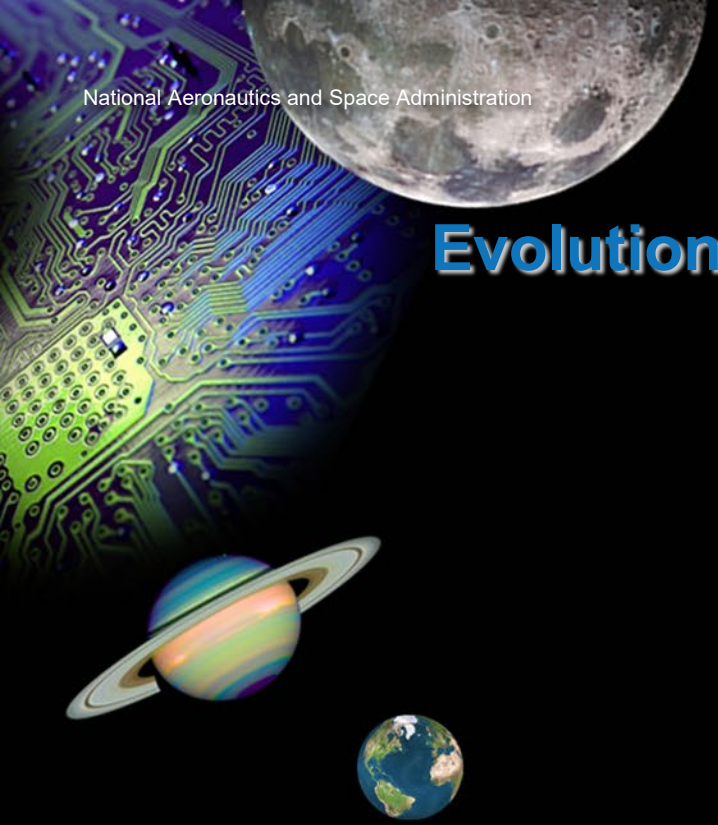
National Aeronautics and Space Administration



Evolution of NASA's core Flight System (cFS) for the Mission of Tomorrow

Goddard Space Flight Center
Flight Software Systems Branch
Code 582

Rich Landau
Product Development Lead, cFS
Richard.D.Landau@nasa.gov

Dr. Ashok Prajapati
Program Manager, cFS
Ashok.K.Prajapati@nasa.gov

Outline

- Goddard's Flight Software Systems Branch
- What is cFS?
- cFS Usage and Impact
- Long-term vision and objectives
- Roadmap for cFS development:
  - Next-Gen Hardware
  - Scripting
  - AI Inferencing
  - Mission Hardening
  - Expanded Suite of Capability

Goddard's Flight Software Systems Branch

As the creators, maintainers, and stewards of NASA's core Flight System, we:

- Orchestrate changes across projects and commonize new capability for the benefit of all
- Provide software support to industry partners to help push the technical envelope and achieve mission success
- Lead all stakeholders in the evolution of cFS to provide more capability out of the box

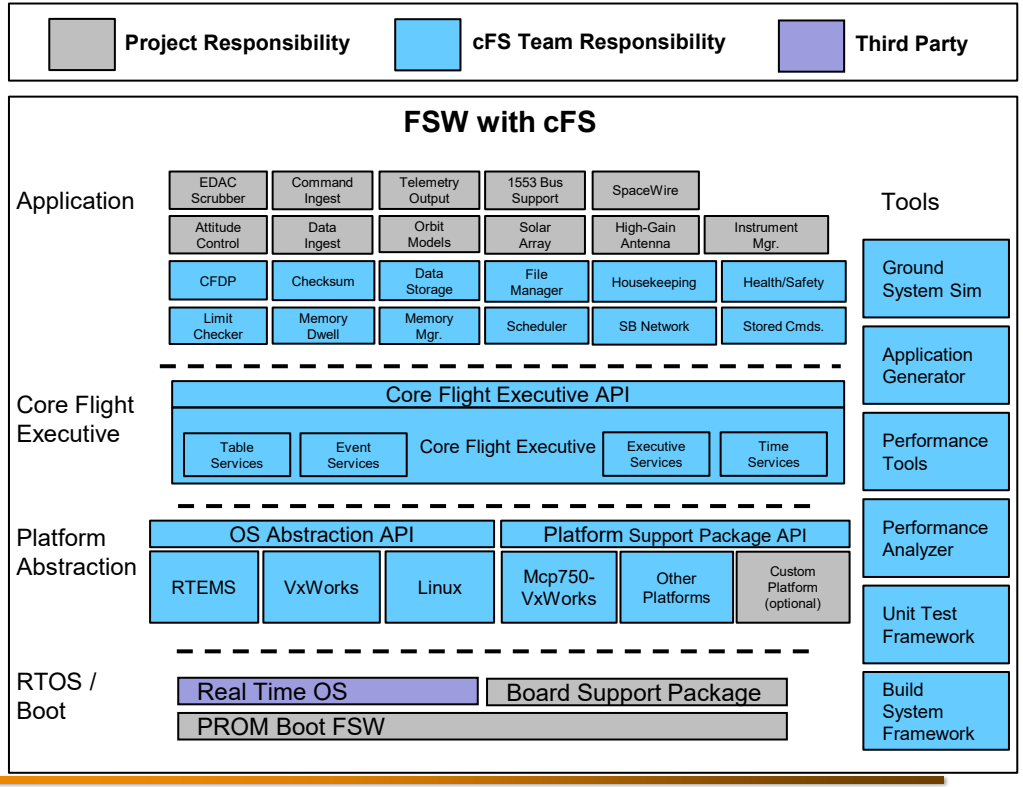
What is cFS?

