Presentation for Spacecraft Flight Software Workshop (FSW 2015),
Johns Hopkins APL, Laurel, MD, Oct. 27-29, 2015

Toward a Suite of Middleware Services for Enhanced Spacecraft Configuration and Capability

Dr. Alan George
Professor of ECE
University of Florida

Dr. Tuba Yavuz
Assistant Professor of ECE
University of Florida

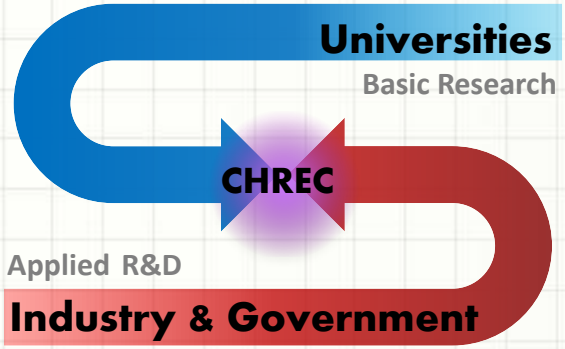
Nelson Psenjen
Antony Gillette
Jacob Joji
Chris Wilson
Research Students
University of Florida

Outline

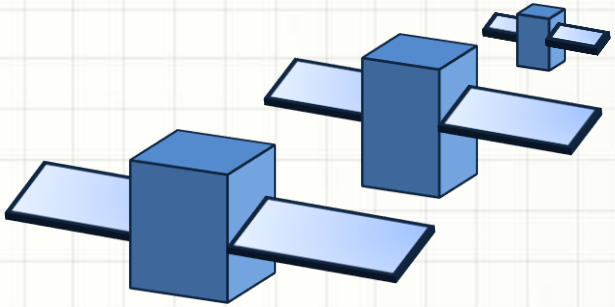
- Acknowledgements
- Key Technologies
- Motivations and Goals
- Architectural Overview
- Prototype
- Conclusions

Acknowledgements

- Space Middleware is a research project at CHREC
  - NSF Center for High-Performance Reconfigurable Computing (**CHREC**)
    - Top national research center founded in 2007
    - Comprises 3 university sites (UF, BYU, VT) and **over 30 industry and government** partners



- Space Middleware investigation is a collaborative CHREC effort
  - Key Partners
    - NASA KSC, GSFC, JSC, LRC, ARC
    - Honeywell
    - Lockheed-Martin SVIL

