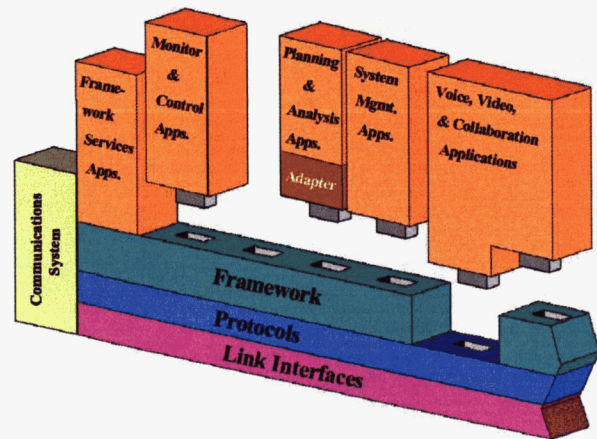


Figure 3.2.1-1 C3I Application Layers

Table 3.2.1-1 Summary of Application Functions

Application	Description
<b>Monitor &amp; Control</b>	This includes the main user interface tools for command and control including applications for telemetry display, command building/selection/ execution, and event monitoring.
<b>Information Processing &amp; Management</b>	This includes processes used for functions like telemetry packing/unpacking, math/EU conversions, recording, playback, archive retrieval/reporting, and information discovery, retrieval, & interpretation.
<b>Planning &amp; Analysis</b>	This includes a large set of existing tools used to analyze and plan things like crew activities, attitudes, trajectories, resources, communications, etc. Initially, to avoid large, re-hosting costs, simple framework adapters would be developed to allow for information interchange with other applications on the framework.
<b>Voice, Video, &amp; Collaboration</b>	The C3I architecture completes the transition of voice and video to the network. These applications would perform traditional voice loop & video services, but also start to leverage the digital streams for more advanced functions. Collaboration tools were also added to provide more effective methods for space to ground communications that will be necessary with the increasingly disconnected communications.
<b>System Management</b>	This includes all of the tools necessary to deploy and manage all aspects of a C3I instance.
<b>Portal Services</b>	This includes the services required to allow secure remote/web based access to mission data, voice, video, or information products.
<b>User Tools</b>	This includes tools that allow the users to log activities, perform handovers, track and issues.
<b>Framework Services</b>	This includes applications related to framework provided services such as security, information mapping, message routing, and link management.
<b>Hardware Control and/or Simulation</b>	This includes the control or interface software to any hardware device that must be controlled. For ground instances this could be GSE or communications hardware. If implemented on-board, this would include software that controls flight hardware. Since these applications are framework based, simulations could be substituted for any controlled component.

#### 6) Framework Layer

At the center of the C3I architecture is the framework layer. The framework standardizes the behavior for how applications interact with the system and with each other. At a basic level, it provides the “publish and subscribe model” for information exchange that frees applications from needing to know the details of the source of the data that they consume or the destination for data they produce. It also provides mechanisms for self-describing data formats. The framework layer also provides integration services such as security control functions and information discovery and request functions. Applications interface to the framework using the framework API. On the back end interface, the framework interfaces to the protocols layer to provide network based communications functions.

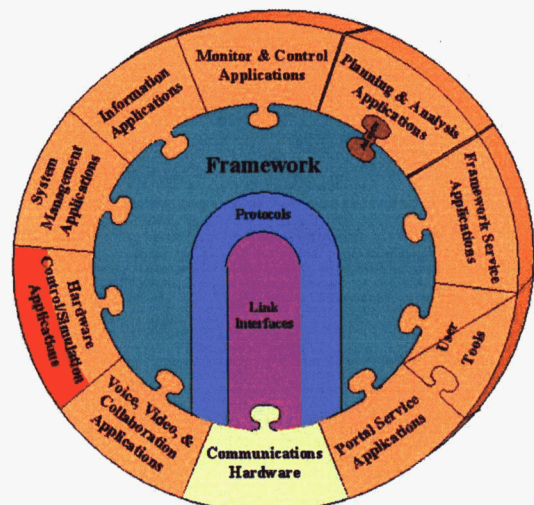


Figure 3.2.2-1 Framework Centered C3I



## 7) Protocols Layer

The protocols layer provides the support for the various protocols that are required to support the diverse communications constraints of Constellation missions ranging from terrestrial links to 40-minute round-trip delays encountered by Mars links. Isolation of this layer also allows for the protocol stack to change as system requirements change without affecting the higher-level frameworks and application layers. This layer provides the protocol implementations that enable an integrated space and ground network that supports packet switching/routing and Quality of Service (QOS). These features are the basis for progressing from a manual switchboard-like communications infrastructure to a full network based implementation.

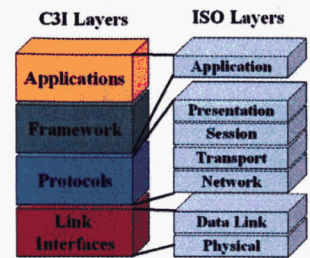


Figure 3.2.3-1 Layer Map to Protocol Stack

## 8) Link Interfaces Layer

Link Interfaces provides a common interface for the rest of the C3I architecture to interface to various communications equipment. Since communications equipment is often sized and selected based on a number of vehicle/element specific factors and communications technology is advancing rapidly, it is impractical to assume common communications hardware on all Constellation elements. Instead, the link interfaces layer attempts to provide a common interface wrapper around the communications hardware to isolate the higher layers from the differences in individual systems.

The concepts behind this layer are discussed further in the section discussing the “Integrated Space and Ground Network”.

### • Messaging Middleware Based Framework

The second element of the architecture is the messaging middleware based **framework**. As introduced in the Layers section, the framework is more than just one of the four layers, but really stands out as the centerpiece of the C3I architecture. It provides the main mechanism that allows for the “loosely coupled system of systems” while also providing a means to unify the different application into an integrated system. The framework is made up of several key parts as shown in Figure 3.3-1.

Starting at the base of the framework is the **messaging middleware**. Messaging middleware provides data distribution throughout the network mainly using the “publish and subscribe” paradigm. This paradigm abstracts communications such that applications no longer have to worry about end-point locations (IP addresses, ports, etc). Instead, applications can simply publish information that they produce onto the network (or message bus) and subscribe to information they need. Figure 3.3-2 illustrates this concept.

Currently, there are several options for messaging middleware as both COTS and Government Off-the-Shelf (GOTS) products. Each has its own features, advantages, and costs. Given that messaging middleware technology is

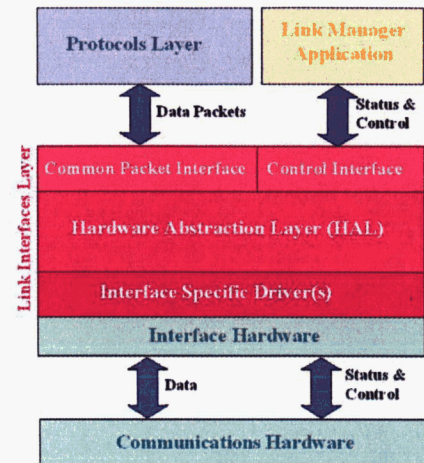


Figure 3.2.4-1 Link Interface Concept

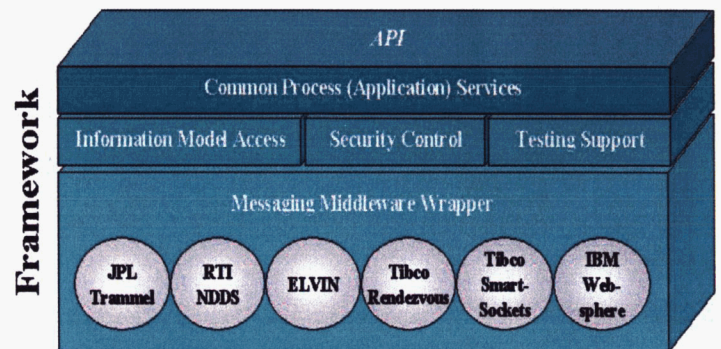


Figure 3.3-1 Framework Internal Overview

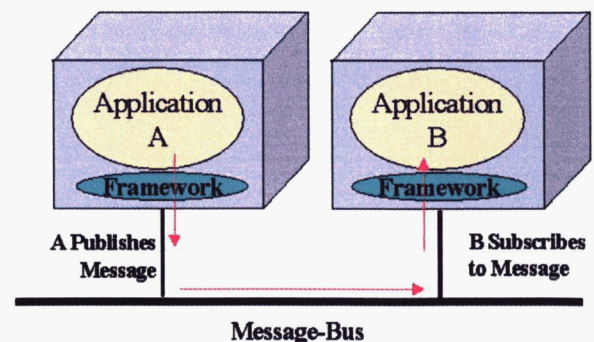


Figure 3.3-2 Publish and Subscribe Message-Bus



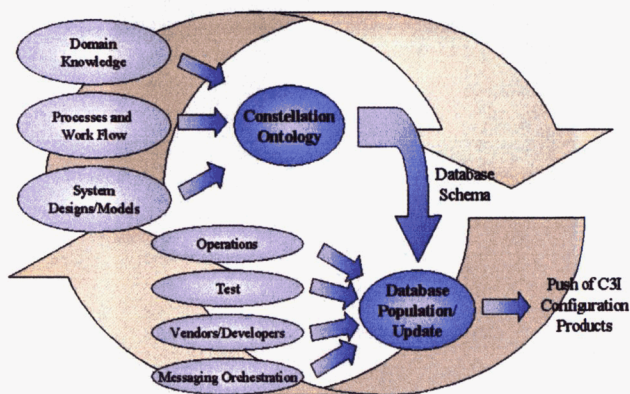
still evolving, instead of attempting to select a single product, the **architecture wraps the middleware** thus preventing vendor lock-in issues and providing flexibility for the end-user for performance, platform, and/or cost.

Above the messaging middleware, the framework provides services that are required for all C3I applications and enable a consistent approach to information access, security control, and testing support. Applications built on top of the framework automatically inherit these features with little or no development. These types of common features for all applications reduce the effort required to configure and deploy C3I instances with large numbers of applications and nodes. On top of the common framework services are the application building blocks that provide consistency for how applications behave and respond. Examples of these services are standard monitor and control interfaces, common configuration setup mechanisms, and standard scripting support.