NASA 2020 Software of the Year Award winner!

Flight Software (FSW) - the "brains" of the spacecraft:
- Controls instruments, thermal, electrical, attitude, etc.
- Ensures safety during anomalies
- Provides autonomy, guaranteed response time, determinism, reliability, maintainability

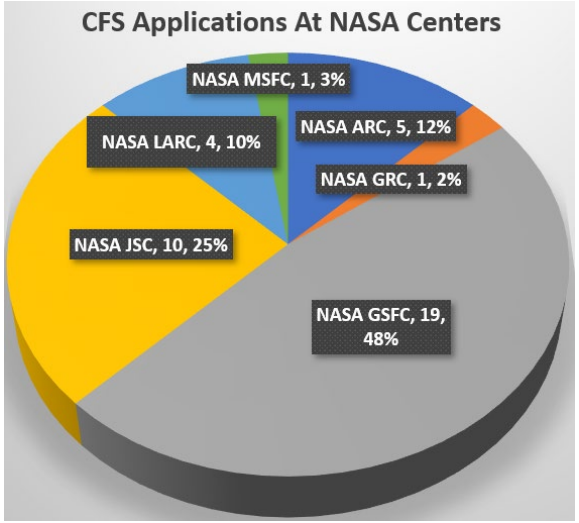
cFS - a TRL 9 FSW framework, providing:
- A dozen modular software applications for common functions (limit checker, file transfer, memory mgmt, etc.)
- Portability via processor and OS abstraction
- Supporting tools (unit test framework, performance, ground system simulator, etc.)

FSW with cFS
- Faster/cheaper development with 100% re-use of common services (time management, event notification, dynamic configurability, …) and applications (CFDP, Checksum, File Manager, …)
- Plug-n-play capability via the consistent interface for common and mission-specific applications

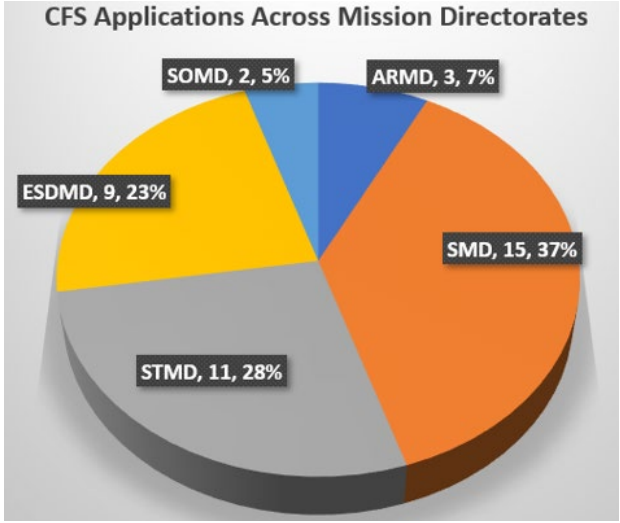


cFS commonizes core services, allowing project teams to focus on project-specific development

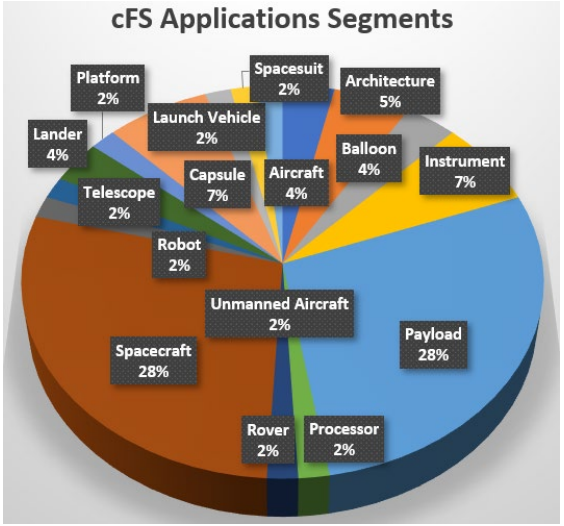
cFS Usage and Impact At NASA



cFS is used at 6 NASA Centers!



cFS is across mission directorates and on 40+ projects historically including RST, CCRS, MAV, LRO, Gateway, Artemis, and many more!



cFS is on embedded systems that span the space ecosystem.

cFS named by ISwSIS (and approved by HEO directorate) as the international standard framework for FSW, and is likely the most used FSW system in the world

cFS External Applications And Summary of Key Benefits

- **Quality**: cFS reuse reduces defects by as much as 95%* over starting from new software.
- **Cost**: cFS reuse reduces cost by as much as 70%* over starting from new software.
- **Risk and Schedule**: Starting from cFS substantially lowers project risk and makes schedules more predictable.
- **Standards**: cFS is now a standard for the space community defined by ISwSIS (and approved by HEO directorate) as standard FSW framework (for NASA, ESA, CSA, JAXA).
- **Tech Transfer**: cFS impact is documented in numerous publications.