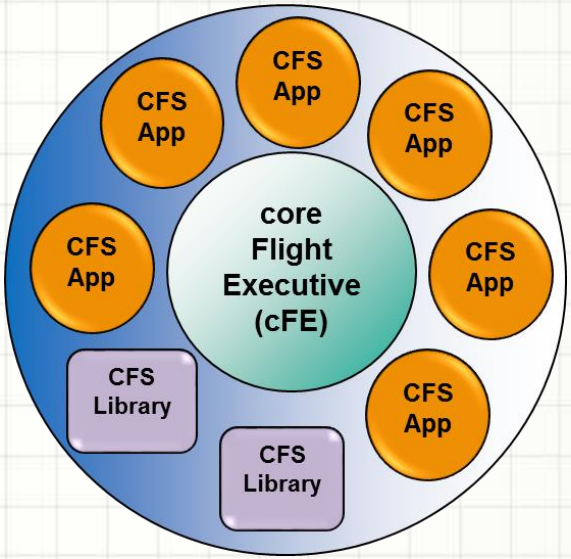


See **www.chrec.org** for more info

KEY TECHNOLOGIES

Core Flight Executive

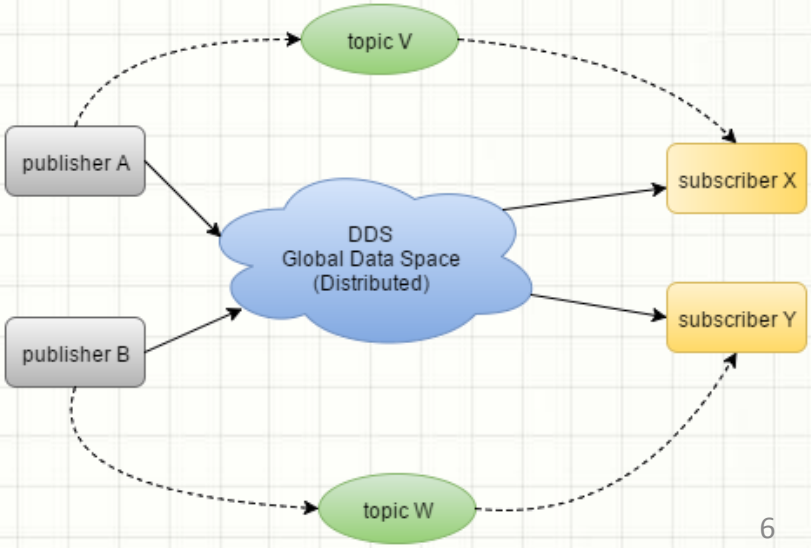
- Integrates with NASA Goddard's reusable flight software framework
  - Open source version of cFE/CFS **available at SourceForge**
  - Perform local device management, software messaging, & event generation
- Core Components
  - Core Flight Executive (cFE)
    - Mission-independent software services
  - Core Flight System (CFS)
    - Applications and libraries running on cFE



Developed by Object Management Group (OMG)

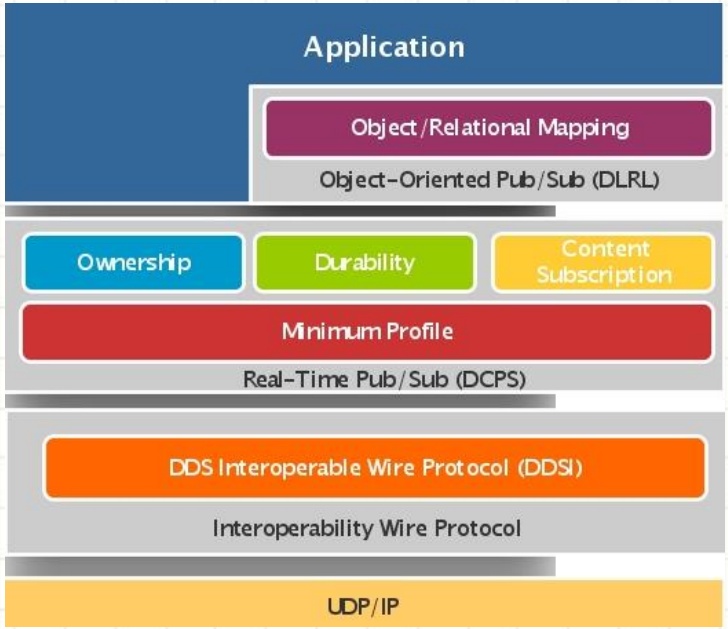
Data Distribution Service (DDS)

- Uses publish-subscribe model for data, events, and command exchange
  - Eliminates complex network programming for distributed applications
- Multiple DDS implementation vendors exist
  - OpenDDS (OMG)
  - OpenSplice (PrismTech)
  - ConnextDDS (RTI)

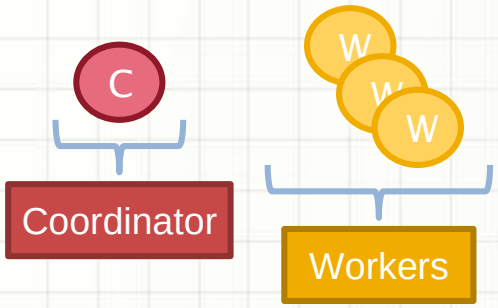


Data Distribution Service (DDS)

