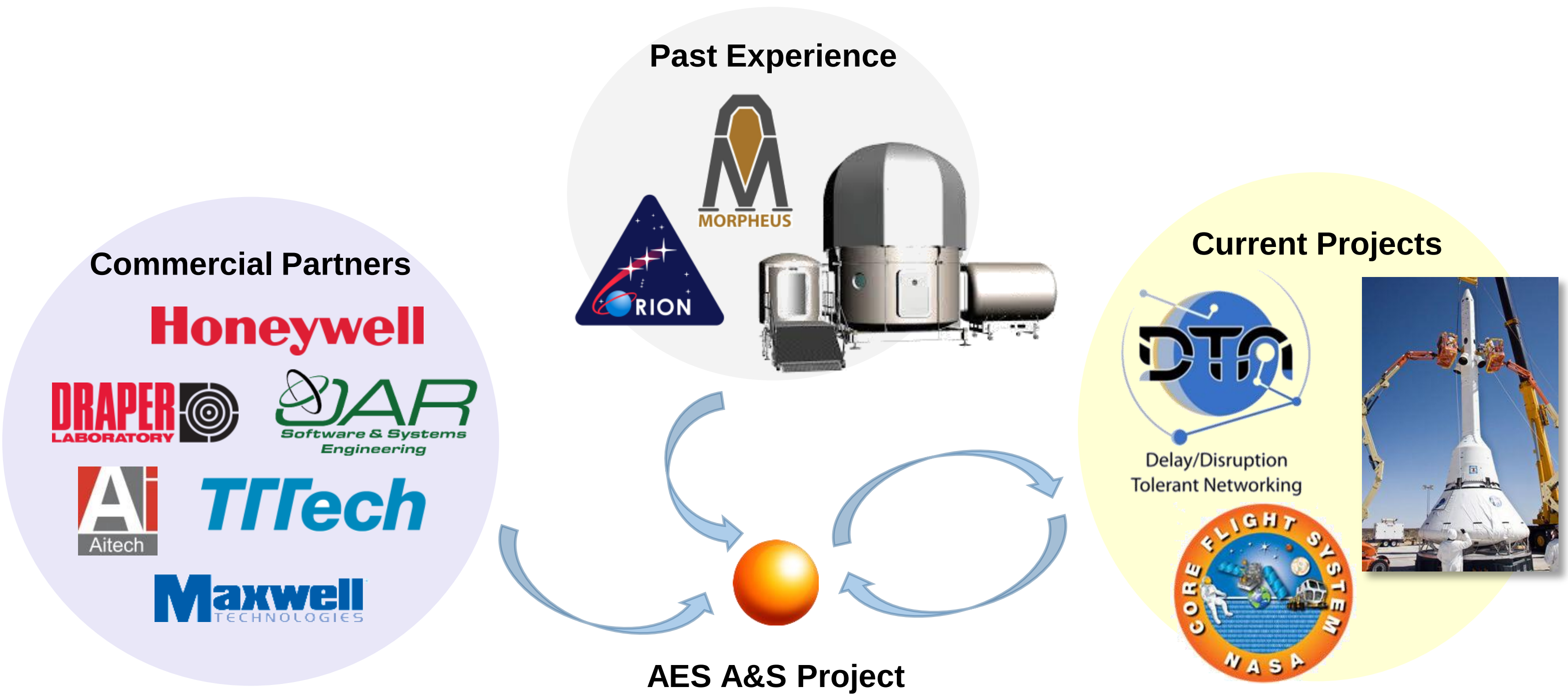
# Overview of TTE Applications and Development at NASA/JSC

CCSDS SOIS SUBNET WG Meeting
ASI, Rome, Italy
17 – 21 October 2016

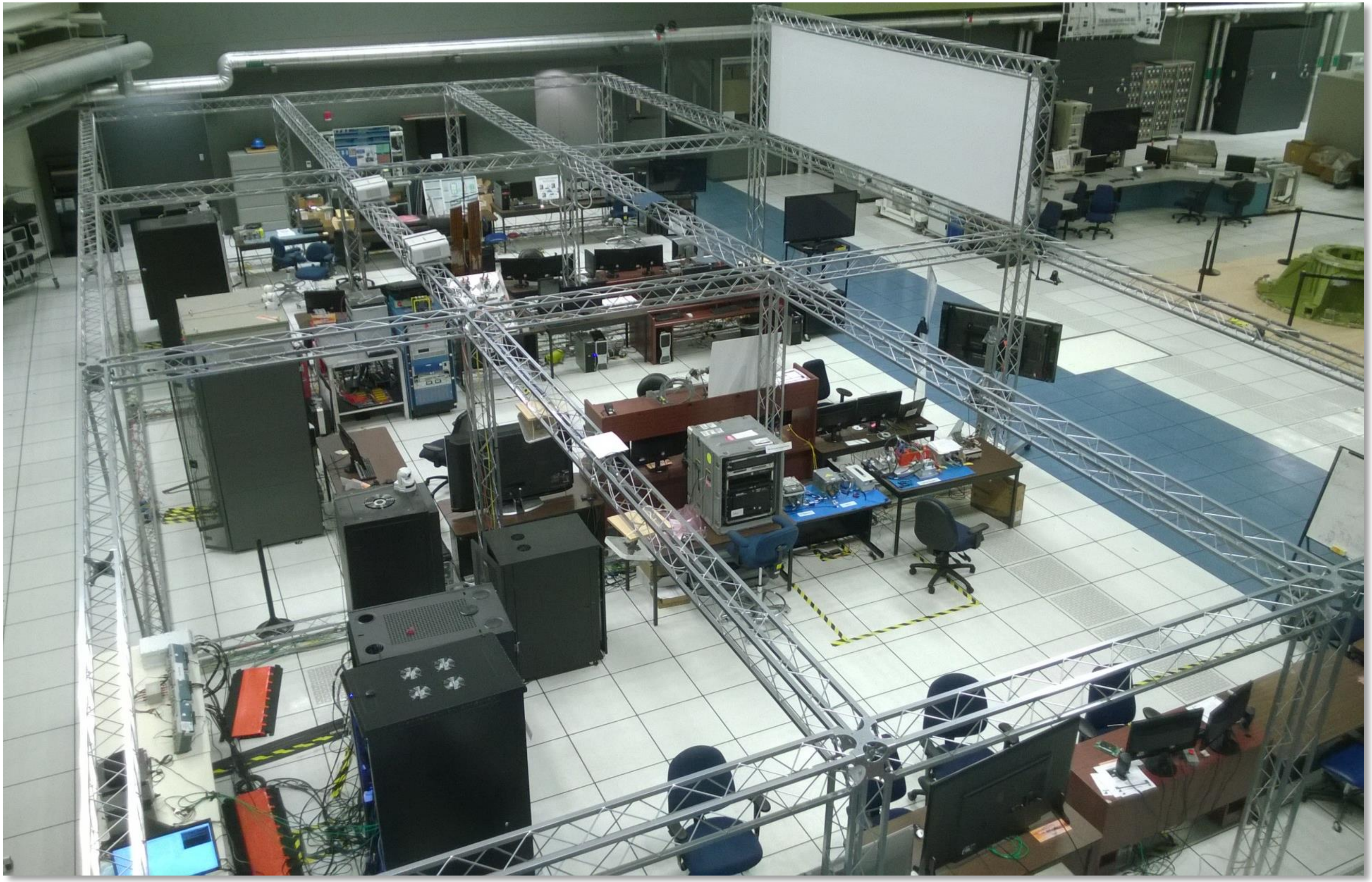
Andrew Loveless (NASA JSC)
andrew.loveless@nasa.gov

# Background

- The **Avionics and Software (A&S) Project** is developing a mission-agnostic architecture applicable to spacecraft or habitats.
  - Chartered by NASA's Advanced Exploration Systems (AES) Program.
  - Includes participation by most NASA centers and several commercial partners.
  - Mature promising architectures for use in other NASA projects.
  - **Approach:** Minimize development time/cost by utilizing COTS technologies.



# IPAS: Integrated Testbed

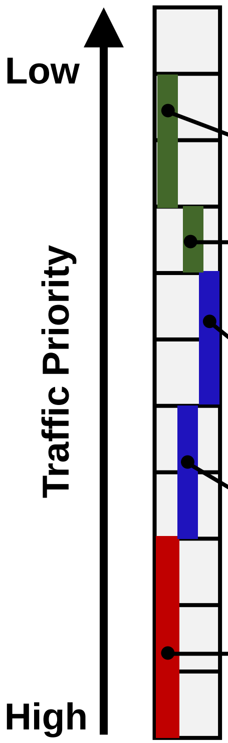


**IPAS testbed located at NASA/JSC in B29**

# TTEthernet Traffic Classes

Time-Triggered Ethernet can help overcome difficulties in realizing an IMA architecture by providing multiple traffic classes for different criticality levels.

## Bandwidth Utilization



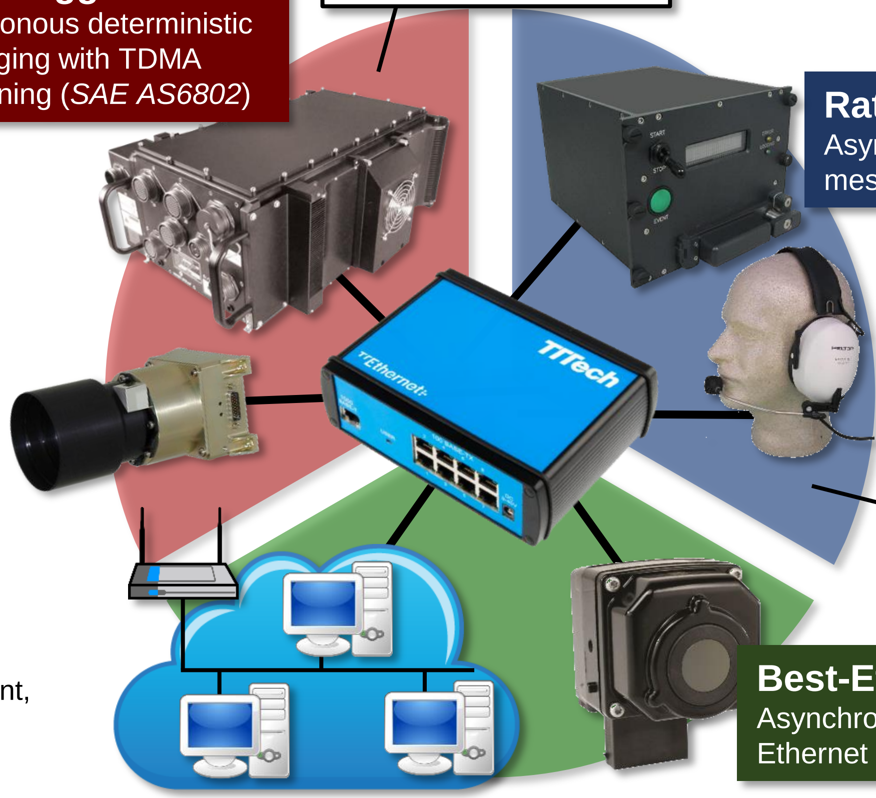

9% diagnostics/config and experiments **(IEEE 802.3)**

18% high-definition video and displays **(IEEE 802.3)**

9% real-time telemetry processing and data recorder **(ARINC 664)**

18% real-time audio/ video streaming **(ARINC 664)**

27% hard real-time vehicle control (vehicle management, IMU, star tracker, power controller) **(SAE AS6802)**




