On Time-Triggered Ethernet in NASA's Lunar Gateway

Andrew Loveless
NASA Johnson Space Center
andrew.loveless@nasa.gov

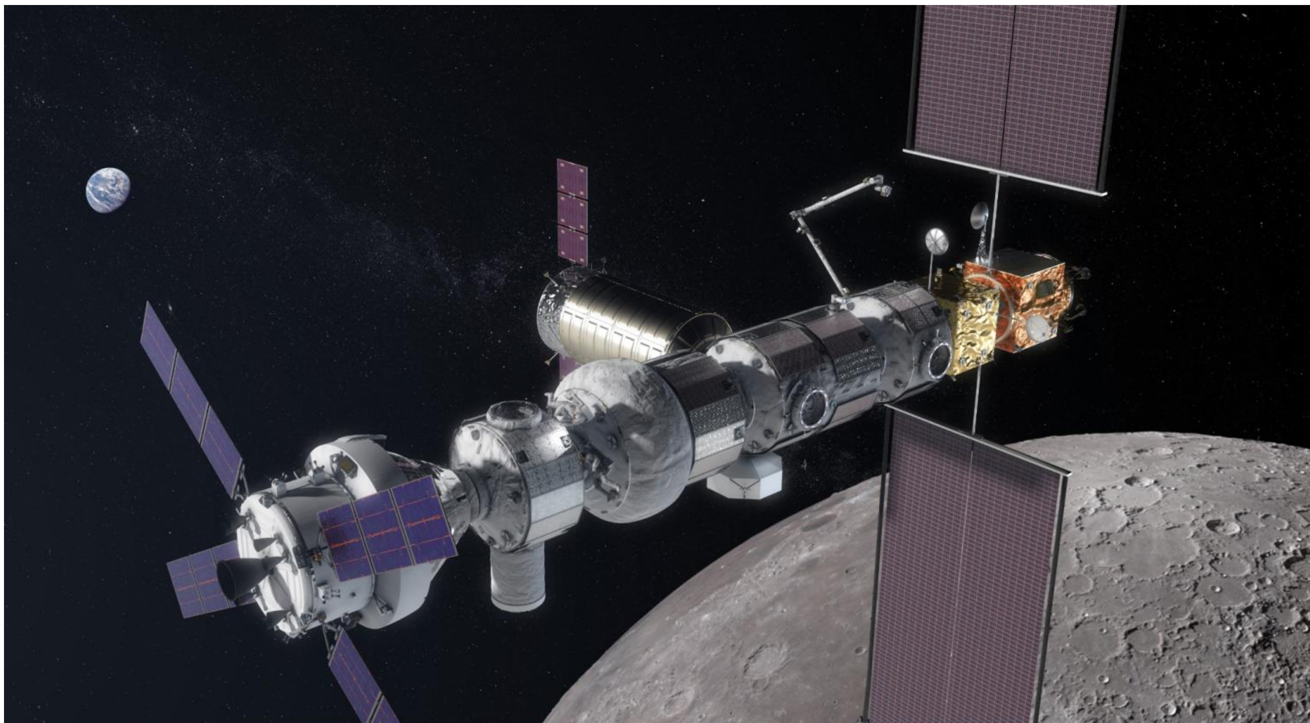
July 30, 2020

Agenda

- Introduction to Gateway
- Time-Triggered Ethernet (TTE) backbone
- TTE, A Fault-Tolerant Interconnect
- TTE, An Integration Framework
- A Unique Challenge, Classical Ethernet
- Conclusion

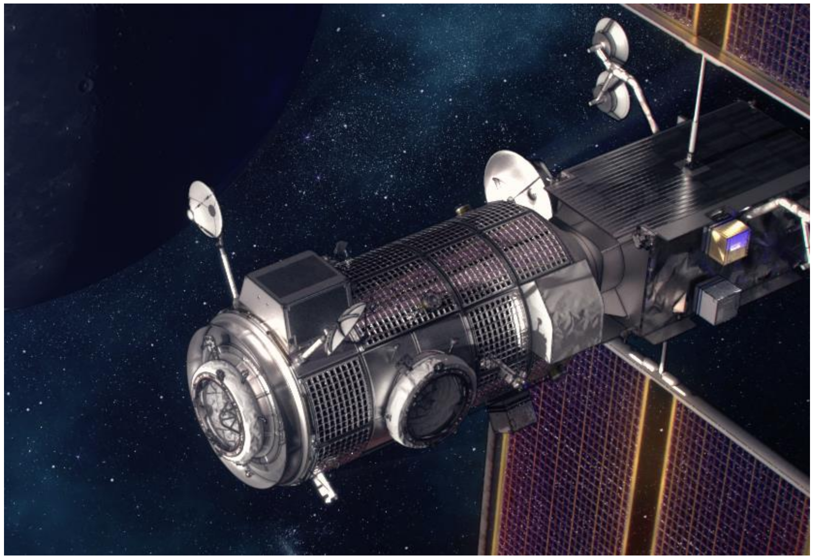
Introduction to Gateway

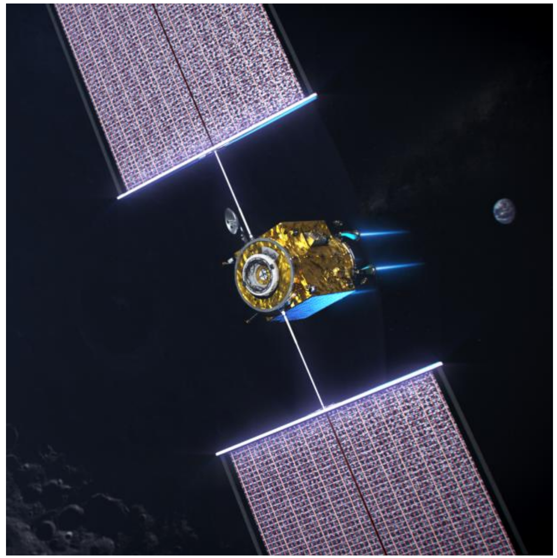
- Lunar outpost under development to support a sustainable human presence on and around the moon



Introduction to Gateway

- Made up of multiple modules:
  - **Power and Propulsion Element (PPE)**
    - Solar electric propulsion spacecraft that will provide power, communications, and attitude control
  - **Habitation and Logistics Outpost (HALO)**
    - Provides life support, command and control, energy storage and power distribution

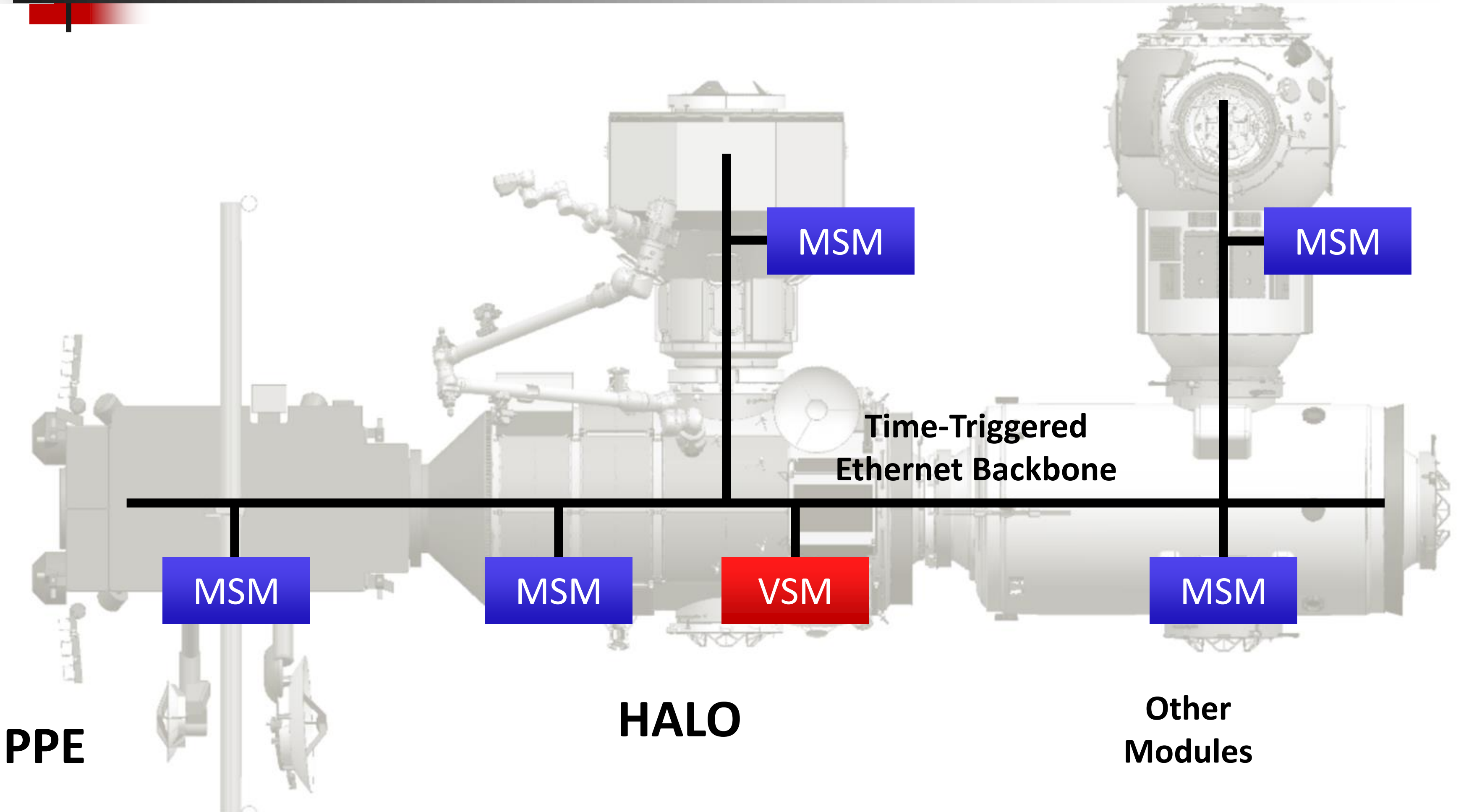




Introduction to Gateway

- Visiting vehicles:
  - Orion, Gateway Logistic Services, Human Landing System (HLS)

Introduction to Gateway



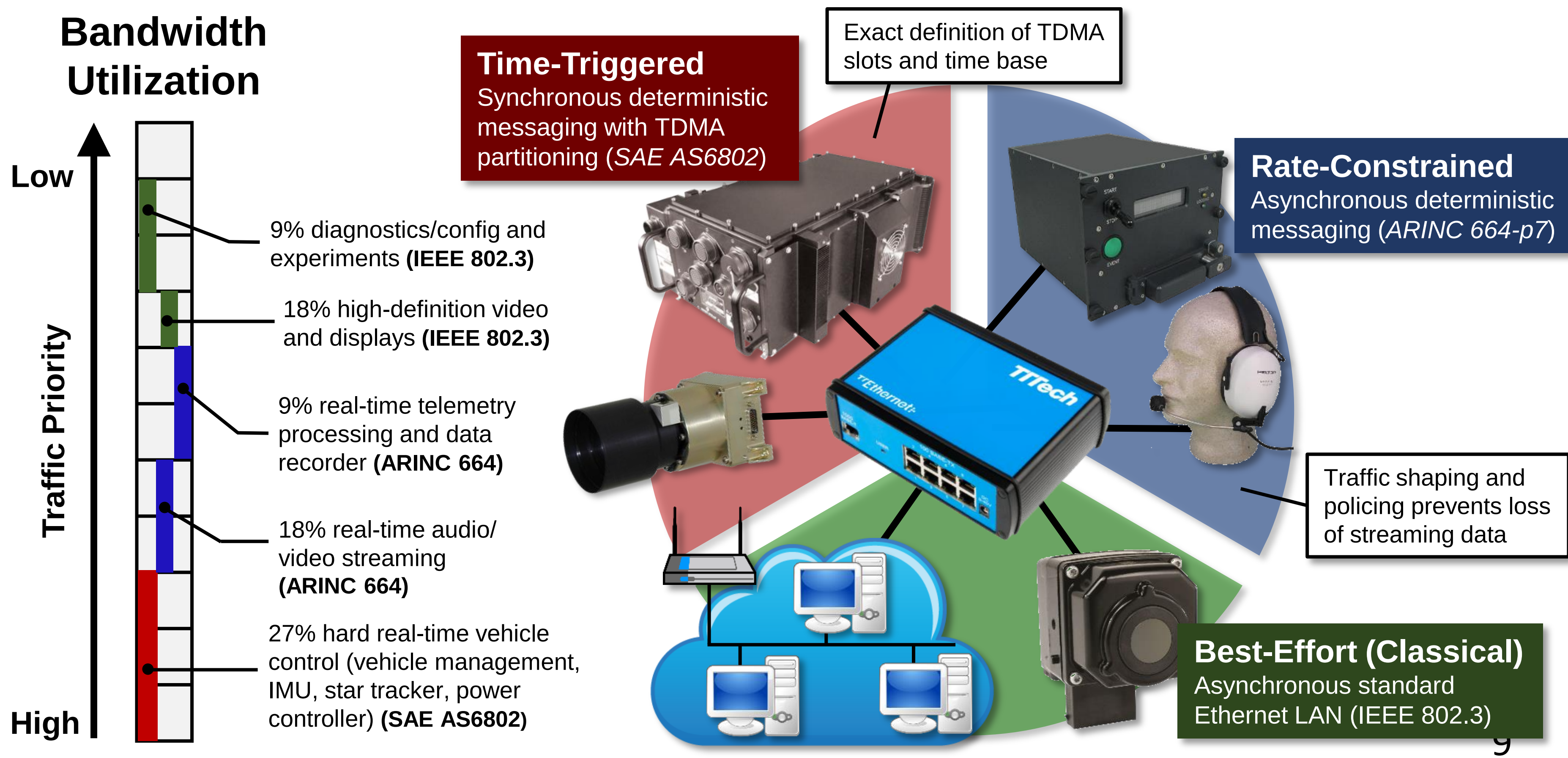
Agenda

- Introduction to Gateway
- Time-Triggered Ethernet (TTE) Backbone
- TTE, A Fault-Tolerant Interconnect
- TTE, An Integration Framework
- A Unique Challenge, Classical Ethernet
- Conclusion

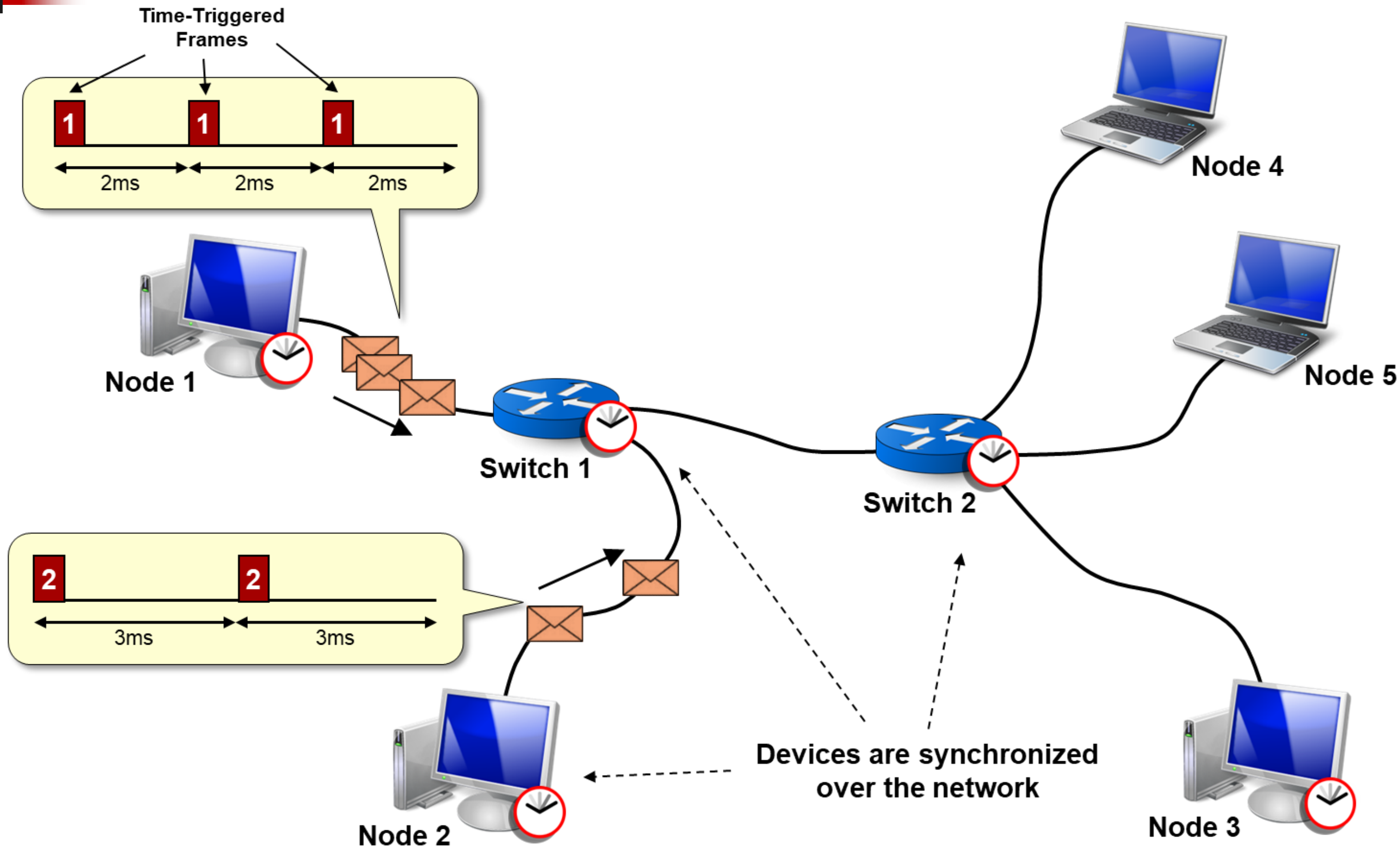
Competing Requirements

- A vehicle like GW puts conflicting requirements on the network
  - It is both a safety-critical vehicle and a research platform
- Control-centric systems need networks with:
  - High integrity and availability
  - Worst case bounded latency and jitter
- Science-centric systems need networks with:
  - Compatibility with COTS devices
  - High throughput
  - Flexibility and expandability
- Often the same computer needs both:
  - E.g. IMA – functions with different criticalities on the same computer