- Main DDS entities
  - Domain
    - Conceptual container: Entities communicate with each other only if they belong to same domain
  - Topic
    - Data object exchanged by subscribers/publishers
  - Publishers
    - Data producers
    - Creates “topics” and publishes data samples
  - Subscribers
    - Data consumers
    - DDS delivers data samples to them
- Interoperability
  - DDSI-RTPS specification
    - Interoperability protocol to allow multi-vendor DDS implementations communicate

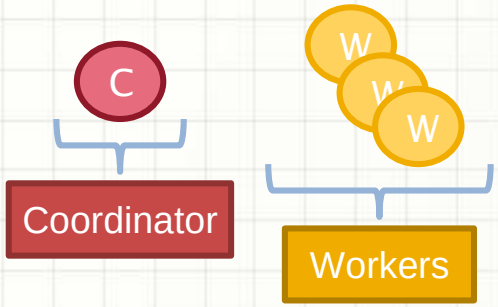


ADDAM

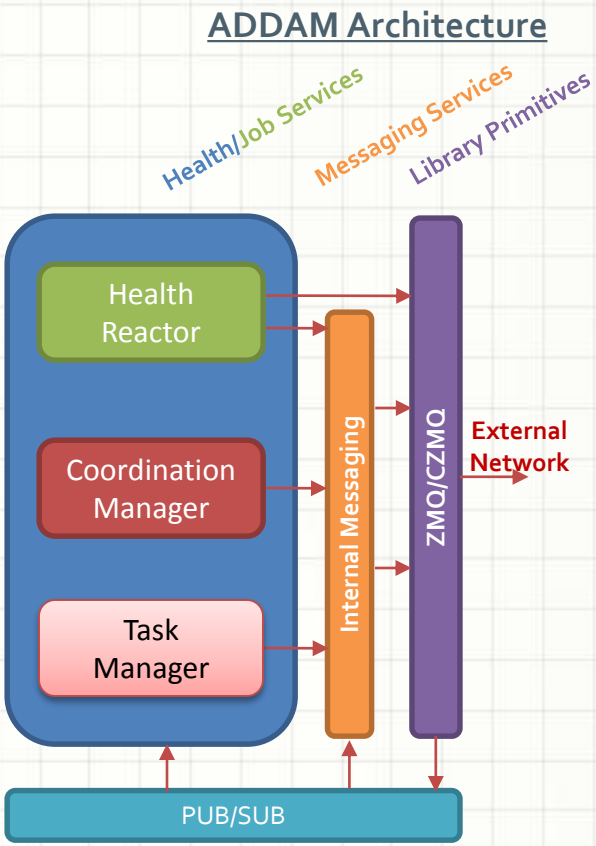


- Adaptive Dependable Distributed Aerospace Middleware (**ADDAM**)
  - System for dependable, distributed, and parallel computation
  - Self-recovering distributed system consisting of worker and coordinator agents
- Aims to provide a platform for dependable processing on parallel and distributed space computers
  - Assists in design and execution of fault-tolerant applications (serial and parallel)