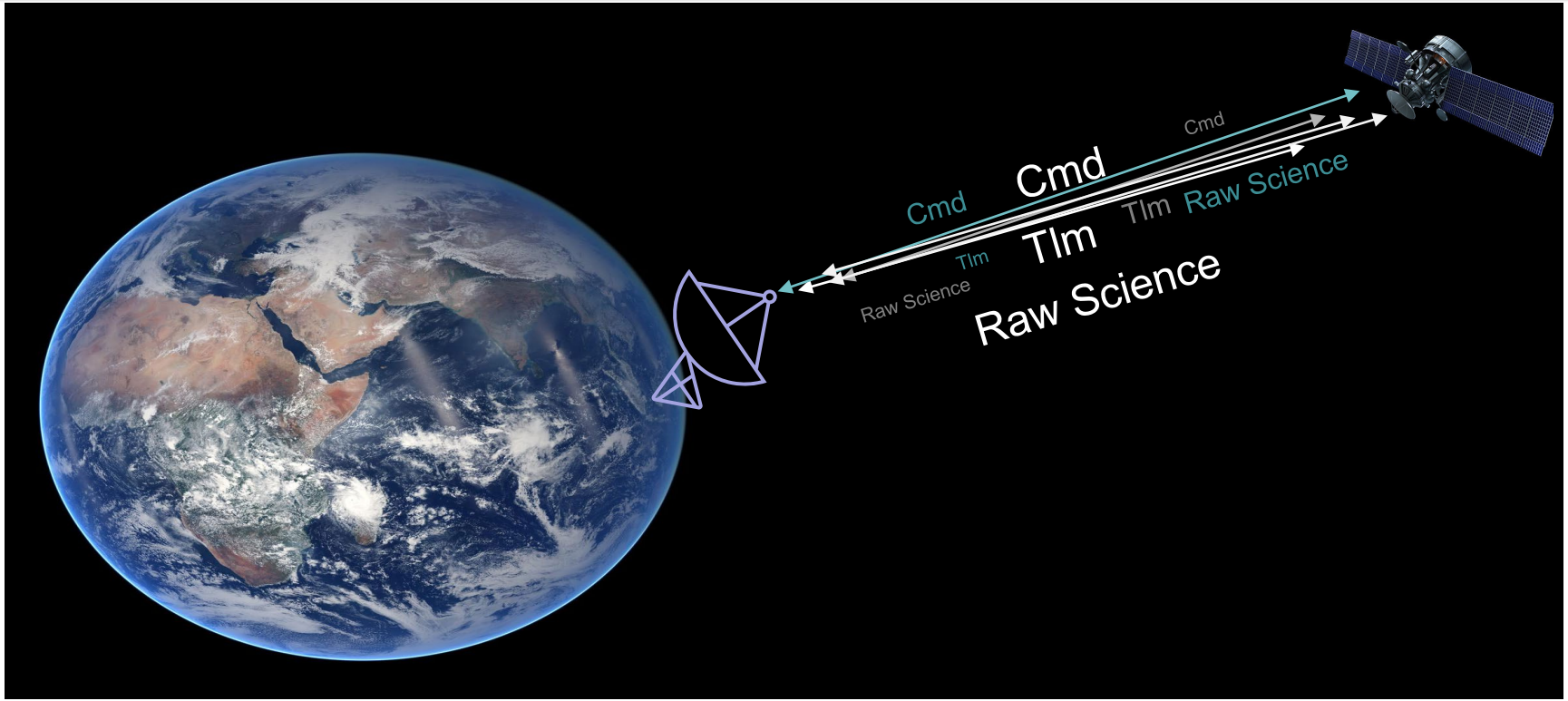
"We stand on the shoulders of giants. The work we were doing was built on the work people had done before us. NASA's Core Flight Software is a big part of what we do on the flight vehicle." - Intuitive Machines CTO

- DoD and other US agencies
- Academia* – CACTUS-1, BlackCAT, EagleCam
- Private/International Sectors*

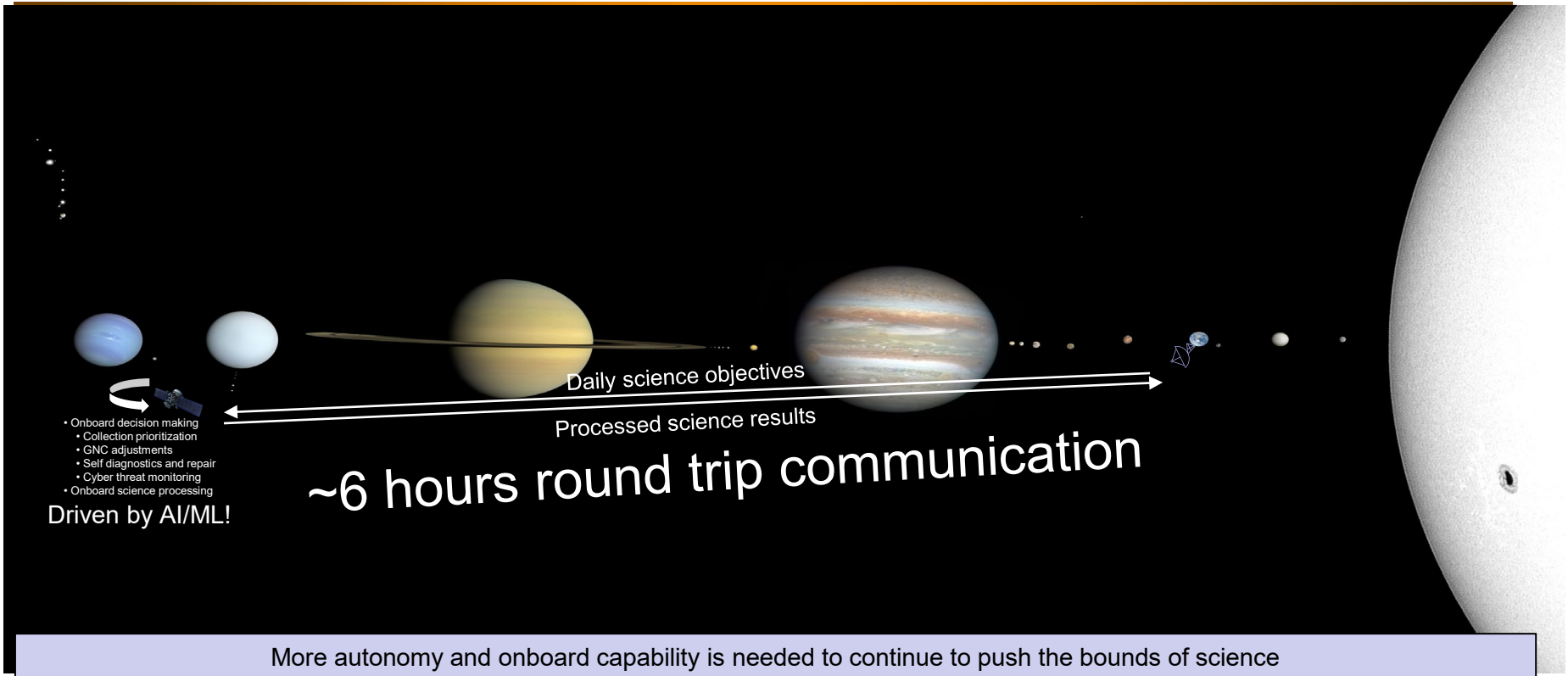
cFS is used across a broad range of systems and applications across the space industry.