The success of the application framework lies in the Software Development Kit (SDK) and API for the framework that application developers will use to develop or adapt their applications. The SDK must provide a simple, easy to use interface with complete documentation, sample code, and testing tools. For applications, the framework standardizes the behavior for how application elements interact with the system and other applications.

### • Information Model

The C3I architecture is based on the concept of “data driven” applications and services. This means that the specifics about the vehicle or equipment that is being controlled are not compiled into applications, but loaded at operations run-time from configuration data using more generalized applications. While this enables the architecture to flexibly support a large number of target vehicles or equipment, the information required to drive each configuration must be effectively managed. Additionally, to build a system that is more responsive to operations needs, there must be a mechanism to access and interpret the large, diverse set of related information products (e.g. plans, analysis, reference information, and test data). To provide these functions, the C3I uses the concept of an information model.



**Figure 3.4-1 Information Model Process**

the reality of design, manufacturing, test, and operations processes. This process must be iterative to allow for the growth and evolution of the enterprise. A sample of the iterative process for developing and growing the information model is illustrated in Figure 3.4-1.

The information model provides the opportunity to use very simple techniques to logically link information and products that have traditionally existed separately. Examples include vendor test reports, specifications, wiring & channelization information, system schematics and drawings, unit specific maintenance logs, and many others. Utilizing simple database/framework adapters, user applications can discover and request information related to a system, parameter, command, etc. Additionally, the information model would be used to associate security rules with information. This capability combined with flexible security management tools can provide more complete and sustainable security features for the architecture. The information model must also provide readily available access of message names and associated message content for application developers. This information becomes the main interface specification for most applications.

An information model is nothing new to the command and control world. Master Measurement Lists, Telemetry and Command Databases, Wiring List, etc. all have provided the tracking and mapping of data and bits in the system to information. The C3I Integrated Information Model leverages the common information exchange platform afforded by the messaging framework to provide an Enterprise level access to related information. To do this, a domain ontology or naming topology must be constructed that maps the domain knowledge into the software engineering technologies (requirements, models, classes, objects, and processes). The process for constructing such an ontology must start simply, plan for growth and change, and must integrate into

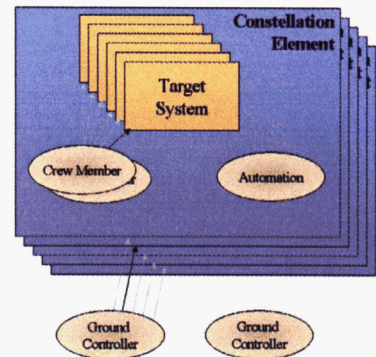


## ● Standard Command and Control Method

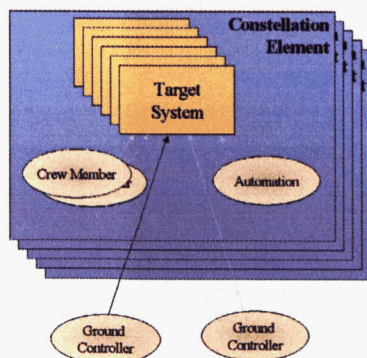
Command and control is central to the C3I architecture and all Constellation elements. The fundamental purpose of a command capability is to provide appropriate *security* and *accuracy*. It is important that the initiator of the command is provided *valid cues* to prompt the issue of a command, the “*right*” command is selected and executed, and the initiator is provided *appropriate feedback* as to the status and execution of the command. Additionally, the initiator must *understand any behavior* that can occur in both nominal and off-nominal execution of commands (i.e. override of closed loop control, communications loss, power loss, processor reboot, etc.). For Constellation operations, this capability must be capable of being performed not only by ground controllers and crewmembers onboard a space vehicle, but also for ground based scientist, EVA crewmembers, or various types of automation (i.e. software, scripts, sequences, etc.).

All of these considerations are extremely important for any single space vehicle. However, when a single set of users must control several space vehicles and systems, it is absolutely critical that a common method is used and their activities don’t conflict. For this reason, the C3I architecture proposes a standard command and control method for all Constellation elements. This commonality must extend from the interface (computer display for humans, API or message interface for automation) to the logic used to execute the commands.

There may be limits to the ability to enforce commonality in the execution of commands due to differences in the systems that implement commands. C3I proposes commonality from the initiator to the interface with the specific system software that executes the command. The method that a command is executed by the vehicle system control software is still important, but falls outside the scope of the C3I task.



**Figure 3.5.2 -  
Controlling Many  
Targets**



**Figure 3.5-1- Possible  
Controllers for a Single  
Target**

The C3I proposes working closely with the astronaut and ground controller communities across the agency to develop a reference implementation and specification that will ensure consistent implementation of Constellation command and control interfaces and logic. The design must address issues such as management of multiple authorized commanders. Not only does this possibility exist with ground operators and multiple crewmembers, but also with situations where one of the commanders is automation (manual override of closed loop control or manual/automated/mixed mode control execution). Any implementations must include adequate authorization management, visibility into actions of other authorized users, and mechanisms for control handover.

One very important aspect of command and control is security. The C3I architecture addresses security implementation in all layers as well as with the Information Model. The security features of the C3I architecture will provide all of the required functions including authentication, authorization, access control, confidentiality (encryption), data integrity, and availability.

Since any common implementation will need to be applied to multiple vehicles and systems, the C3I proposes the use of data driven applications. Data from the C3I information model/management system will provide a “virtual vehicle” or system that command tools will use to configure for commanding of a specific target vehicle or system. The system will need to provide a common method for navigation and selection of target vehicles and systems, security authorization, and integrated procedure execution and automation.

This capability is very important to successful operation of the Constellation program. Not only will this provide a safer, more reliable command capability, it will also save in operations costs such as training and procedures development.

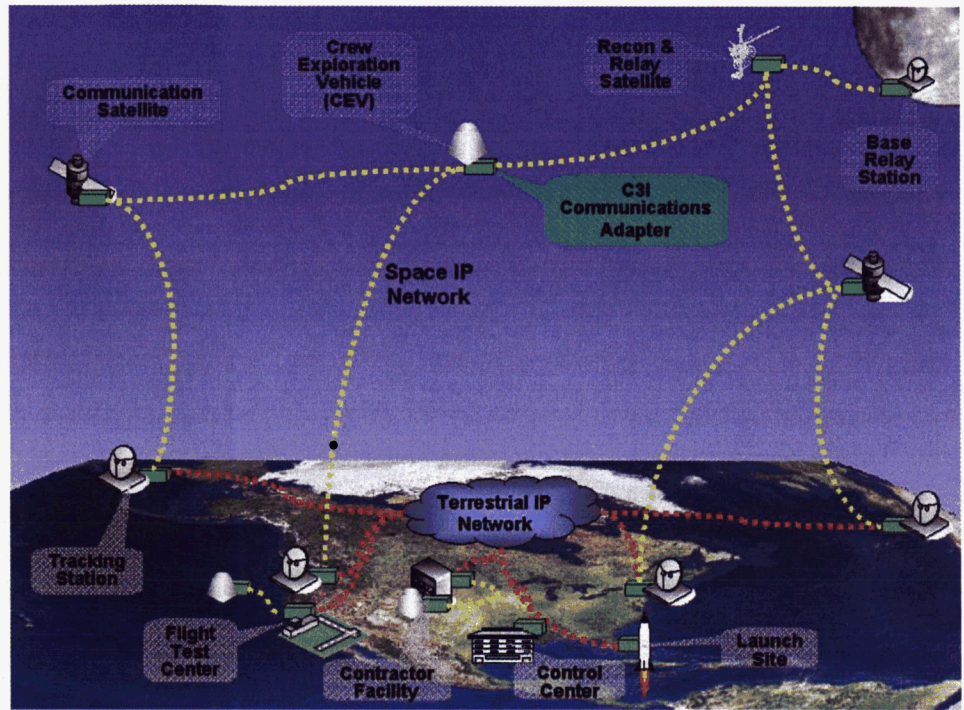


- **Integrated Space and Ground Network**

The final major element of the C3I architecture is the integrated space and ground network. To provide this network functionality, the architecture implements the C3I Communications Adapter (CCA). The CCA provides the major communications functions between Constellation elements. As more Constellation elements are deployed that include the CCA, a switched/routed communications network emerges.

As elements are added, they become participating network nodes providing relay, routing, store & forward, and QOS that make up the network.

The result is the effective networking of the C3I instances via the terrestrial and space Internet Protocol (IP) networks (using appropriate CCSDS, industry standard protocols and framing). Since the current space communications network is manually switched, this implies the implementation of a Space IP network. The pursuit of an automatically switched and routed Space IP network is important for the growing infrastructure of space-based elements that require communications. Such networks will eventually allow for more efficient use of the space communications assets while reducing the required planning and management resources. Note that this IP network approach also includes the transmission of voice and video streams. Figure 3.6-1 illustrates a basic network of space and ground based assets. The CCA provides the “network adapter” that allows this type of network to emerge.



**Figure 3.6-1 Integrated Space and Ground Communications Network**

Functionally, the CCA must be able to provide link management, routing, packet/file store and forward relay, communications negotiations, telemetry prioritization & compression, and encryption functions. Like the applications set, the CCA functionality can be tailored depending on the Constellation element. Not all functions are mandatory. While each node on the network is not required to contain all of the CCA functions, the common set of functionality can provide a network that is fully interoperable and efficient per Figure 3.6-2. The current proposal for the CCA is for NASA to develop and own a reference implementation and resulting specification. This specification would allow implementation by any Constellation element contractor and still ensure interoperability between elements.

- **The Views of the C3I Architecture from Multiple Perspectives**

Although the architecture elements provide a description of the technical constructs that make up the C3I architecture, it may still be unclear “what” the C3I architecture is. The following is a very brief attempt to clarify this with a set of perspectives that may provide some context to better understand the architecture.

