



An Open Avionics and Software Architecture to Support Future NASA Exploration Missions

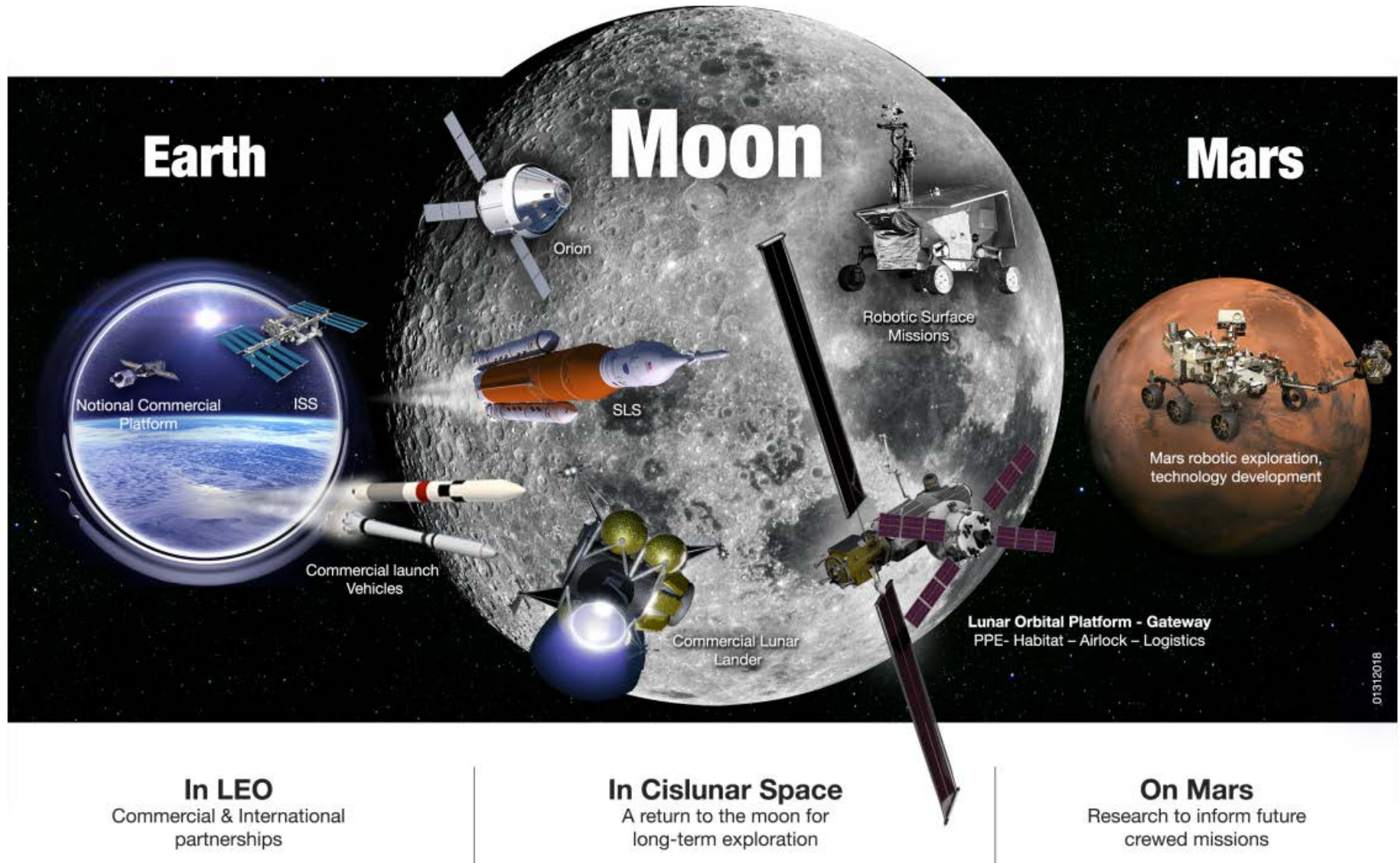
SPACE COMPUTING & CONNECTED
ENTERPRISE RESILIENCY
CONFERENCE 2018

June 5, 2018

*Adam Schlesinger, Manager (Acting), AES Avionics & Software Project
NASA-Johnson Space Center (JSC), Houston, TX*



NASA Exploration Campaign



Lunar Orbital Platform – Gateway Development



LUNAR ORBITAL PLATFORM-GATEWAY DEVELOPMENT

Establishing leadership in deep space and preparing for exploration into the solar system

FOUNDATIONAL GATEWAY ELEMENTS

2022

2023

2024+



These foundational gateway capabilities can support multiple U.S. and international partner objectives in cislunar space and beyond.