*Note: There are many other users; for proprietary and security reasons, not all are listed.

The Current Paradigm



But as we explore…



More autonomy and onboard capability is needed to continue to push the bounds of science

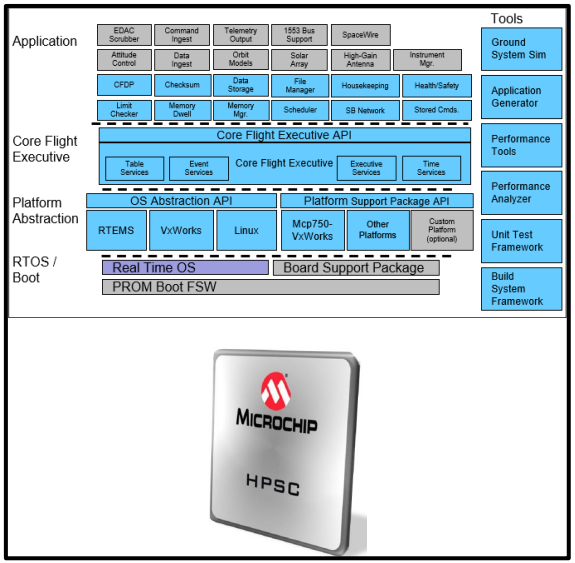
Image credit: CactiStaccingCrane under Creative Commons license
(Satellite, antenna, and text overlaid)

cFS Evolution Objectives

- Grow the suite of capability available for projects with both existing NASA-developed capabilities and proactively developed technologies
- Expand the cFS Project Team using current NASA/GSFC FSW experts to better support the FSW and NASA communities; includes certification, training, development, consulting, etc.
- Support cutting-edge operating systems and hardware
- Enhance deep space mission support
- Provide edge computing and reduce reliance on Earth-space links
- Add advanced capabilities for industry and academia
- Protect assets from adversaries

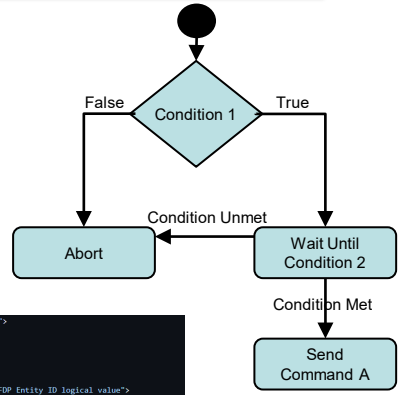
Support for Next-Gen Hardware

- NASA is developing the High Performance Spaceflight Computing (HPSC) hardware platform, providing:
  - 8-core CPU capable of heavily parallelized operations
  - Hardware vector acceleration enabling onboard AI / ML
  - Hypervisor enabling split architectures such as real-time (command & data handling, ACS, etc) and non-real-time science processing on a single chip
  - Hardware fault tolerance, cryptography, next-gen timers
- The cFS team is looking to integrate with HPSC and provide standardized mission access to HPSC features and capabilities



Onboard Scripting

- High level mission control is historically done by the ground
- For deep space missions and to increase efficiency, move some decisional mission logic to the flight system
- Scripts are updatable by engineers, flight ops, or crew
- Standardized capability that can interface with core apps and mission apps with an appropriate CCSDS EDS database
- Scripts written in Lua, with other languages as potential future expansions