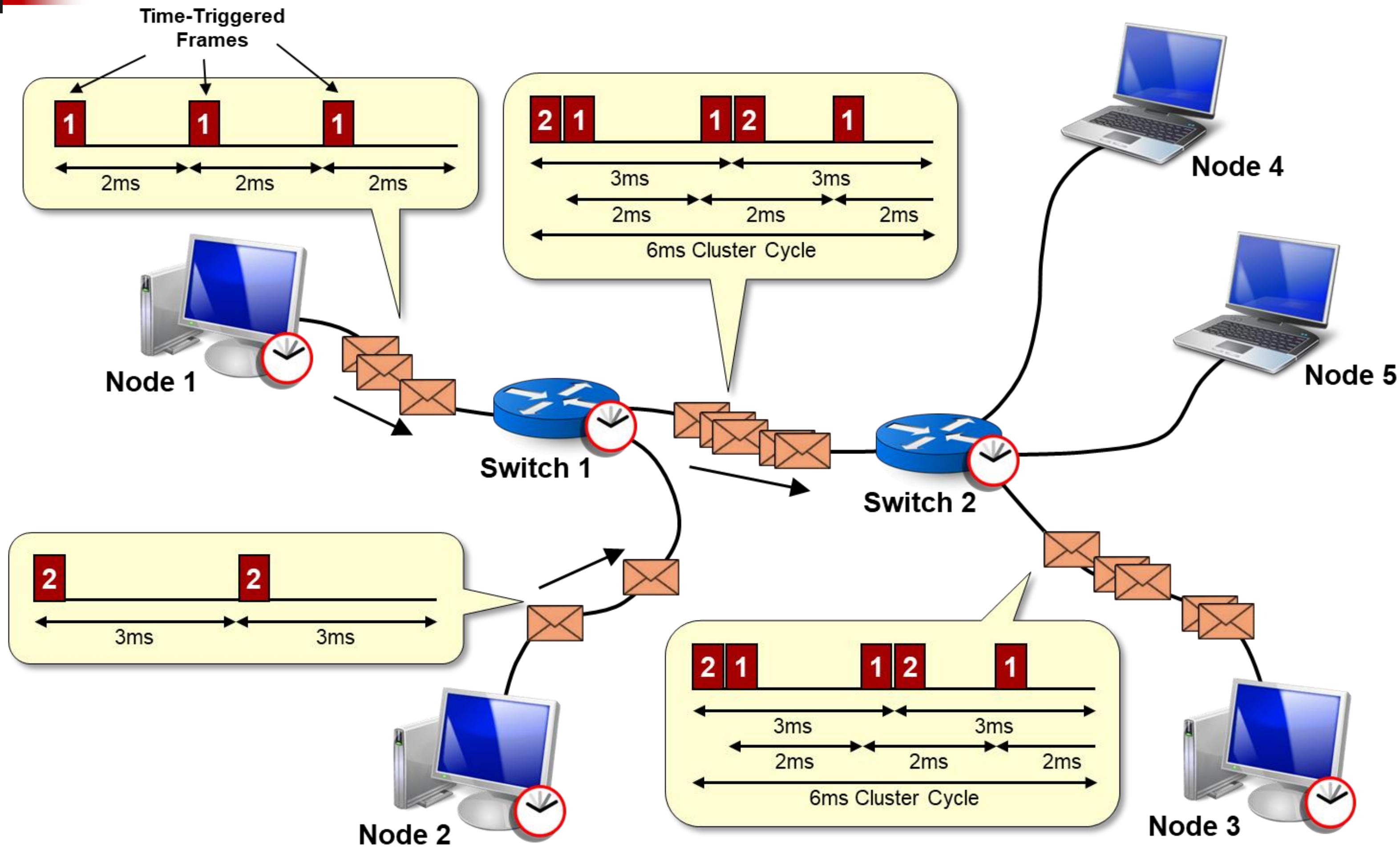
Time-Triggered Ethernet (TTE)



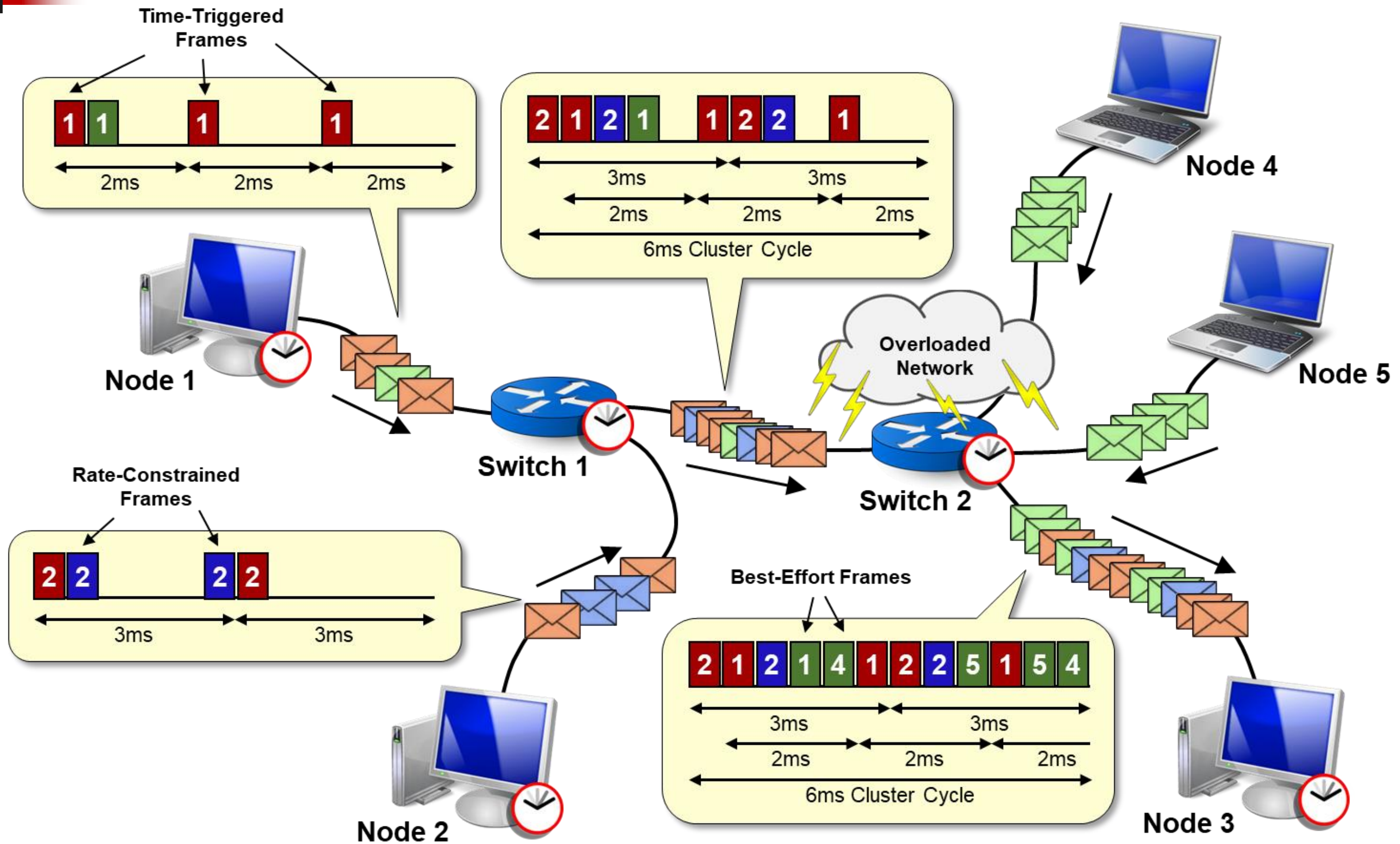
TTE Traffic Integration



TTE Traffic Integration

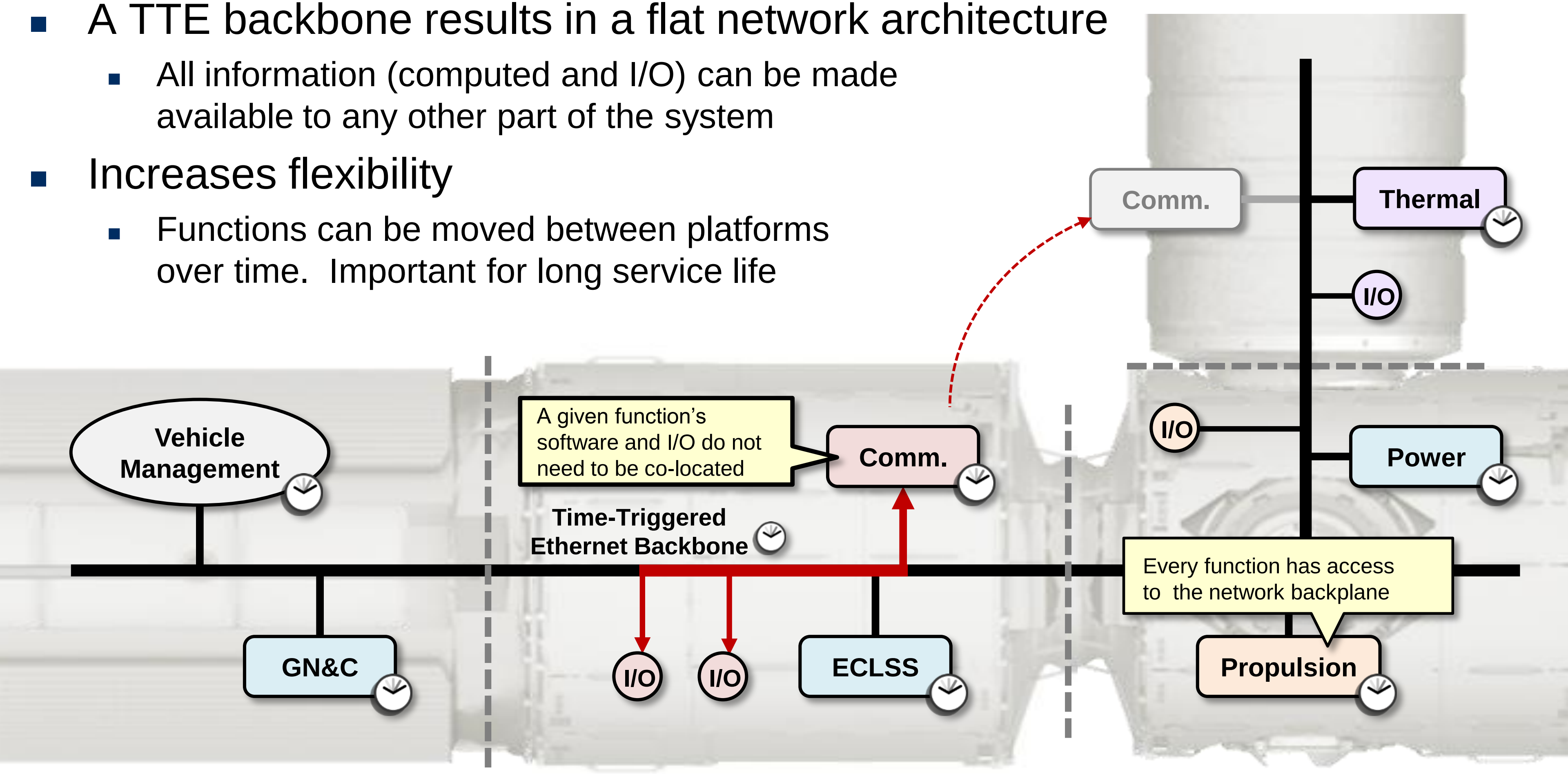


TTE Traffic Integration



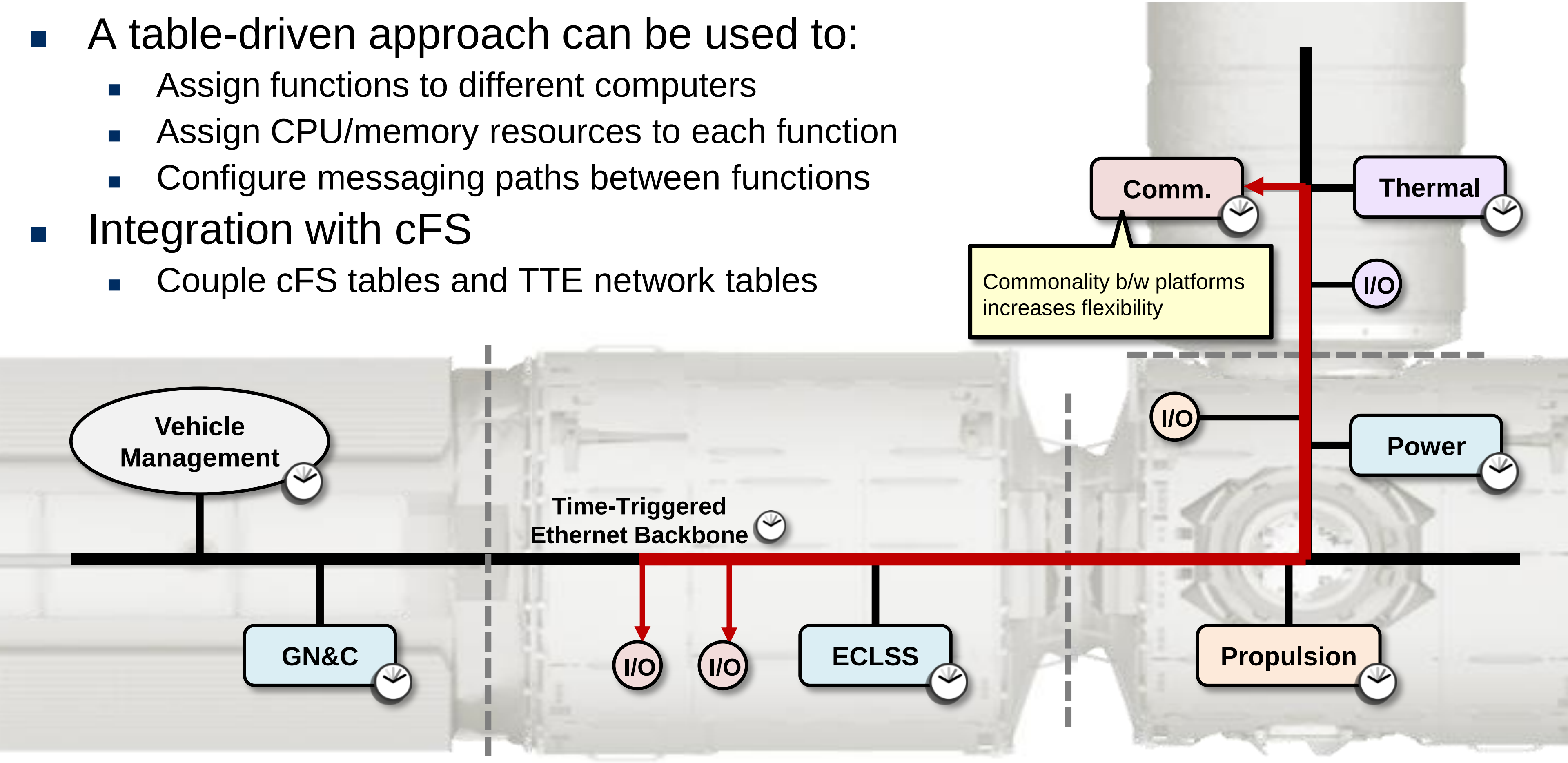
A Flat Network Architecture

- A TTE backbone results in a flat network architecture
  - All information (computed and I/O) can be made available to any other part of the system
- Increases flexibility
  - Functions can be moved between platforms over time. Important for long service life

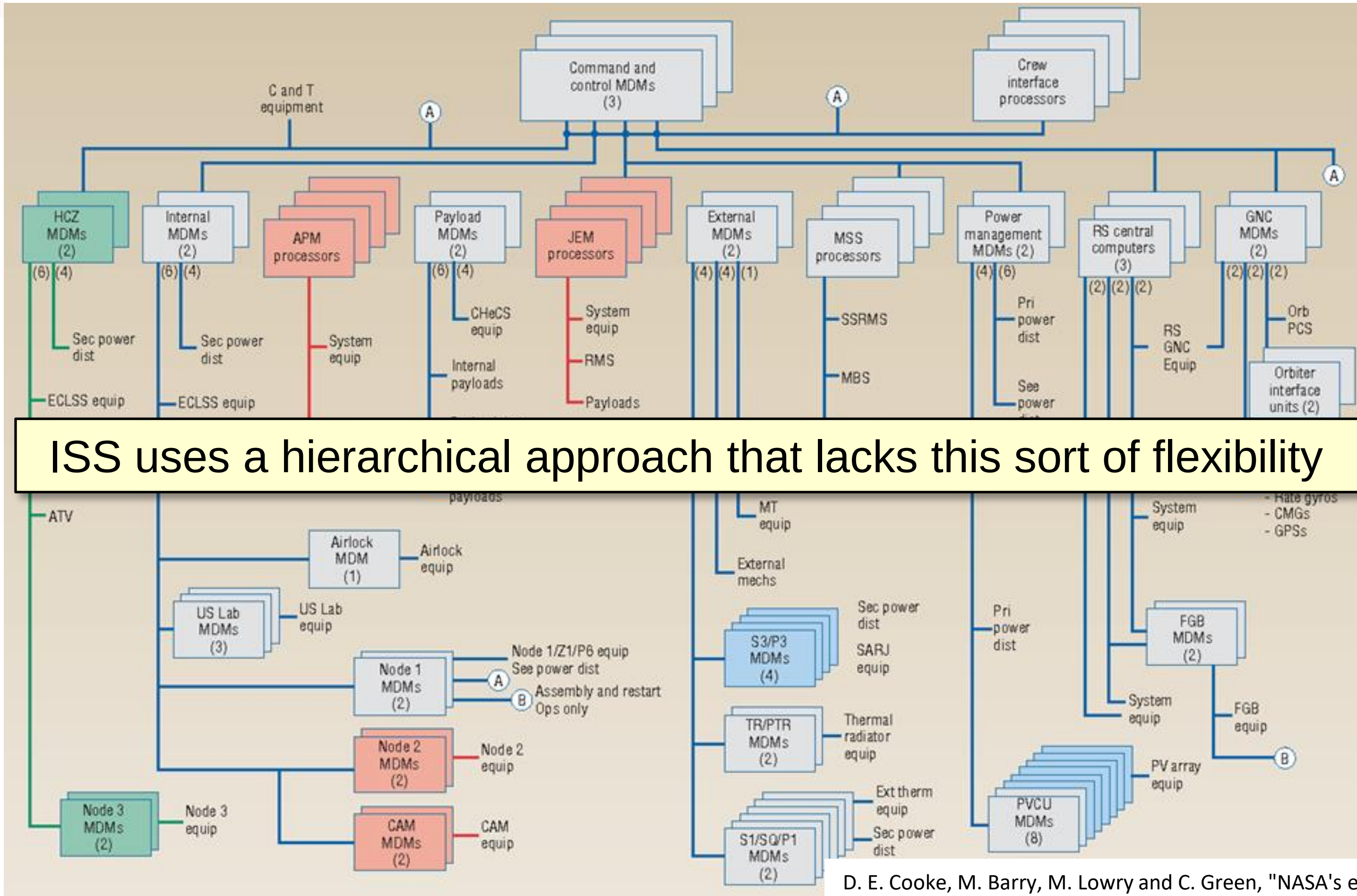


A Flat Network Architecture

- A table-driven approach can be used to:
  - Assign functions to different computers
  - Assign CPU/memory resources to each function
  - Configure messaging paths between functions
- Integration with cFS
  - Couple cFS tables and TTE network tables