20180327

CAPABILITIES

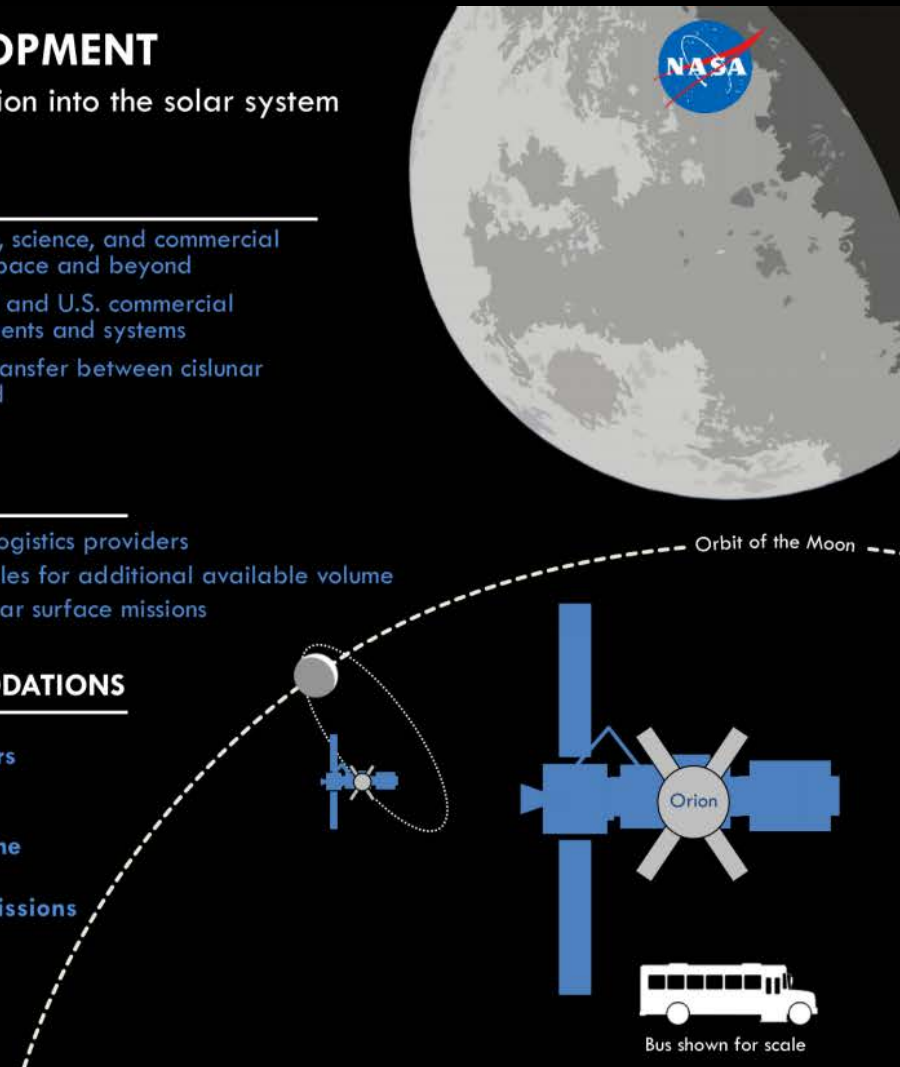
- Supports exploration, science, and commercial activities in cislunar space and beyond
- Includes international and U.S. commercial development of elements and systems
- Provides options to transfer between cislunar orbits when uncrewed

OPPORTUNITIES

- Logistics flights and logistics providers
- Use of logistics modules for additional available volume
- Ability to support lunar surface missions

INITIAL ACCOMMODATIONS

-  **4 Crew Members**
-  **At least 55 m³ Habitable Volume**
-  **30 Day Crew Missions**



Advanced Exploration Systems (AES) Division



- **NASA's Advanced Exploration Systems (AES) division is pioneering innovative approaches and public-private partnerships to rapidly develop prototype systems, advance key capabilities, and validate operational concepts for future human missions beyond Earth orbit**
- **AES activities are related to crew mobility, habitation, vehicle systems, robotic precursors, and foundational systems for deep space**
- **AES infuses new technologies developed by the Space Technology Mission Directorate and partners with the Science Mission Directorate to address Strategic Knowledge Gaps for multiple destinations**
- **AES is leading NASA's Lunar Orbital Platform – Gateway (LOP-G) Efforts**





- **AES Avionics & Software (A&S) Project Goal:**

- Define and exercise an avionics architecture that is open-source, highly reliable with fault tolerance, and utilizes standard capabilities and interfaces, which are scalable and customizable to support future exploration missions

- **A&S Drivers:**

- **Technology Transparency**

- The underlying hardware should not have any impact on an application either during development or execution
- Code reuse and portability

- **Reliability and Maintenance**

- Operate in the presence of failures so that Maintenance Free Operating Periods (MFOPS) can be achieved
- Provide autonomous operations
- Minimal number of unique spares

- **Incremental Update & Certification - Designed for Growth and Change**

- Applications can be inserted/alterd with minimum impact on other systems and on the supporting safety case
- Flexible scheduling to meet the deadlines of all the applications for each viable configuration and when system is upgraded