From a **management perspective**, the C3I architecture provides an approach and process to developing capability for an evolving program that can flexibly adapt to continually provide more capability, enabling lower

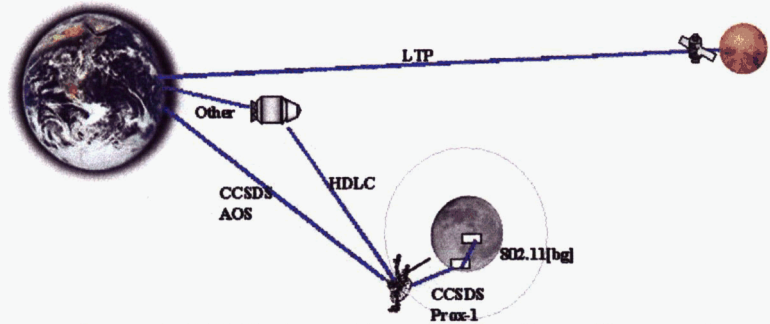


operations costs. The C3I also finally provides a common architecture across NASA that maximizes interoperability and significantly reduces costly duplication of command and control functions.

From a **user perspective**, the C3I architecture can provide a common set of tools for monitor and control of a given facility, vehicle, or science experiment. The architecture can provide the ability to abstract monitoring and commanding to a common interface while still allowing for tailoring where required. It also provides a platform that can support the evolution of operations from a real-time command and control role for near-earth operations, to a planning and support role for Mars operations.

From an **application developer perspective**, the C3I architecture provides a simple set of tools (software development kit) to build applications that can leverage the power of the information interchange and communication features of the framework. These tools allow application developers to reduce their direct interdependencies on other applications and to rapidly be able to “plug in” to the C3I framework.

From a **deployment perspective**, the C3I architecture allows for framework-based applications to be distributed, configured, and operated independent of platform specifics. Additionally, functions that are built into the framework allow system managers to monitor and control the entire system with a robust understanding of the platform and network performance.



**Figure 3.6-2 Various Communications Protocols Required to Make the Network Function**

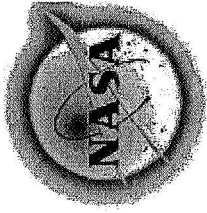
From a **configuration management perspective**, the C3I architecture allows for full command and control support of multiple target systems or vehicles. A robust information model, implemented using modern database and information management tools provides for the simplified management of the large amount of configuration information required for multi-vehicle support.

## **V. CONCLUSION**

The Command, Control, Communications, and Information (C3I) Architecture is an important component of NASA's Constellation Program. The architecture provides communications interoperability between all of the space and ground elements. It also provides a common command and control approach for all elements. The architecture proposes a layered architecture centered on a message-based framework. This network centric approach provides long-term flexibility, scalability, and interoperability. The C3I architecture is not proposed as a point solution, but rather an on-going development process designed to constantly adapt to technology advancements in an effort to decrease costs and increase capability and reliability.

## Appendix A - Acronyms and Abbreviations

ACI	Administratively Controlled Information
API	Application Programming Interface
ARC	AMES Research Center
BAA	Broad Agency Announcement
CCA	C3I Communications Adapter
CCSDS	Consultative Committee for Space Data Systems
CEV	Crew Exploration Vehicle
C3I	Command, Control, Communications, and Information
DFRC	Dryden Flight Research Center
COTS	Commercial Off-the-Shelf
GFE	Government Furnished Equipment
GMSEC	GSFC Mission Services Evolution Center
GOTS	Government Off-the-Shelf
GSE	Ground Support Equipment
FEP	Front End Processor
GRC	Glenn Research Center
GSFC	Goddard Space Flight Center
HOSC	Huntsville Operations and Support Center
IDT	Integrated Discipline Teams
IP	Internet Protocol
IPT	Integrated Product Team
ISS	International Space Station
JPL	Jet Propulsion Laboratory
JSC	Johnson Space Center
KSC	Kennedy Space Center
LCC	Launch Control Center
LEO	Low Earth Orbit
MCC	Mission Control Center
MDS	Mission Data System
MOM	Message Oriented Middleware
MSFC	Marshall Space Flight Center
NASA	National Aeronautics and Space Administration
QOS	Quality of Service
RFP	Request for Proposal
SCAWG	Space Communication Architecture Working Group
SDK	Software Development Kit
SSC	Stennis Space Center
TRMM	Tropical Rainfall Measuring Mission
WBS	Work Breakdown Structure
XTCE	XML Telemetry and Command Exchange



# **Moving Towards a Common Ground and Flight Data Systems Architecture for NASA's Exploration Missions**

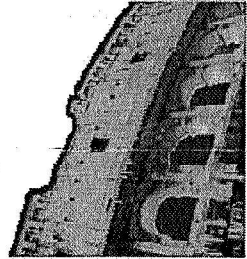
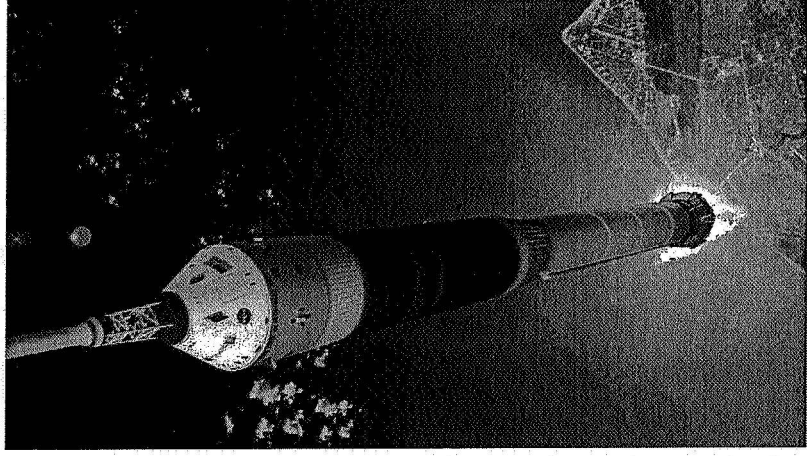
**Co-Authors:**

**Steve Rader, NASA JSC**

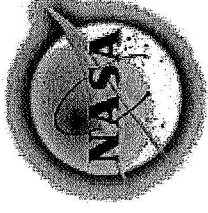
**Mike Kearney, NASA MSFC**

**Thom McVittie, NASA JPL**

**Dan Smith, NASA GSFC**

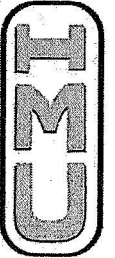
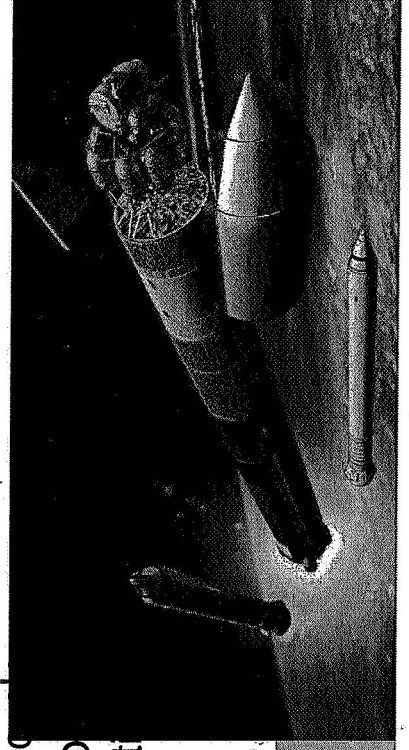


**SpaceOps 2006 Symposia  
Rome, Italy June 19-23**

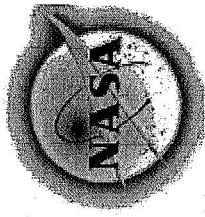


# Program Background

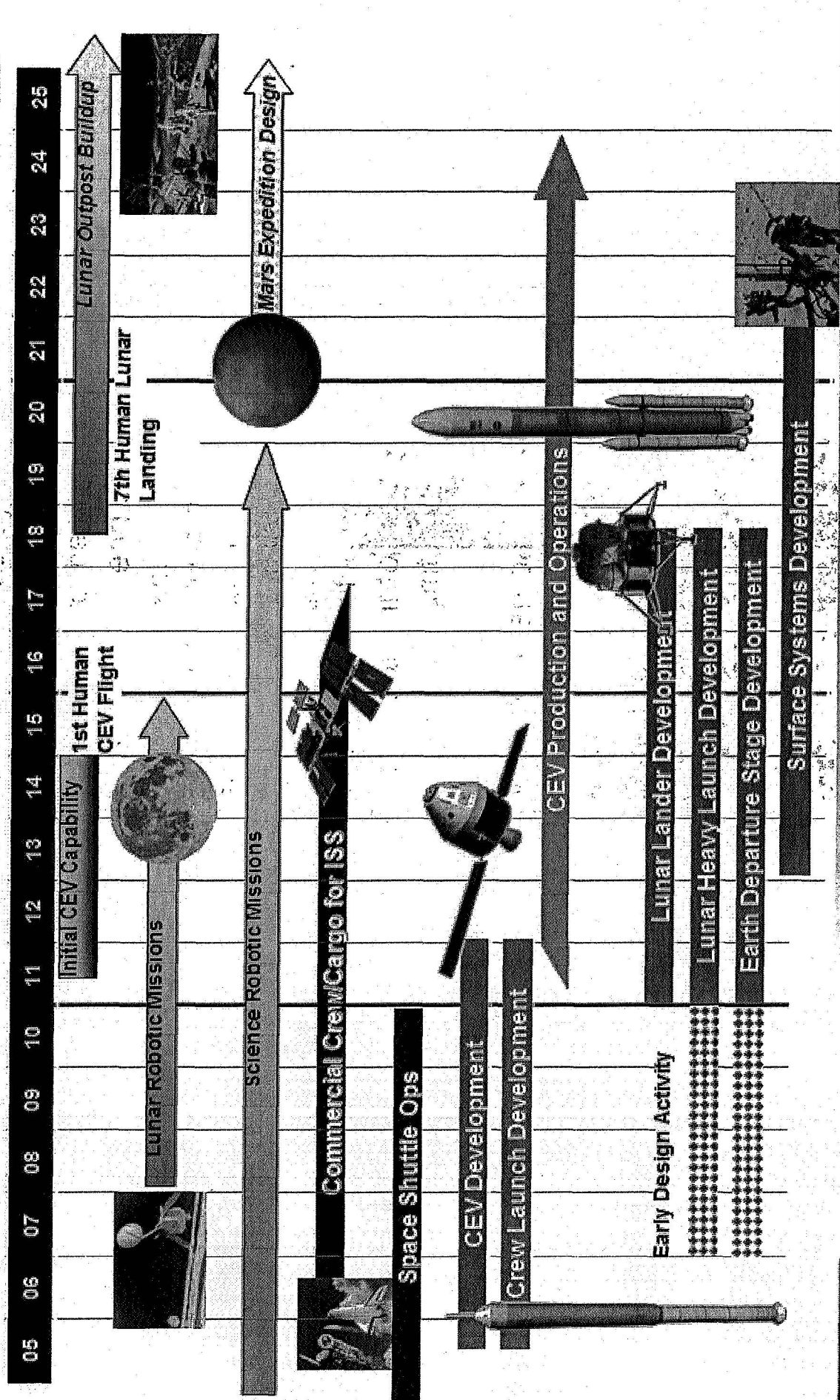
- ◆ NASA's Exploration programs will take man back to the moon, Mars and beyond.
- ◆ NASA is anticipating many new vehicles for robotic and lunar exploration
  - ◆ Crew Exploration and Crew Launch vehicles
  - ◆ Earth Departure Stage and Lunar Surface Access Module
  - ◆ Surface habitats, robotic rovers, communications relays
- ◆ Mission operations will require complex interaction between distant vehicles, and a new communications and data systems approach.
- ◆ NASA vehicles will be supplied by many different contractors
  - ◆ It is possible that other agencies may add to the heterogeneous mix of vehicles
- ◆ NASA has formed a team to address end-to-end data and comm standards
  - ◆ Work is centered in the Constellation program
    - ★ NASA's exploration program to perform manned missions to the moon and Mars.
  - ◆ Focus is on Interoperability using current technologies
  - ◆ Follows approach of net-centric architecture initiatives
  - ◆ Applies to Flight systems and ground facilities
  - ◆ Plan is to standardize early in the program life cycle
- ◆ Communications, Command, Control and Information
  - ◆ Team is the Constellation C3I System Integration Team