**Time-Triggered**
Synchronous deterministic messaging with TDMA partitioning (*SAE AS6802*)

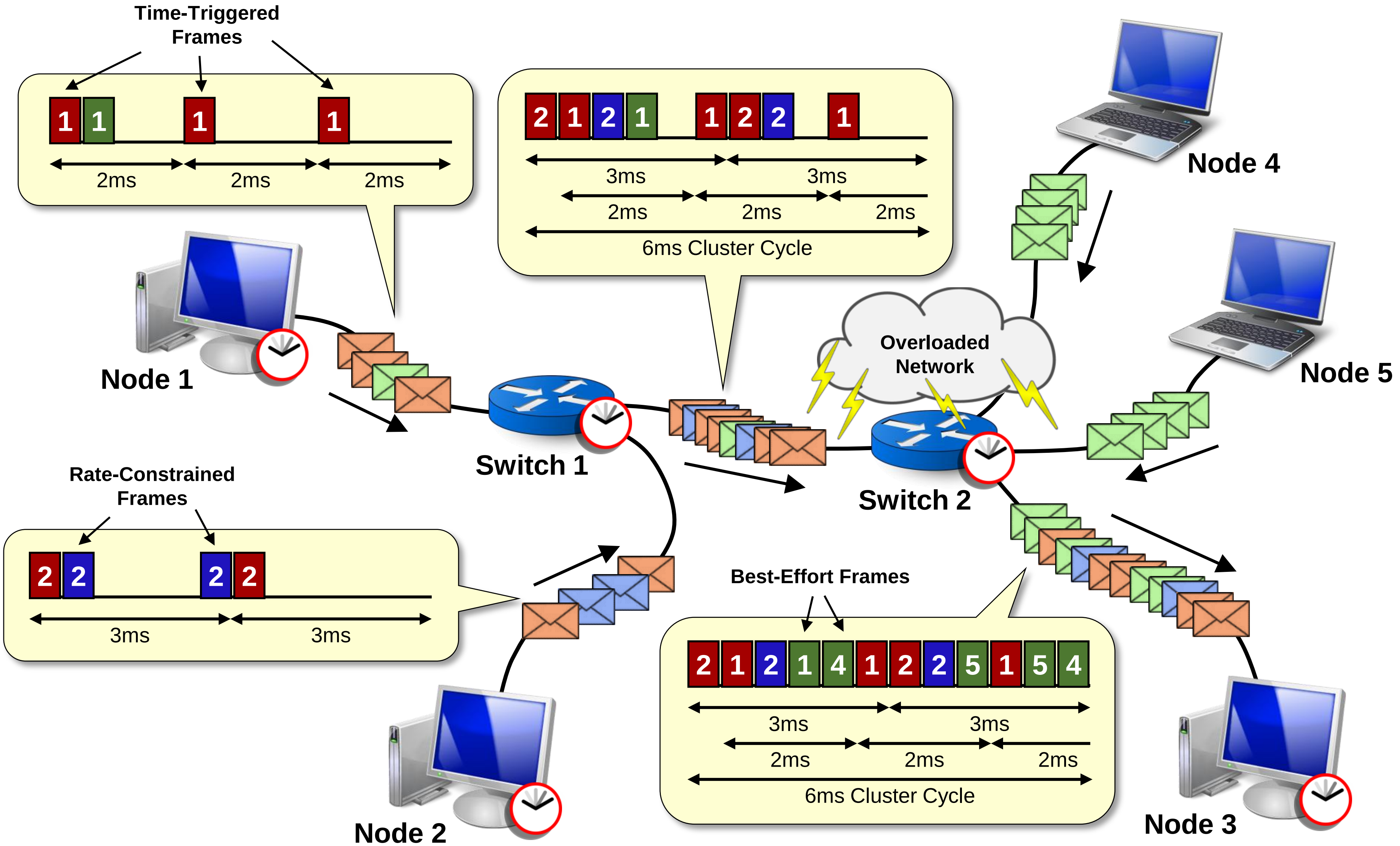
Exact definition of TDMA slots and time base

**Rate-Constrained**
Asynchronous deterministic messaging (*ARINC 664-p7*)

Traffic shaping and policing prevents loss of streaming data.

**Best-Effort (Classical)**
Asynchronous standard Ethernet LAN (IEEE 802.3)

# TTEthernet Traffic Integration



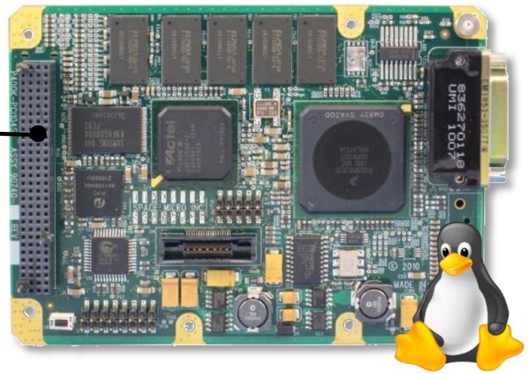
# Past Work: Technology and Tools

## NASA-JSC has a long history of using Time-Triggered Ethernet.

- Collaborated with Honeywell on application of TTGbE for the Orion MPCV (2007).
- Have worked with every major iteration of TTTech's TTEthernet (2008 – Present).

## Example Projects:

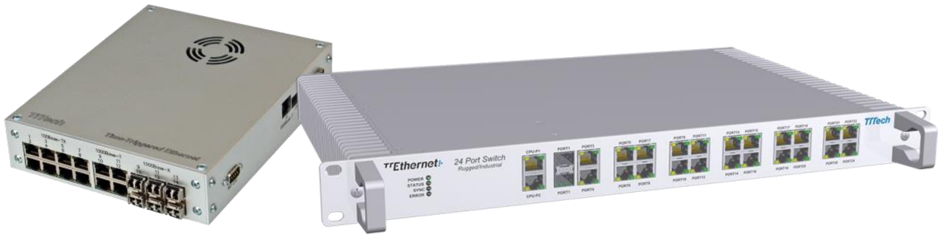
- Driver development to support TTEthernet on a wide range of different platforms and OSs.
  - **Chip-IP Versions:** Phoenix (Gen 2), Pegasus (Gen 3)
  - **Platforms:** Space Micro Proton-400K, Aitech SP0-100
  - **Operating Systems:** RT-Linux and VxWorks RTOS
- Developed scripts to automate scheduling and deployment.
- Built tools for network loading, visualization, and analysis.
- Built libraries for Core Flight Software (CFS) supporting network-based FSW scheduler, synchronization, and voting.
- Wrote extensions to stock TTE implementations, including:
  - Network stack for Phoenix Chip-IP - including UDP and IP layers.
  - Wrapper APIs with abstraction over DMA/PIO transfer mechanisms.
  - Abstraction layer for Pegasus Chip-IP on VxWorks.
- Developed tools for report generation and metric collection.



# Past Work: Technology and Tools

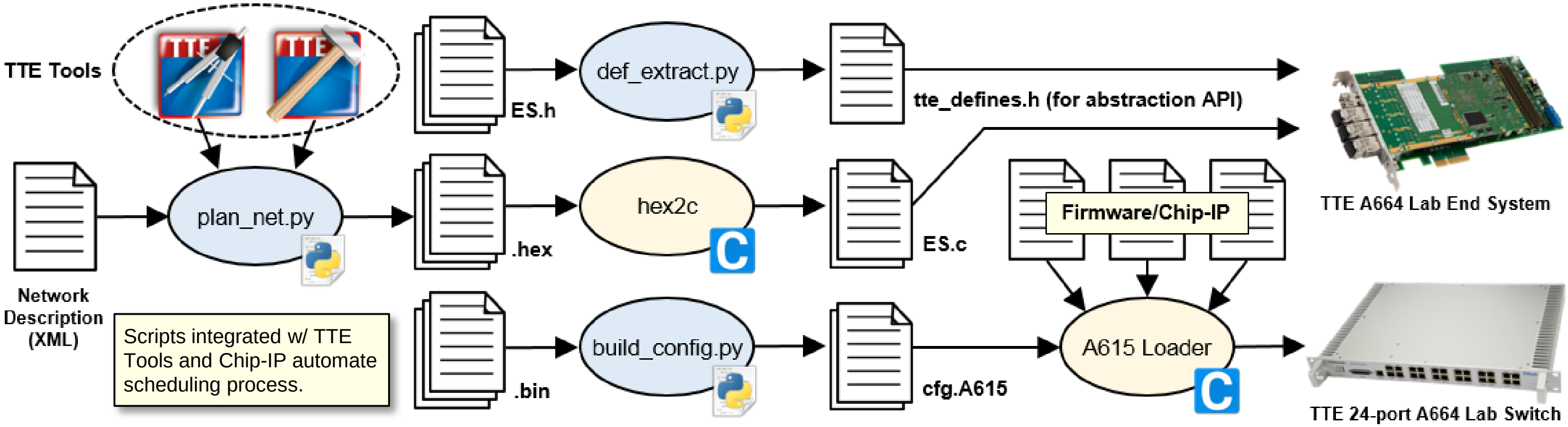
## NASA-JSC has a long history of using Time-Triggered Ethernet.

- Collaborated with Honeywell on application of TTGbE for the Orion MPCV (2007).
- Have worked with every major iteration of TTTech's TTEthernet (2008 – Present).



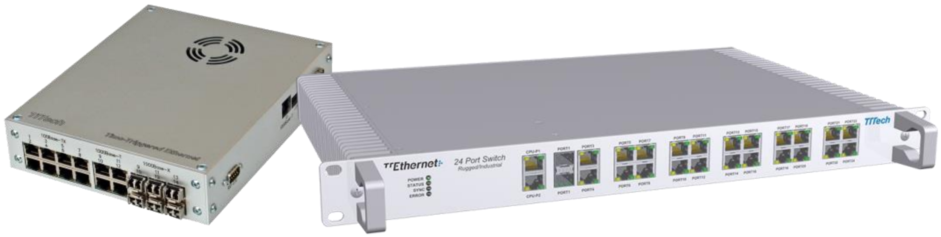
## Example Projects:

- Driver development to support TTEthernet on a wide range of different platforms and OSs.
  - **Chip-IP Versions:** Phoenix (Gen 2), Pegasus (Gen 3)
  - **Platforms:** Space Micro Proton-400K, Aitech SP0-100
  - **Operating Systems:** RT-Linux and VxWorks RTOS
- Developed scripts to automate scheduling and deployment.