AES Avionics & Software (A&S) Project



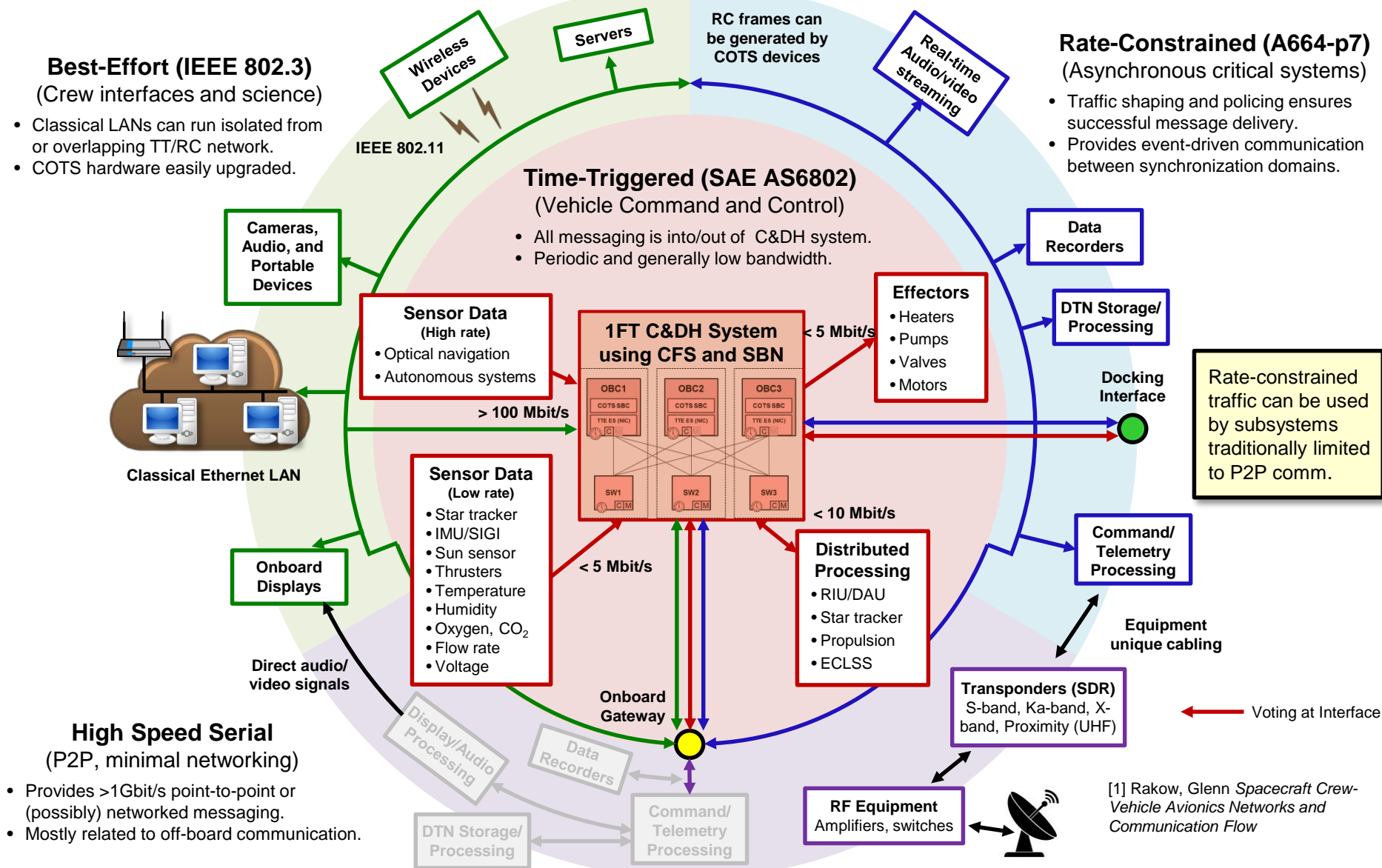
- **A&S Focus Areas and Objectives:**

- **Command & Data Handling (C&DH)** - Define a reliable, high-performance & modular C&DH architecture and build HW catalog
- **Software** - Provide a reusable software architecture and tools suitable for human-rated missions
- **Human Interfaces** - Identify, integrate & test human interface technologies that are scalable, sustainable, and evolvable to support future exploration
- **Communication and Wireless Systems** - Enable interoperable, wireless & networked communication for inter/intra-vehicle systems
- **Systems Engineering and Integration (SE&I)** - Model, build & test flexible and robust integrated vehicle systems

- **A&S Benefits:**

- Results in an open architecture that allows use of hardware from multiple vendors
- Enables use of evolving (near-launch) technology
- Ability to upgrade capabilities and infuse new technologies with cost-effective validation

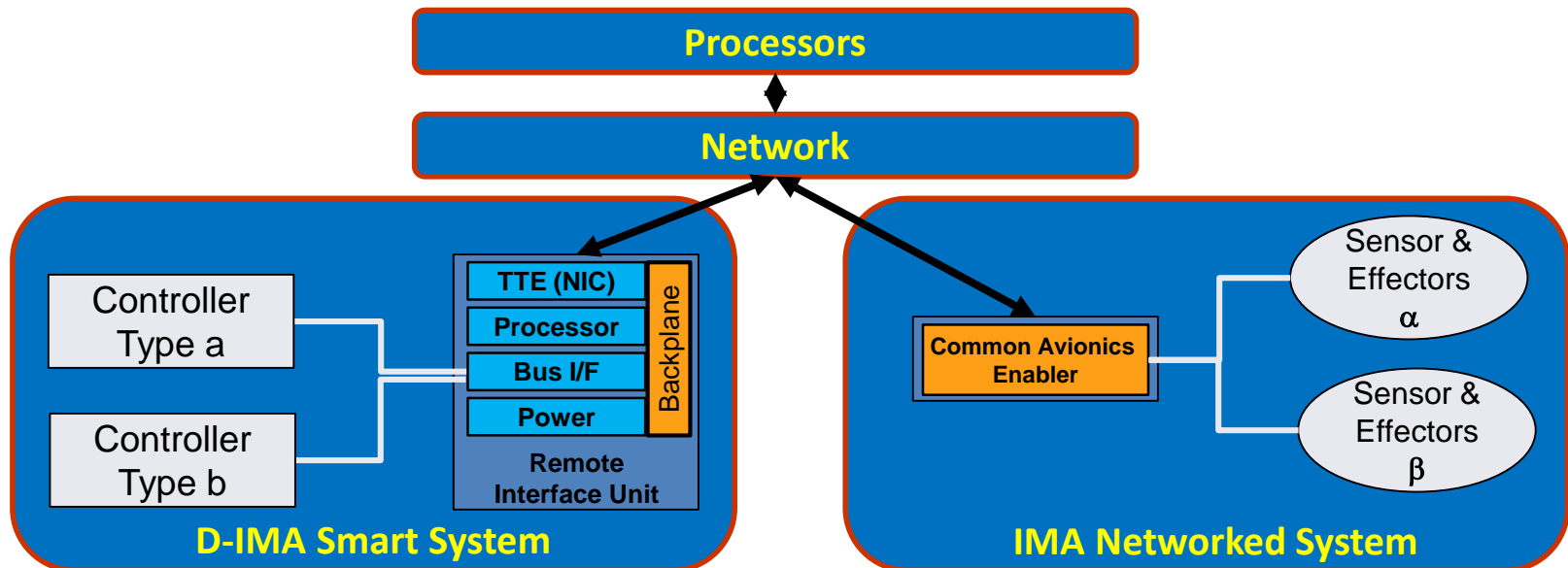
Avionics & Software Architecture



Blueprint of the Architecture: Distributed Integrated Modular Avionics (D-IMA)