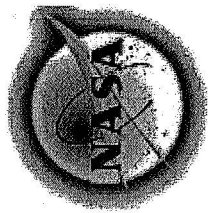






# NASA's Exploration Roadmap



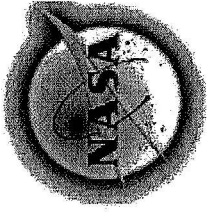


# Caveats and Limitations

- ◆ The C3I Architecture is the current direction of the Constellation Program, but it is still in a formative stage.
- ◆ Further details on this C3I Architecture are not releasable because of competition sensitivity for upcoming procurements.
- ◆ This paper and presentation does not constitute guidance or direction to future program participants.
- ◆ Future direction will be to publish an "Interoperability Specification" that applies to all vehicles and facilities in the Constellation program.
  - ◆ When the program is able to release that specification, broader participation and review will be possible.
- ◆ Currently "Interoperability" is addressed within the NASA program only because there are no international agreements for participation in the Exploration programs.
- ◆ Where functions can be met by current international standards, they are given high priority for inclusion in the C3I approach.







# Technical Background and Environment

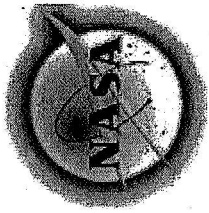
- ◆ NASA anticipates that the Exploration programs will include a large assemblage of free-flying spacecraft that require connectivity

Launch vehicles	Manned transports	Logistics supply craft	Robotic rovers
Lunar ascent/descent	Mars ascent/descent	Surface habitats	Orbital relays
EVA (spacesuits)	Moonbuggies	Surface science	Surface relays

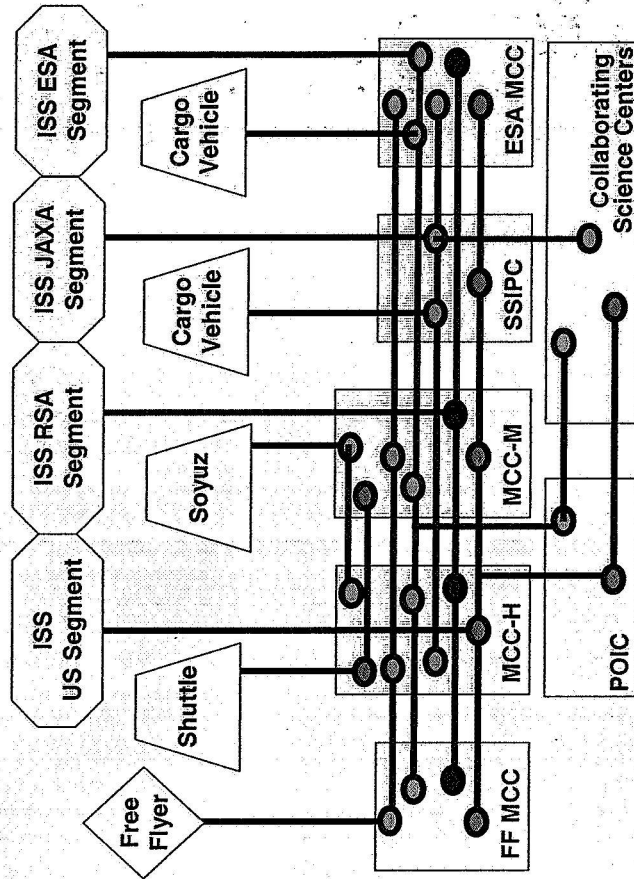
- ◆ Interoperability between mobile vehicles must be greater than prior programs.
- ◆ Evolutionary nature of program requires “loose coupling”
  - ◆ So that newly designed vehicles can arrive at destination and “link up” without redesigning interfaces.
  - ◆ Expect interoperability specification evolution over time
  - ◆ Need to tailor spec for some spacecraft (robots don’t need voice)
- ◆ Many good standards exist
  - ◆ Adopt existing int’l and de facto standards when possible
  - ◆ Adapt (modify) them when necessary
  - ◆ Develop new standards in areas that existing standards don’t cover
- ◆ Interoperability should solve some problems encountered in prior programs (e. g. ISS).



# Interoperability – the problem statement



Current (ISS Program)



Current ISS telemetry flow.

Each color is a different telemetry format.

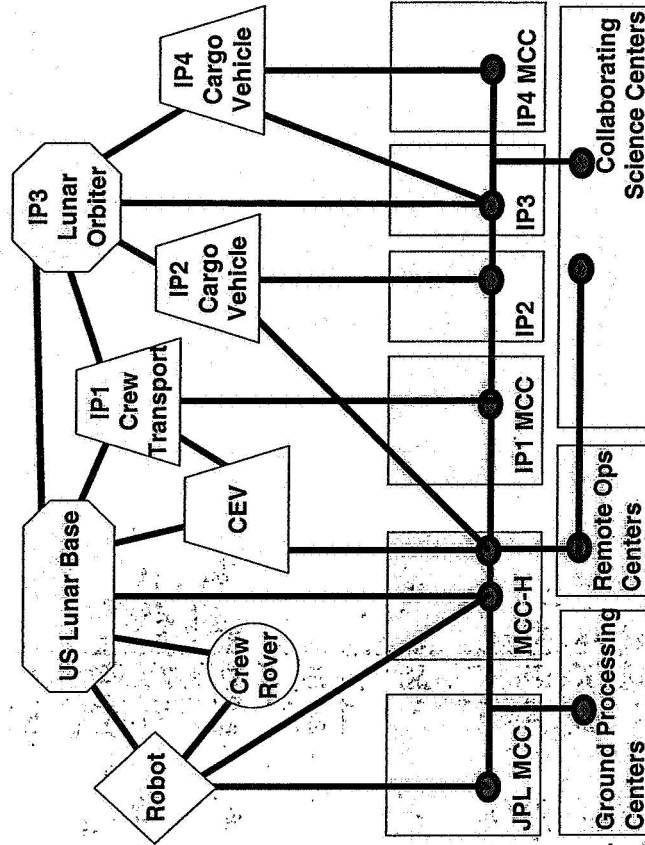
Partial – the real world is actually more complex.

Other data types (command, etc.) are equally incompatible.

Multiple colors in any one box imply excessive cost.

Colors missing from any box imply operations handicaps.

Future (Exploration programs)

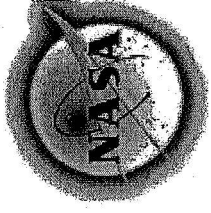


Conceptual architecture for the future with interoperability problem solved.

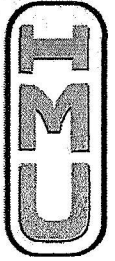
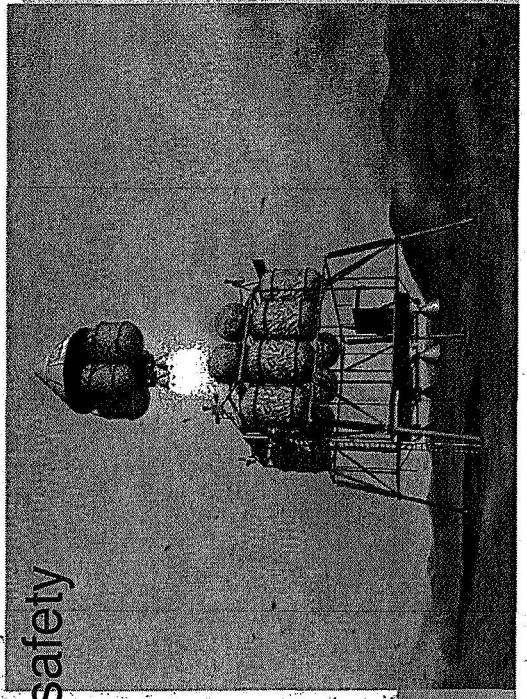
IP1, IP2, etc. represent potential International Partners in Exploration.