A Flat Network Architecture

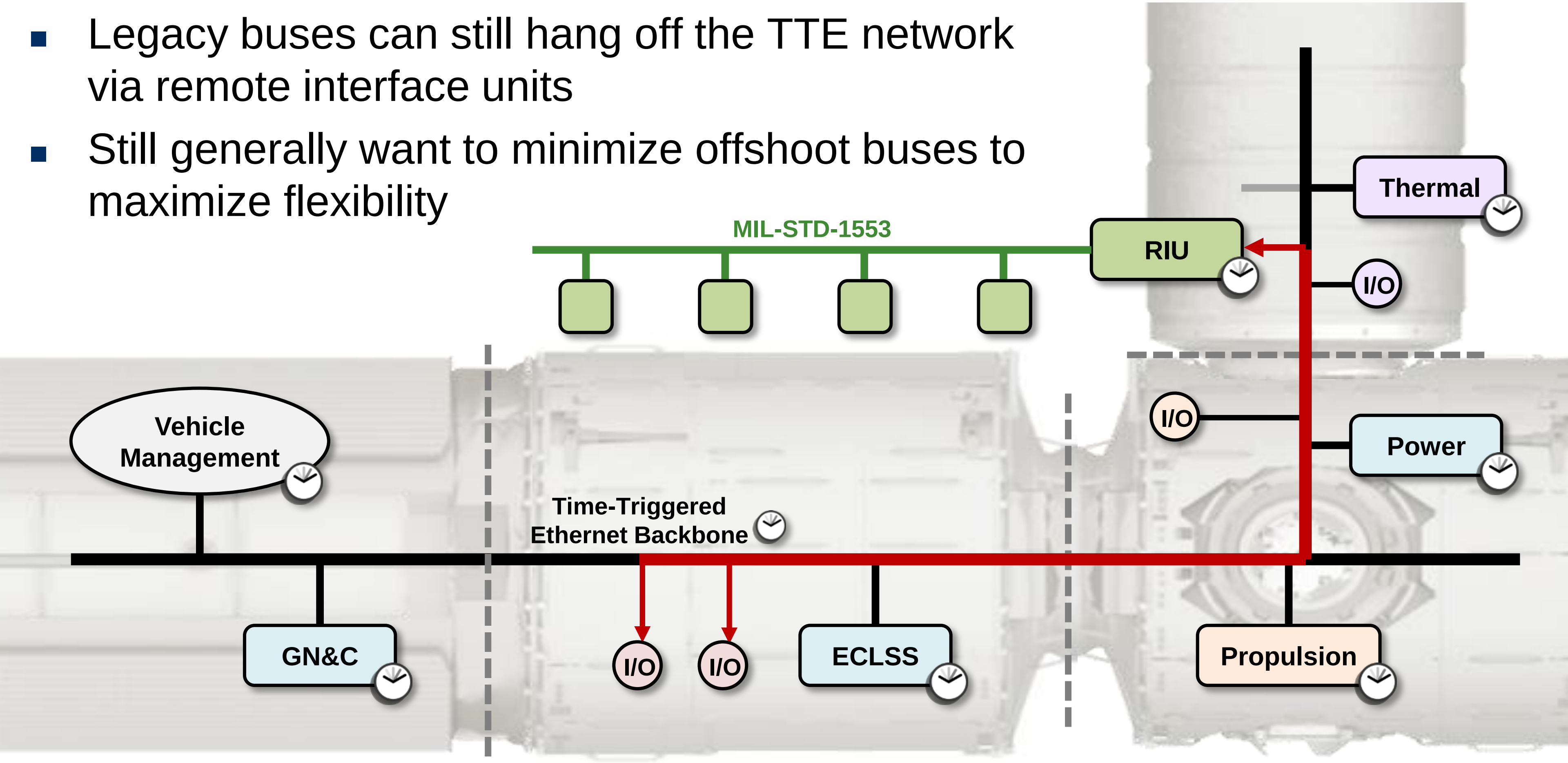


ISS uses a hierarchical approach that lacks this sort of flexibility

D. E. Cooke, M. Barry, M. Lowry and C. Green, "NASA's exploration agenda and capability engineering," in *Computer*, vol. 39, no. 1, pp. 63-73, Jan. 2006

A Flat Network Architecture

- Legacy buses can still hang off the TTE network via remote interface units
- Still generally want to minimize offshoot buses to maximize flexibility

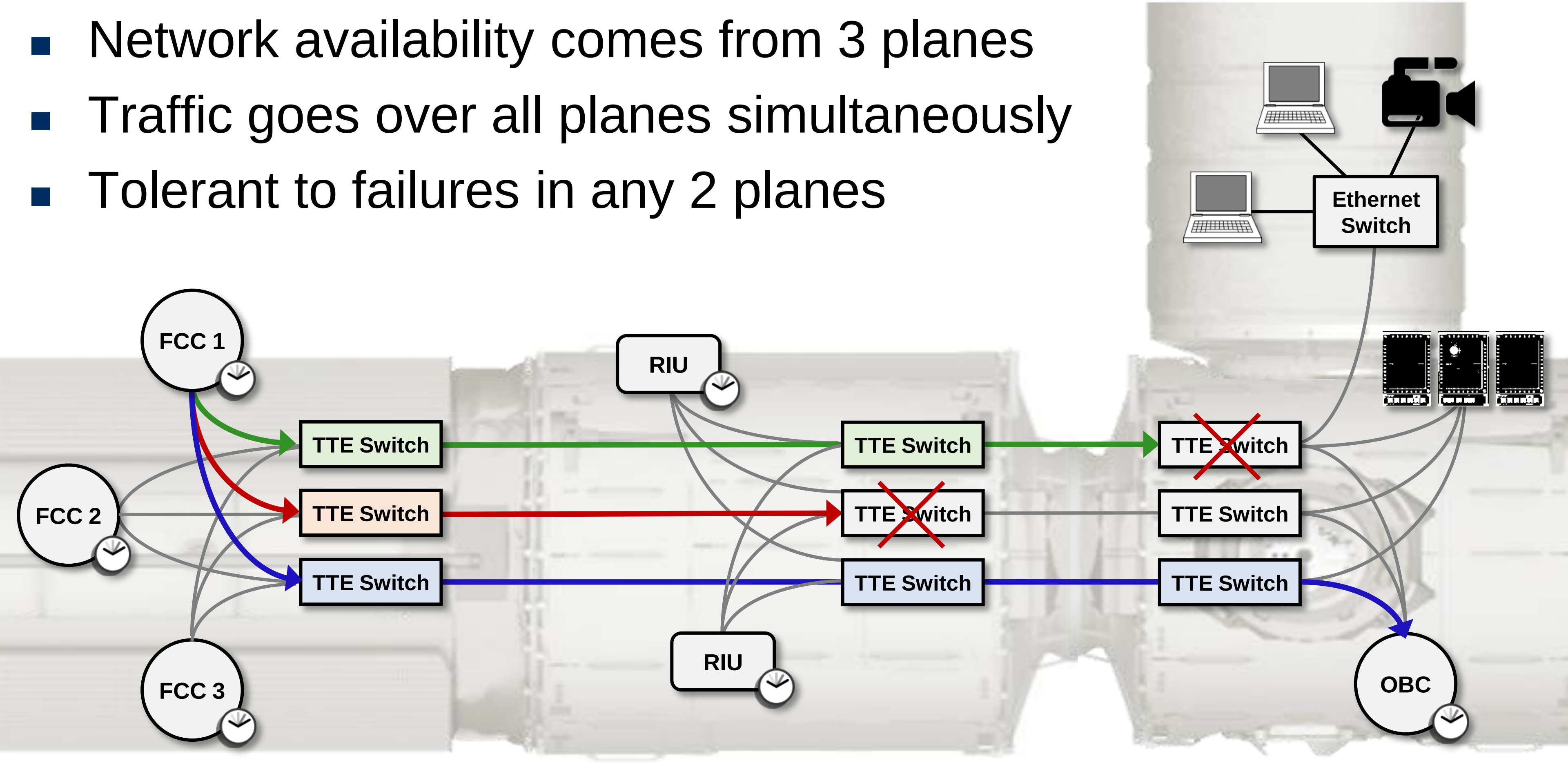


Agenda

- Introduction to Gateway
- Time-Triggered Ethernet (TTE) backbone
- TTE, A Fault-Tolerant Interconnect
- TTE, An Integration Framework
- A Unique Challenge, Classical Ethernet
- Conclusion

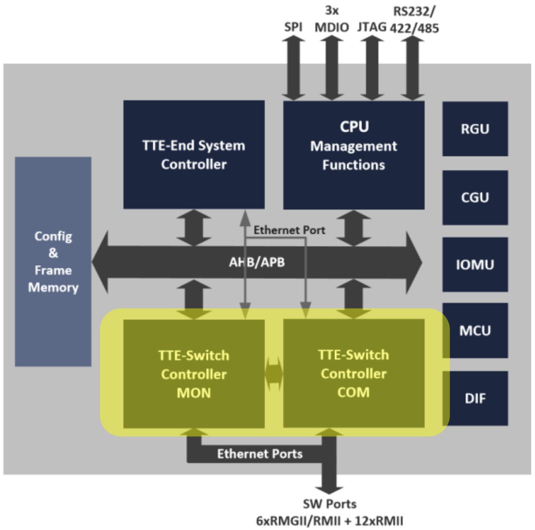
TTE Network Availability

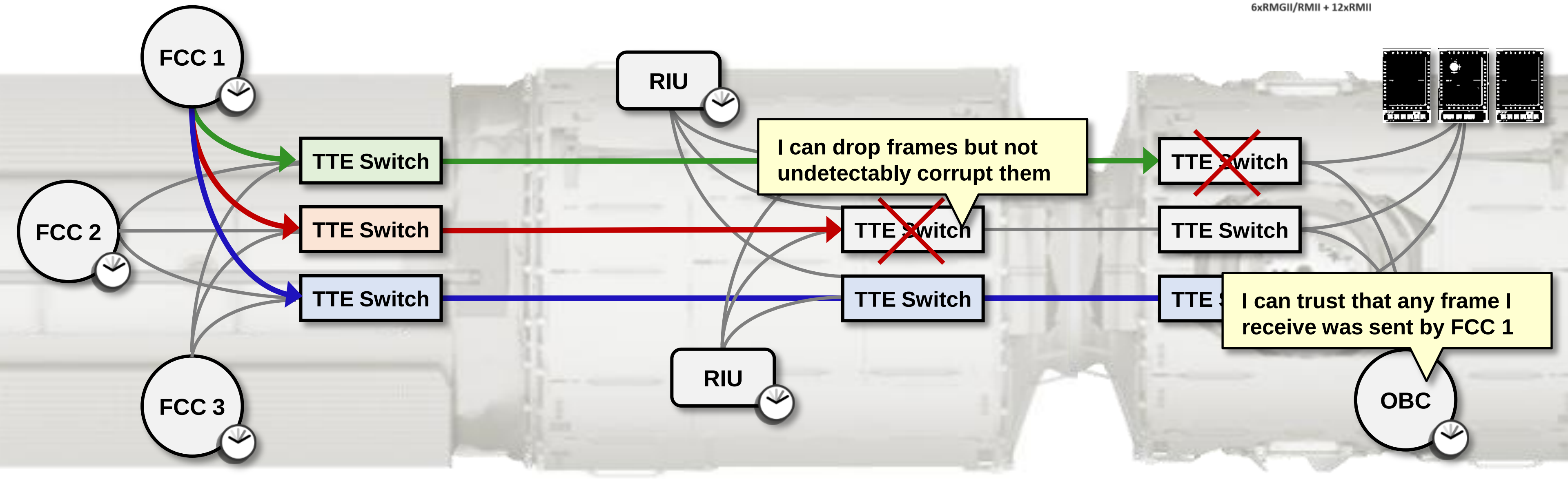
- Network availability comes from 3 planes
- Traffic goes over all planes simultaneously
- Tolerant to failures in any 2 planes



TTE Network Integrity

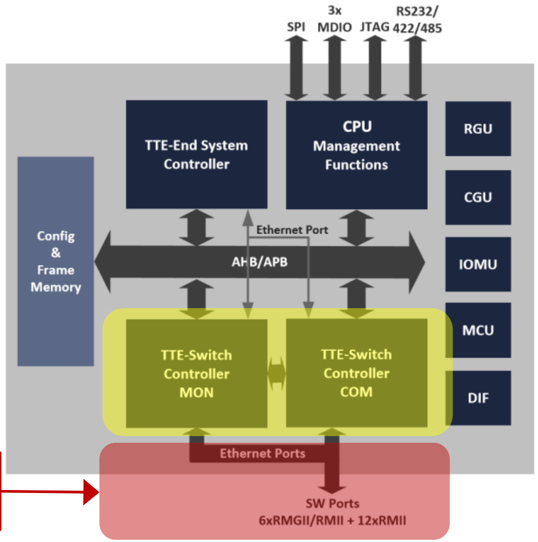
- High-integrity switch design prevents undetectable frame corruption
- Uses COM/MON with two switch IPs



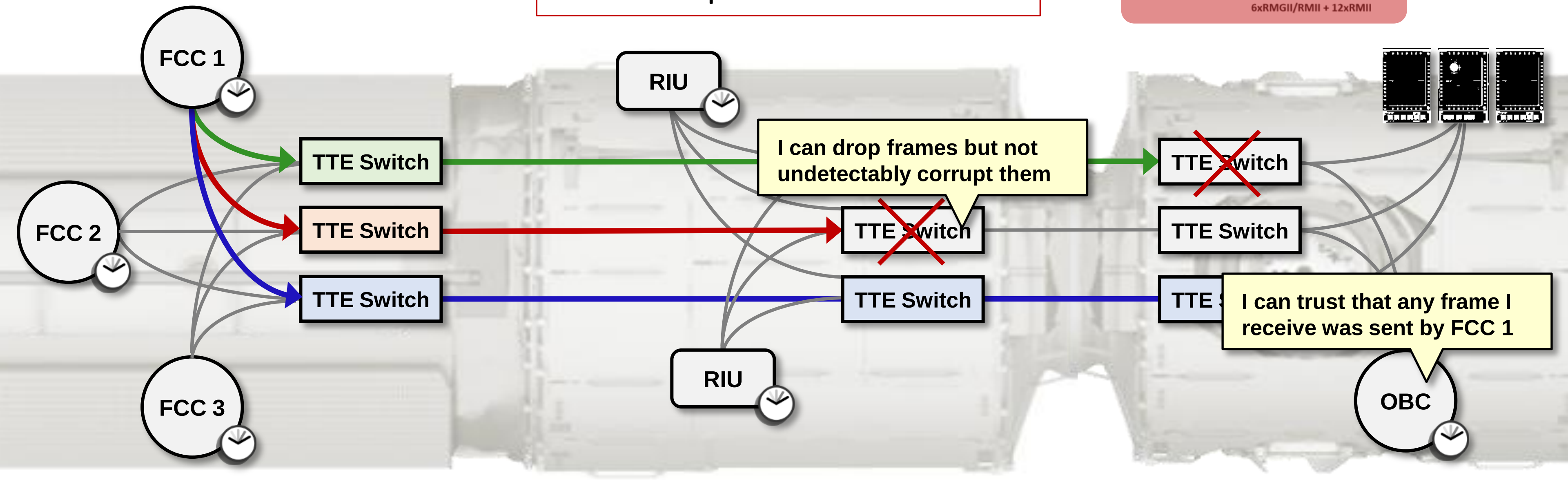


TTE Network Integrity

- High-integrity switch design prevents undetectable frame corruption
- Uses COM/MON with two switch IPs

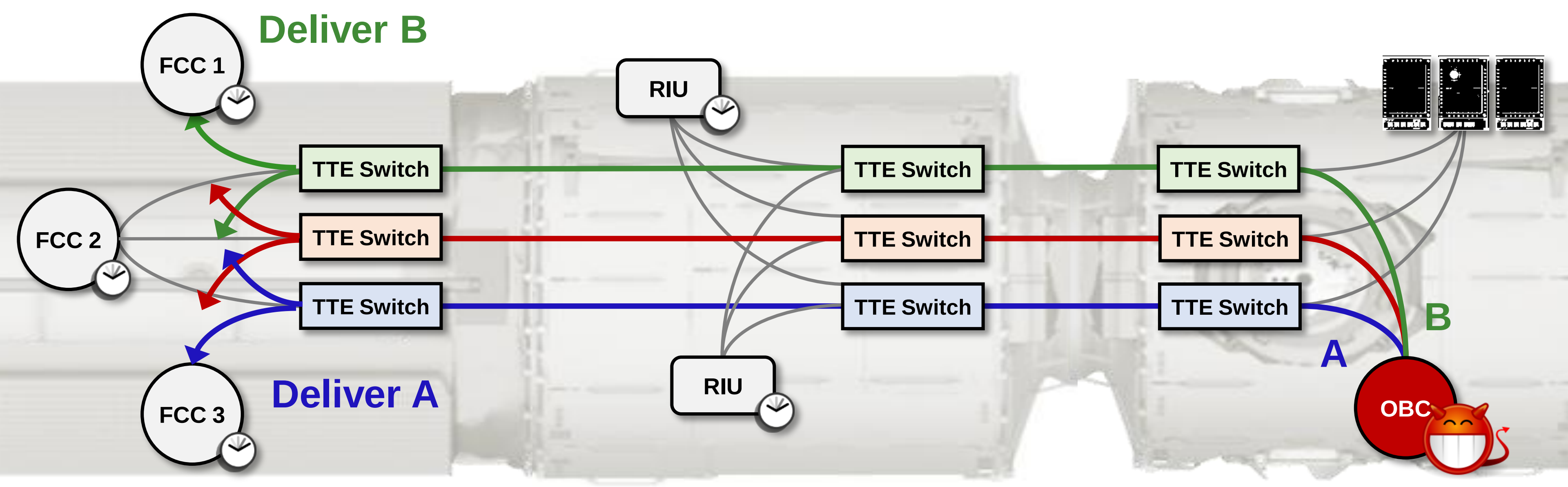


Still susceptible to faults here



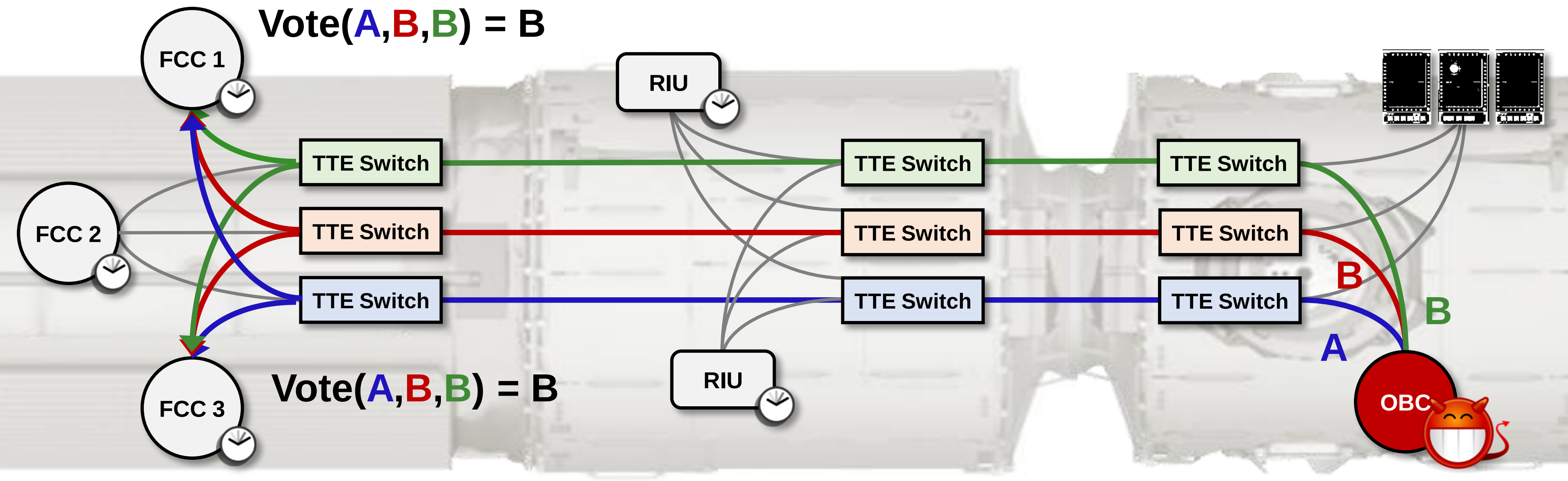
TTE Asymmetric Transmissions

- A faulty ES can send different frames to different planes
- Receivers deliver the first valid frame
- Means different receivers can deliver different frames!