	Description	Rationale
Distributed	<ul style="list-style-type: none"> Resources (both hardware and software) are physically distributed 	<ul style="list-style-type: none"> Reduce harness mass Provide for local control of local functions Lowers flight computer load
Integrated	<ul style="list-style-type: none"> Multiple functions of different criticalities running on separate, high integrity, partitions 	<ul style="list-style-type: none"> Re-locatable functions not limited to a single line replaceable unit (LRU) boundary
Modular	<ul style="list-style-type: none"> Standard interfaces/Virtual Backplane Avionics boxes built up of hub card(s), power supply(s) and subsystem slices Software constructed of re-locatable modules 	<ul style="list-style-type: none"> Provides for composability Allows for hardware reuse Allows for software reuse
Avionics	<ul style="list-style-type: none"> Board level building blocks used to assemble boxes into systems 	<ul style="list-style-type: none"> Allows for multiple vendor hardware components

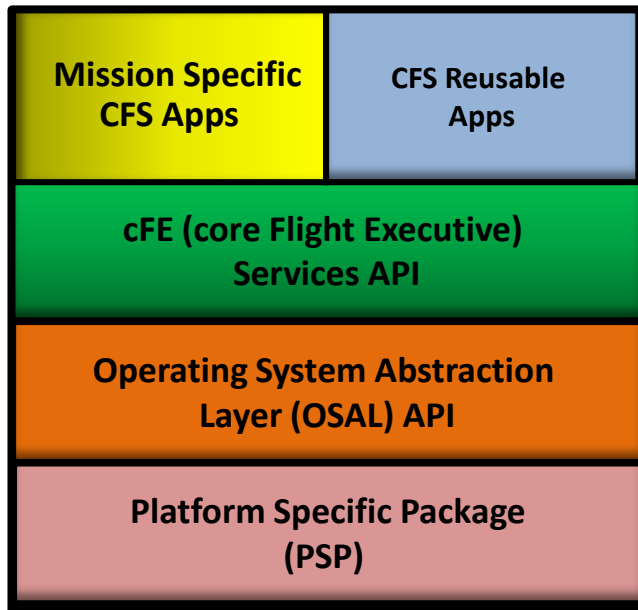


Brain of the Architecture: NASA's Core Flight Software (CFS)