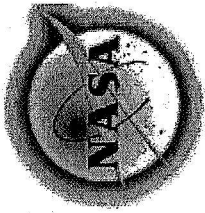


# Architecture Drivers



- ◆ Unification – Mission elements must be unified by a foundation of interoperability.
- ◆ Common data formats for primary ops data types
  - ✦ Command, telemetry, voice, etc.
- ◆ Data structures defined separately from processing logic
- ◆ General purpose data-driven applications support multiple vehicles via data reconfiguration (vs. SW development)
- ◆ Higher level of abstraction for communications protocols
  - ✦ Standardized well-understood interface
  - ✦ Draw from commercial approaches (message bus, etc.)
- ◆ Result: ↑flexibility = ↑capability = ↑safety

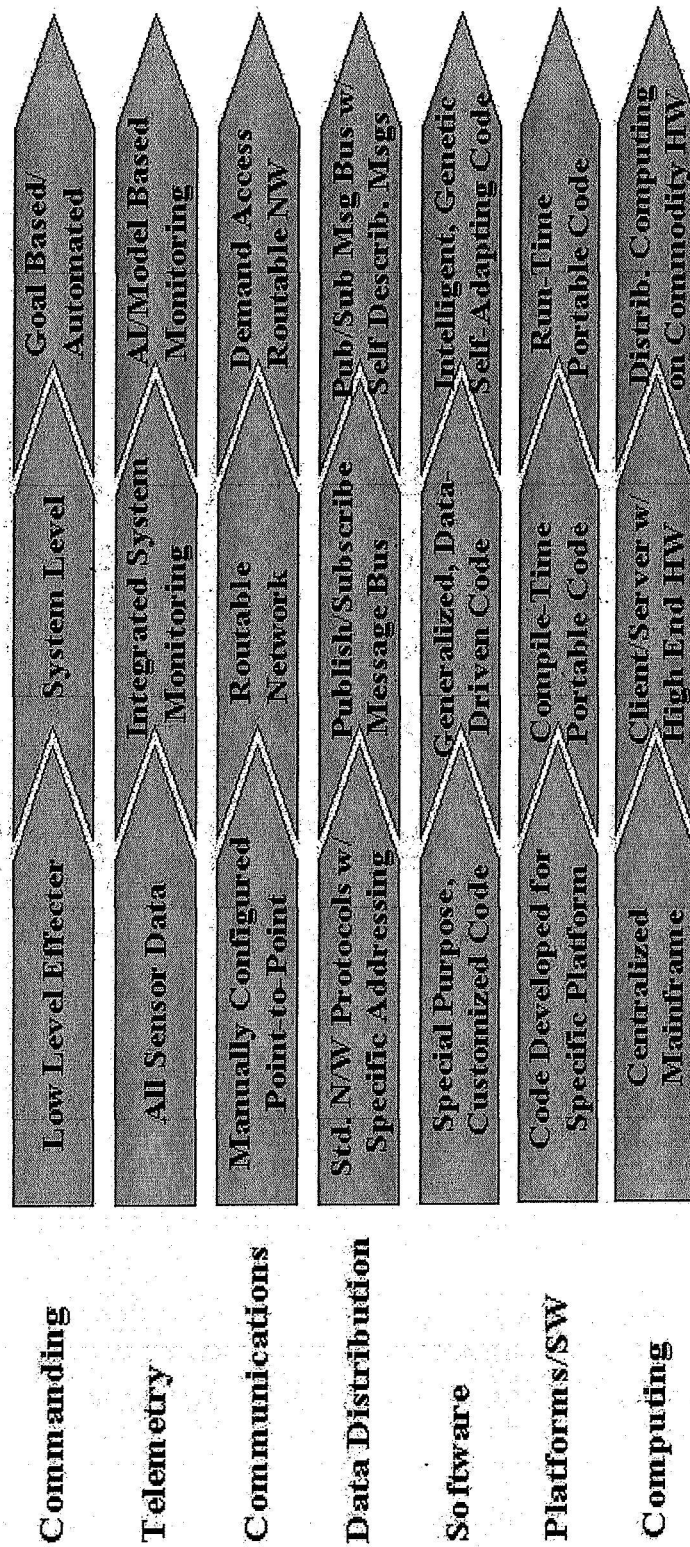




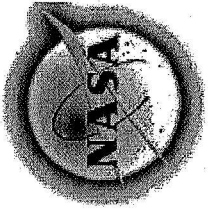
# Architecture Drivers (Cont.)

- ◆ Unification (Cont)
- ◆ This approach parallels current technology trends in the commercial world
  - ✦ Higher levels of abstraction → flexible building blocks
  - ✦ Reducing unique interfaces → simpler interoperability

## Technology Trends to Higher Levels of Abstraction

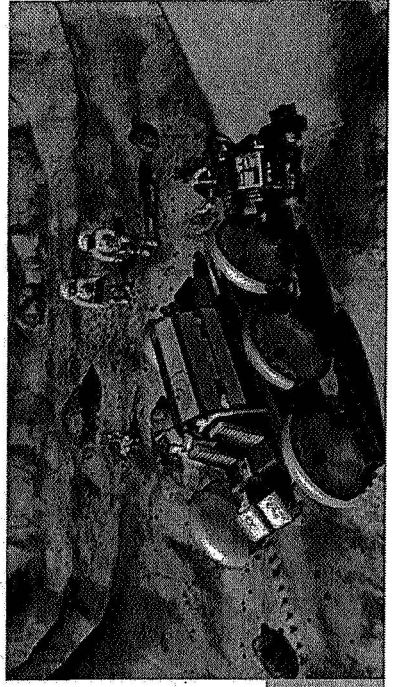


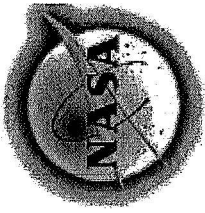




# Architecture Drivers (Cont.)

- ◆ Flight Worthiness
  - ◆ Unification means architecture components must work for both flight and ground systems
  - ◆ Flight qualifiable requires small memory footprint for spacecraft operating systems, etc.
  - ◆ Must support direct interoperation of flight elements without ground system involvement (many light-minutes away, etc.)
- ◆ Adaptability – Near Earth to Mars environment
- ◆ Flexibility – Allow introduction of new technology
  - ◆ Layered, modular approach allows “swap out” of layers
- ◆ Scalability – Supports large mission operations facility or small single remote user needs



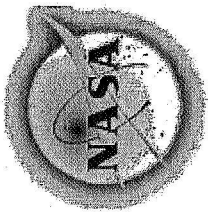


# Architecture Drivers (Cont.)

- ◆ Cost Effectiveness
- ◆ Development – Must allow separately developed components to be integrated easily
  - ✦ Provides layer of isolation
  - ✦ Reduces inter-component dependencies
- ◆ Operations – Must allow streamlined reconfiguration (in days) to replace software development (in months or years)
- ◆ Must enable (but not require) common equipment and software usage at multiple locations (facilities, spacecraft)
  - ✦ Early usage of multiple innovative tools at multiple locations is allowed.
  - ✦ Later, interoperability supports consolidation down to most effective tool used at multiple locations.
- ◆ Common use of open standards wherever possible

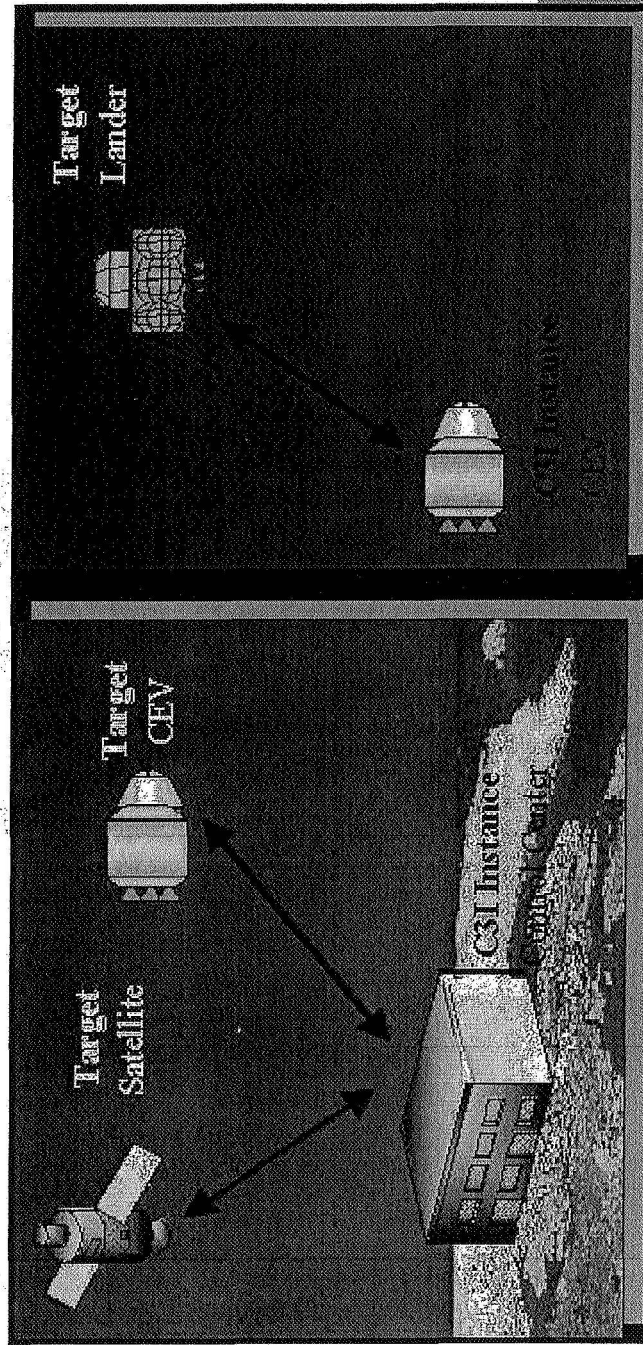


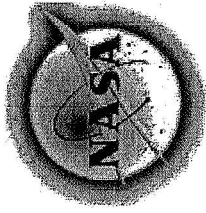




# Architecture Overview

- ◆ “Instance” of C3I architecture may be control center, vehicle, test/checkout stand, etc.
- ◆ A controlling instance can control target instances (vehicles, other ground equipment, etc.)
- ◆ C3I compliant targets have full interoperability
  - ◆ “Legacy” targets may use C3I “adapters” and may not have full interoperability.





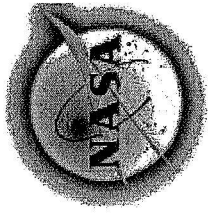
# Architecture Overview

## ◆ Main Architectural Features

Feature	Description	Benefit
Layered Partitioning of Functionality	Isolates changes and simplifies implementation and control of common services.	Increased capability over time due to technology advancements that can be more easily incorporated into the systems.
C3I Messaging Middleware based Framework	Standardizes application's interfaces with the system and with each other while providing efficient network data distribution.	Reduces development integration costs, enables more efficient small development teams, and provides more efficient information distribution.
Integrated Information Model	Provides enterprise information access and multiple vehicle support with the use of data driven apps.	Reduced ops. costs from workflow automation and more efficient information management.
Standard Command & Control Method	Provides a standard interface and logical protocol for implementing command and control functions.	Reduces training and simplifies operations (increasing safety, ops. reliability) across the increased number of vehicles, systems.
Integrated Space and Ground Network	Same protocols flight/ground allows elements to exchange information efficiently and effectively.	Reduced cost of operating the network with smarter, more efficient utilization of the communications links.