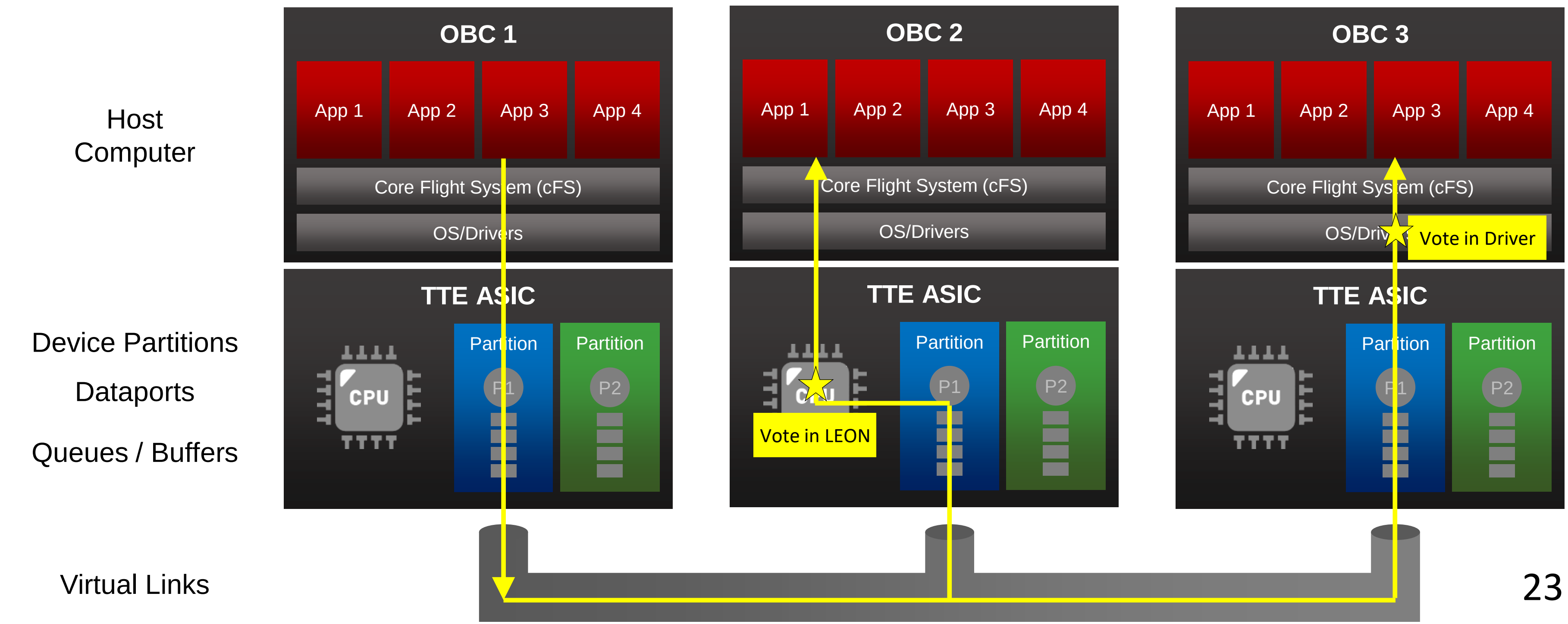
TTE Asymmetric Transmissions

- Return all redundant frames to the host
- Hybrid majority vote (exclude manifest faulty frames)
- Guarantees all correct receivers deliver the same message



Voting Realization

- Voting is configurable on a VL-by-VL basis – groups redundant frames by arrival time
- Can be performed in the Leon CPU in TTE ASIC, or in the driver software



TTE Broadcast Channel

- Lets you realize a **broadcast channel** abstraction

- Given 1 faulty end system:
  - If the sender is correct and broadcasts v, all correct receivers get v
  - All correct receivers deliver the same value

- Given 1 faulty end system + 1 faulty switch:
  - If the sender is correct and broadcasts v, all correct receivers get v
  - All correct receivers ***that deliver a value*** deliver the same value

TTE Broadcast Channel

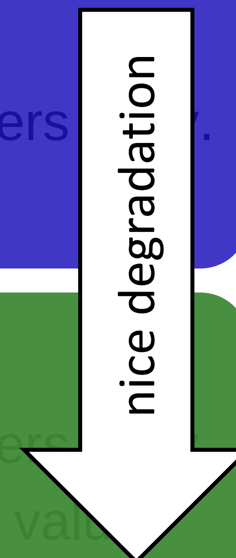
- Lets you realize a **broadcast channel** abstraction

- Given 1 faulty end system:
  - If the sender is corre… al… correct receivers … v.
  - All correct receivers deliver the same value.

Byzantine Agreement

- Given 1 faulty end system + 1 faulty switch:
  - If the sender is corre… correct receive…
  - All correct receivers <u>**that deliver a value**</u> deliver the same val…

Crusader Agreement

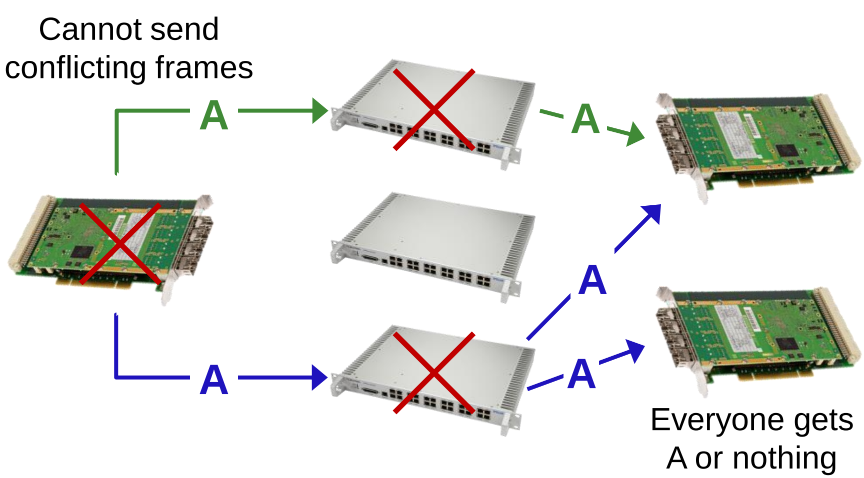


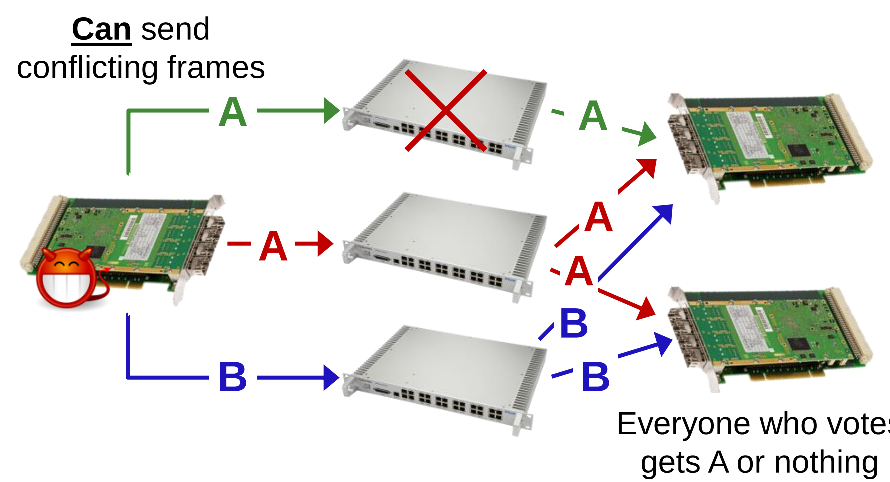
Integrity/Availability Tradeoff

Orion (Phoenix IP) → **Gateway (Pegasus IP)**

- Has **high-integrity** end systems
- Crusader broadcast channel with faulty ES and **two faulty switches**

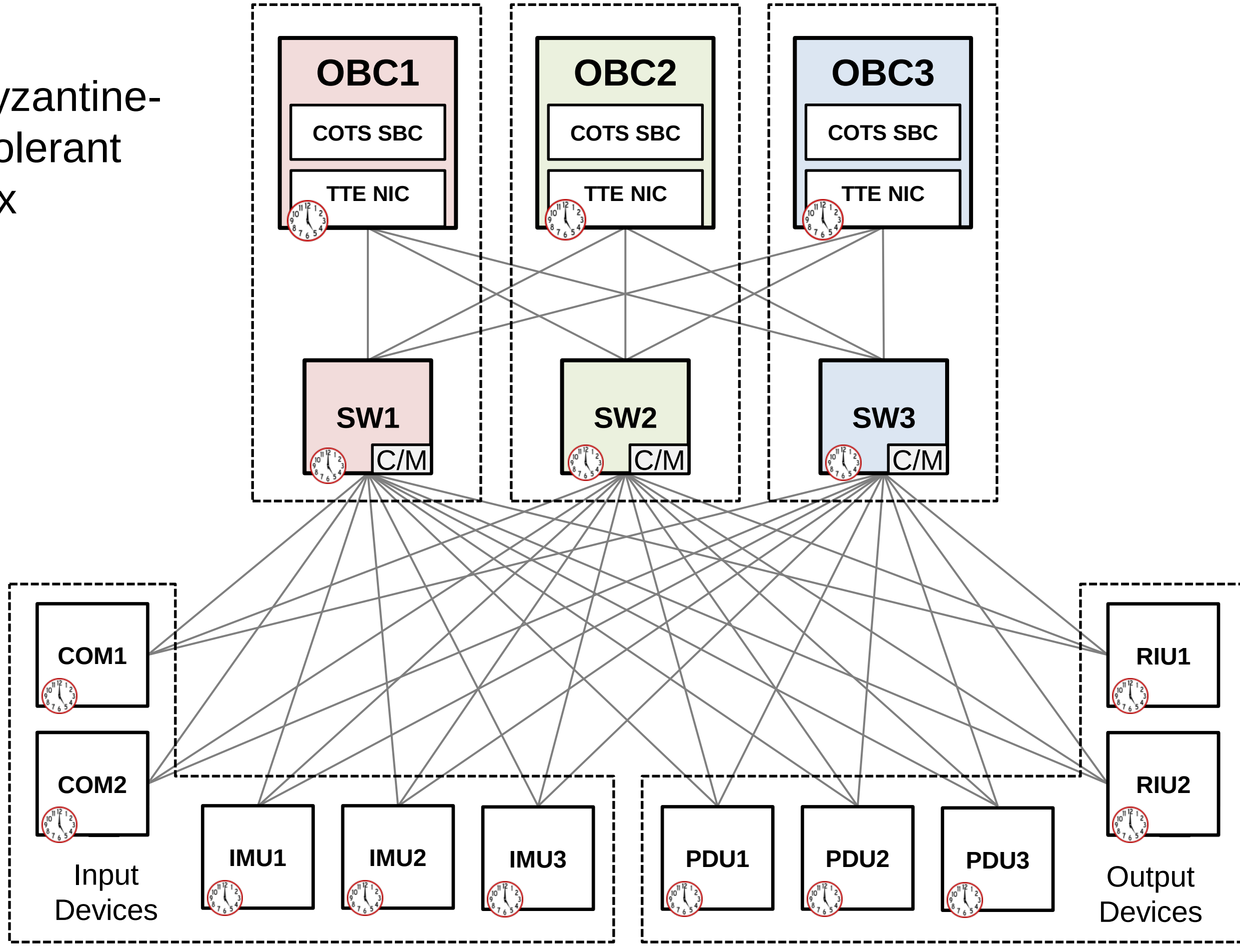
- Only **standard-integrity** end systems
- Crusader broadcast channel with faulty ES and **one faulty switch**





Building on Broadcast Channels

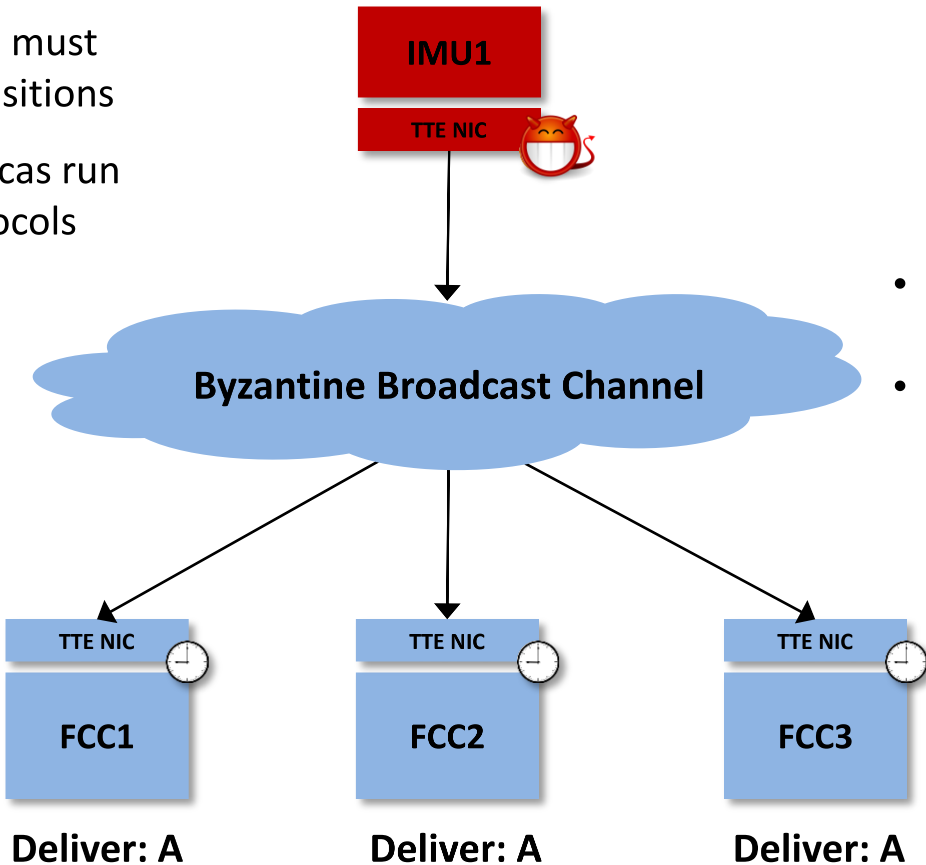
Example: 1 Byzantine-resilient fault-tolerant switched triplex



Agreeing on External Data

Key tenet of SMR: Replicas must go through same state transitions

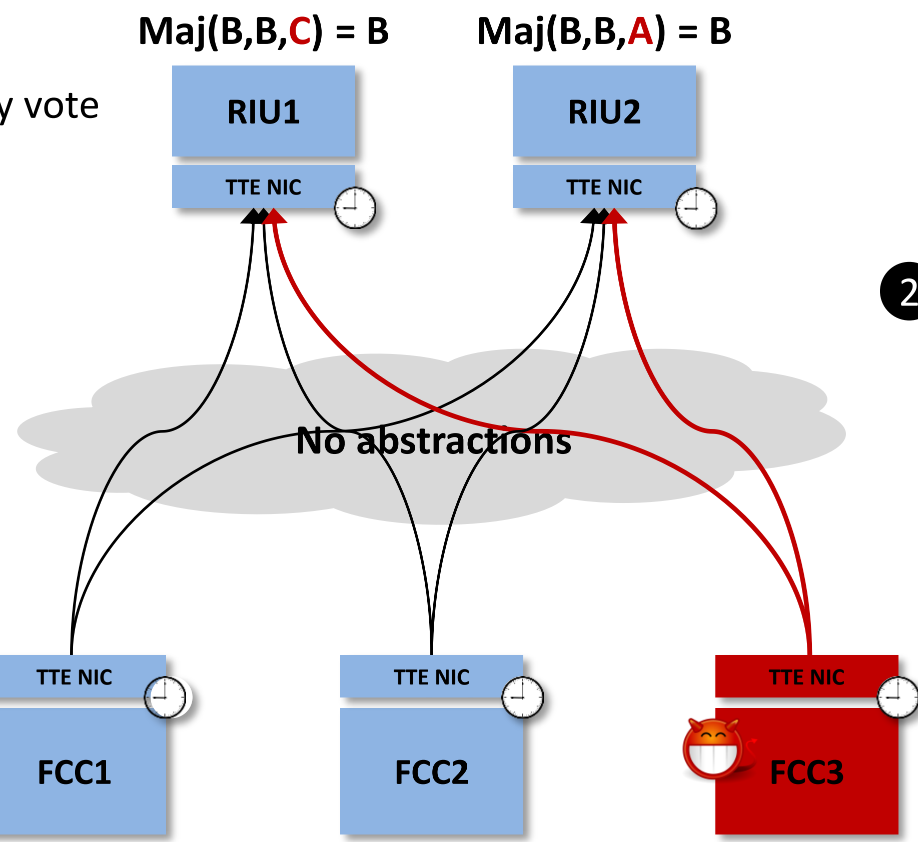
How to accomplish?: Replicas run Byzantine Agreement protocols on all input data



- Makes agreement on inputs (and state) trivial
- No explicit 2 round exchange needed

Must be consistent

Commanding Actuators



3 Receivers majority vote the commands

2 Receivers use normal first valid approach

1 A faulty NIC sends different frames to different planes

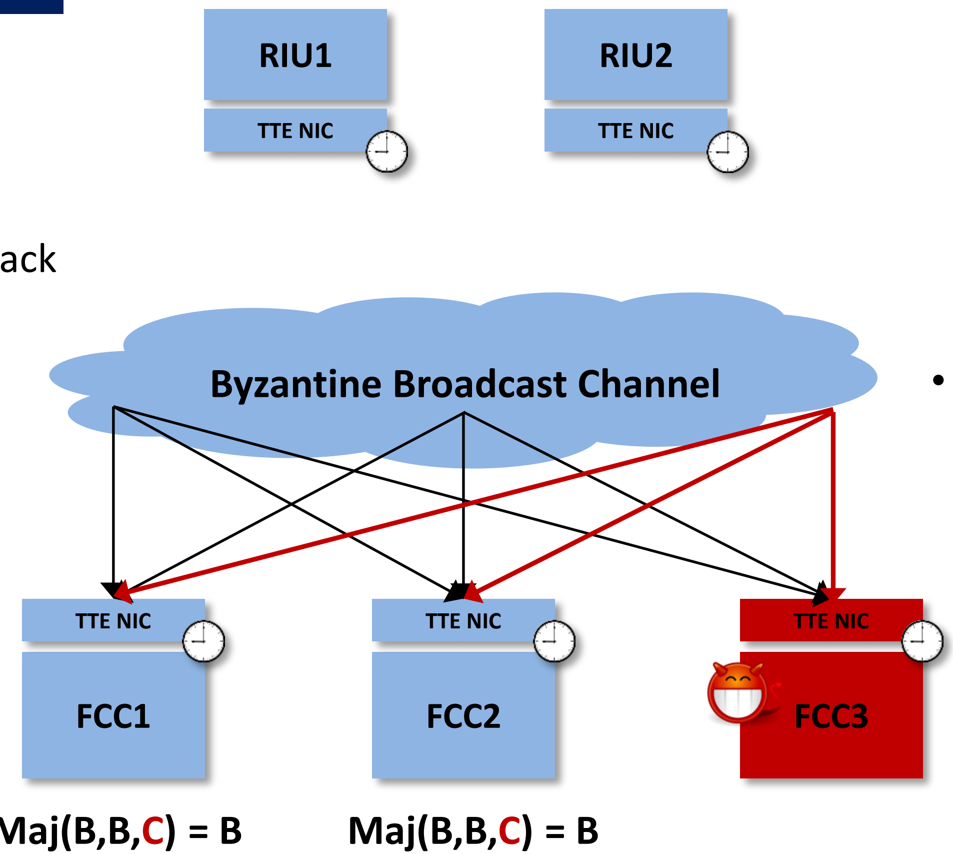
All correct replicas have same state so generate same output