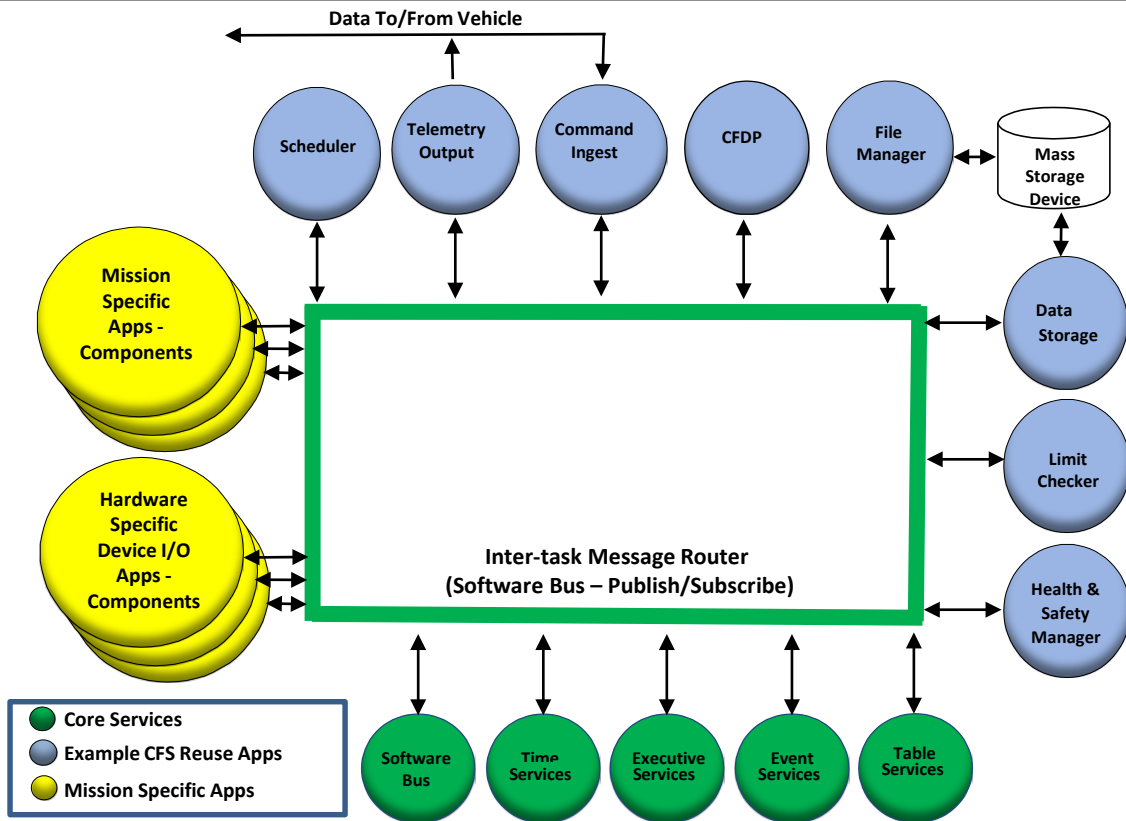


- **Core Flight Software (CFS) is a NASA-developed asset for spacecraft flight software reuse that is available as open-source:**
 - <http://sourceforge.net/projects/coreflightexec/>
- **Productized real-time flight software developed over several years by Goddard Space Flight Center to serve as reusable software framework basis for spacecraft missions, test missions, real-time systems**
- **AES has since advanced the product line, including achieving human-rating, as a reusable software solution for future exploration missions**
- **CFS provides a published service layer (cFE) and an Operating System Abstraction Layer (OSAL) for common services to run on multiple platforms and with several operating systems**
 - Pub/Sub message bus, time services, events, tables, file, task execution
 - <http://sourceforge.net/projects/osal/>
- **CFS provides common reusable spacecraft functions as open-source or government-purpose applications**
 - Scheduler, commanding, telemetry, communication, data recording, limits, system health, sequences

Core Flight Software Framework- Architectural Layers



Notional CFS Application Software Architecture

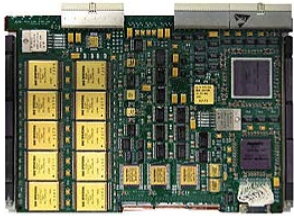


- CFS also provides a mechanism to link distributed CFS instances called Software Bus Network (SBN), including an SBN library (SBN-lib) for non-CFS applications that need access to software bus data

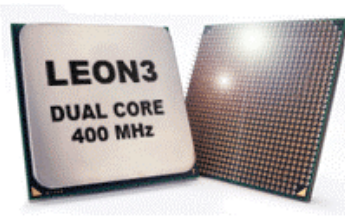
Some CFS Supported Platforms: Non-Exhaustive



- CFS has been ported to work on many processors...



BAE RAD750



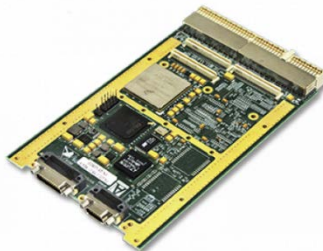
LEON3



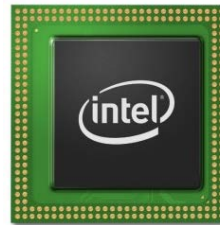
Space Micro Proton 400K



Raspberry Pi



AITECH SP0-100

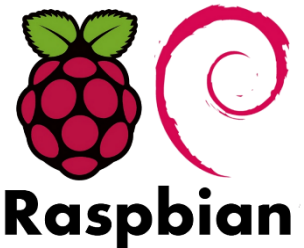


Intel x86



Maxwell SCS750

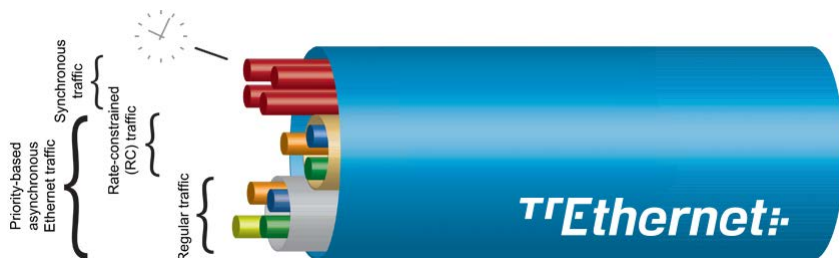
- ...and with many operating systems, both real- and non-real-time



Backbone of the Architecture: Time-Triggered Ethernet



- **Time-Triggered Ethernet (TTE) is compatible with, can coexist with on the same physical media, and expands classical Ethernet with services to meet time-critical or deterministic conditions, supporting three message types:**
- **Time-triggered (SAE AS6802): Sent over the network at predefined times and take precedence over all other message types**
 - Occurrence, delay and precision of messages are predefined and guaranteed
- **Rate-constrained (ARINC 664 p7): Used for applications with less stringent determinism and real-time requirements**
 - Bandwidth is predefined and guaranteed for each application and delays/jitter have defined limits
- **Best-effort (IEEE 802.3): Follow classical Ethernet policy**
 - Use the remaining network bandwidth and have lower priority than TT or RC messages
- **TTE Standards exist or are in-work and NASA supports development of an open TTE Standard**
 - SAE AS6802 – Time-Triggered Ethernet
 - European Cooperation for Space Standardization (ECSS) ECSS-E-ST-50-16 – Time-Triggered Ethernet
 - Consultative Committee for Space Data Systems (CCSDS) Sub-Network Services WG