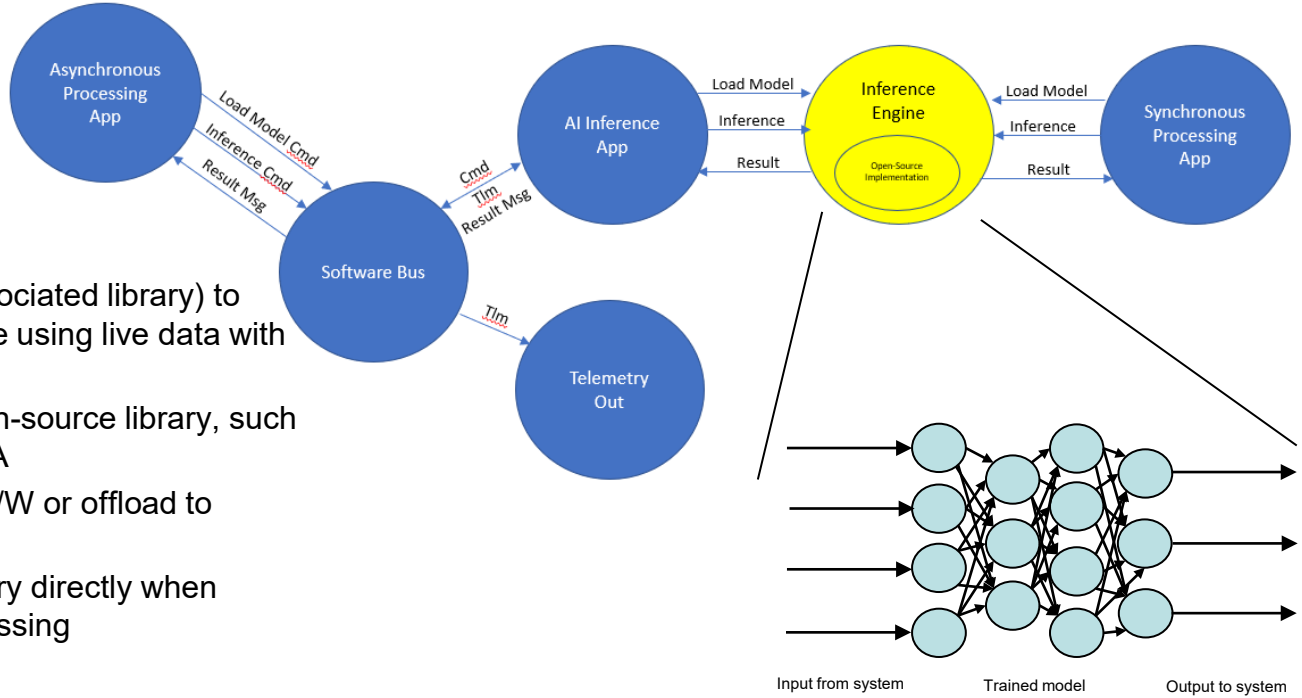



```
<PackageFile xmlns="http://www.ccsds.org/schema/sois/seds">
  <Package name="CF" shortDescription="CFDP">

    <DataTypeSet>

      <IntegerDataType name="EntityId" shortDescription="CFDP Entity ID logical value">
        <IntegerDataEncoding sizeInBits="32" encoding="unsigned" />
      </IntegerDataType>

      <IntegerDataType name="TransactionSeq" shortDescription="transaction sequence number size">
          <LongDescription>
@par Description
      The max size of the transaction sequence number as expected for all CFDP packets.
      CF supports the spec's variable size of TSN, where the actual size is
      selected at runtime, and therefore the size in CFDP PDUs may be smaller
      than the size specified here.  This type only establishes the maximum
      size (and therefore maximum value) that a TSN may be.

@note This type is used in several CF commands, and so changing the size
       of this type will affect the following structure:
        CF_Transaction_Payload_t, any command that selects a transaction based on TSN

@par Limits
         Must be one of uint8, uint16, uint32, uint64.
          </LongDescription>
        <IntegerDataEncoding sizeInBits="32" encoding="unsigned" />
      </IntegerDataType>

      <EnumeratedDataType name="EnableFlag" shortDescription="Enable/Disable logical value">
        <EnumerationList>
          <Enumeration label="NO"  value="0" shortDescription="Option Disabled" />
```

Example of an EDS for cFS's CFDP app

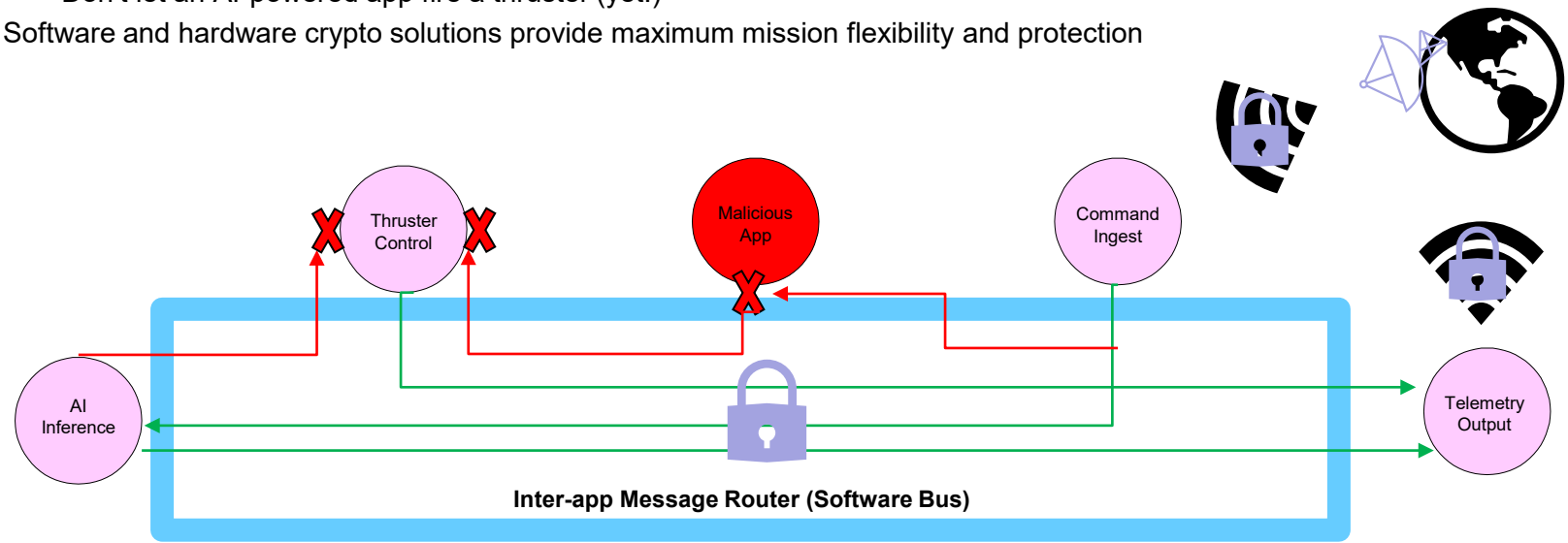
AI Inferencing

- Provide platform for running AI Inference using ground-trained models, enabling:
  - General spacecraft autonomy
  - Onboard data processing
  - Cyber threat detection
  - Self diagnostics and repair
  - Much more!
- Develop AI Inference app (and associated library) to load a model and perform inference using live data with a standardized cmd/tlm API
  - Standard API will wrap an open-source library, such as Tensorflow Lite or OpenXLA
  - Run on advanced multi-core H/W or offload to GPU/FPGA
  - Custom apps can use the library directly when needed for synchronous processing