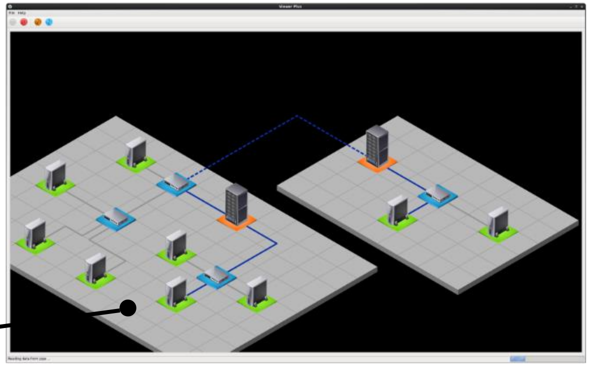


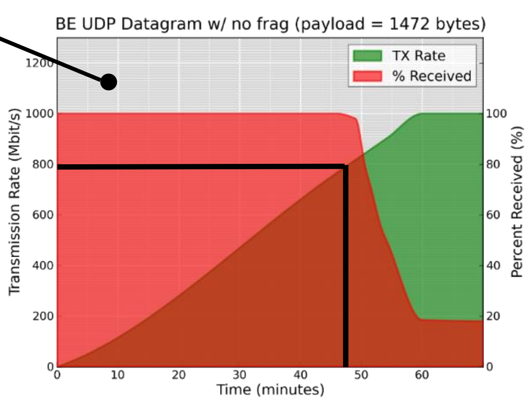
# Past Work: Technology and Tools

- **NASA-JSC has a long history of using Time-Triggered Ethernet.**
  - Collaborated with Honeywell on application of TTGbE for the Orion MPCV (2007).
  - Have worked with every major iteration of TTTech's TTEthernet (2008 – Present).



- **Example Projects:**
  - Driver development to support TTEthernet on a wide range of different platforms and OSs.
    - **Chip-IP Versions:** Phoenix (Gen 2), Pegasus (Gen 3)
    - **Platforms:** Space Micro Proton-400K, Aitech SP0-100
    - **Operating Systems:** RT-Linux and VxWorks RTOS
  - Developed scripts to automate scheduling and deployment.
  - Built tools for network loading, visualization, and analysis.
  - Built libraries for Core Flight Software (CFS) supporting network-based FSW scheduler, synchronization, and voting.
  - Wrote extensions to stock TTE implementations, including:
    - Network stack for Phoenix Chip-IP - including UDP and IP layers.
    - Wrapper APIs with abstraction over DMA/PIO transfer mechanisms.
    - Abstraction layer for Pegasus Chip-IP on VxWorks.
  - Developed tools for report generation and metric collection.

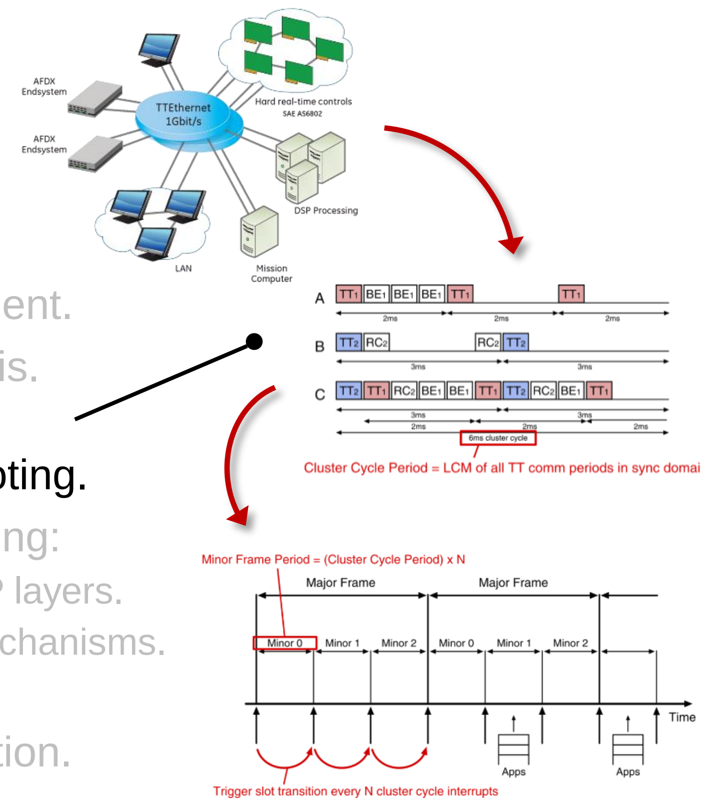




# Past Work: Technology and Tools

- **NASA-JSC has a long history of using Time-Triggered Ethernet.**
  - Collaborated with Honeywell on application of TTGbE for the Orion MPCV (2007).
  - Have worked with every major iteration of TTTech's TTEthernet (2008 – Present).

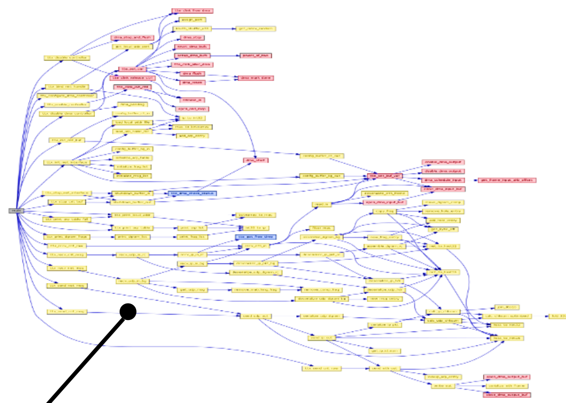
- **Example Projects:**
  - Driver development to support TTEthernet on a wide range of different platforms and OSs.
    - **Chip-IP Versions:** Phoenix (Gen 2), Pegasus (Gen 3)
    - **Platforms:** Space Micro Proton-400K, Aitech SP0-100
    - **Operating Systems:** RT-Linux and VxWorks RTOS
  - Developed scripts to automate scheduling and deployment.
  - Built tools for network loading, visualization, and analysis.
  - Built libraries for Core Flight Software (CFS) supporting network-based FSW scheduler, synchronization, and voting.
  - Wrote extensions to stock TTE implementations, including:
    - Network stack for Phoenix Chip-IP - including UDP and IP layers.
    - Wrapper APIs with abstraction over DMA/PIO transfer mechanisms.
    - Abstraction layer for Pegasus Chip-IP on VxWorks.
  - Developed tools for report generation and metric collection.

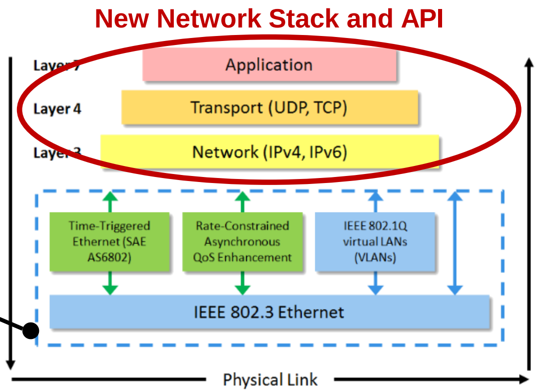


# Past Work: Technology and Tools

- **NASA-JSC has a long history of using Time-Triggered Ethernet.**
  - Collaborated with Honeywell on application of TTGbE for the Orion MPCV (2007).
  - Have worked with every major iteration of TTTech's TTEthernet (2008 – Present).


- **Example Projects:**
  - Driver development to support TTEthernet on a wide range of different platforms and OSs.
    - **Chip-IP Versions:** Phoenix (Gen 2), Pegasus (Gen 3)
    - **Platforms:** Space Micro Proton-400K, Aitech SP0-100
    - **Operating Systems:** RT-Linux and VxWorks RTOS
  - Developed scripts to automate scheduling and deployment.
  - Built tools for network loading, visualization, and analysis.
  - Built libraries for Core Flight Software (CFS) supporting network-based FSW scheduler, synchronization, and voting.
  - Wrote extensions to stock TTE implementations, including:
    - Network stack for Phoenix Chip-IP - including UDP and IP layers.
    - Wrapper APIs with abstraction over DMA/PIO transfer mechanisms.
    - Abstraction layer for Pegasus Chip-IP on VxWorks.
  - Developed tools for report generation and metric collection.

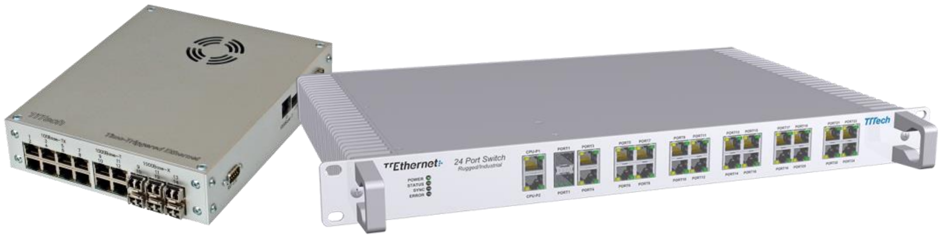




# Past Work: Technology and Tools

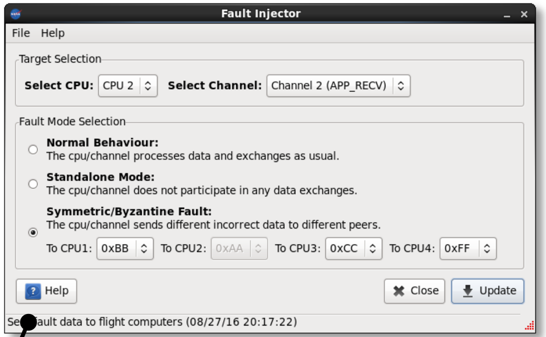
## NASA-JSC has a long history of using Time-Triggered Ethernet.

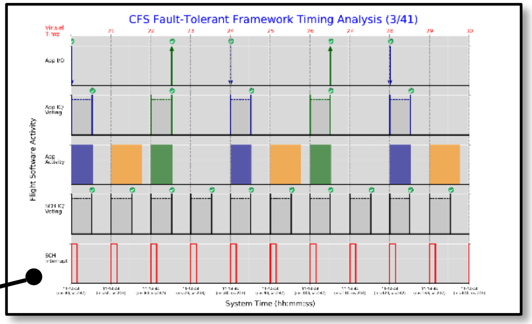
- Collaborated with Honeywell on application of TTGbE for the Orion MPCV (2007).
- Have worked with every major iteration of TTTech's TTEthernet (2008 – Present).




## Example Projects:

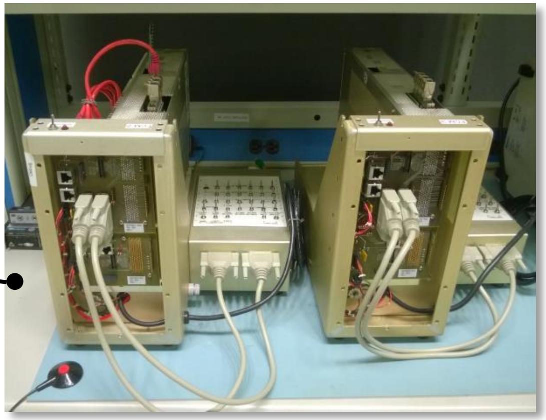
- Driver development to support TTEthernet on a wide range of different platforms and OSs.
  - **Chip-IP Versions:** Phoenix (Gen 2), Pegasus (Gen 3)
  - **Platforms:** Space Micro Proton-400K, Aitech SP0-100
  - **Operating Systems:** RT-Linux and VxWorks RTOS
- Developed scripts to automate scheduling and deployment.
- Built tools for network loading, visualization, and analysis.
- Built libraries for Core Flight Software (CFS) supporting network-based FSW scheduler, synchronization, and voting.
- Wrote extensions to stock TTE implementations, including:
  - Network stack for Phoenix Chip-IP - including UDP and IP layers.
  - Wrapper APIs with abstraction over DMA/PIO transfer mechanisms.
  - Abstraction layer for Pegasus Chip-IP on VxWorks.
- Developed tools for report generation and metric collection.