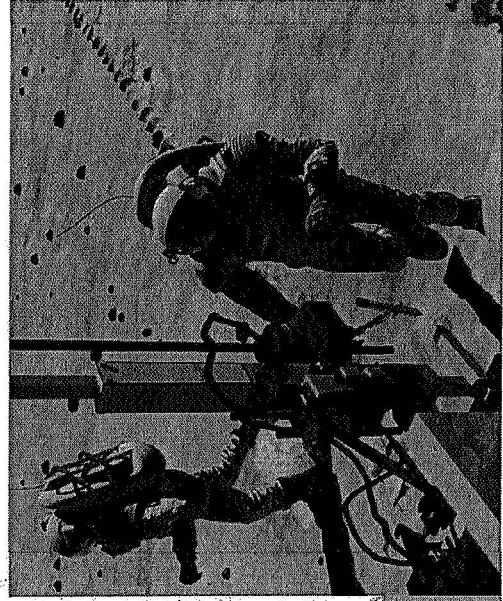






# Approaches and Concepts

- ◆ Develop loosely coupled interoperable systems.
- ◆ Allows small development team “cells”, reduces integration costs, increases response time for system changes.
- ◆ Layered partitioning of functionality isolates the effects of changes between system components.
- ◆ Allows applications that send data to be “unaware” of applications that receive data. Reducing interface costs.
- ◆ Use systems based on open standards.
- ◆ Standardizing comm protocols for space-based usage and ground-based usage effectively creates integrated space/ground network.
- ◆ Migrate towards commodity-based systems where possible



## Slide 13

---

K1

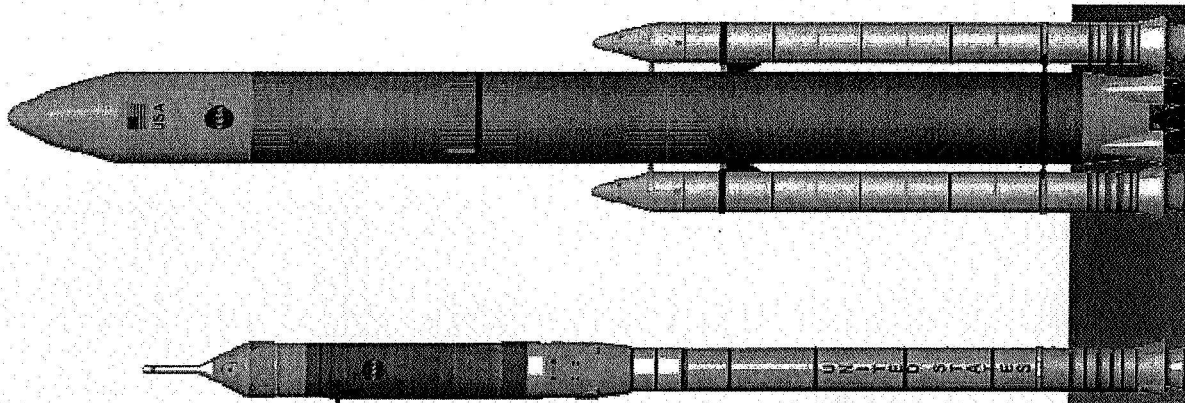
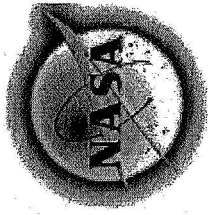
This stuff seems to be a repeat of the architecture driver material, at a time when we should be getting into the technical meat.  
Should maybe delete or consolidate with earlier section.

Kearney, 5/20/2006



## Approaches and Concepts (Cont.)

- ◆ Consistent Command and Control across vehicles and facilities.
- ◆ Standard model applied to all C3I-compliant elements.
- ◆ New approach: use virtual system representations that exist as data applied to generalized data-driven applications.
- ◆ Switched and Routed Networks
  - ◆ The terrestrial IP routed network is effective and ubiquitous. It enables capabilities and cost-effectiveness that is needed in the Exploration programs.
  - ◆ Automated routing makes the entire space/ground network addressable and routinely accessible.



## Slide 14

---

K2 I need help explaining "virtual system representations that exist as data".  
Kearney, 5/20/2006



# Architecture Layers

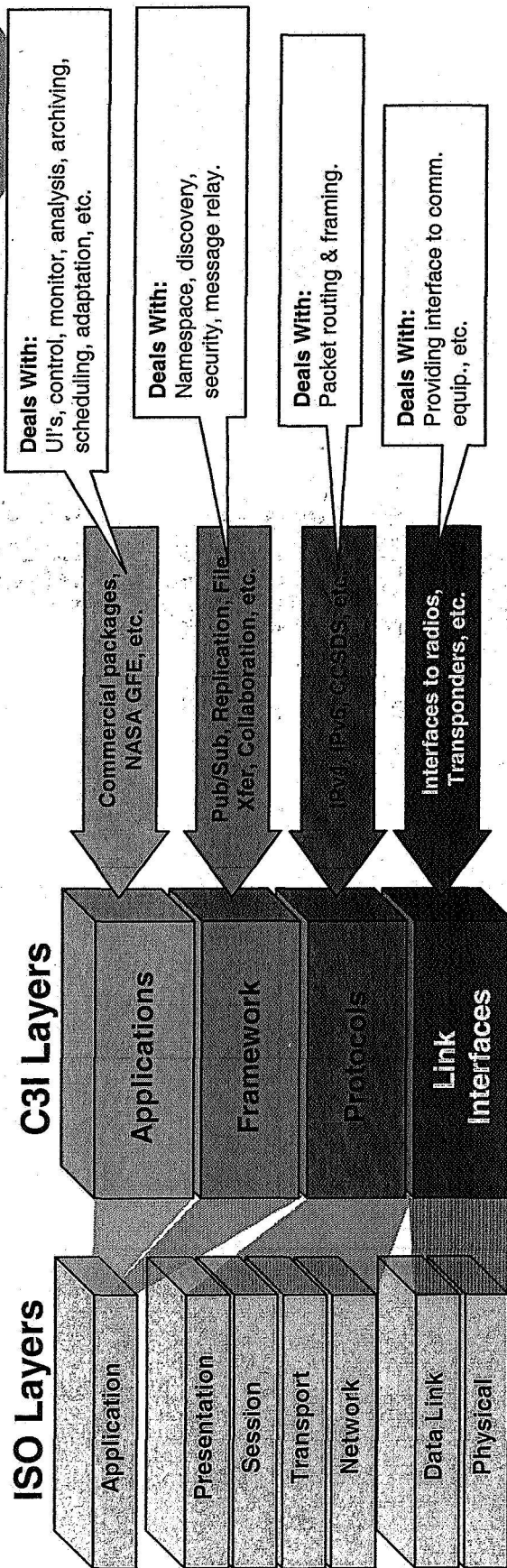
- ◆ Objective: Simplify, decouple functions, leverage new technologies
- ◆ Allow one layer to change with new technology, not affecting other layers or applications except when using new functions.
- ◆ Framework is critical new layer (ISO application function)

## Tenets

**Keep it simple.**

**Decouple functions.**

**Leverage modern technologies.**



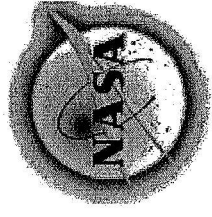
## Slide 15

---

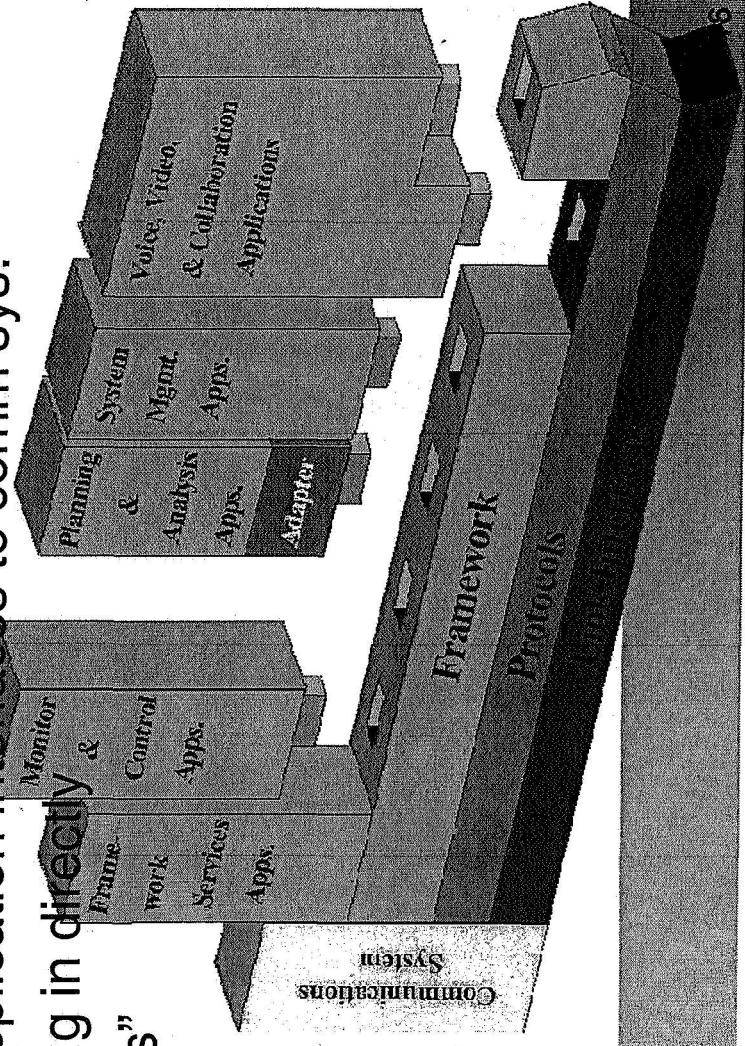
- K3**      Need to consolidate these two figures into one. Need original artwork.  
Kearney, 5/20/2006