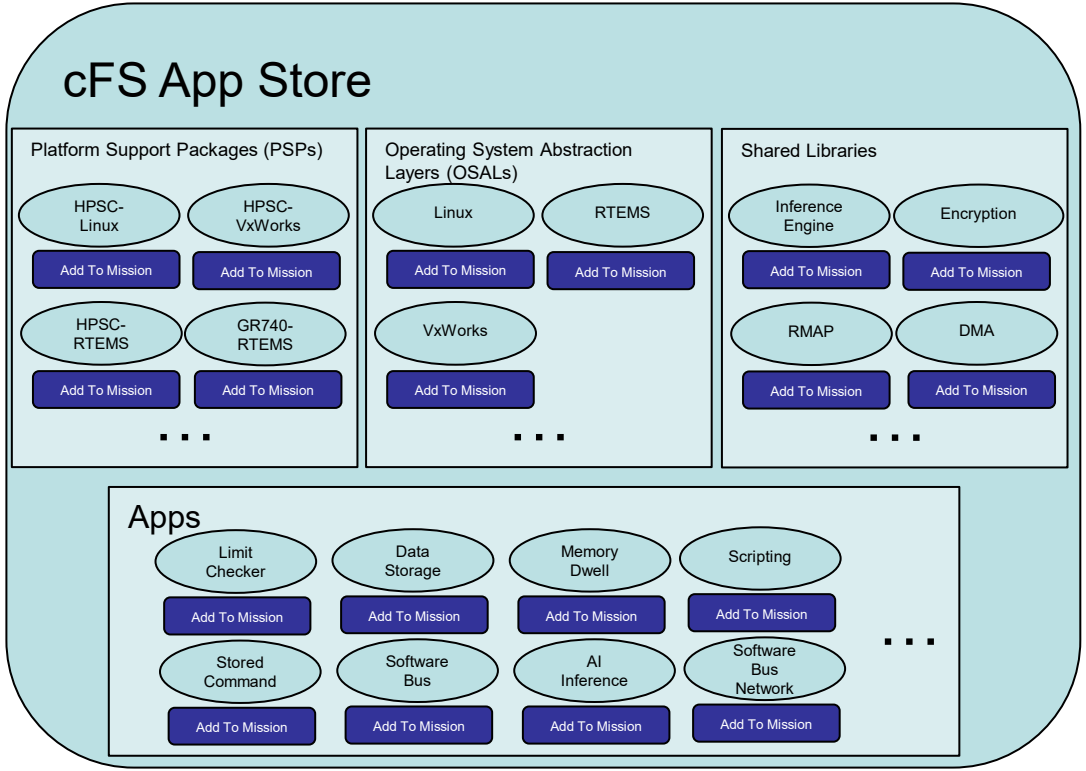
Flight System Hardening

- As cyber threats evolve, missions require more protection to minimize risk to their science
- The flight system can be further protected with sophisticated message authentication, encryption, and routing control
  - Protect against unauthorized access to the system *and* unauthorized actions once onboard
- Capability also allow safe “sandboxing” of low-TRL apps such as AI Inferencing to raise the technology’s TRL level
  - Don’t let an AI-powered app fire a thruster (yet!)
- Software and hardware crypto solutions provide maximum mission flexibility and protection



cFS App Store

- GSFC / Code 582 develops and maintains core cFS functionality with community involvement
- The cFS team is working to make it easier for mission teams to pick and choose which functionality they need for their missions based on requirements and hardware capabilities
- Concept furthers cFS's founding principles of standardizing complex capability and enable sharing between missions



Conclusion

- cFS was designed from the start with evolution and adaptability in mind to provide a software platform with cutting edge capability.
- The framework is actively being used widely on various NASA missions, by OGAs, commercial, and other international space agencies.
- Its development is led by GSFC, who is engaging in partnerships with the larger community to address a wide range of use cases.
- The cFS team is adding key features to accommodate advances in technology.
- Next generation hardware is coming, enabling computationally intensive science missions not previously possible. Our software is evolving to be ready for this new age in spaceflight.

The cFS team is looking for partners to help us drive technology forward to support the science missions to come