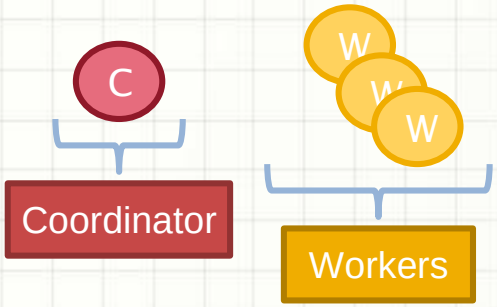
ADDAM



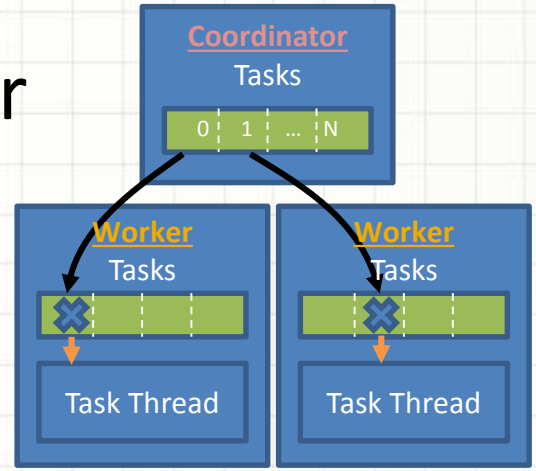
- Uses heartbeats from agents to interpret system events
  - Newly joined agents
  - Disconnected agents
  - Reconnected agents
- Peer discovery
  - Broadcast and receive availability beacons; doubles as heartbeats
- Health reactor
  - Track changes in peer availability
- Task manager
  - Handle task dispatching and re-issue
- Coordination manager
  - Perform coordinator election



ADDAM

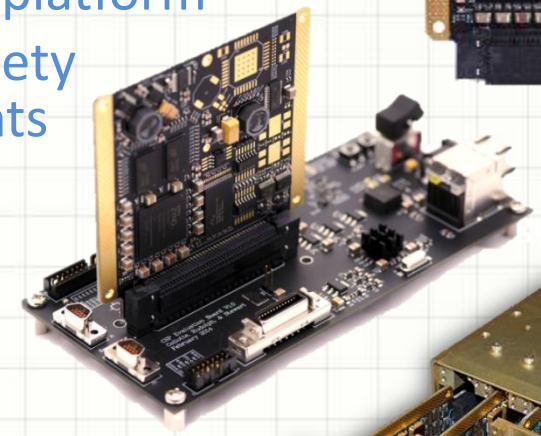


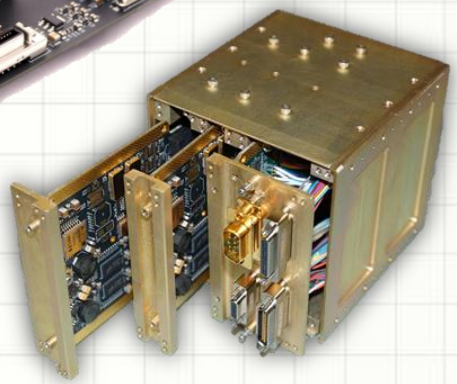
- Task redundancy
- Task reassignment on worker failure
- Coordinator failover
  - Using RAFT[1] consensus algorithm
- Per-device/node process failover
  - By use of an external daemon to monitor ADDAM processes and respawn them upon failure



[1] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in Proc. USENIX Annual Technical Conference, 2014, pp. 305–320.

CHREC Space Processor (CSP)



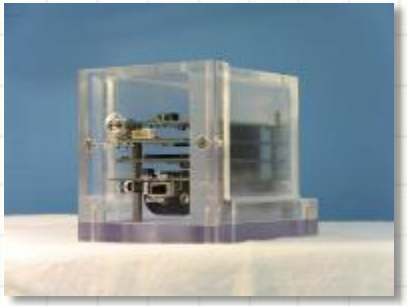


- Goal
  - High-performance, energy-efficient, low-cost, and dependable space-computing platform
  - Scalable and flexible to fulfill a variety of demands in mission requirements
  - Low power, high performance, and high reliability/availability
- Concept
  - **Multifaceted** hybrid computer
    - Hybrid system (commercial + rad-hard)
    - Hybrid processor (multicore CPU + FPGA subsystem) via Xilinx Zynq
  - Selective population scheme
    - **Pick-and-choose** commercial or rad-hard components
  - Flexible architecture (mission interfaces, algorithm acceleration)

Modular Integrated Stackable Layers

- Goals
  - Controller and processor "rack and stack" instrumentation technology with flexible design
  - Features reusable modules to meet varying mission requirements

- Objectives
  - Microcontroller systems and apps
  - Modular, scalable, reconfigurable
  - Quick to reconfigure for new apps
  - Industrial temperature environments
  - LEO environments