# Architecture Layers (Cont.)

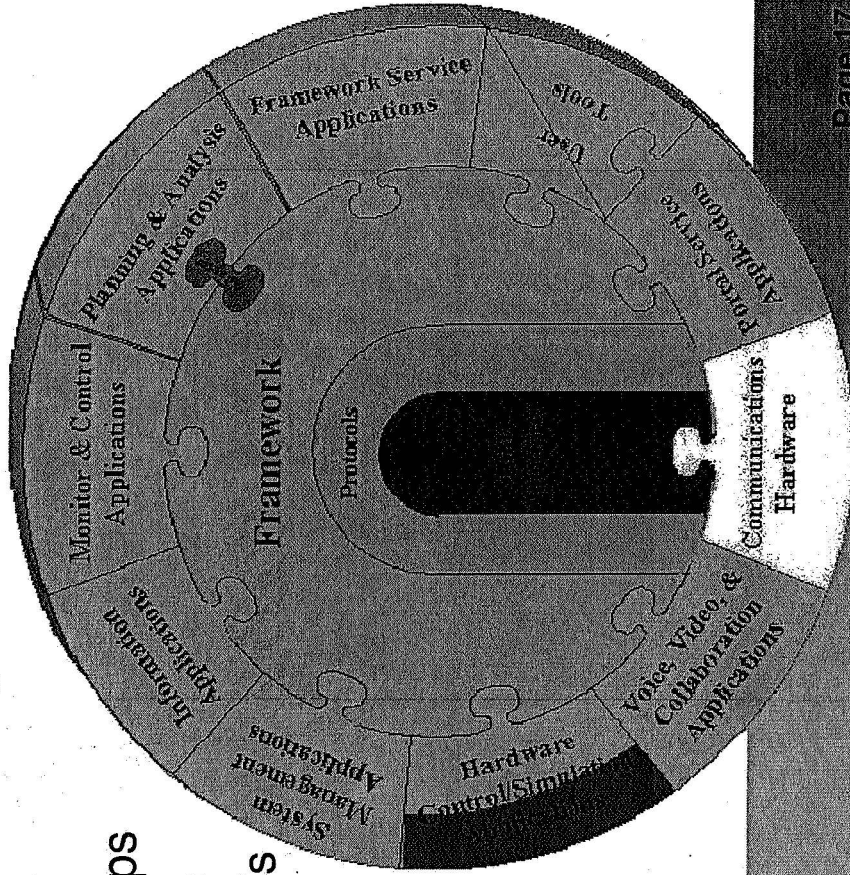
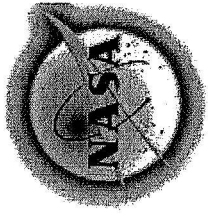


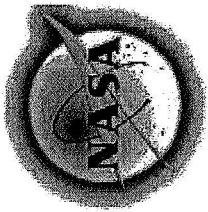
- ◆ Application Layer
- ◆ C3I will enable ability to standardize core applications
  - ✦ For Command and Control, analysis, planning and info mgt.
  - ✦ For voice, video, collaboration, workflow automation, etc.
- ◆ C3I also enables a potentially large library of certified apps available for “shopping list”
  - ✦ Similar to results experienced in NASA’s GMSEC environment
- ◆ Plug-n-Play approach to application interfaces to comm sys.
- ◆ Native C3I applications plug in directly
- ◆ Legacy apps use “adapters”
- ◆ Expect ability to “tailor” to support unique needs
  - ✦ Not all instances need all applications



# Architecture Layers (Cont.)

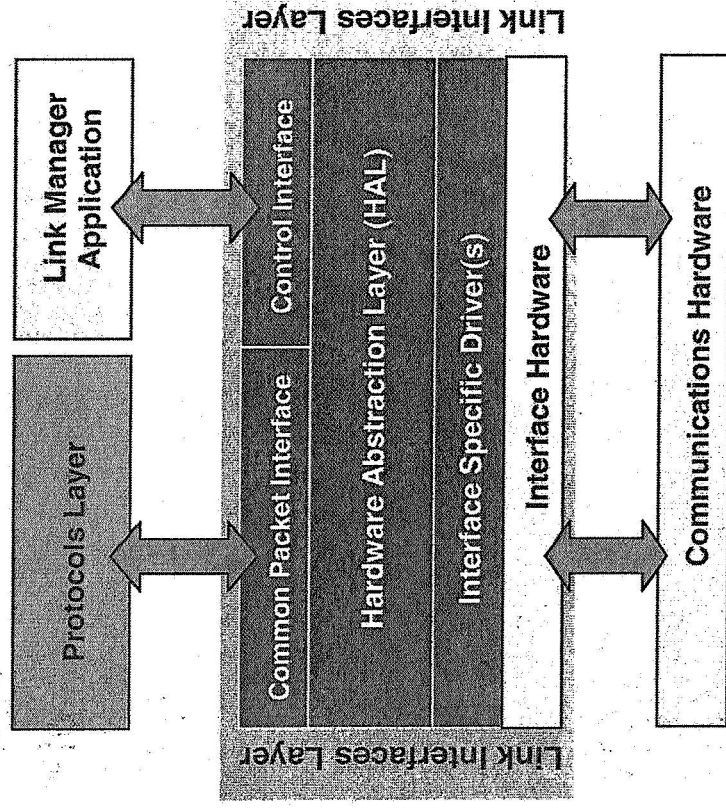
- ◆ Framework Layer
  - ◆ Standardizes application interaction with the system
  - ◆ Publish and subscribe model of message bus technology
    - ✦ Transmitters are not sensitive to receiver interfaces
  - ◆ Provides mechanisms for self-describing data formats
  - ◆ Provides integration services
    - ✦ Security “hooks” for security apps
    - ✦ Discovery and request services
    - ✦ Standard API for application I/Fs
  - ◆ “Adapters” may be needed for legacy applications





# Architecture Layers (Cont.)

- ◆ Protocols Layer
  - ◆ Integrates various protocols (ISO layers 3 to 6) needed for the diverse comm needs of Exploration.
  - ◆ Layering allows selected protocol stack to change without affecting higher-level frameworks and applications
- ◆ Link Interfaces Layer
  - ◆ Foundation communications services and equipment
  - ◆ Will not be a common set on all elements
  - ◆ Will evolve to future SW based radios, optical comm links, etc.
  - ◆ Common “wrapper” around communications hardware

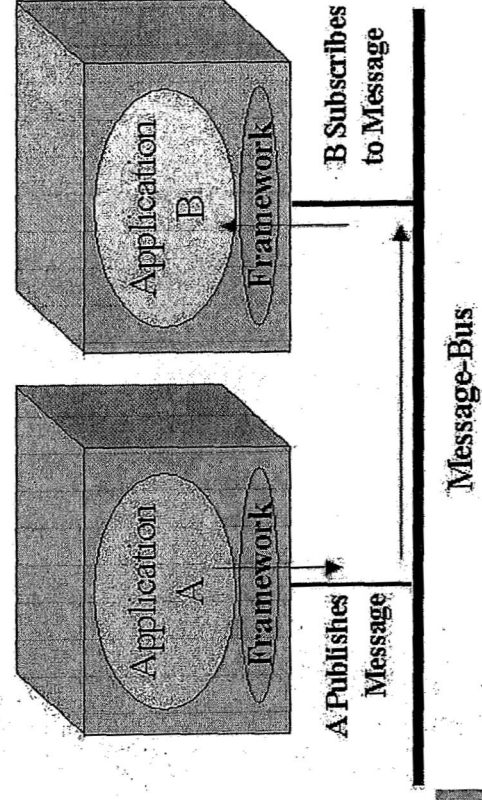






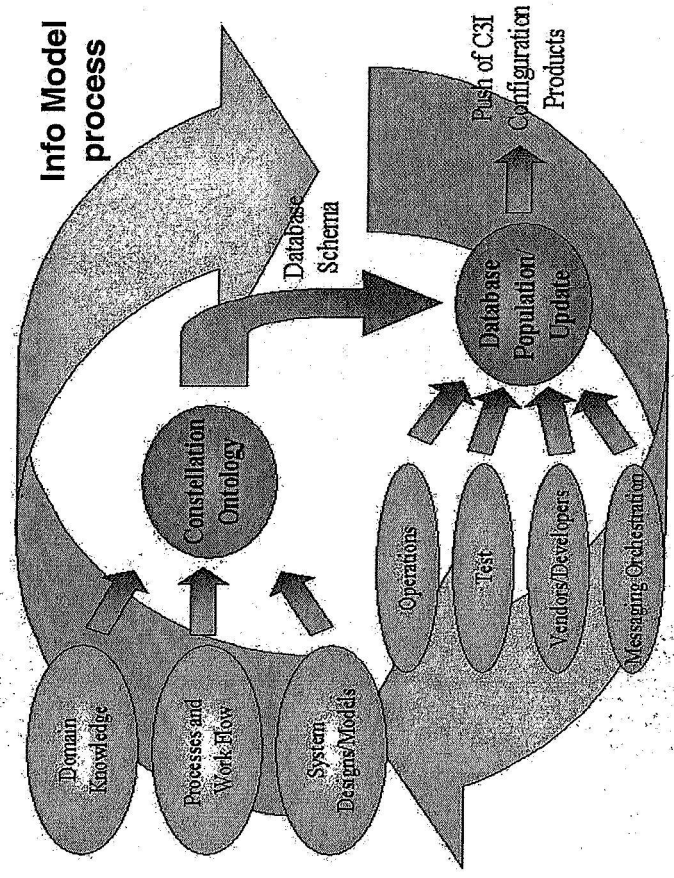
# Messaging Middleware-based Framework

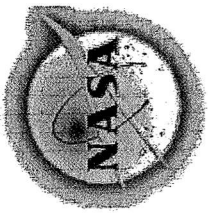
- ◆ Centerpiece of new C3I architecture concept
- ◆ Middleware between protocols and applications layers
- ◆ Key functions: Raises communication to higher level of abstraction, provide benefits of loosely coupled architecture
- ◆ Multiple message busses are available (CCSDS AMS, GMSEC, Commercial products), so framework will provide an insulating layer between message bus and applications.
- ◆ SDK and API definitions will support application development teams
- ◆ Applications inherit framework services, simplifying appl. development
- ◆ Info access, security control, testing, configuration setup, scripting, monitor and control interfaces