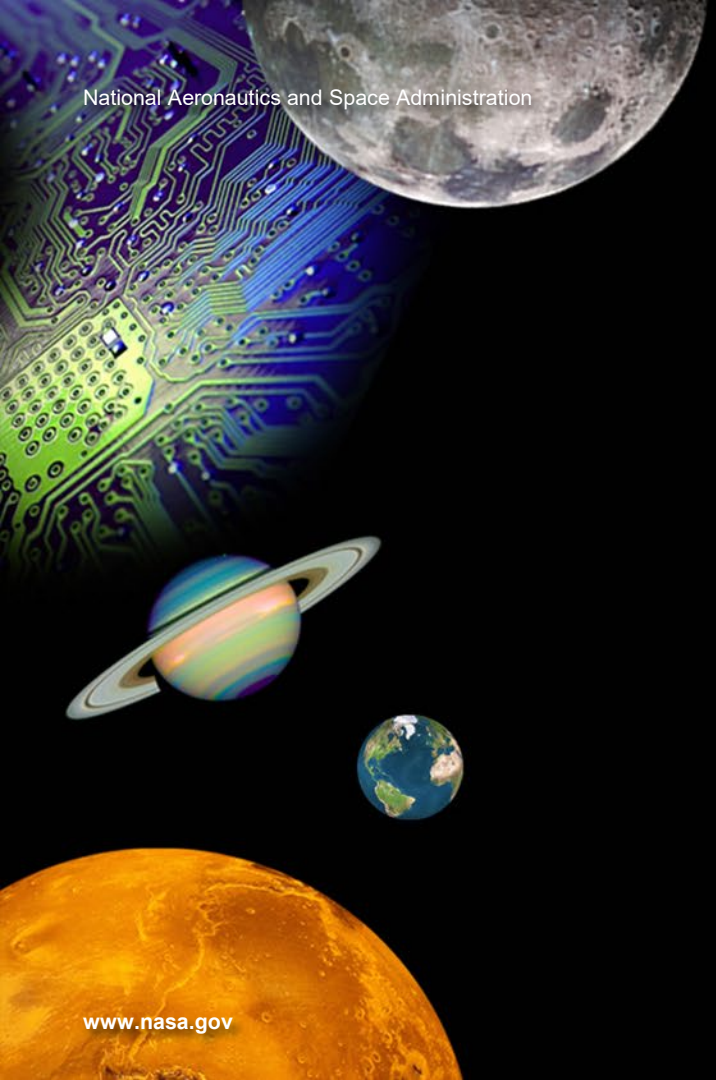
Engage

The cFS team welcomes and encourages:

- Comments
- Questions
- Ideas
- Discussion

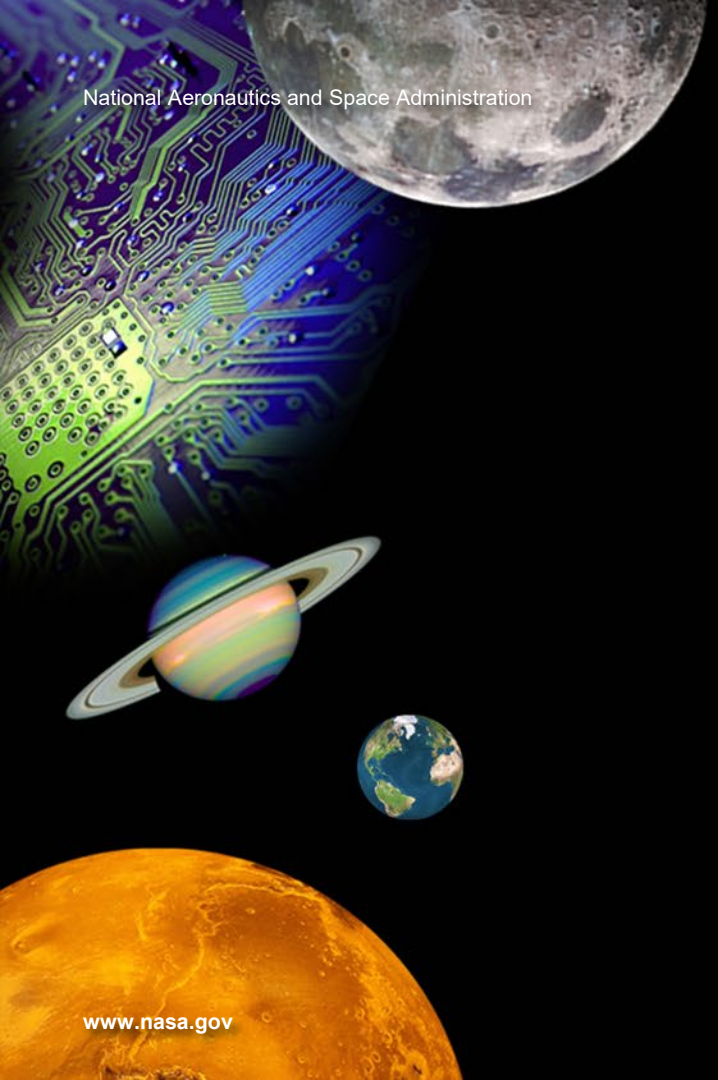
If your project is using cFS, please let us know!!

- cfs-program@lists.nasa.gov
- Ashok.K.Prajapati@nasa.gov, cFS PM
- Richard.D.Landau@nasa.gov, cFS PDL