# Past Work: Fault-Tolerance

- **NASA-JSC has a focus fault-tolerance for human-rated vehicles.**
  - Experience from the Space Shuttle, ISS, and X-38 CRV has influenced the design of several fault tolerance approaches.
  - We have used Time-Triggered Ethernet to realize multiple architectures accommodating different fault classifications.
- **Different Approaches:**
  - Boeing 787 **Self-Checking Pair (SCP)** with lockstep IBM 750FX processors and TTGbE interface.
    - Comparable to Orion Vehicle Management Computer (VMC).
  - **Warm-Backup** redundant computers (shadowing).
    - Comparable to ISS Command and Control MDMs.
  - **Triplex Voting** with Master/Slave synchronization.
    - Demonstrated running Ascent Abort 2 (AA2) mission scenario with Orion GN&C flight software.
  - **Quad-Voting** with message-based synchronization.
    - Realized on 4x Aitech SP0-100 SBCs running VxWorks.
  - **Quad-Voting** with real-time network synchronization and 1-byzantine fault tolerance.
    - Showed ability to transparently vote all input and output data between apps in 100Hz FSW schedule table.



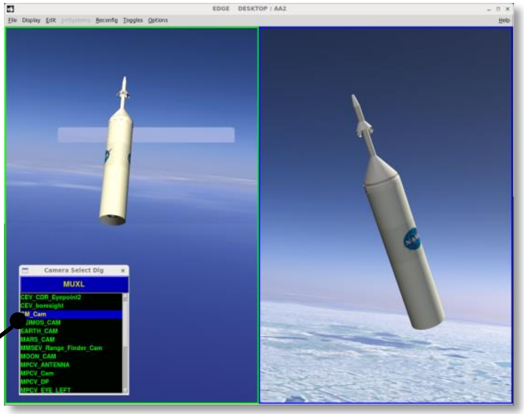
# Past Work: Fault-Tolerance

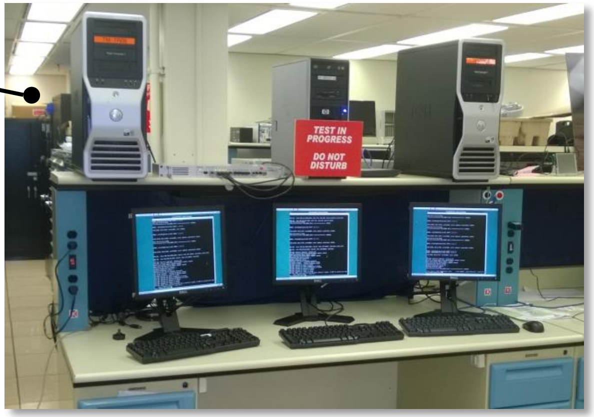
## NASA-JSC has a focus fault-tolerance for human-rated vehicles.

- Experience from the Space Shuttle, ISS, and X-38 CRV has influenced the design of several fault tolerance approaches.
- We have used Time-Triggered Ethernet to realize multiple architectures accommodating different fault classifications.

## Different Approaches:

- Boeing 787 **Self-Checking Pair (SCP)** with lockstep IBM 750FX processors and TTGbE interface.
  - Comparable to Orion Vehicle Management Computer (VMC).
- **Warm-Backup** redundant computers (shadowing).
  - Comparable to ISS Command and Control MDMs.
- **Triplex Voting** with Master/Slave synchronization.
  - Demonstrated running Ascent Abort 2 (AA2) mission scenario with Orion GN&C flight software.
- **Quad-Voting** with message-based synchronization.
  - Realized on 4x Aitech SP0-100 SBCs running VxWorks.
- **Quad-Voting** with real-time network synchronization and 1-byzantine fault tolerance.
  - Showed ability to transparently vote all input and output data between apps in 100Hz FSW schedule table.



# Past Work: Fault-Tolerance

## NASA-JSC has a focus fault-tolerance for human-rated vehicles.

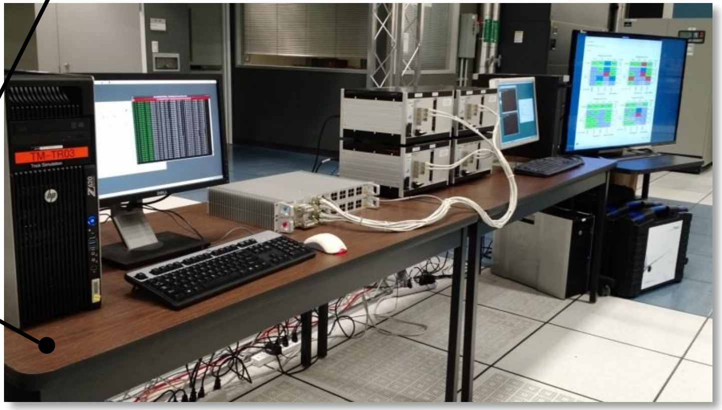
- Experience from the Space Shuttle, ISS, and X-38 CRV has influenced the design of several fault tolerance approaches.
- We have used Time-Triggered Ethernet to realize multiple architectures accommodating different fault classifications.

## Different Approaches:

- Boeing 787 **Self-Checking Pair (SCP)** with lockstep IBM 750FX processors and TTGbE interface.
  - Comparable to Orion Vehicle Management Computer (VMC).
- **Warm-Backup** redundant computers (shadowing).
  - Comparable to ISS Command and Control MDMs.
- **Triplex Voting** with Master/Slave synchronization.
  - Demonstrated running Ascent Abort 2 (AA2) mission scenario with Orion GN&C flight software.
- **Quad-Voting** with message-based synchronization.
  - Realized on 4x Aitech SP0-100 SBCs running VxWorks.
- **Quad-Voting** with real-time network synchronization and 1-byzantine fault tolerance.
  - Showed ability to transparently vote all input and output data between apps in 100Hz FSW schedule table.

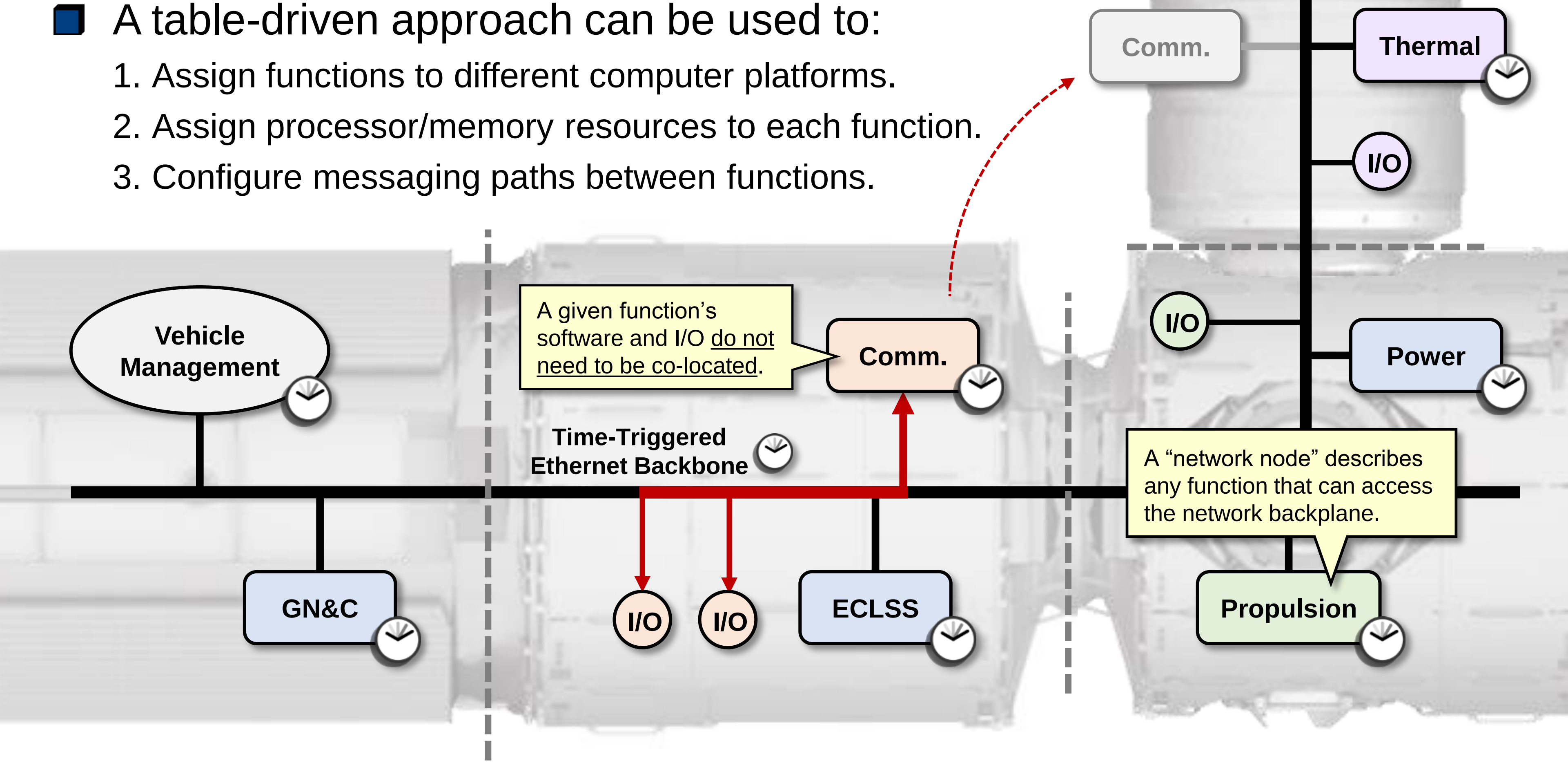

# Past Work: Fault-Tolerance

- **NASA-JSC has a focus fault-tolerance for human-rated vehicles.**
  - Experience from the Space Shuttle, ISS, and X-38 CRV has influenced the design of several fault tolerance approaches.
  - We have used Time-Triggered Ethernet to realize multiple architectures accommodating different fault classifications.
- **Different Approaches:**
  - Boeing 787 **Self-Checking Pair (SCP)** with lockstep IBM 750FX processors and TTGbE interface.
    - Comparable to Orion Vehicle Management Computer (VMC).
  - **Warm-Backup** redundant computers (shadowing).
    - Comparable to ISS Command and Control MDMs.
  - **Triplex Voting** with Master/Slave synchronization.
    - Demonstrated running Ascent Abort 2 (AA2) mission scenario with Orion GN&C flight software.
  - **Quad-Voting** with message-based synchronization.
    - Realized on 4x Aitech SP0-100 SBCs running VxWorks.
  - **Quad-Voting** with real-time network synchronization and 1-byzantine fault tolerance.
    - Showed ability to transparently vote all input and output data between apps in 100Hz FSW schedule table.