Sikorsky S-97 RAIDER

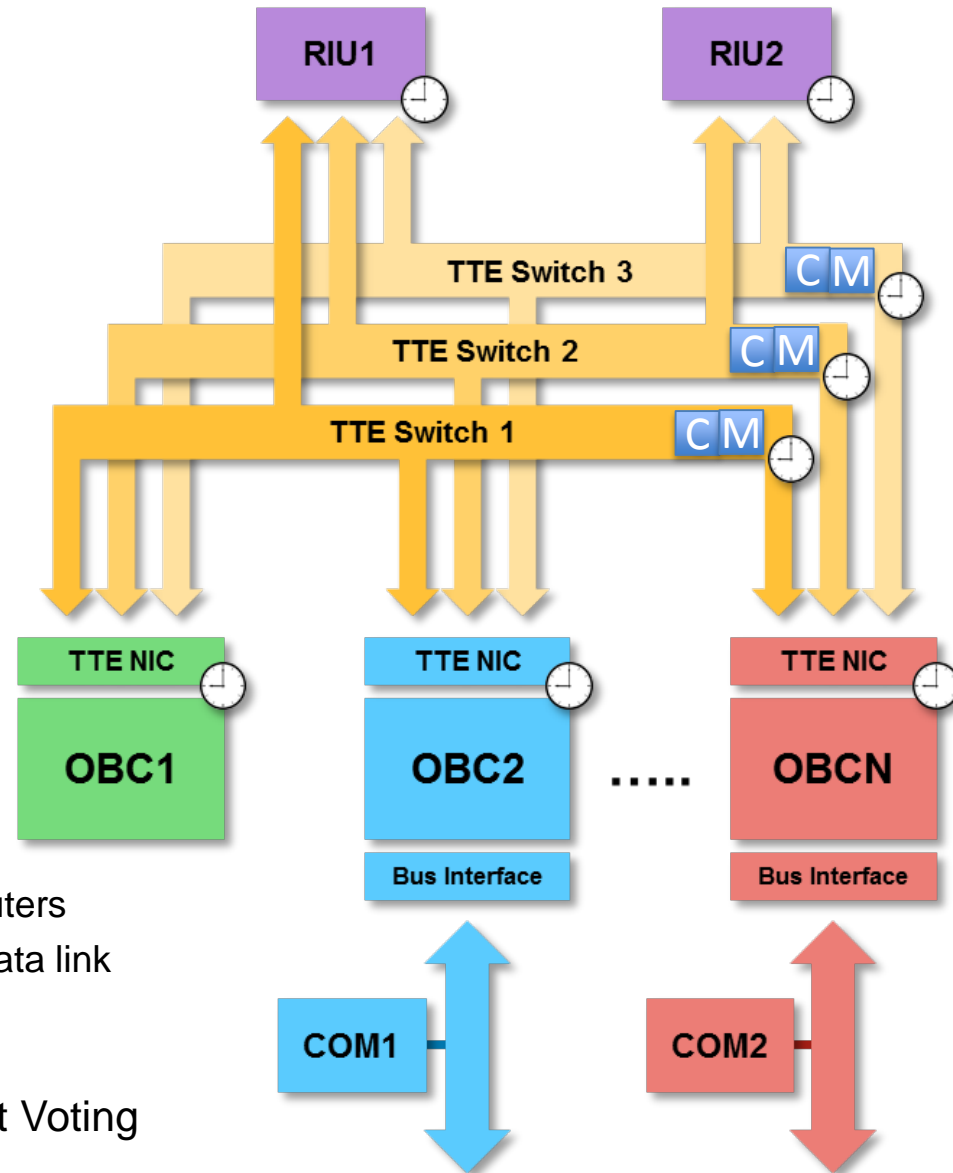


NASA's Orion Spacecraft

Reliability and Robustness: Triplex Voting Architecture



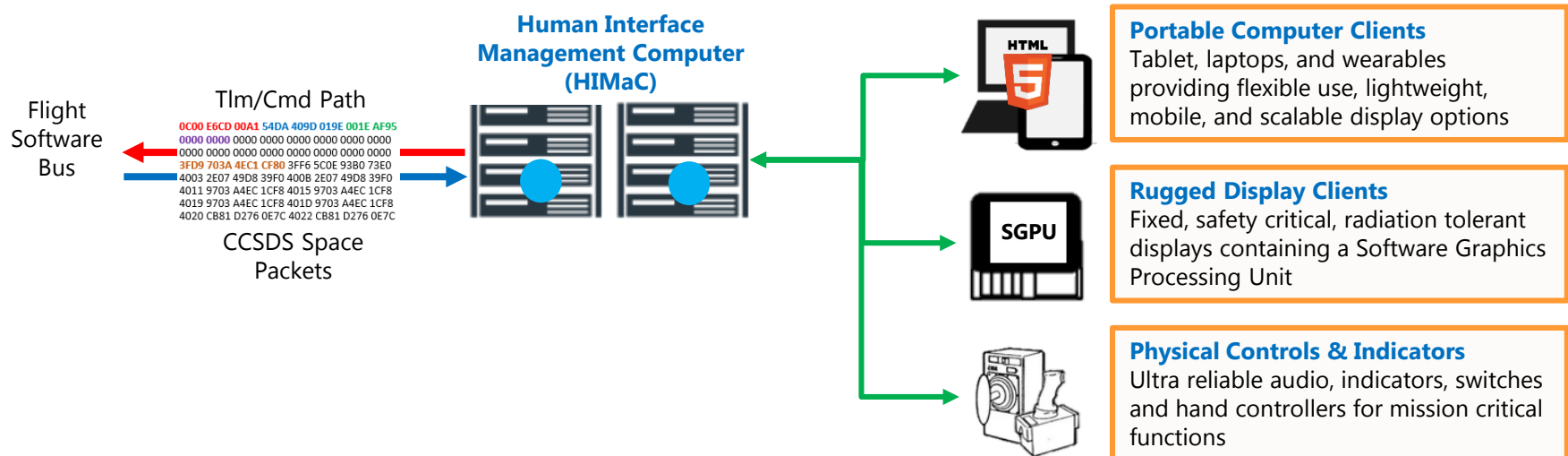
- **Developed a 1-Byzantine Fault tolerant voting architecture using TTE and CFS using current COTS technologies**
 - Three Onboard Computers (OBC)
 - Three High-Integrity (command/monitor) TTE Switches
 - Remote Interface Units (RIU)
- **Architecture is scalable through additional network planes, high-integrity devices, etc.**
- **Approach uses TTE for data distribution and sync and built CFS apps to do so**
- **Benefits of the voting architecture:**
 - Enables the use of COTS single board computers
 - Eliminates need for separate cross-channel data link
 - Eliminates need for separate timing hardware
- **Paper:** “A Proposed Byzantine Fault-Tolerant Voting Architecture using Time-Triggered Ethernet”