National Aeronautics and Space Administration

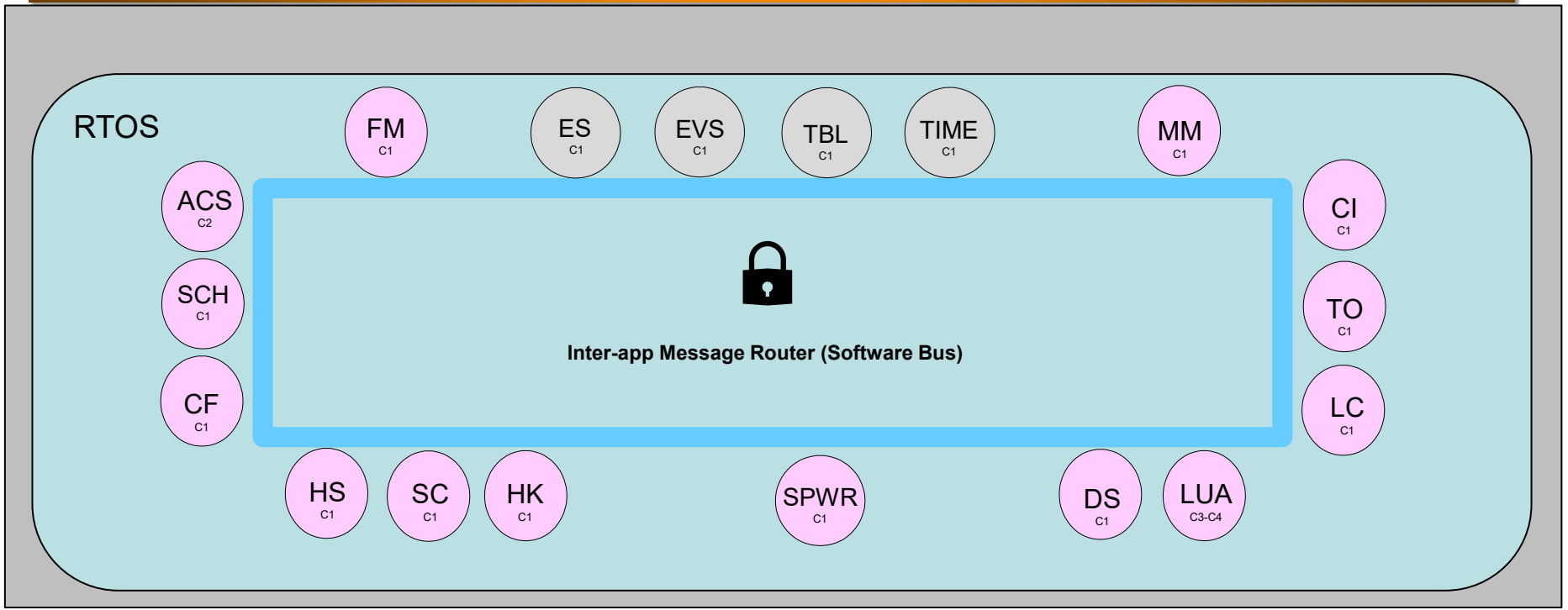
Questions?



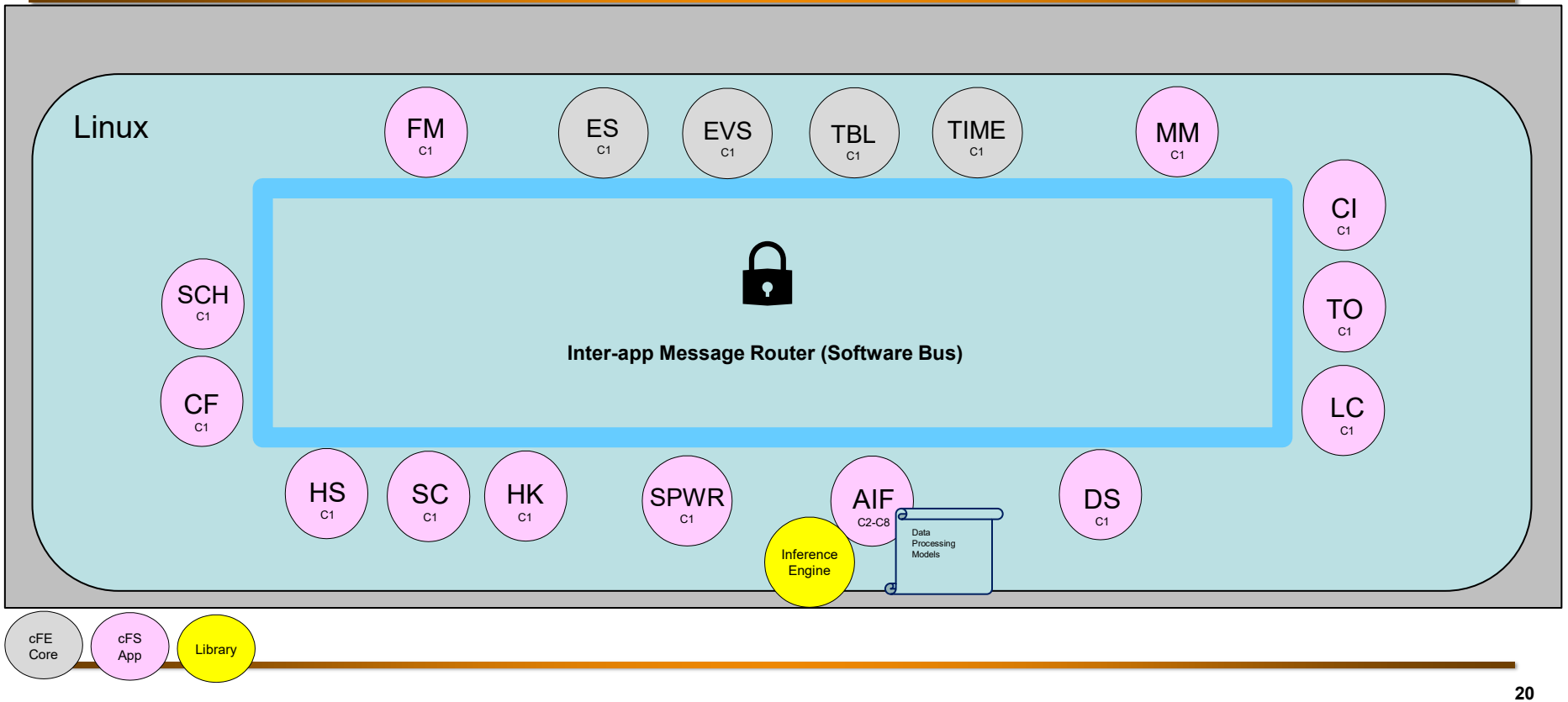
National Aeronautics and Space Administration

Backup

Example: Simple Spacecraft C&DH



Example: Instrument Data Processing



Example: Entire Complex Mission Solution