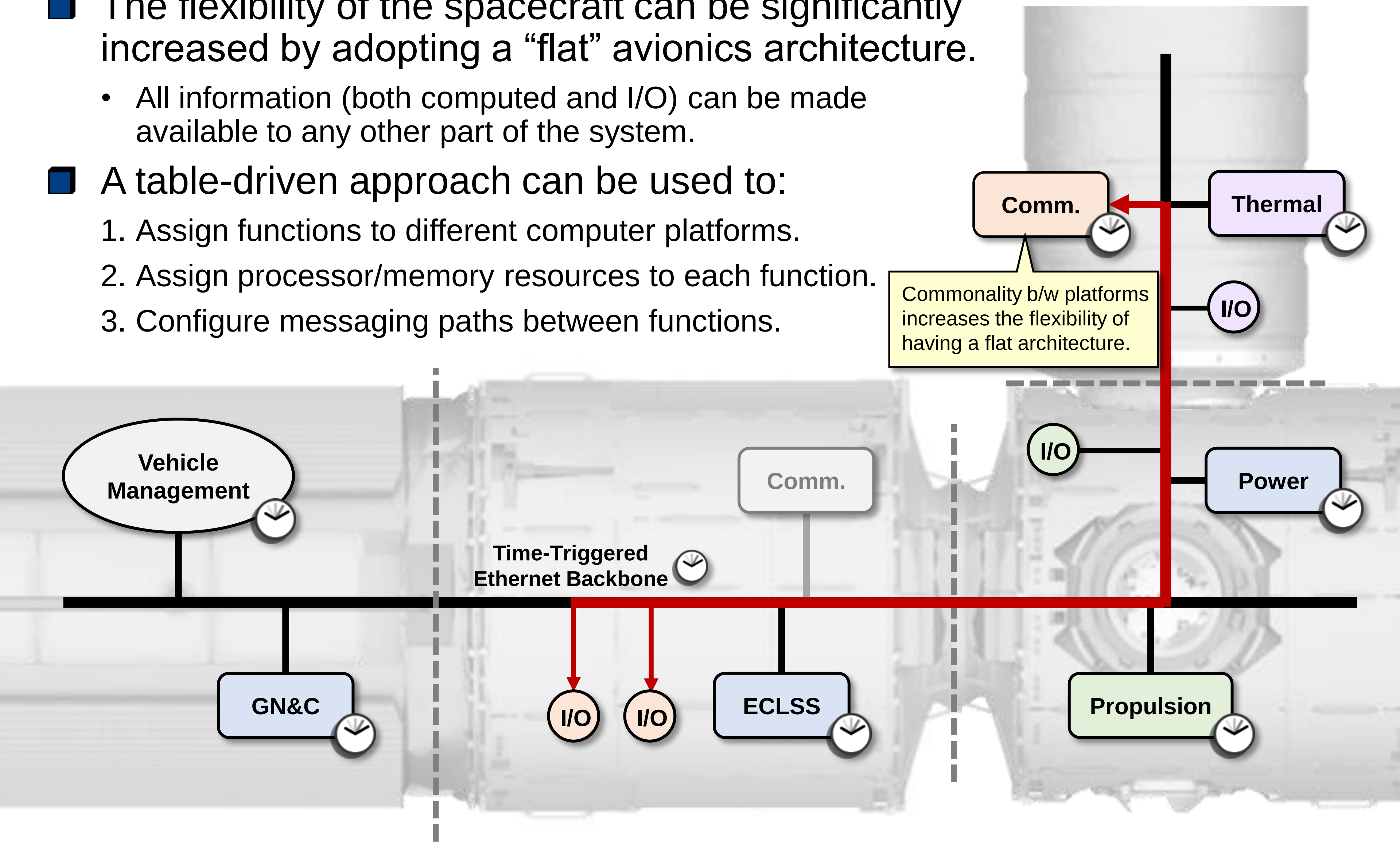
# Conceptual – Network Backplane

- The flexibility of the spacecraft can be significantly increased by adopting a “flat” avionics architecture.
  - All information (both computed and I/O) can be made available to any other part of the system.
- A table-driven approach can be used to:
  1. Assign functions to different computer platforms.
  2. Assign processor/memory resources to each function.
  3. Configure messaging paths between functions.



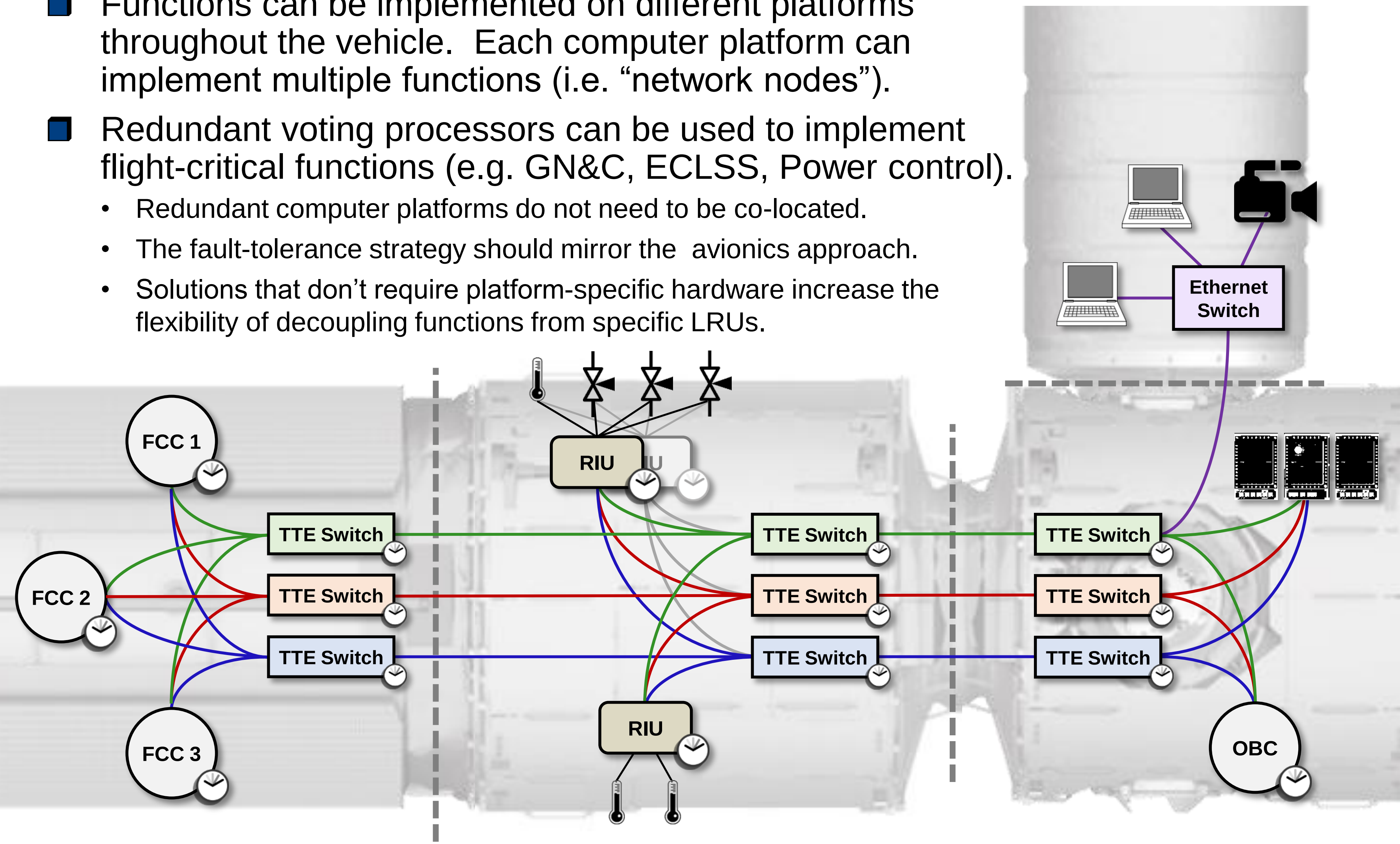
# Conceptual – Network Backplane

- The flexibility of the spacecraft can be significantly increased by adopting a “flat” avionics architecture.
  - All information (both computed and I/O) can be made available to any other part of the system.
- A table-driven approach can be used to:
  1. Assign functions to different computer platforms.
  2. Assign processor/memory resources to each function.
  3. Configure messaging paths between functions.



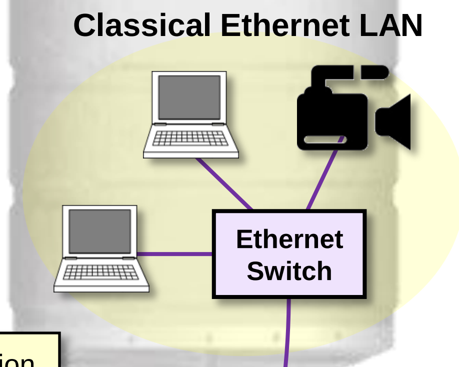
# Physical – Network Backplane

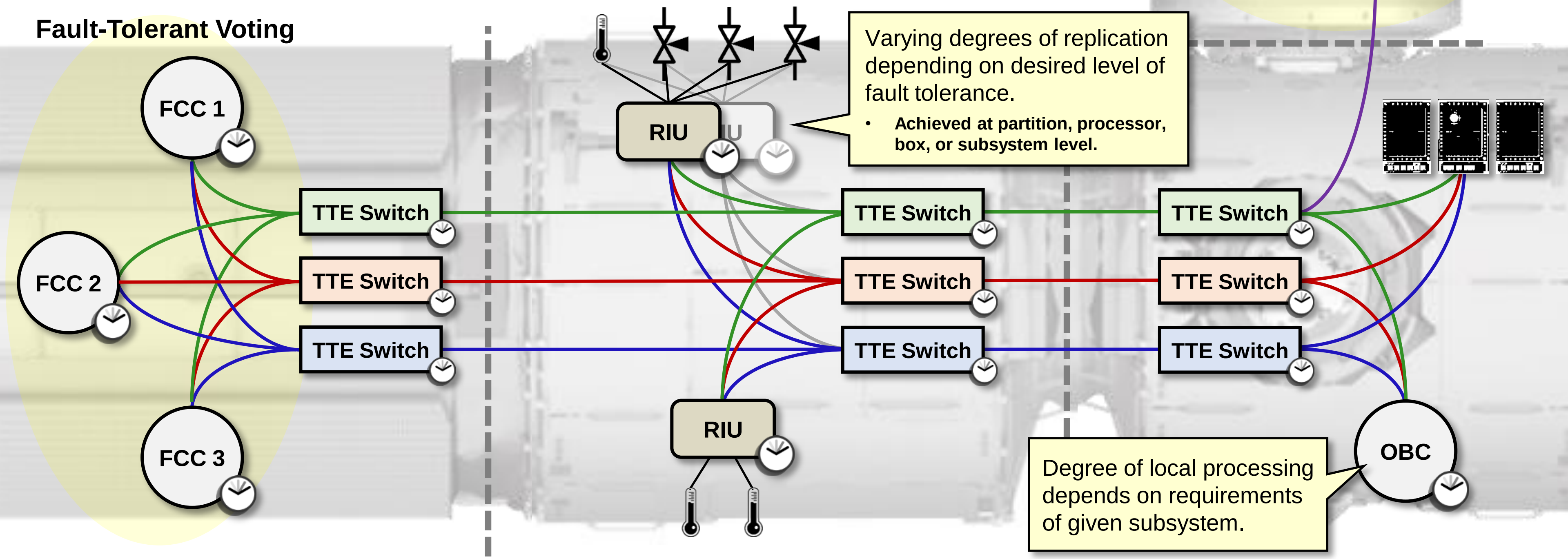
- Functions can be implemented on different platforms throughout the vehicle. Each computer platform can implement multiple functions (i.e. “network nodes”).
- Redundant voting processors can be used to implement flight-critical functions (e.g. GN&C, ECLSS, Power control).
  - Redundant computer platforms do not need to be co-located.
  - The fault-tolerance strategy should mirror the avionics approach.
  - Solutions that don’t require platform-specific hardware increase the flexibility of decoupling functions from specific LRUs.