Fault Diagnosis

Happening Simultaneously …

1 Outputs are reflected back via broadcast channel

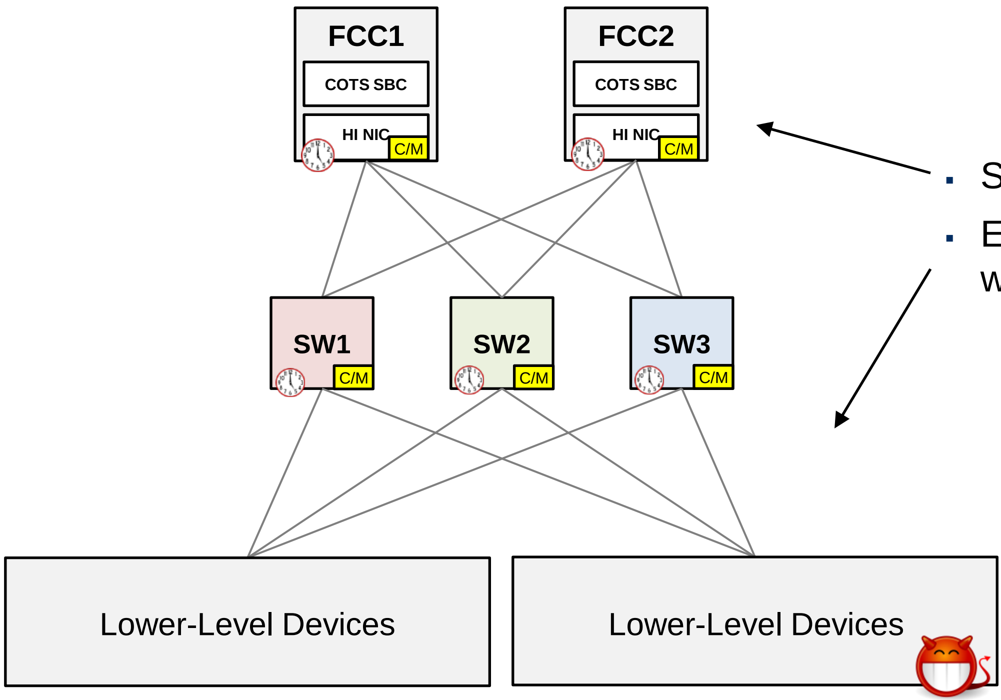
2 Replicas vote the **consistent** outputs from the replicas. Any replica that disagrees *is faulty.*



- Uses same VLs as RIUs are reading

Building on Broadcast Channels

- Broadcast channels are applicable to many fault models and fault tolerance approaches
- E.g. A primary-backup system constrained to symmetric transmissive failures



- Still need to maintain consistent internal state
- Even if OBC1/2 can be assumed to fail in benign ways, the lower-level devices may not

Agenda

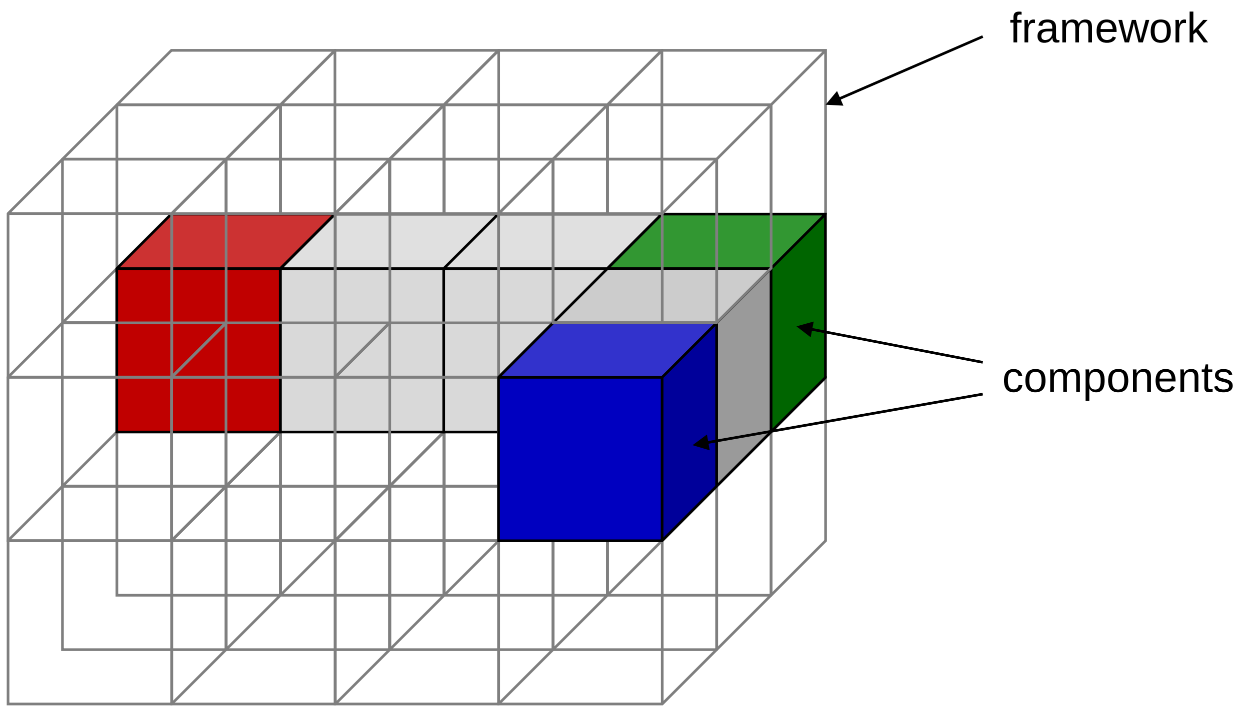
- Introduction to Gateway
- Time-Triggered Ethernet (TTE) backbone
- TTE, A Fault-Tolerant Interconnect
- TTE, An Integration Framework
- A Unique Challenge, Classical Ethernet
- Conclusion

TTE: It’s a Framework

- TTE is not just an interconnect, it is an **integration framework**

Framework provides the supporting structure for integrating components into the system

- Components can rely on the framework
- Framework doesn’t rely on the components



TTE: It's a Framework

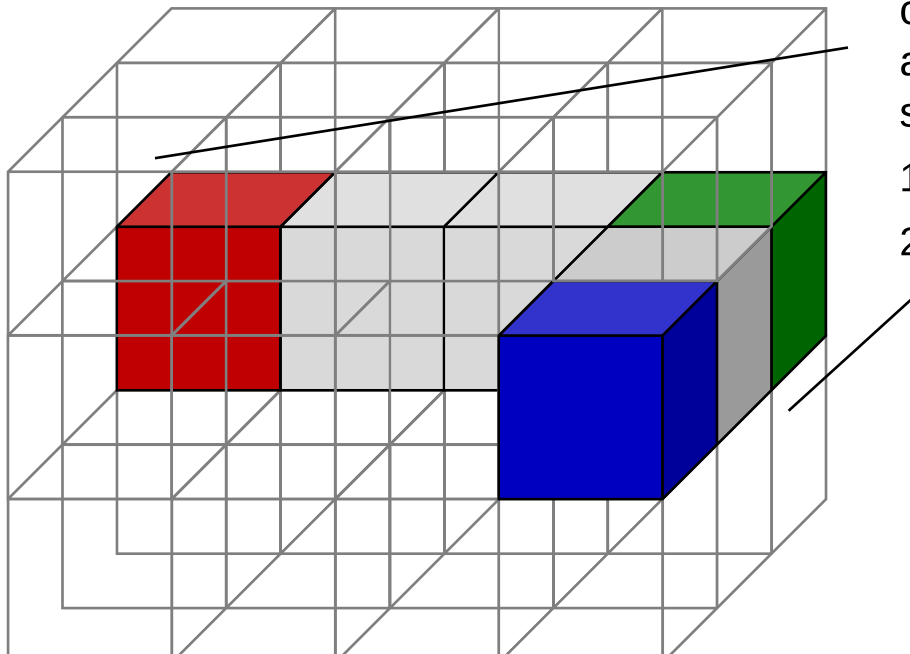
- TTE is not just an interconnect, it is an **integration framework**

TTE provides multiple formally verified services:

- Clock synchronization
- Clique detection
- Group membership
- Startup and Restart

… Which in combination provide services to the components (e.g. SW, subsystems):

- Partitioned Messaging
- Scheduled Execution

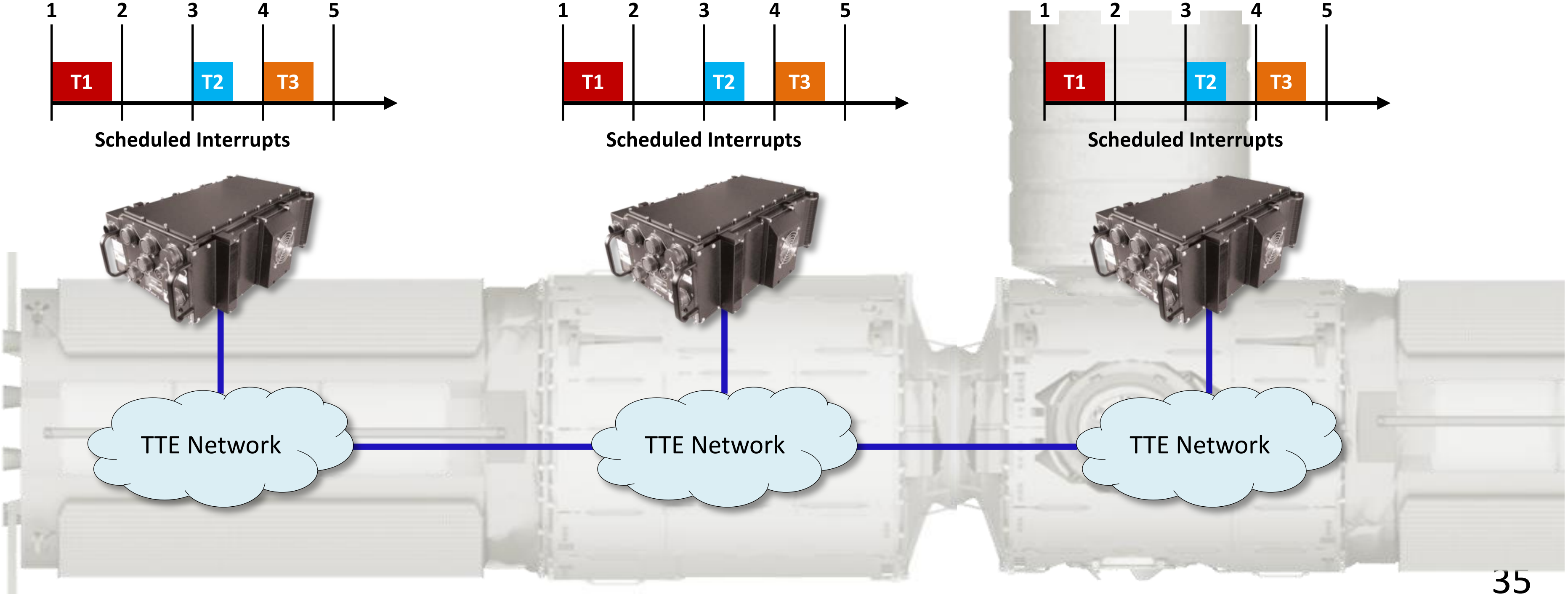


… That guarantee certain properties about the integrated system:

1. Composability
2. Compositionality

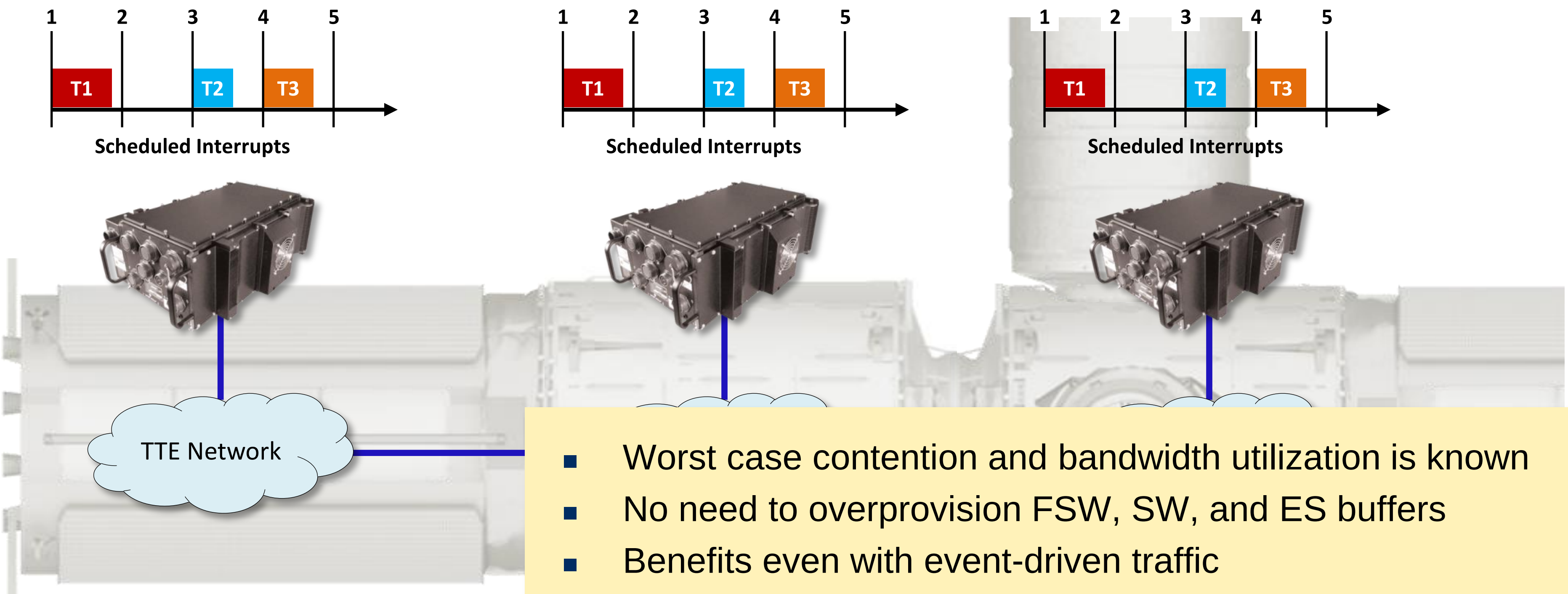
Scheduled Execution

- Schedule interrupts based on global synchronized clock
- Cross-platform alignment between task scheduling and TT network scheduling



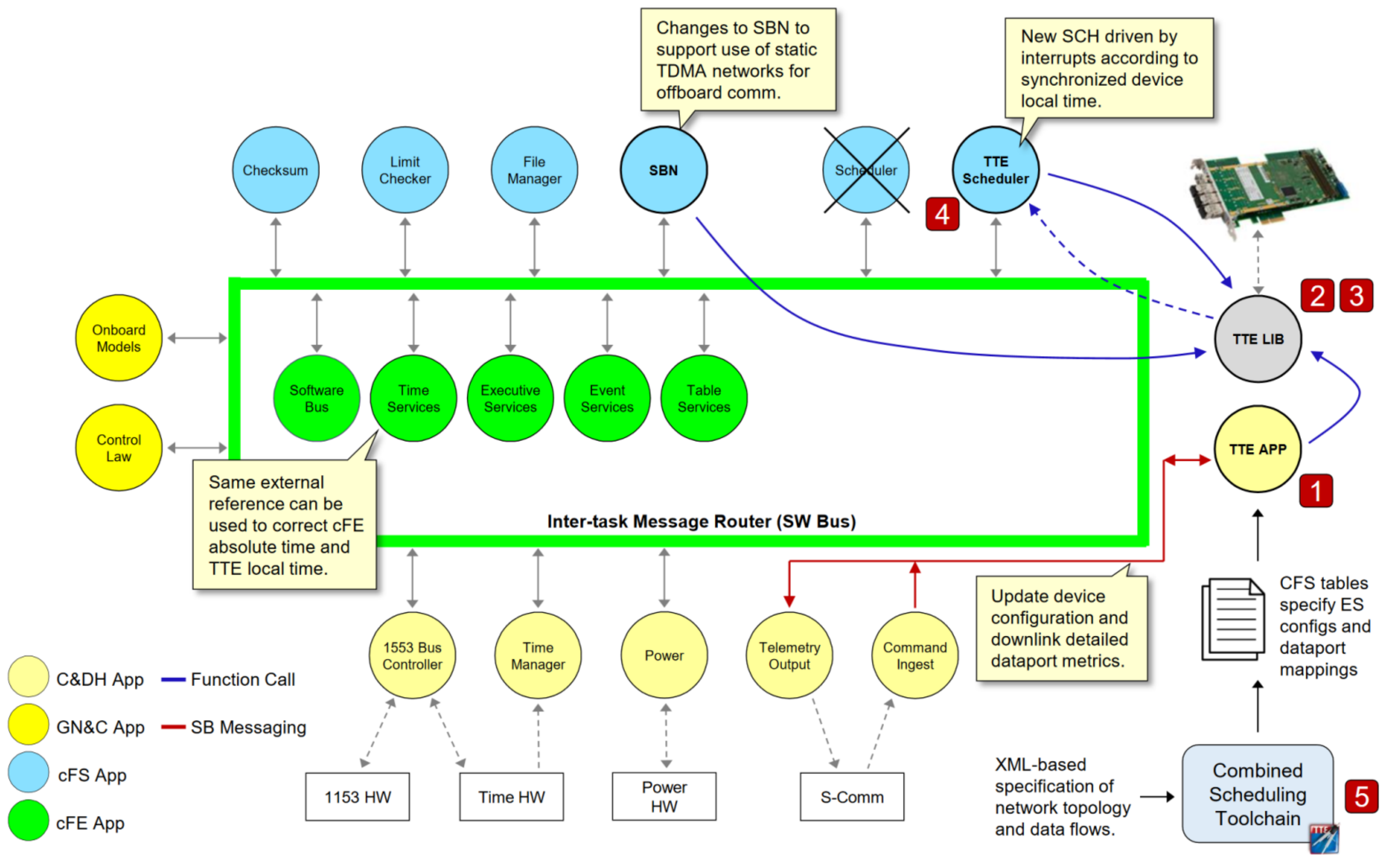
Scheduled Execution

- Schedule interrupts based on global synchronized clock
- Cross-platform alignment between task scheduling and TT network scheduling



- Worst case contention and bandwidth utilization is known
- No need to overprovision FSW, SW, and ES buffers
- Benefits even with event-driven traffic