The Crew Element: Human Interface Architecture



- Developed a human interface architecture to reduce barriers between the crew and the vehicle
- Core component is the Human Interface Management Computer (HIMaC) that acts as a Display Server, Telemetry/Command Exchange Server and Audio/Video Server
 - HIMaC is tied to flight software bus network and supports different traffic classes
- Architecture consists of open interface standards to provide flexible and reconfigurable peripherals
 - Displays, Controls, Wearables, Audio, Video, Virtual/Augmented Reality
- Provides a robust approach to data security
- Designed to be scalable, sustainable, and evolvable enabling support for system build up, upgrades and extensions



Can You Hear Me Now?: Wireless and Communication Architecture



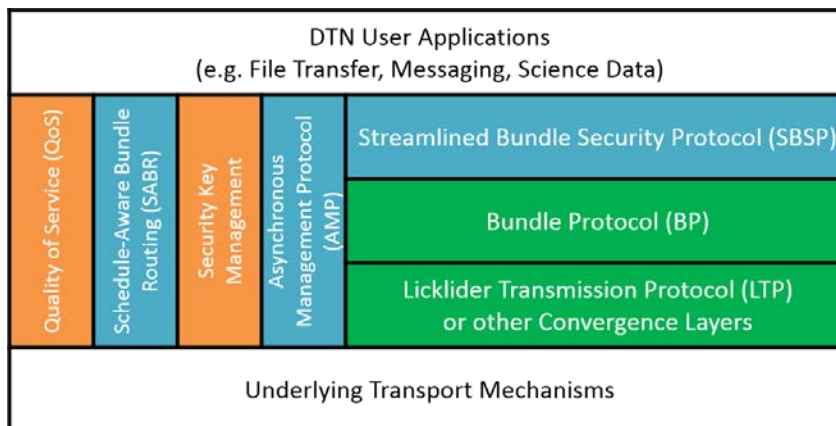
- The communication links that the architecture is designed to support include:
 - LOP-G ↔ Earth
 - LOP-G ↔ Lunar Surface
 - LOP-G ↔ Visiting Vehicle
 - LOP-G ↔ Proximity/Wireless Communications (i.e. Extra-Vehicular Activity (EVA))
- Architecture supports several standard wireless standards and technologies for internal spacecraft and proximity communications
 - IEEE 802.11 Family
 - 5G Technology (LTE)
 - Wireless Sensor Networks
 - Radio Frequency Identification (RFID) for both logistics and sensing
- Candidate standards and technologies have been identified, are still being evaluated, and have not been finalized
 - Optical communication is also being looked at for DSG
- Will leverage the Interagency Operations Advisory Group (IOAG) Service Catalog and Consultative Committee for Space Data Systems (CCSDS) Standards
- Internetworking capabilities are a requirement, and must operate in the presence of time delays and outages
 - Delay/Disruption Tolerant Networking (DTN) is the solution



Connecting to the Solar System Internet (SSI): Delay/Disruption Tolerant Networking (DTN)