Related Work

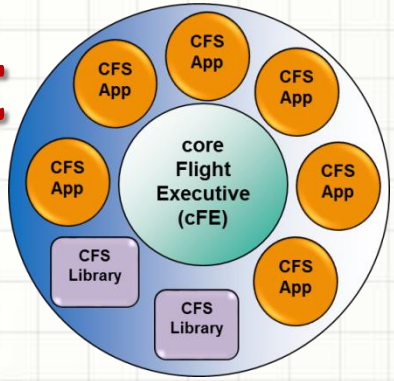
- In pursuit of this research, several other software solutions and middleware objects were studied
- **BioNET** – BioServe Space Tech., UC, NASA JSC
  - BioNet: A developer-centric middleware architecture for heterogeneous devices and protocols
- **CCSDS SOIS** – NASA / ESA
  - Spacecraft Onboard Interface Services
- **DMM** – Honeywell, UF, NASA
  - Dependable Multiprocessor Middleware
- **EDS** – CCSDS SOIS WG, ESA TRP
  - Electronic Data Sheets for Onboard Devices
- **Zero Configuration Networking (Zeroconf)**
  - Universal Plug and Play (UPnP)
  - Apple Bonjour
  - Avahi
- **Common Avionics Enabler (CAE)** – NASA JSC
  - Electronic endpoint capable of providing and/or utilizing the services of CCSDS SOIS

MOTIVATIONS AND GOALS

Why Middleware?

- **Middleware for Space**
  - Software services for reliable, interoperable, portable, reusable, efficient, and scalable discovery, management, and use of flight hardware and software resources
- **Example Use Cases**
  - Smart spacecraft, space supercomputing
- **Strategic Questions**
  - What future needs are well met by existing tools?
    - **Flight Computer Management (cFE/CFS toolset)**
  - What future needs are beyond existing tools?
    - **Flight System Management**
    - *Operating atop cFE/CFS, spanning multiple computers & modules*
    - *Leveraging of Spacecraft Onboard Interface Services (SOIS)*

Flight System Management



- **Flight Computer Management (existing)**
  - Focus upon space computer with its attached units
  - Core (cFE) and extended (CFS) services, API, apps
- **Flight System Management (notional)**
  - **Focus upon system-wide resources and management**
    - Broader scope for higher reliability, performance, configurability, adaptability, and scalability w/ space computers and smart modules
    - Variety of interfacing specifications, as defined in SOIS of CCSDS
  - **Multiple space processors and computers**
    - Spanning multiple devices, boards, chassis, and even spacecraft
    - To enable ***dependable computing*** (redundancy), ***distributed computing*** (cooperation), and ***parallel computing*** (collaboration)
  - **Assortment of smart modules**
    - Each capable of computing and networking, however modest
    - SBCs (e.g., CSP, SpaceCube) for C&DH and/or data processing, smart thrusters, smart comm, smart power, smart instruments, et al.
    - Each running some form of flight management core (cFE, cFE-lite)

SBC = single-board computer

Ex: Smart Modules on SmallSats?