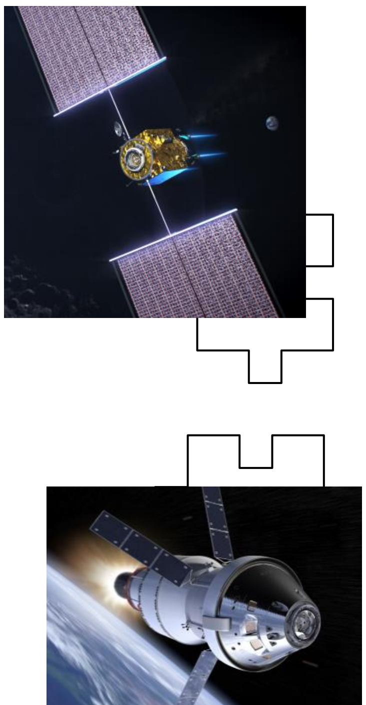
Scheduled Execution: cFS

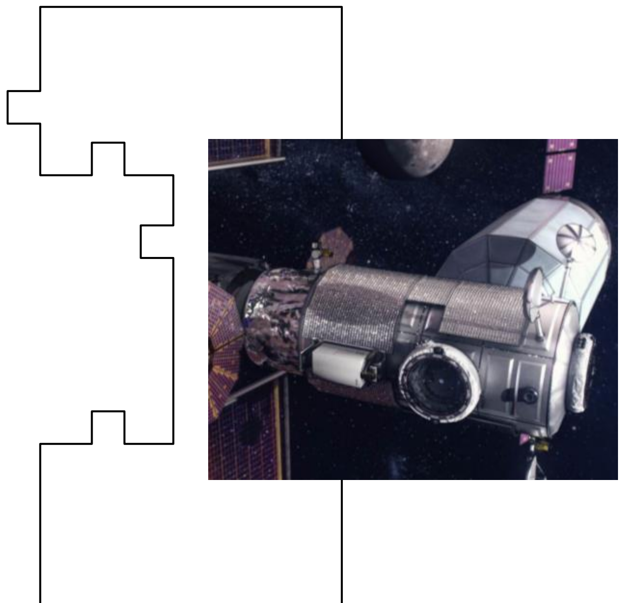


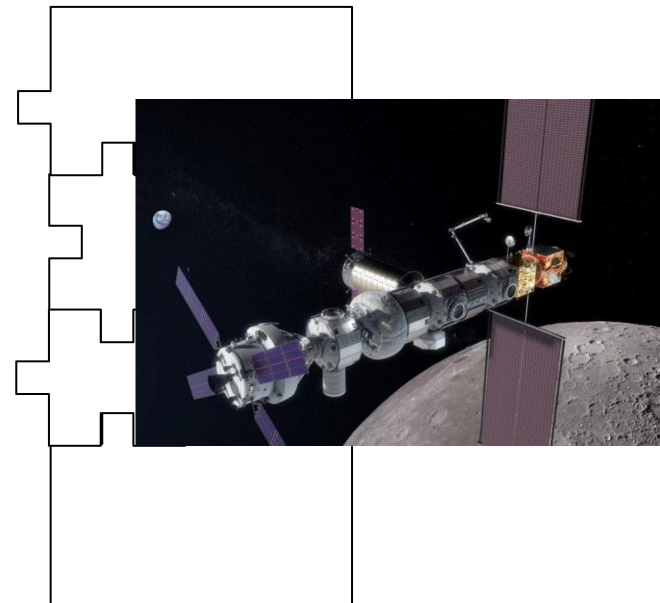
Composable Systems

Because we know how these behave in isolation …

We know how this behaves when fully integrated



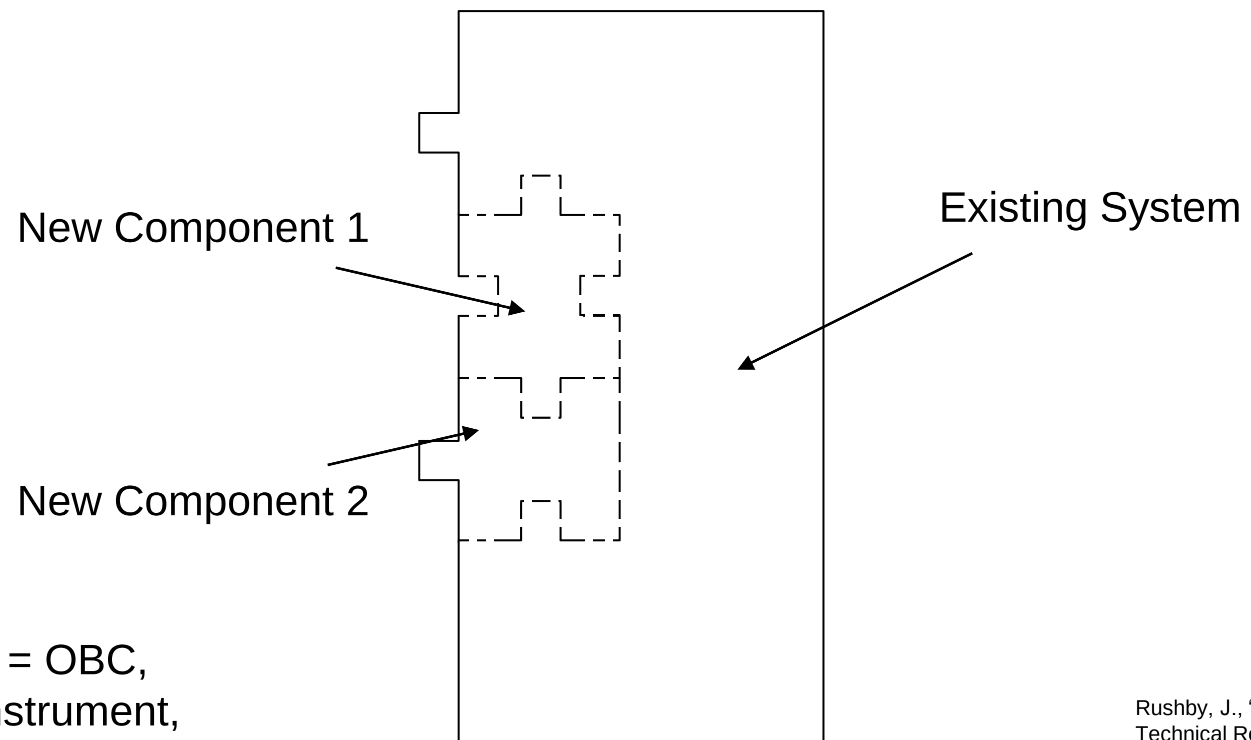




Module Integration

Composable Systems

- Certification traditionally requires us to view the system as an indivisible whole

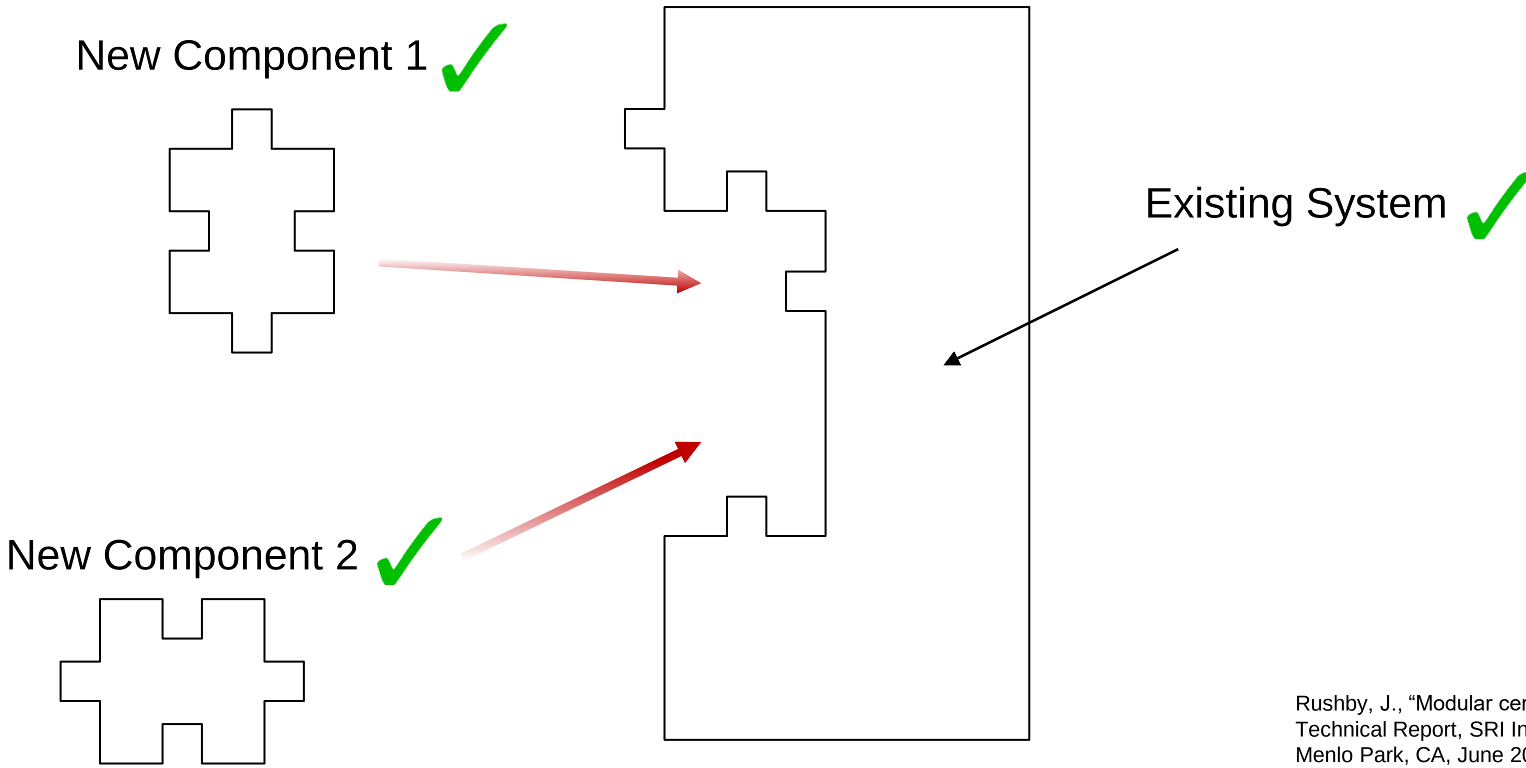


Component = OBC,
Software, Instrument,
Subsystem, Function

Rushby, J., “Modular certification,” Technical Report, SRI International, Menlo Park, CA, June 2002.

Composable Systems

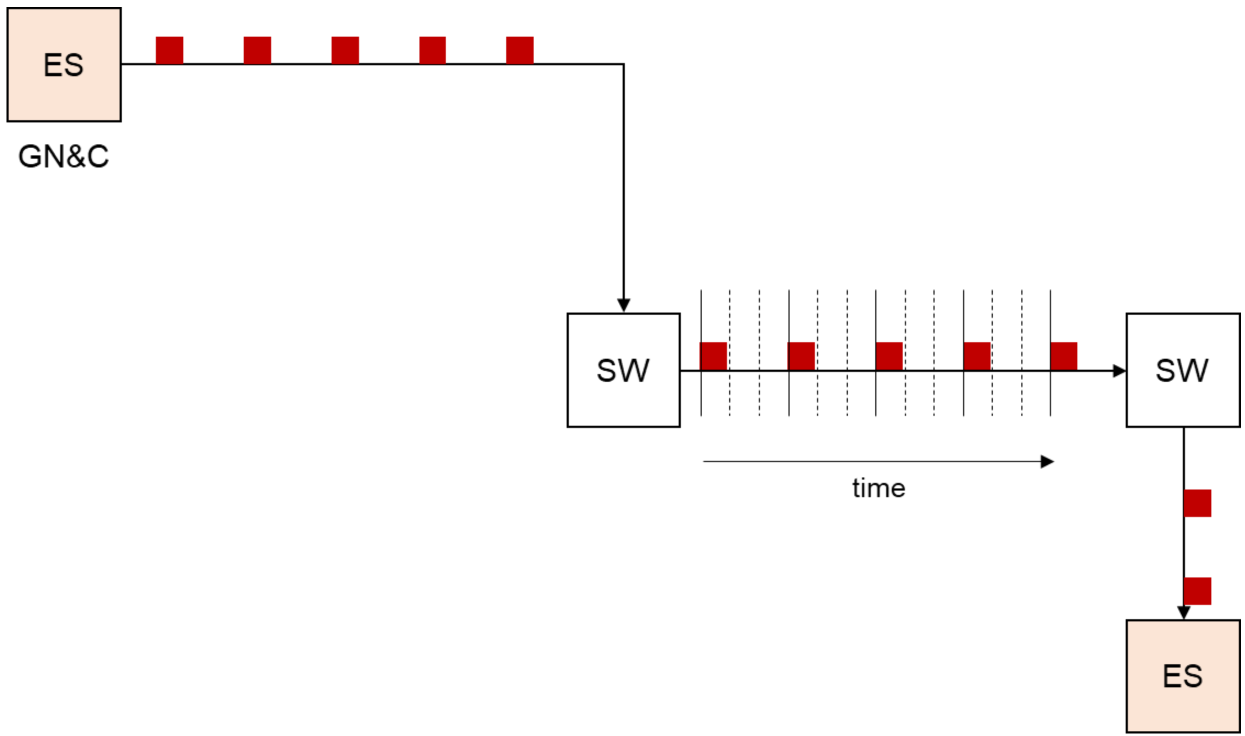
- Desirable to certify the whole by integrating smaller certified components



Rushby, J., "Modular certification," Technical Report, SRI International, Menlo Park, CA, June 2002.

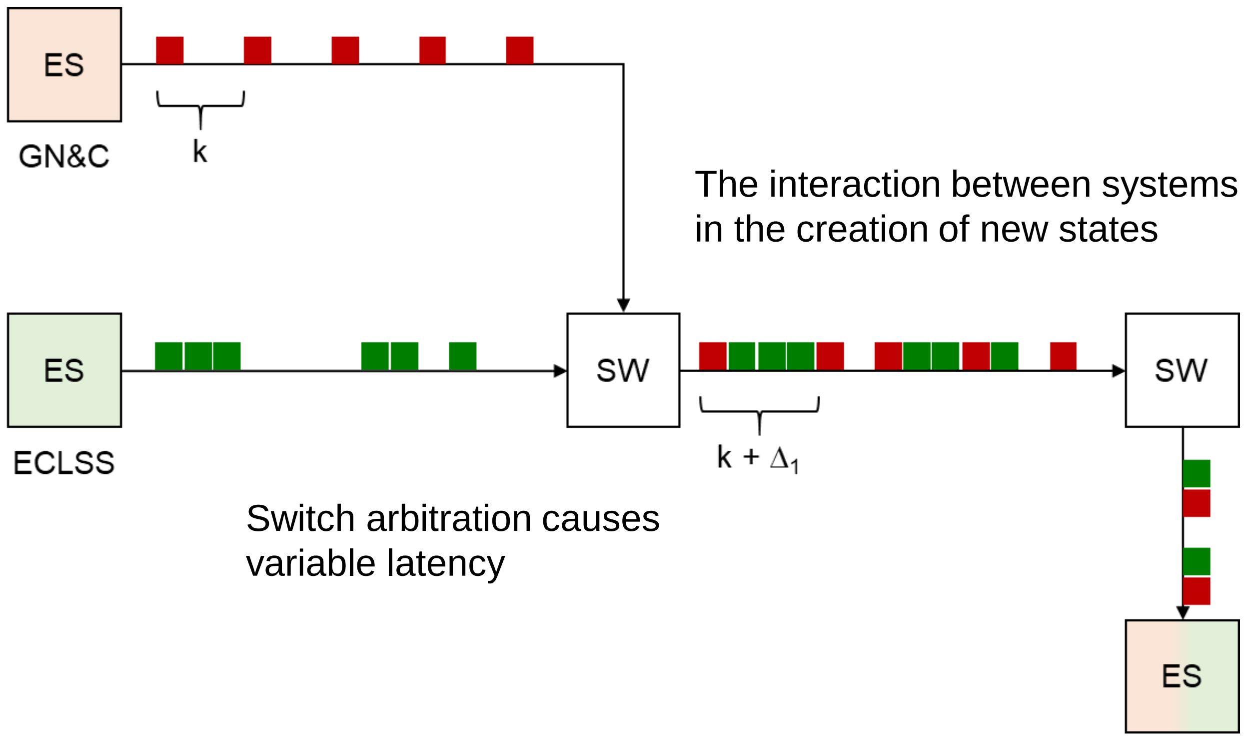
Why is it Hard?

- Network traffic between modules can interact in unanticipated ways
- Makes it impossible to consider modules in isolation



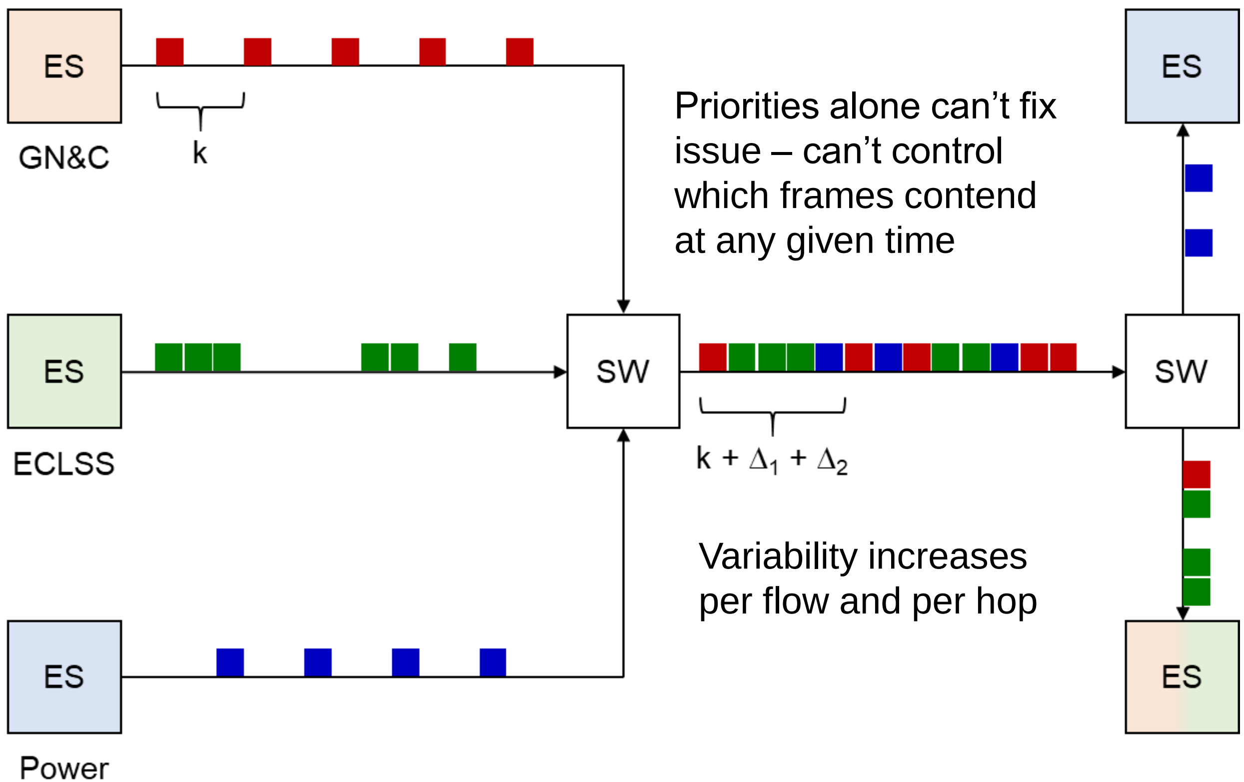
Why is it Hard?

- Network traffic between modules can interact in unanticipated way
- Makes it impossible to consider modules in isolation



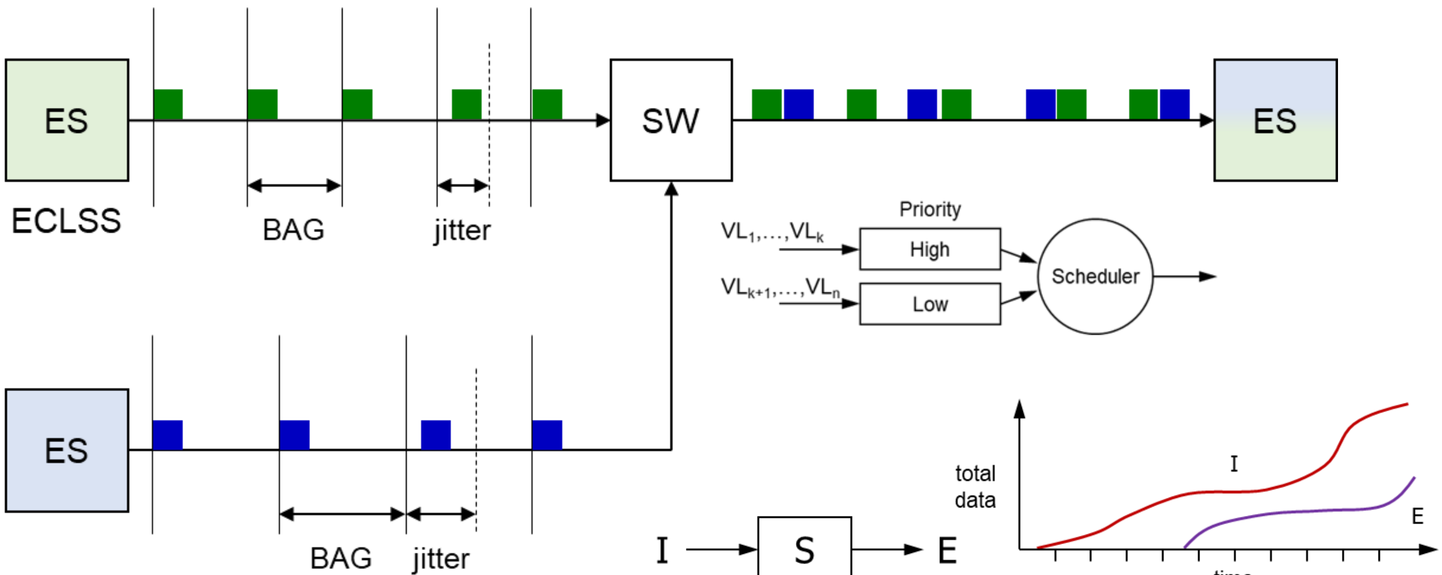
Why is it Hard?