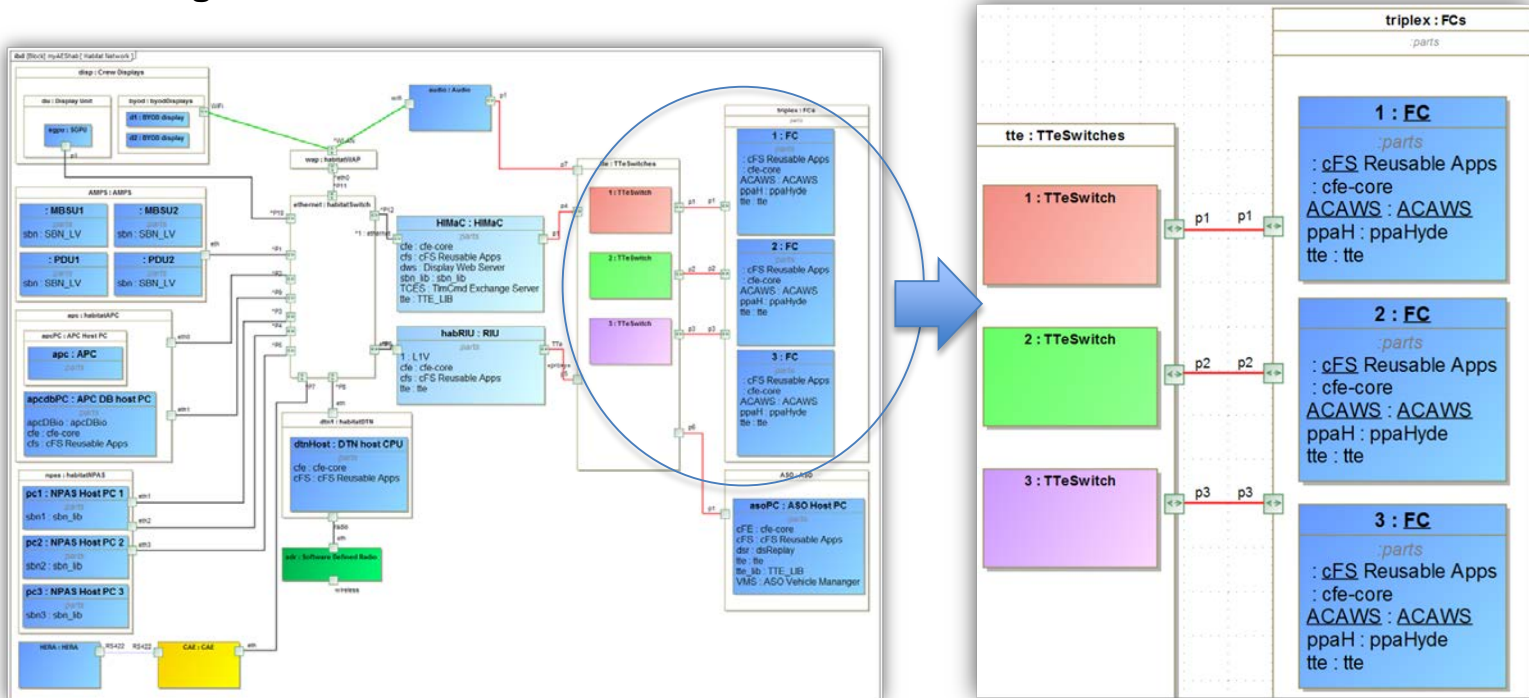
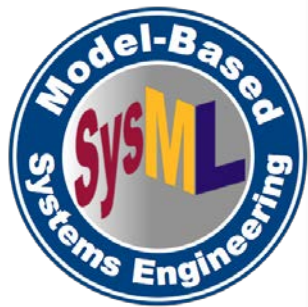
- Delay/Disruption Tolerant Networking (DTN) is an AES developed protocol suite that extends the terrestrial Internet capabilities into highly stressed data communication environments where the conventional Internet does not work
 - These environments are typically subject to frequent disruptions, unidirectional/asymmetric links, long delays and high error rates
- DTN is being standardized by the Consultative Committee for Space Data Systems (CCSDS) and the Internet Engineering Task Force (IETF) DTN Working Groups
- NASA's Interplanetary Overlay Network (ION) DTN implementation is open-source software:
 - <https://sourceforge.net/projects/ion-dtn/>
- Some of the benefits of DTN include improved operations and situational awareness, interoperability and reuse, space link efficiency, utilization and robustness, security and quality-of-service



Putting it All Together: Systems Engineering and Integration (SE&I)



- Determined the necessary avionics functions for architecture, allocated the functions to abstract systems and implemented the systems to perform the functions
- Modeled the avionics and software architecture using Model-Based Systems Engineering (MBSE) tools using the Systems Modelling Language (SysML) throughout life-cycle
- Led the migration of other spacecraft subsystems to run CFS applications on path-to-flight processors and connect to the architecture
 - Power, Environmental Control and Life Support System (ECLSS), Vehicle Autonomy applications, etc.
- Conceptualized mission scenarios to exercise/stress the architecture through both simulation and testing



An Enabling Architecture: Supporting Future Autonomous Systems



- **As human exploration moves farther out into space, the need for autonomous systems significantly increases**
 - Many functions of the current Mission Control Center (MCC) will need to move onto the spacecraft
- **AES, STMD and others within NASA are researching various autonomy applications that could be used as part of the Lunar Orbital Platform – Gateway efforts**
- **NASA is also closely tracking commercial developments that could support autonomous systems**
 - AI and Cognitive Computing, Deep-Learning Algorithms, Model-based Condition Monitoring, Industrial and Home Automation, IoT, etc.
- **The developed avionics and software architecture will serve as a platform to exercise autonomy applications and concepts**
 - Exercise onboard autonomous Integrated Vehicle Health Management (IVHM) applications
 - Explore distributed and centralized autonomy concepts
 - Build crew and ground operator familiarity and comfort with autonomy applications
 - Provide reliable command/control capabilities for spacecraft subsystems
 - Provide additional processing/storage for less-capable systems
 - Monitor subsystems and serve as an operations advisor
- **Open architecture will also serve as a technology development platform to help establish partnerships and collaborations to further enhance architecture**
 - Support Academia, International Partner or commercial technologies

What's Next?: Future Architecture Needs/Gaps



- **Availability of high-performance, radiation-tolerant processors, memory and display hardware**
- **Ability to use Commercial-Off-The-Shelf (COTS) hardware/software from multiple vendors**
- **Consensus on common interconnects for space applications**
- **Support for high-bandwidth applications, including audio, video, autonomy, high-resolution imagery, etc.**
- **Maturation and publication of relevant standards**
- **Increased cybersecurity awareness and protection**

Key Takeaways



- **The AES A&S project has developed and implemented an Avionics & Software architecture that is:**
 - Open-source, with standard capabilities and interfaces
 - Highly reliable with 1-Byzantine fault tolerance
 - Scalable and customizable to support future exploration missions such as the LOP-G
 - Built on a foundation of NASA's Core Flight Software (CFS) and Time-Triggered Ethernet (TTE)
 - Realizable with currently available COTS technology and supports multi-vendor hardware
 - Fully modeled in SysML, implemented and tested in relevant environments
 - Designed to support various autonomy technologies that will be needed for deep-space human exploration
- **The Lunar Orbital Platform – Gateway will be built up in phases, with individual elements likely being provided by both International and commercial partners, making interoperability more of a challenge for the overall avionics architecture of the integrated spacecraft**
- **Open standards and abstraction layers allow for late selection of hardware platforms and in-flight upgrades as advances occur in the areas of high-performance, radiation-tolerant processors, memory and display hardware**