# Conceptual – Network Backplane

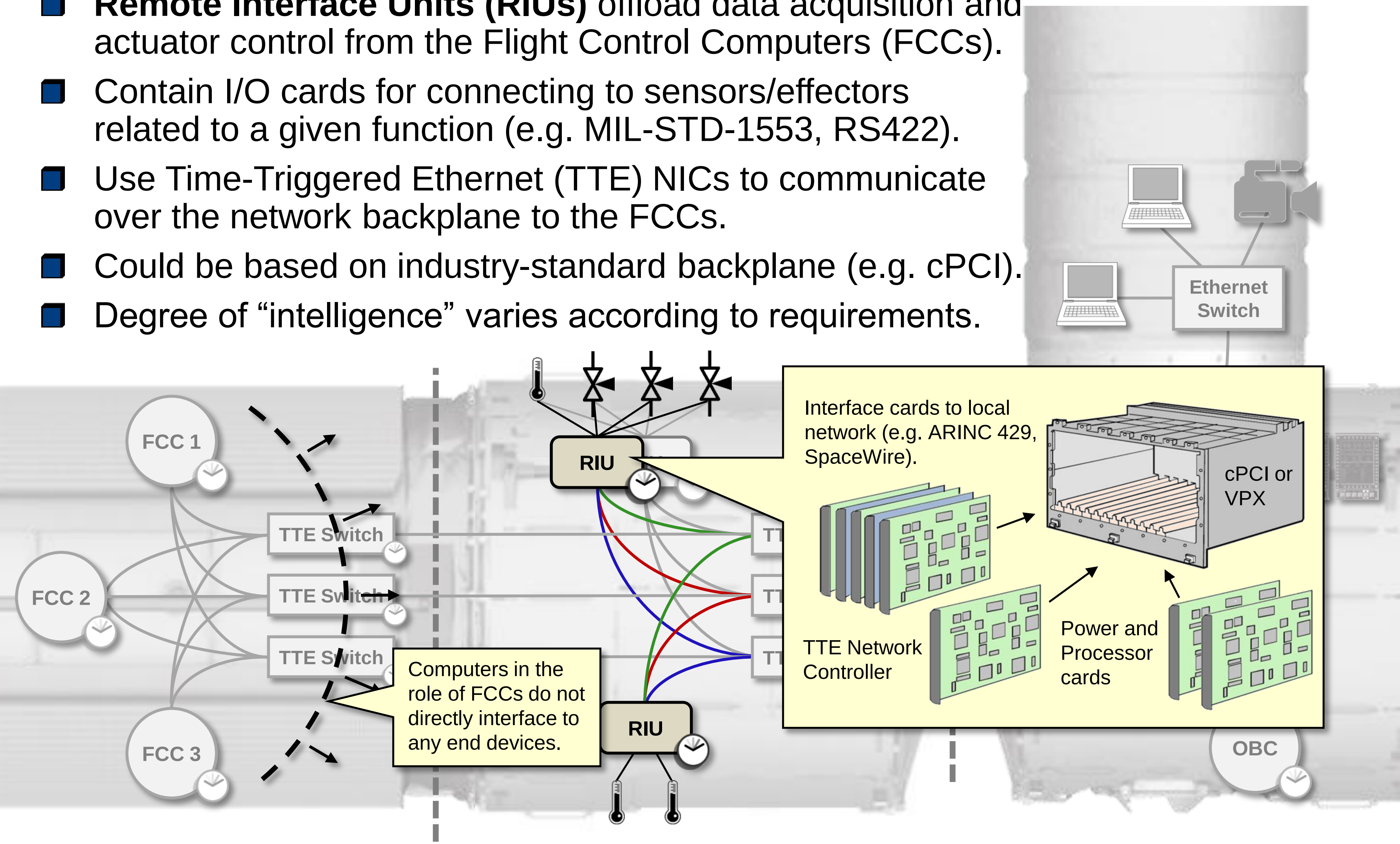
- Functions can be implemented on different platforms throughout the vehicle. Each computer platform can implement multiple functions (i.e. “network nodes”).
- Redundant voting processors can be used to implement flight-critical functions (e.g. GN&C, ECLSS, Power control).
  - Redundant computer platforms do not need to be co-located.
  - The fault-tolerance strategy should mirror the avionics approach.
  - Solutions that don’t require platform-specific hardware increase the flexibility of decoupling functions from specific LRUs.



# Remote Interface Units (RIUs)

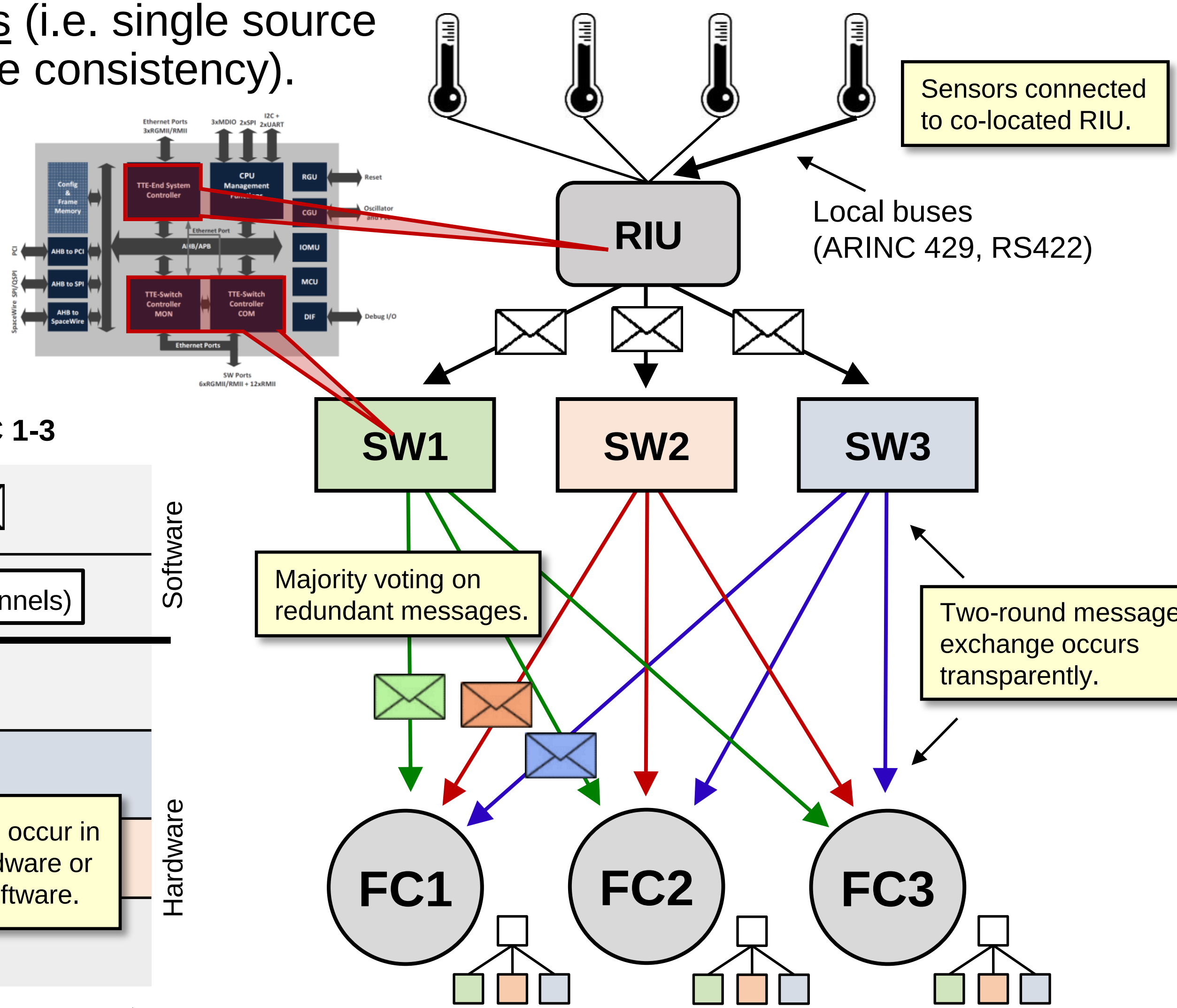
- **Remote Interface Units (RIUs)** offload data acquisition and actuator control from the Flight Control Computers (FCCs).
- Contain I/O cards for connecting to sensors/effectors related to a given function (e.g. MIL-STD-1553, RS422).
- Use Time-Triggered Ethernet (TTE) NICs to communicate over the network backplane to the FCCs.
- Could be based on industry-standard backplane (e.g. cPCI).
- Degree of "intelligence" varies according to requirements.



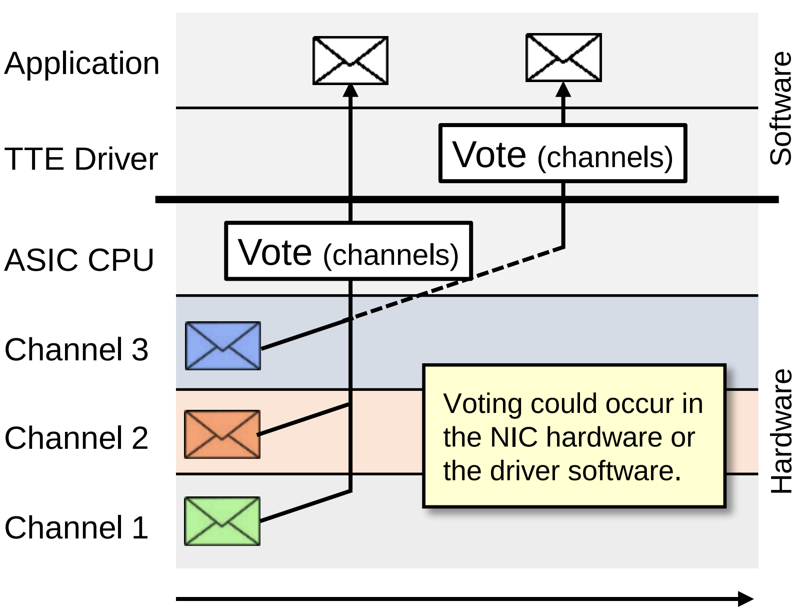
# Notional Fault Tolerance Approach

- Voting could be integrated into the process of <u>reading sensors/RIUs</u> (i.e. single source data requiring interactive consistency).

Voting over 3x redundant channels could mask asymmetric TX by the RIU and be realized with COTS SBCs and TTE space ASICs.