- Network traffic between modules can interact in unanticipated ways
- Makes it impossible to consider modules in isolation



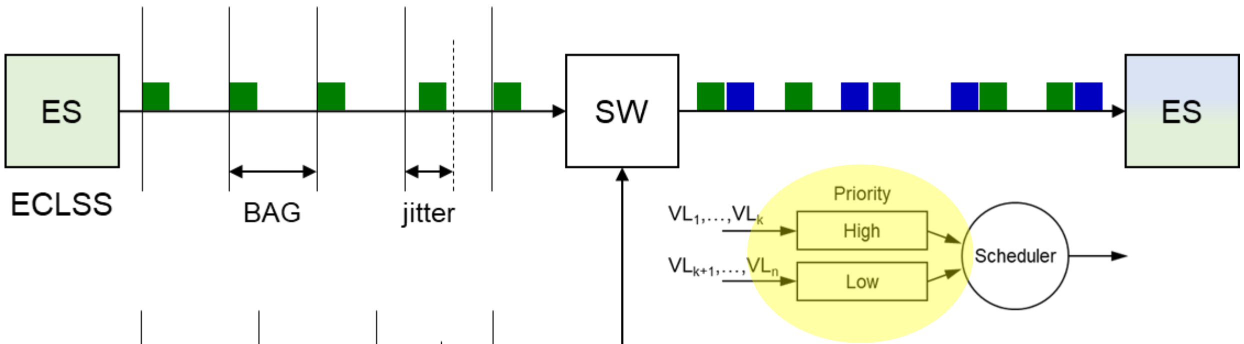
One Approach: A664 P7

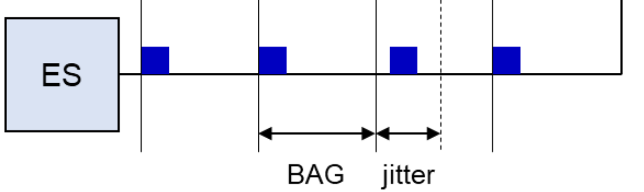
- **Bounding Variability:** ARINC 664 P7 can bound by mathematical proof:
  - 1) per VL end-to-end delay, 2) per port/VL jitter, 3) buffer sizes (from port waiting times)
- Use network calculus to calculate bounds on latency and queue sizes
  - Then use those metrics to determine VL priorities, BAGs, and frame sizes

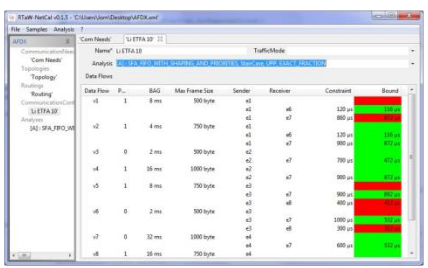


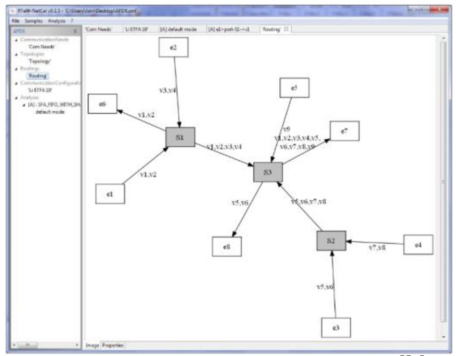
One Approach: A664 P7

- But changing the priority or BAG of a given VL could have little impact, or huge impact
  - Depends not only on that VL, but <u>what it contends with</u>
- Therefore still requires consideration of the entire integrated system
- Increases complexity of maintaining a network that needs to grow/evolve over time





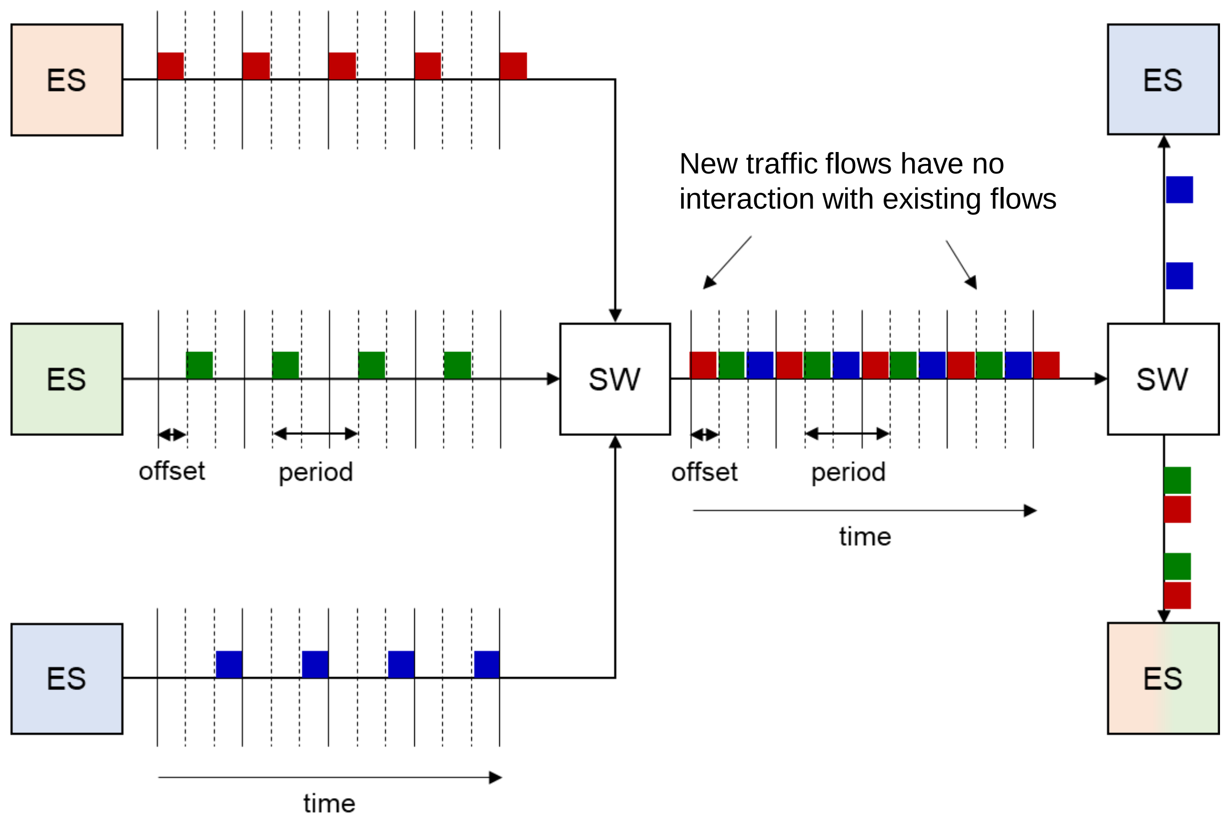




Boyer, M., Migge, J., and Fumey, M., “PEGASE – A Robust and Efficient Tool for Worst-Case Network Traversal Time Evaluation on AFDX,” SAE Technical Paper 2011-01-2711, 2011.

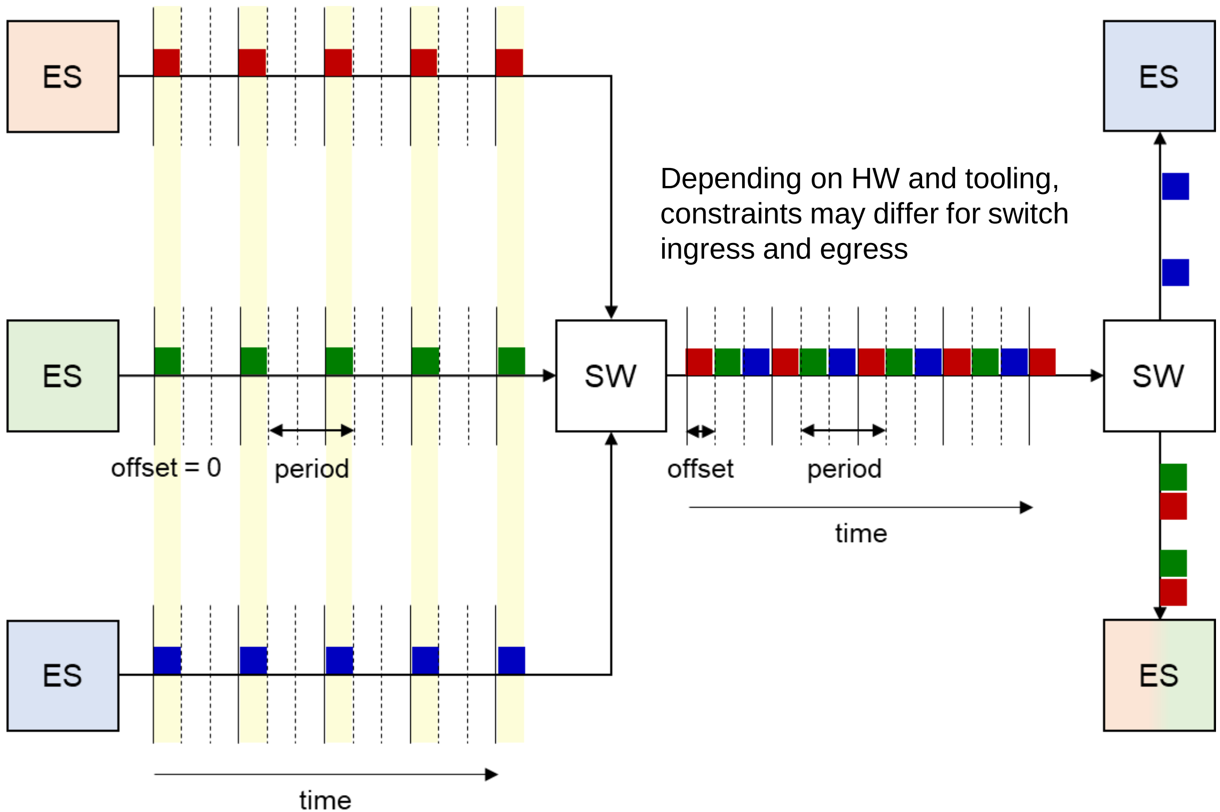
Alternative Approach: TTE

- Prevent the contention that causes the timing variability (e.g. for buffers, ports)



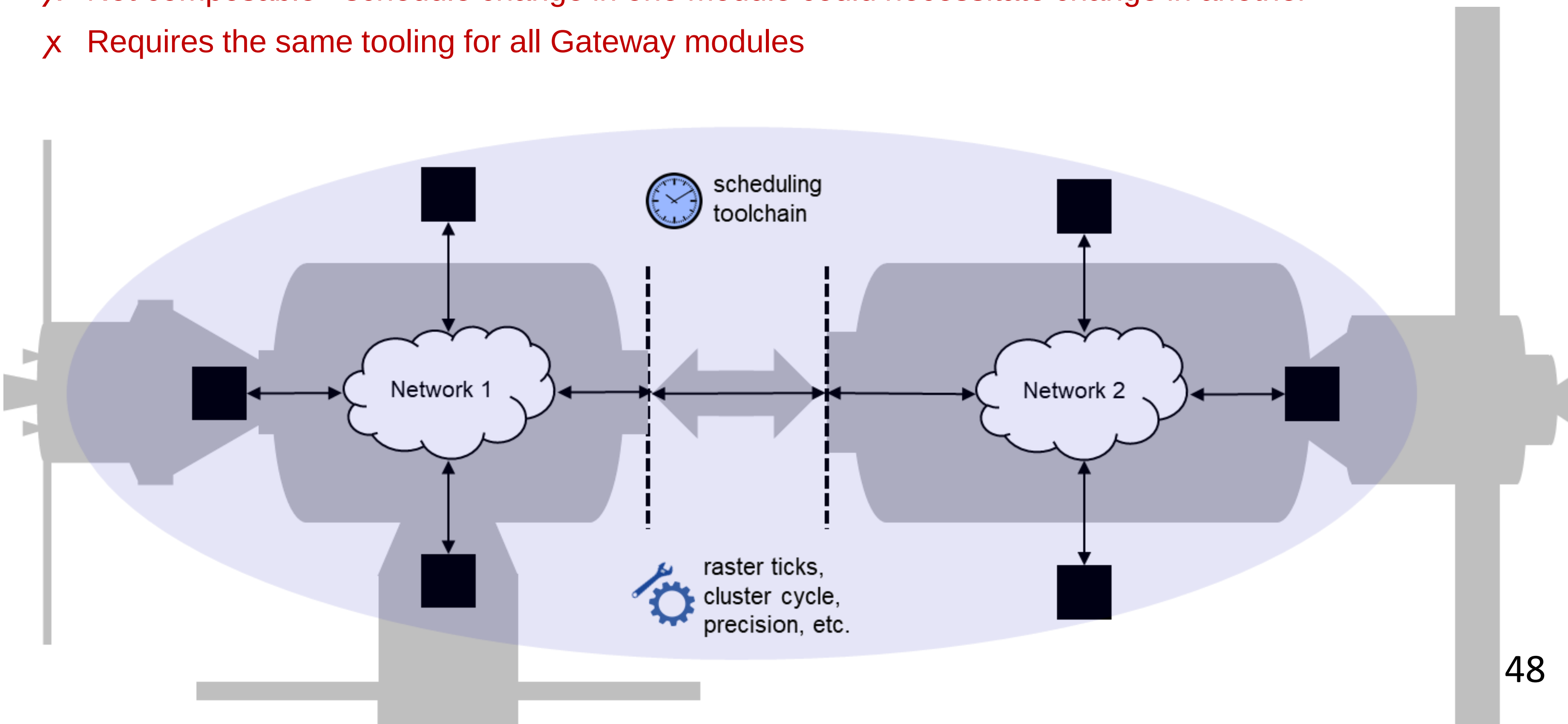
Alternative Approach: TTE

- Prevent the contention that causes the timing variability (e.g. for buffers, ports)



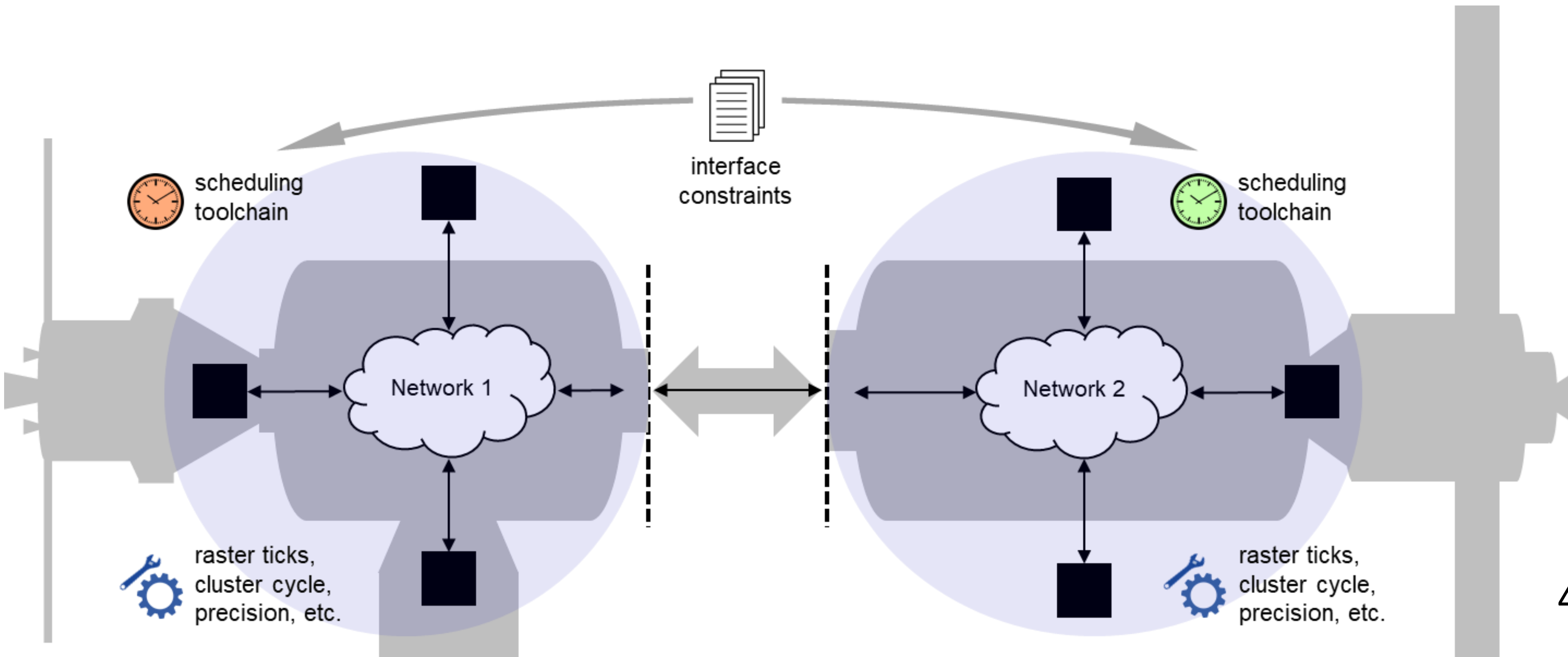
Incremental Buildup

- **One Option:** Schedule the whole integrated vehicle at once
  - ✗ Not composable - schedule change in one module could necessitate change in another
  - ✗ Requires the same tooling for all Gateway modules



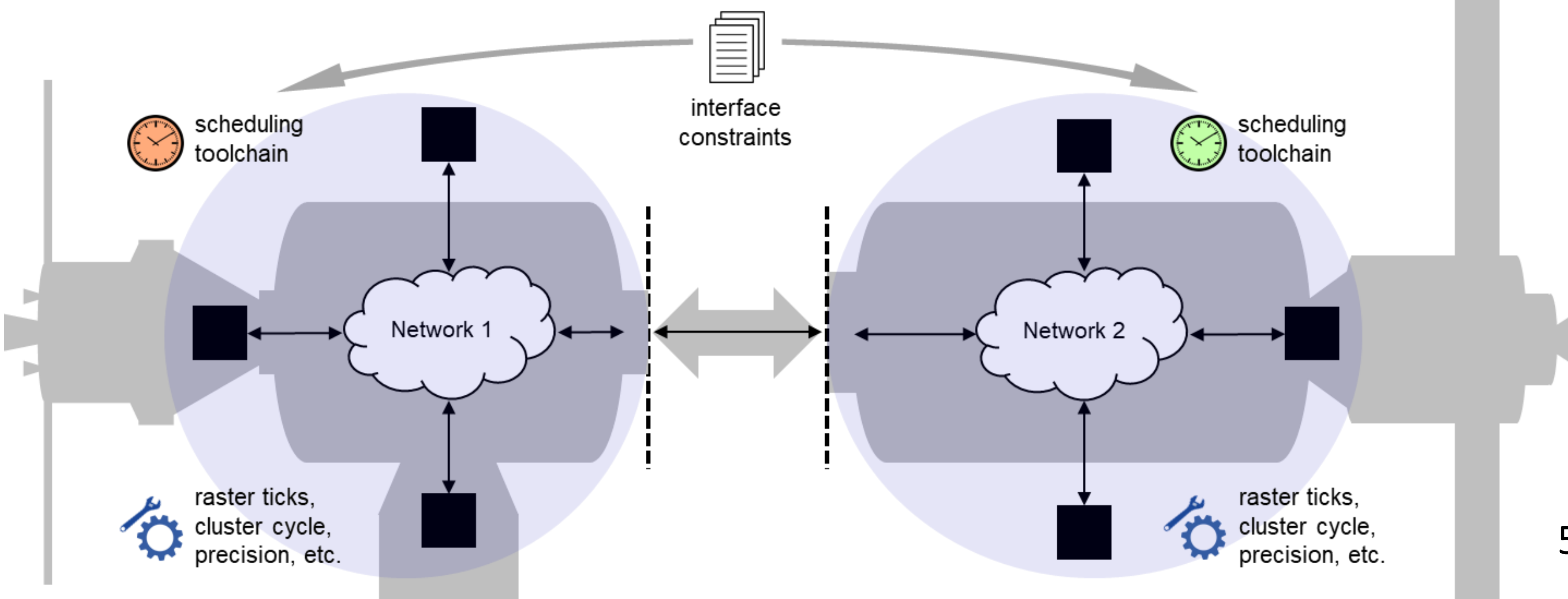
Incremental Buildup

- **Another Option:** Vehicle has one network, scheduled piecewise with different tooling
  - Agree on consistent timing parameters for each module (e.g. raster granularity, cluster cycle, precision)
  - Define message properties at every module interface (e.g. direction, timing, max size)
  - Use these properties as constraints for scheduling each module