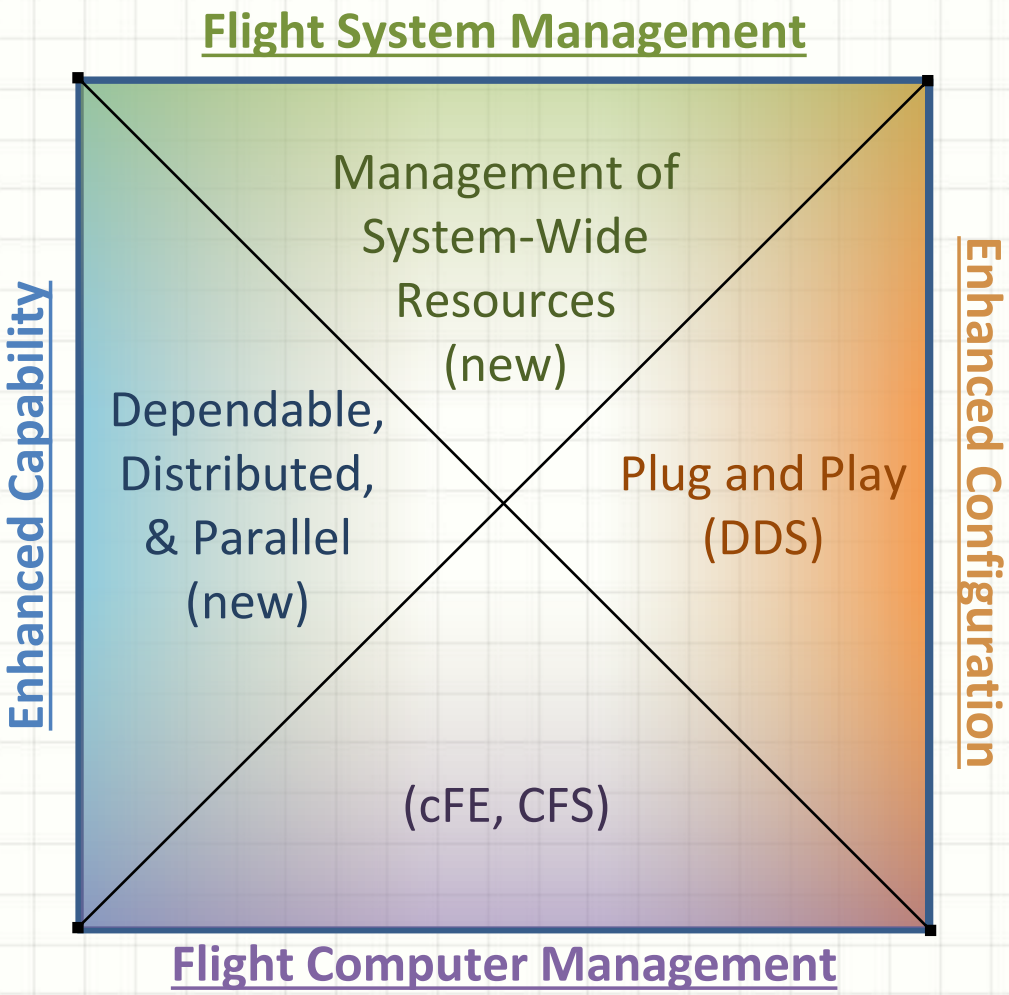
- Which devices are potential candidates as smart modules?
- Could a device type represent multiple network resources?
- Could multiple resources represent a single service?

- **Power**
  - Solar cells
  - Batteries
  - Power generator
- **Propulsion**
  - Thruster
  - Solar sail
  - Attitude determination and control
- **Communication**
  - Transmitters
    - Data rate, duplex
  - Flight terminal
    - Camera, fast-steering mirror
- **Instruments**
  - Photography camera
  - Gamma ray detector
  - Photometer
  - Optical spectrometer

- **Attitude determination and control**
  - Reaction wheels
    - Angular momentum
    - Maximum torque
    - Power
    - Microvibrations
  - Magnetorquer
  - Control moment gyros
  - Aerodynamic wing
  - Star tracker
    - Accuracy
    - Output rate
    - First tracking time
    - Maximum allowable slew rate (attitude maneuver rate)
  - Sun sensors
  - Earth sensors
  - Angular rate sensors
    - Precision: bias instability, angle random walk
  - GPS receiver and antennas
  - Active thermal control systems

Source: "Small Spacecraft Technology State of the Art" (NASA/TP–2014–216648/REV1)