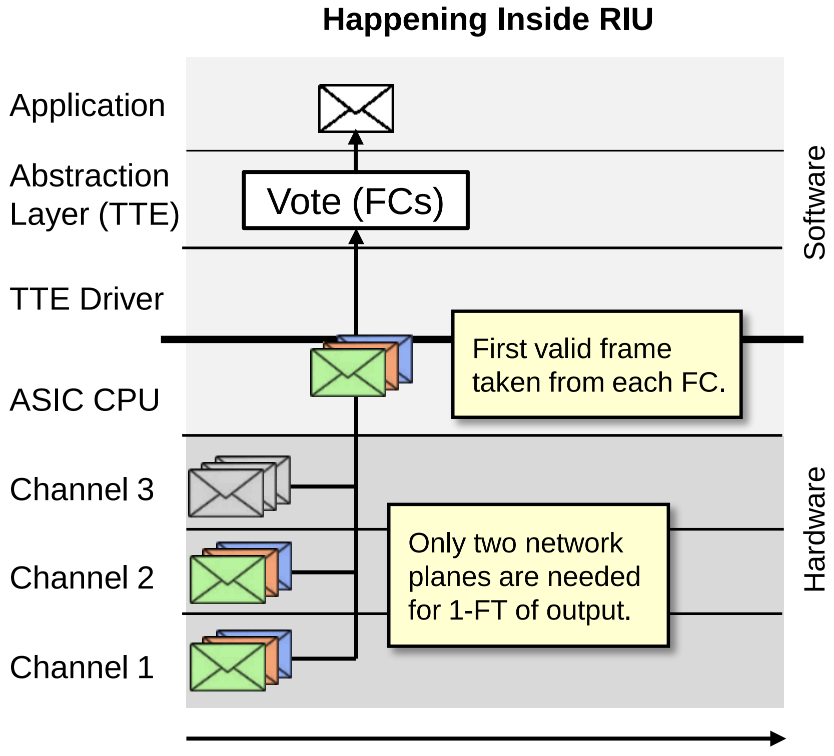

**Happening Inside FC 1-3**

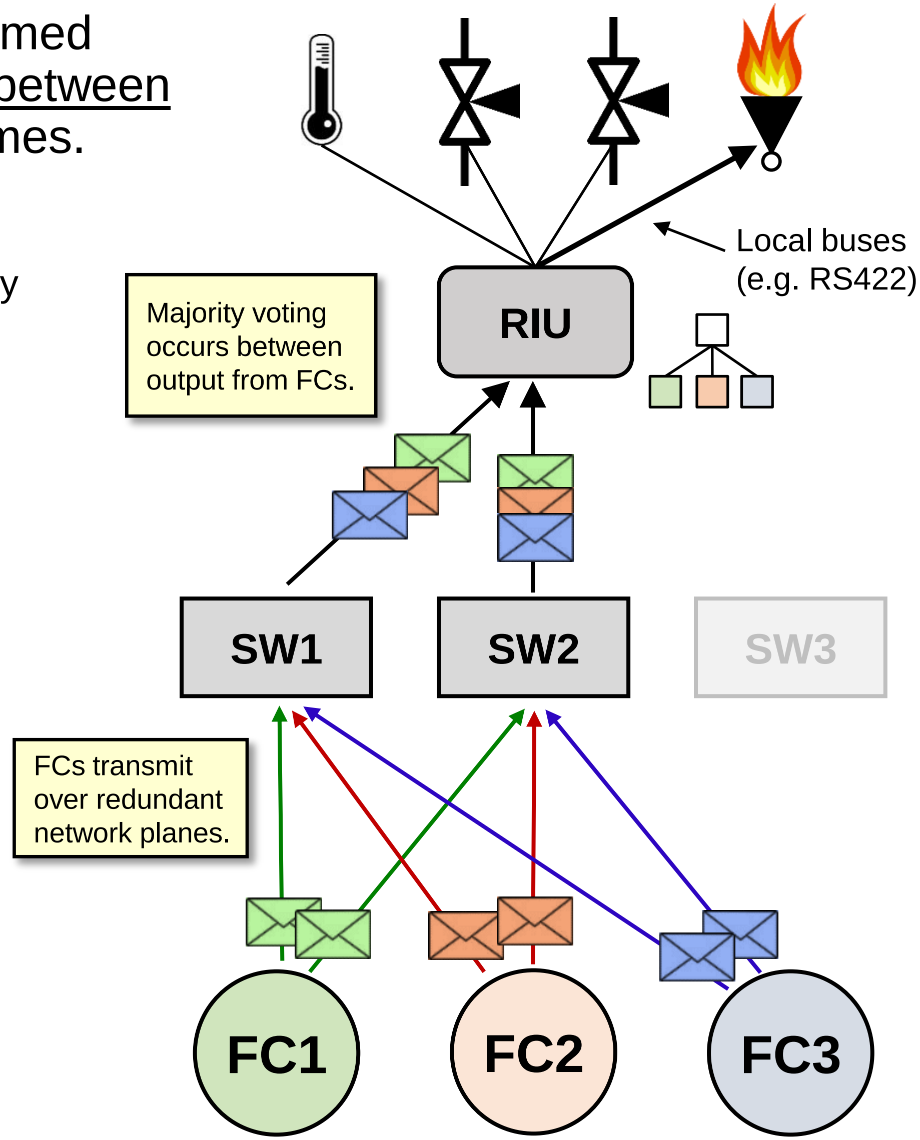


# Notional Fault Tolerance Approach

- Voting of FC commands could be performed at the RIU. The final vote is performed between processors' opinions, not redundant frames.

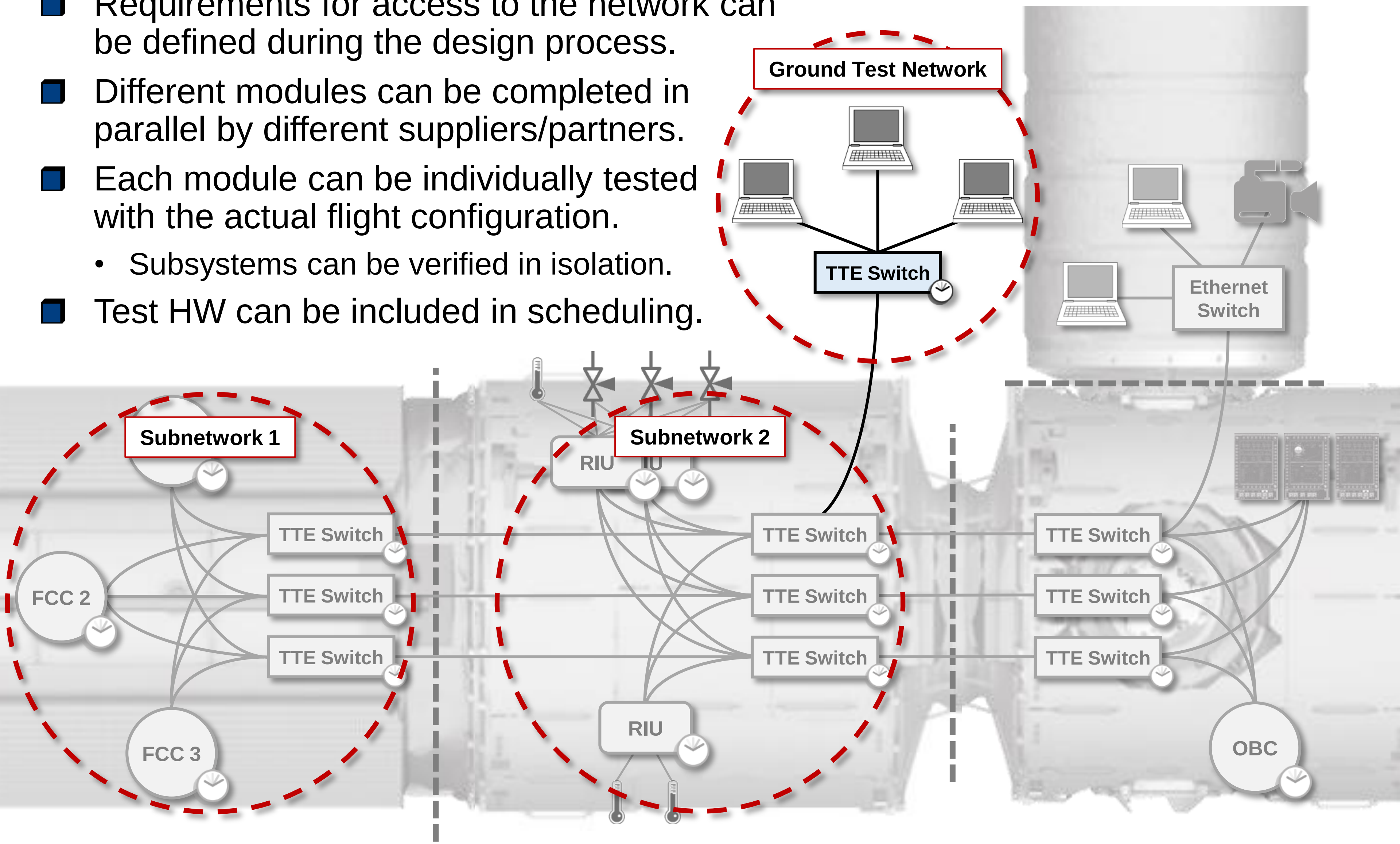
Only two redundant network planes are required for commanding, provided that switches are high-integrity (i.e. COM/MON) and ensure fail-silence.



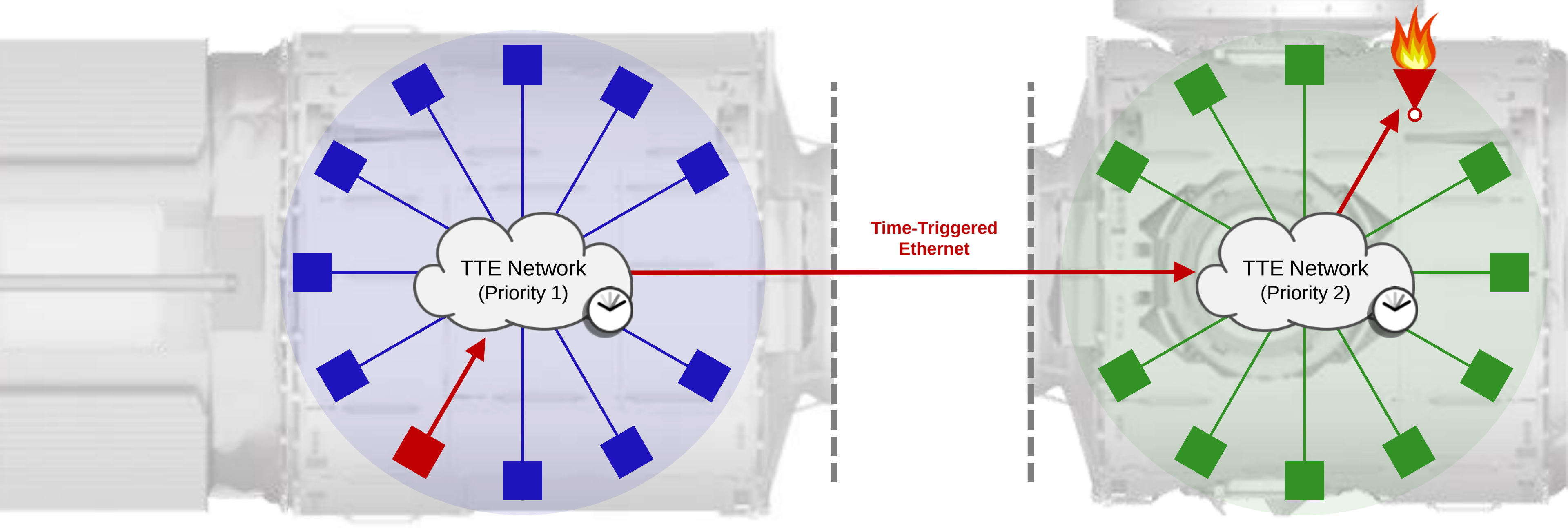
# Network Backplane Composability

- Requirements for access to the network can be defined during the design process.
- Different modules can be completed in parallel by different suppliers/partners.
- Each module can be individually tested with the actual flight configuration.
  - Subsystems can be verified in isolation.
- Test HW can be included in scheduling.