# Information Model

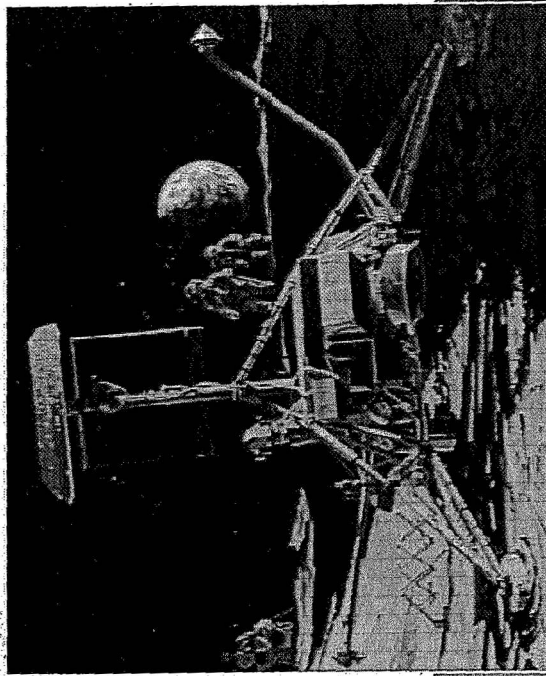
- ◆ Data Driven approach: specific vehicle or equipment characteristics are not “hard-wired” in applications, rather contained in data loaded at “run time” by generic applications.
- ◆ Managing that data is the purpose of the information model.
- ◆ Integrates traditional spacecraft data (e.g. measurement lists) with other operations data (e.g. planning data and other enterprise data (e.g. design and cost models).
- ◆ Adds metadata under total program ontology
- ◆ Provides enterprise level access to data via framework message bus.
- ◆ Applications can discover and request data using Info Model.
- ◆ Info Model data definitions become the main interface definitions between applications.





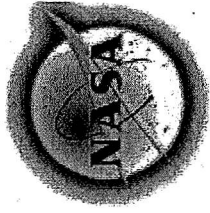
# Standard Command and Control Method

- ◆ The Exploration scenario of many vehicles managed by multiple teams on the ground absolutely requires a common C&C mechanism.
- ◆ For end-to-end system up to the vehicle interface.
  - ✦ human interfaces, command logic, automated scripts, etc.
- ◆ Standardization goes to the interface with the onboard system but not inside the onboard system.
- ◆ Common approach for C&C of one-to-many and many-to-one
- ◆ Data from the Info Model is used to configure command applications for command sessions.
  - ✦ Total info model support enhances accuracy of control action
- ◆ C3I Infrastructure provides security
  - ✦ Authentication, authorization, etc.



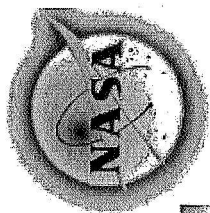


# Integrated Space and Ground Network

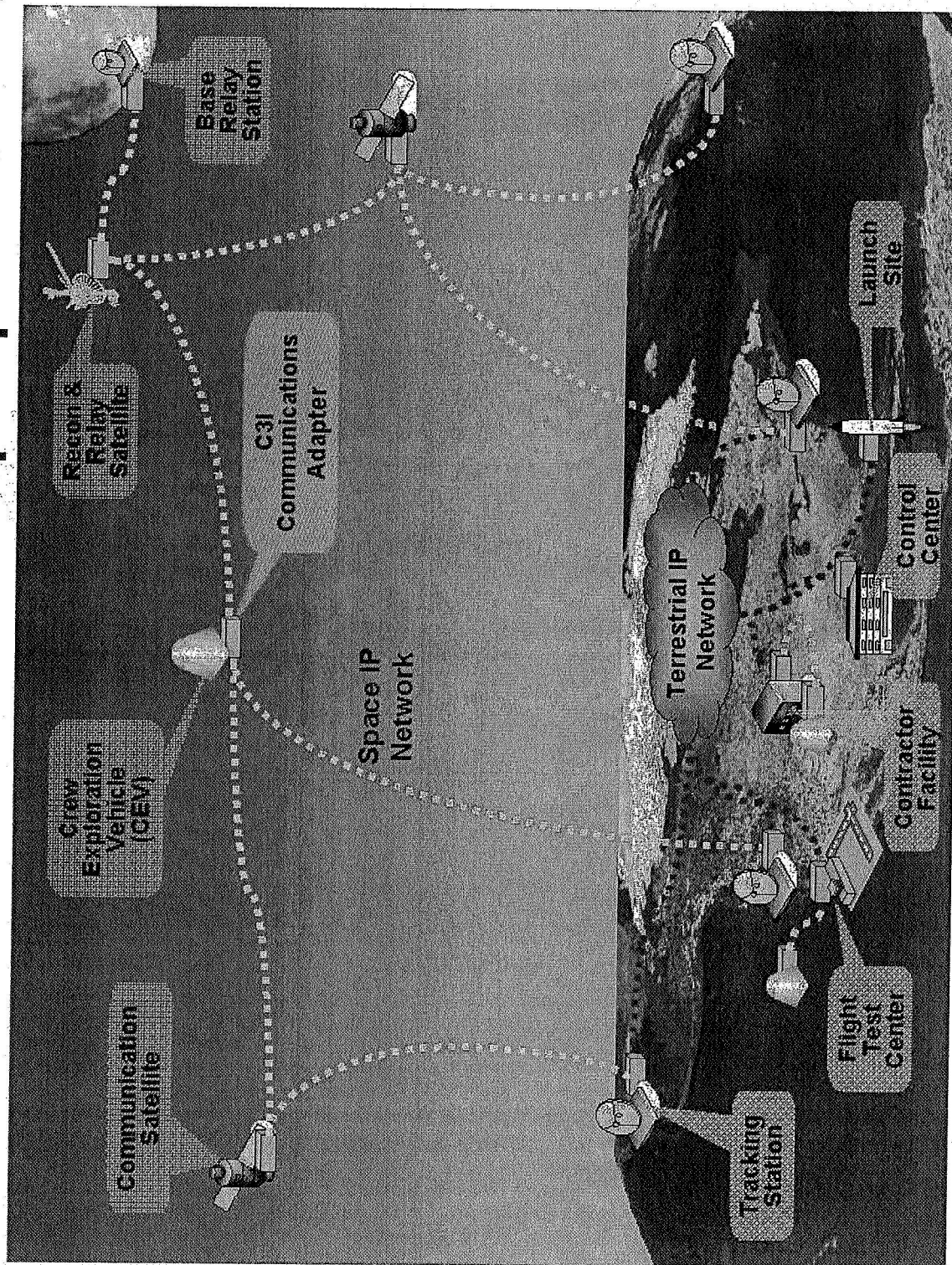


- ◆ C3I Communications Adapter (CCA) will be deployed on elements of the C3I architecture.
- ◆ Provides relay, routing, store & forward, Quality of Service and link management capabilities within each element.
- ◆ As more space and ground elements adopt the CCA, the integrated end-to-end space and ground network emerges.
- ◆ Will use appropriate industry and CCSDS protocols and framing.
- ◆ Will develop the space IP network as more nodes are deployed.





# Integrated network – Terrestrial perspective

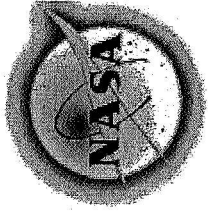




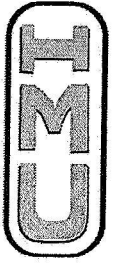
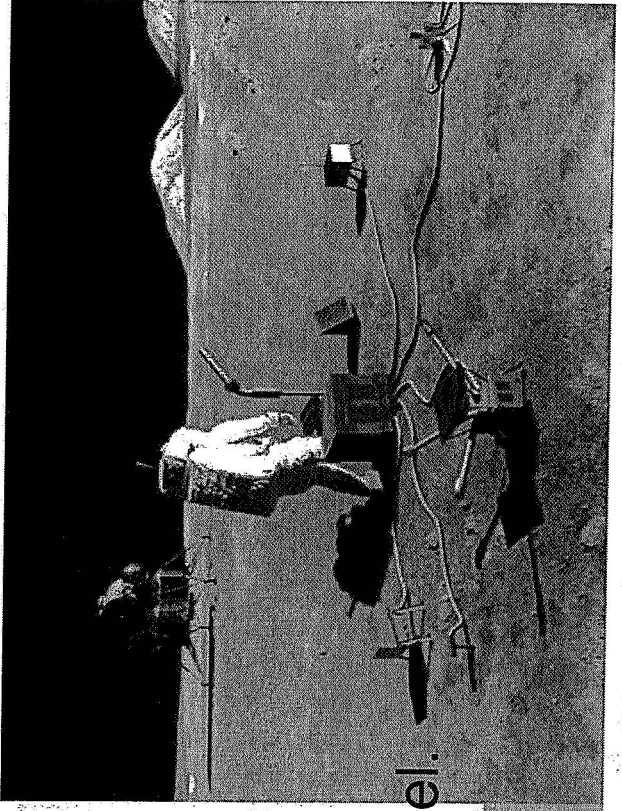


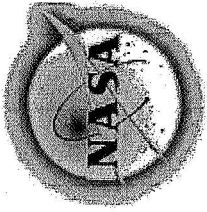


# The Architecture In Perspective



- ◆ Management perspective: Flexible support for an evolving program. Max interoperability allows minimum duplication in control facilities, reducing cost.
- ◆ User perspective: Common tools for all environments (large/small, Earth/Mars), less retraining, more automation.
- ◆ Developer perspective: Data-driven simpler application development, simple and universal external interface.
- ◆ Deployment perspective: Simpler to deploy anywhere there is a framework interface, multi-center, multi-agency, etc.
- ◆ Configuration management perspective: More capable yet equally secure under the info model.





## Conclusion

- ◆ Proposal for C3I constitutes new approach for new era of Space Exploration.
- ◆ Loosely-coupled data-driven interoperability
- ◆ More flexible, more capable and safer operations at a reduced cost

It is time for spaceflight development and operations to step forward using (and adapting when necessary) the methods and tools that are widely recognized for their benefits in other industries.