Incremental Buildup

- **Another Option:** Vehicle has one network, scheduled piecewise with different tooling
  - ✓ Scheduling can be done separately per module (potentially with different tooling)
  - ✓ Scheduling within a module must only change if the interface constraints change
  - ✓ Allows different modules to be tested independently – interactions are completely controlled
  - ✗ Messaging constraints at interface must be identified up front

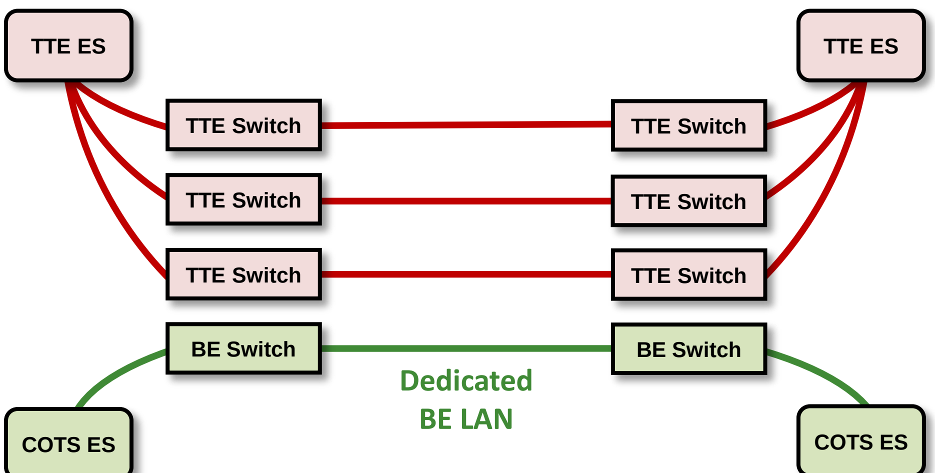


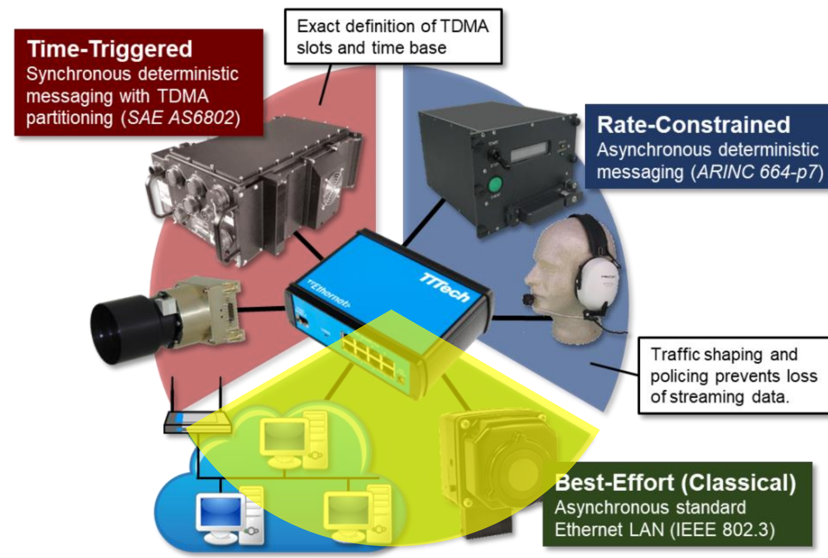
Agenda

- Introduction to Gateway
- Time-Triggered Ethernet (TTE) backbone
- TTE, A Fault-Tolerant Interconnect
- TTE, An Integration Framework
- A Unique Challenge, Classical Ethernet
- Conclusion

Best-Effort Ethernet

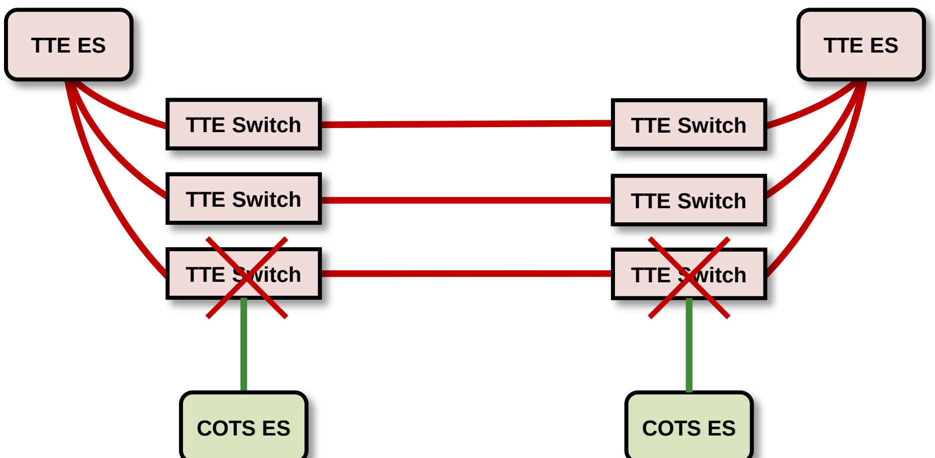
- Gateway has **no dedicated** best-effort plane
  - Standard Ethernet/COTS devices connect to the same TTE network
  - Supported by TTE standard – should be no problem

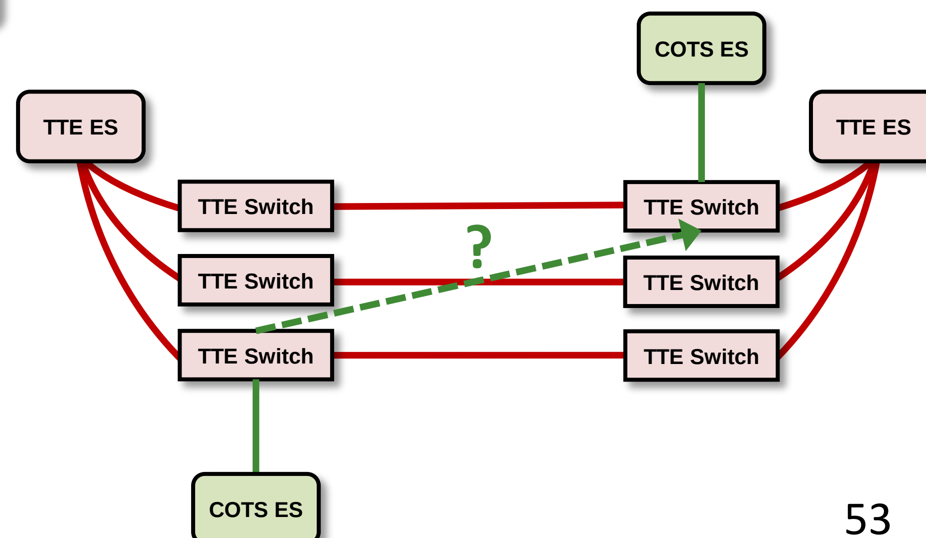




Best-Effort Ethernet

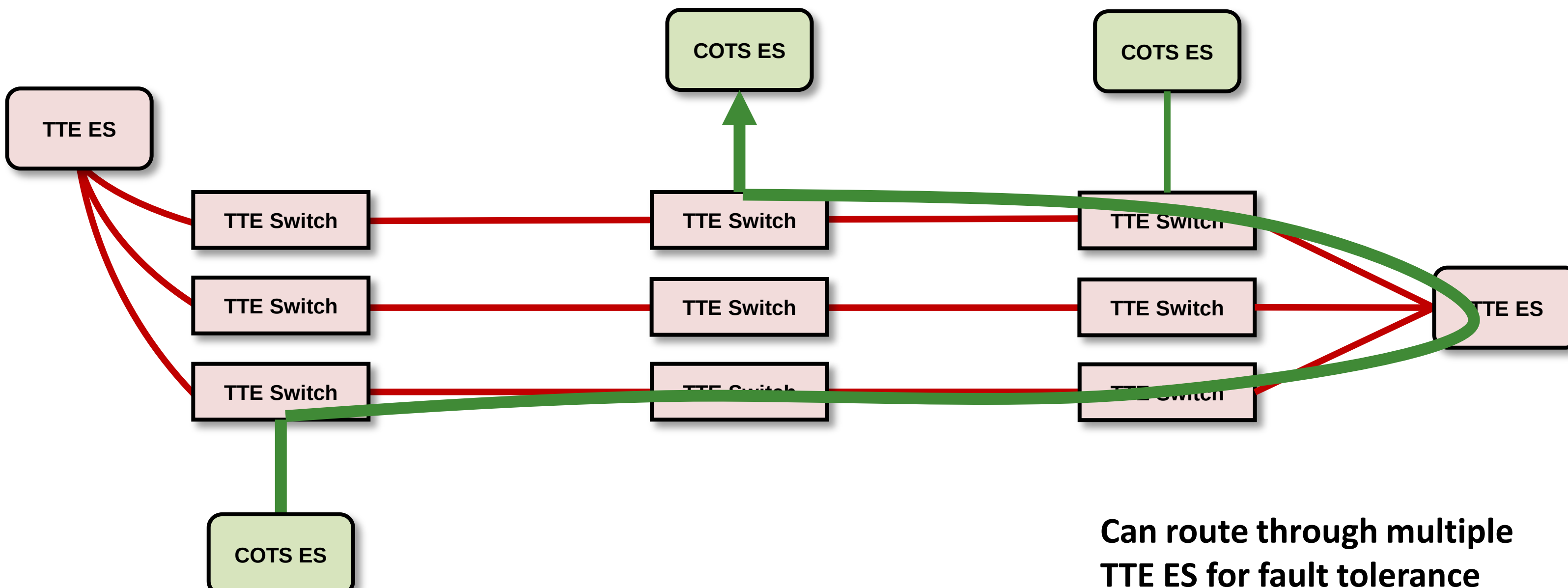
- **A Challenge:** Many COTS devices only connect to one plane
  - If all COTS devices are on same plane, you lose them after one fault
    - Some “non-critical” devices are critical. E.g. Cameras, crew laptops
  - But if COTS devices are on different planes, how do they talk?





A Few Options

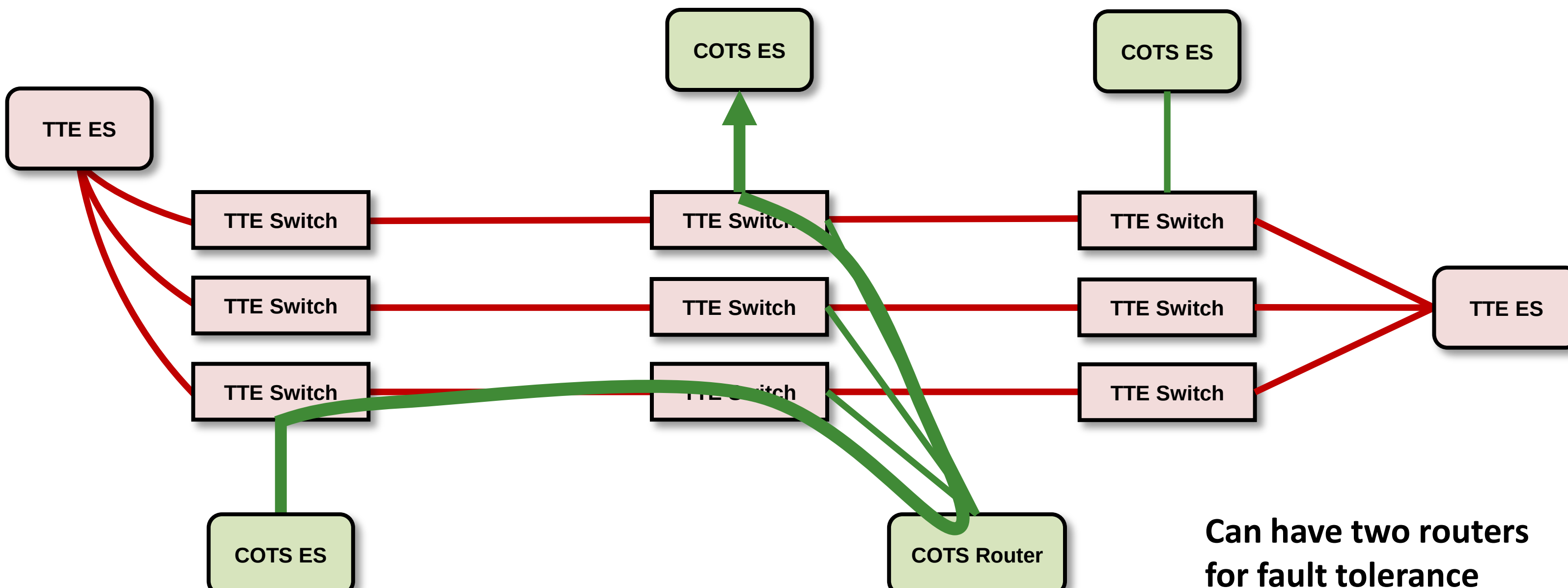
- Retransmit frames through a TTE end system
  - **Pros:** Adds no SWaP, needs no switch ports
  - **Cons:** Needs extra CPU time, SW changes, custom routing rules



Can route through multiple TTE ES for fault tolerance

A Few Options

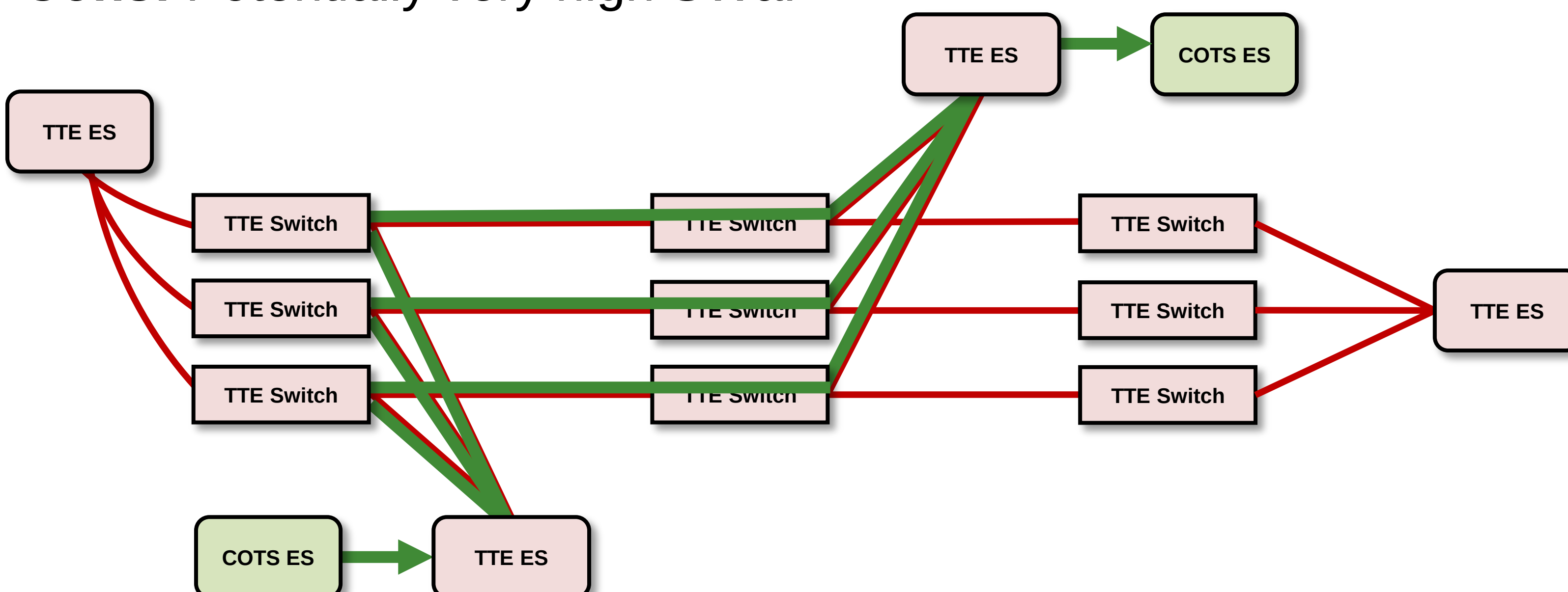
- Add a separate device to route frames
  - **Pros:** No SW changes, can use COTS routers
  - **Cons:** Adds SWaP, needs more switch ports, higher layers (L3+)



Can have two routers for fault tolerance

A Few Options

- Connected devices through a remote interface unit (RIU)
  - **Pros:** Can convert BE to TT/RC, increases availability of BE, reduces switch ports on main backbone
  - **Cons:** Potentially very high SWaP



Best-Effort Ethernet

- **Takeway:** No single best way to integrate COTS devices with a multi-plane TTE network
- Fact that TTE supports best-effort Ethernet is only **part** of the battle

Agenda

- Introduction to Gateway
- Time-Triggered Ethernet (TTE) backbone
- TTE, A Fault-Tolerant Interconnect
- TTE, An Integration Framework
- A Unique Challenge, Classical Ethernet
- Conclusion

Conclusion

- Gateway is based on a redundant TTE backbone
- TTE allows mixed-criticality traffic to exist on same network
- TTE can be used to realize a reliable broadcast abstraction
- TTE should be thought of as an integration framework
- Using COTS devices on TTE is not fool proof

Sources

- **Content**
  - https://nasasitebuilder.nasawestprime.com/wp-content/uploads/sites/45/2019/09/avionics_baseline_final_3-2019.pdf
  - https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170001652.pdf
  - https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170004599.pdf
  - https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170008862.pdf
  - https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170010131.pdf
  - https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20160012363.pdf

- **Images**
  - https://www.nasa.gov/sites/default/files/atoms/files/cislunar-update-gerstenmaier-crusan-v5a.pdf
  - https://www.nasa.gov/sites/default/files/atoms/files/loverro_nac-open-session_2020_final2.pdf
  - https://www.nasa.gov/sites/default/files/atoms/files/gateway_nac_charts_v6.pdf
  - https://www.nasa.gov/sites/default/files/atoms/files/nac-charts-hls-overview-may-2020-heoc.pdf
  - https://www.nasa.gov/press-release/nasa-administrator-to-make-artemis-moon-program-announcement-media-teleconference-set/
  - https://www.nasa.gov/sites/default/files/thumbnails/image/gateway_ppehalo_angles_003.png
  - https://www.nasa.gov/sites/default/files/thumbnails/image/phase01-gateway-2024_00003.jpg
  - https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20160012363.pdf
  - https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170008862.pdf
  - https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170001652.pdf
  - https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170004599.pdf
  - https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170008862.pdf
  - https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170010131.pdf