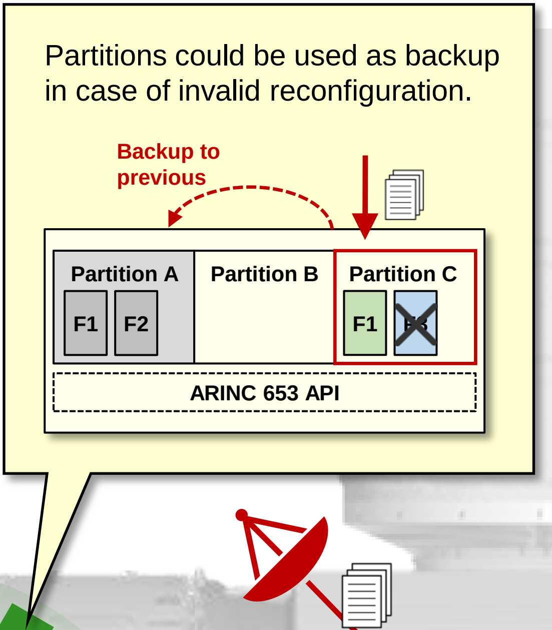
# Incremental Build-Up Approach

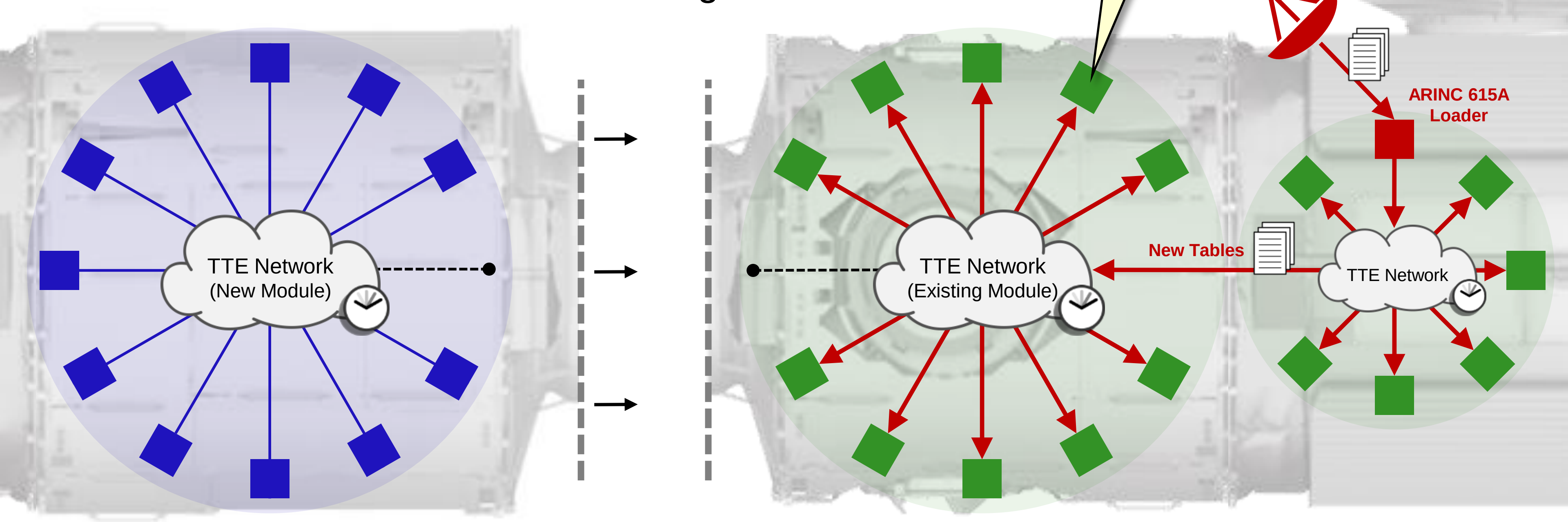
- Vehicles/modules that were developed in cooperation can be launched at different times, perform distinct missions, and join together to form a different system.
  - This approach is taken by networks in the Orion CM and SM.
- I.e. A "super schedule" that accommodates both systems.
- Networks are integrated during docking, and systems in one module are accessible from the other.
- Devices can synchronize to the higher priority network.



# Incremental Build-Up Approach

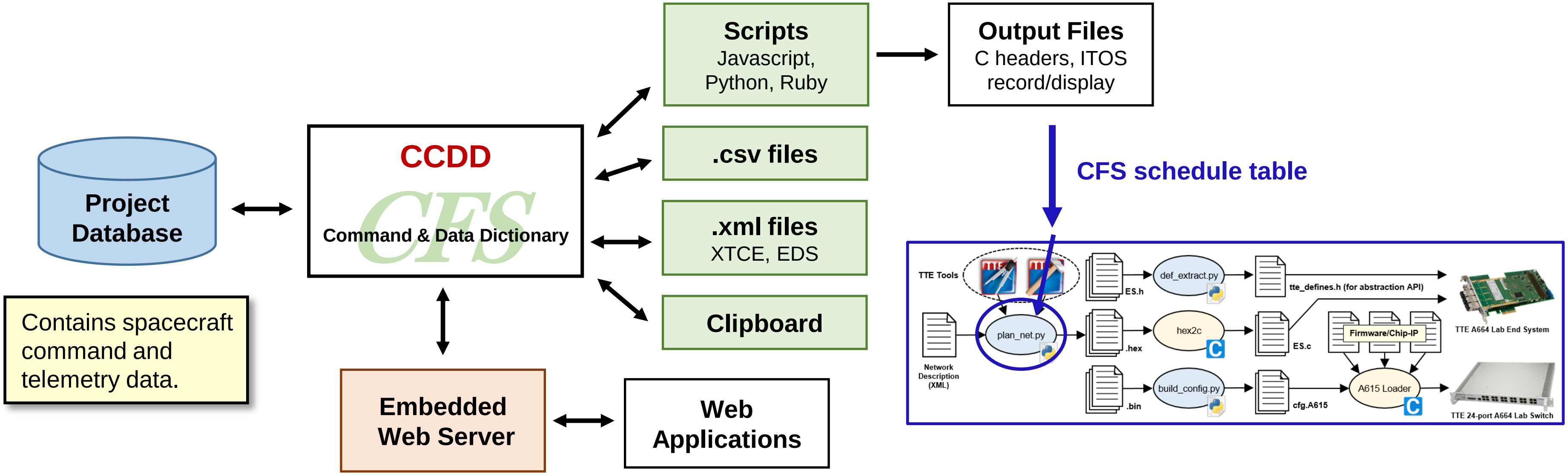
- Modules may also be developed years apart (therefore there is no common scheduling).
- The new module can be launched with updated software and network configurations.
- Existing module receives updated tables from ground.
- Industry-standard loader (e.g. ARINC 615A) distributes configurations over network backplane.
- Existing module verifies correct operation before the new module arrives and is integrated.



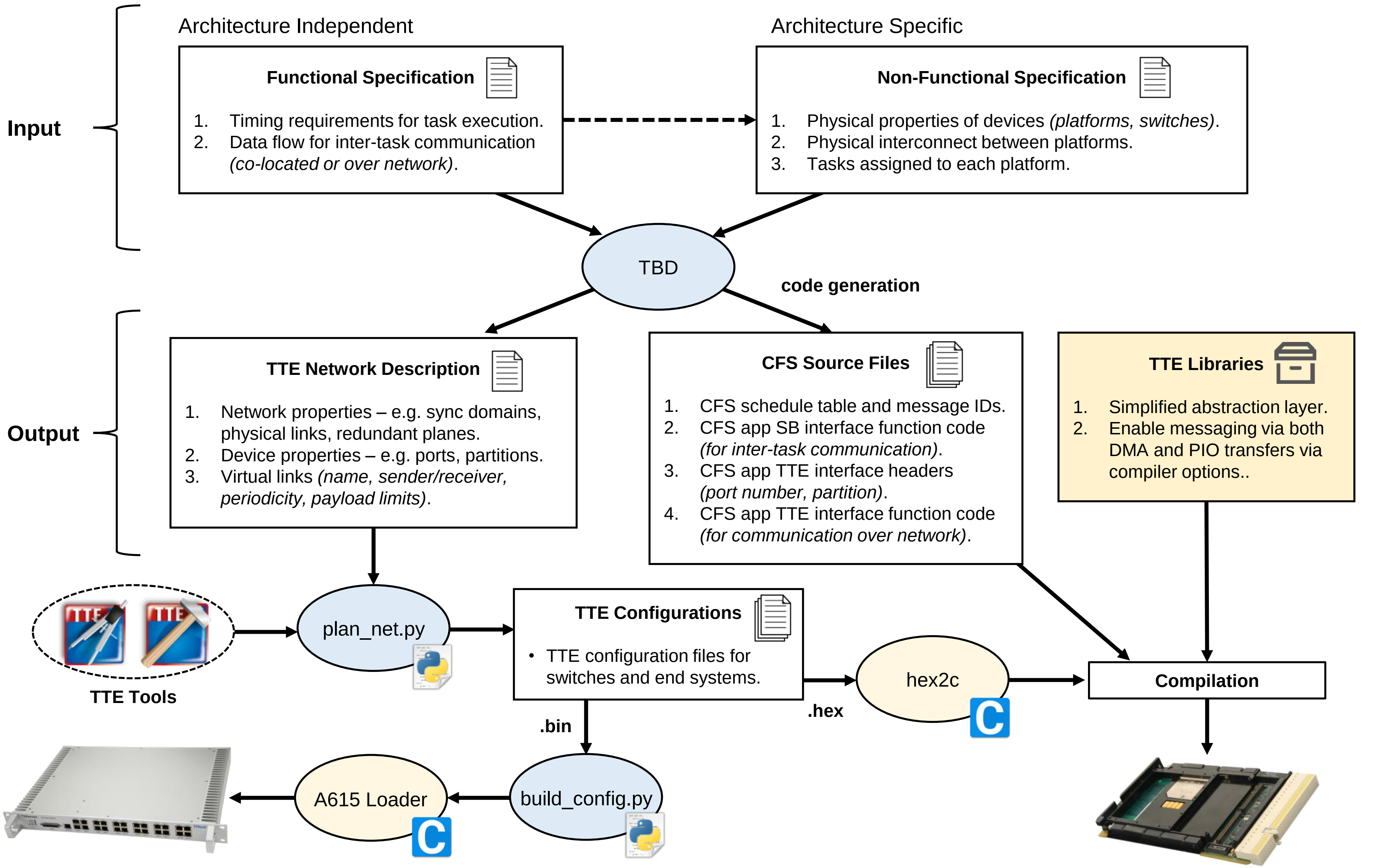



# FSW and Network Scheduling

- **There is no open standard method to couple the scheduling of the flight software and the network.**
  - The latency/determinism benefits of a time-triggered network are lost if the host software is not properly scheduled in relation to the network.
  - Orion solutions for FSW→TTGbE are Third Party Proprietary Information (TPPI).
  - **Planned FY17 Work:** Cooperation with TTTech to demonstrate the ability to use Core Flight Software (CFS) schedule tables as inputs to TTE toolchain.
    - Can potentially generate tables using Command and Data Dictionary (CDD) tool.



# Towards an Integrated Toolchain



# Questions?