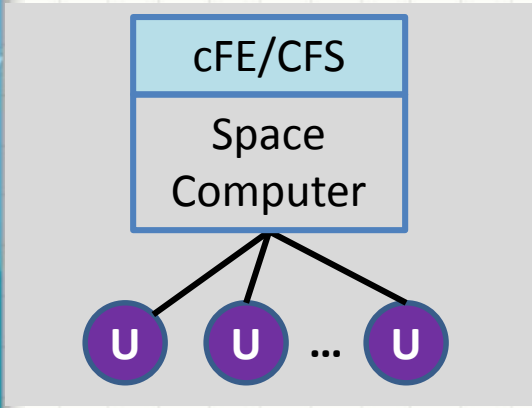
Outlining Areas of Work & Interest



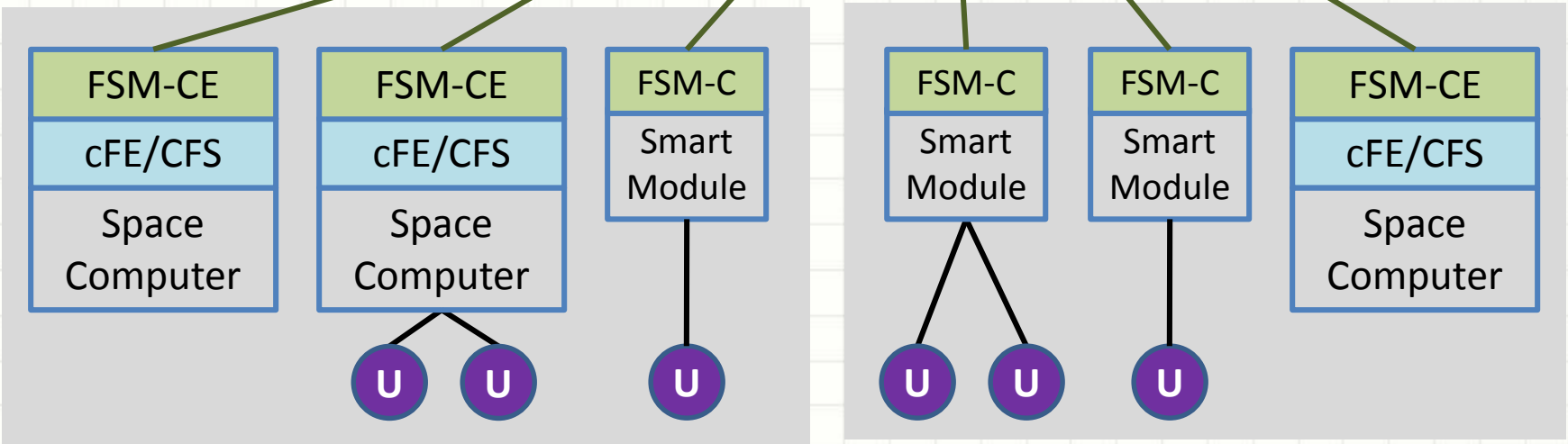
Example Comparison

U = unit (sensor, NAS, comm, power, thruster, et al.)
FSM-C = core services of FSM
FSM-CE = core & extended services of FSM

Centralized Architecture

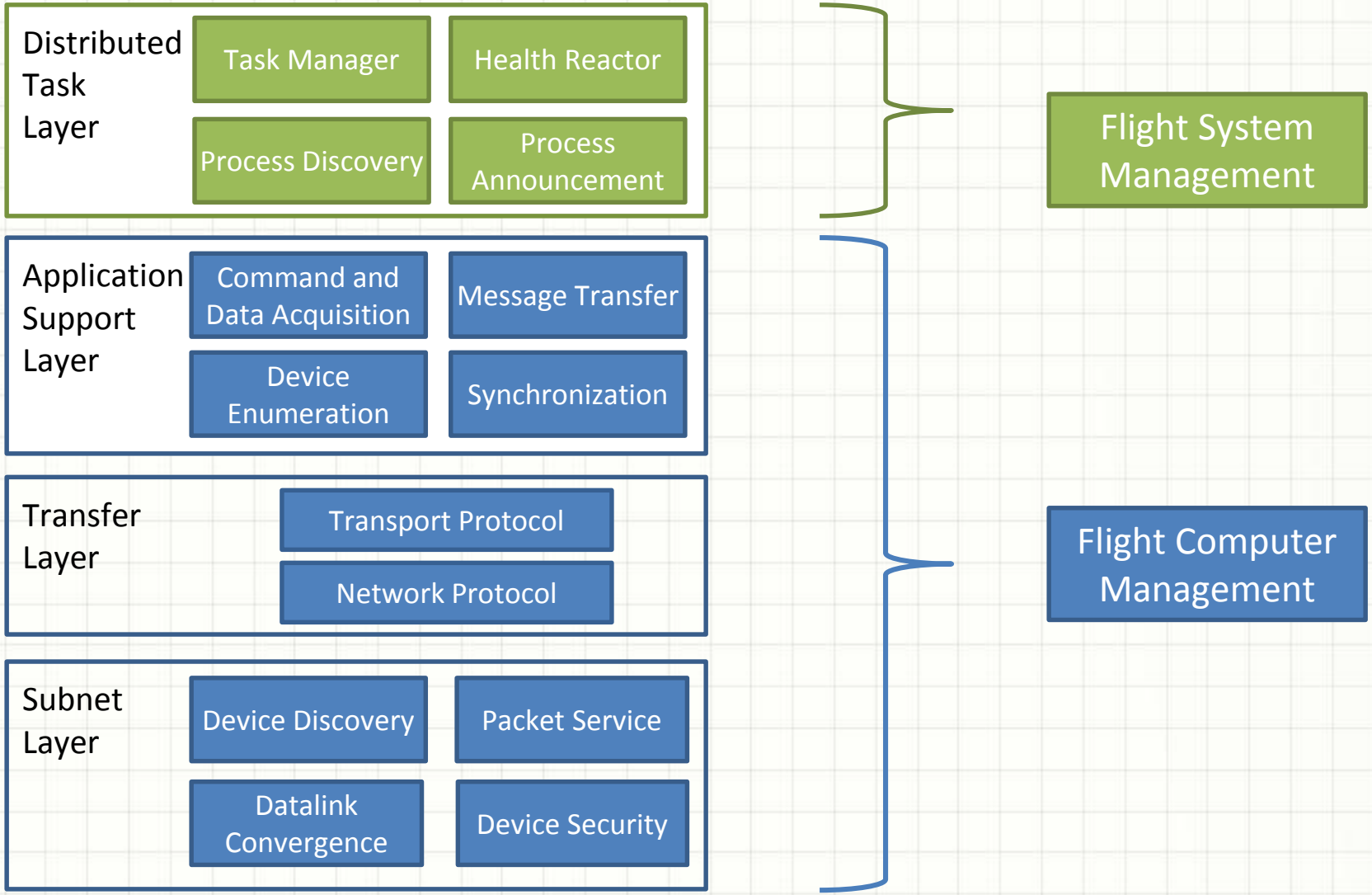


Distributed Architecture



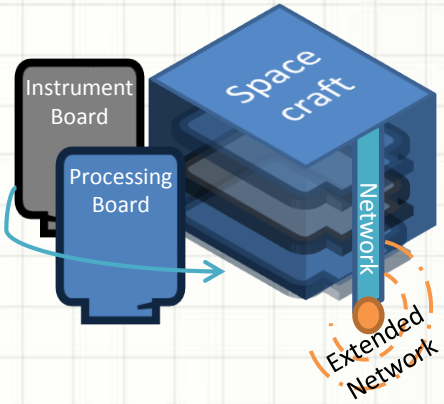
Middleware spanning multiple devices, boards, chassis, or even spacecraft

Enhanced Services

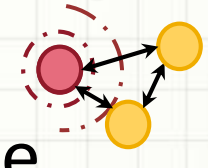


Goal: Enhanced Spacecraft Configuration

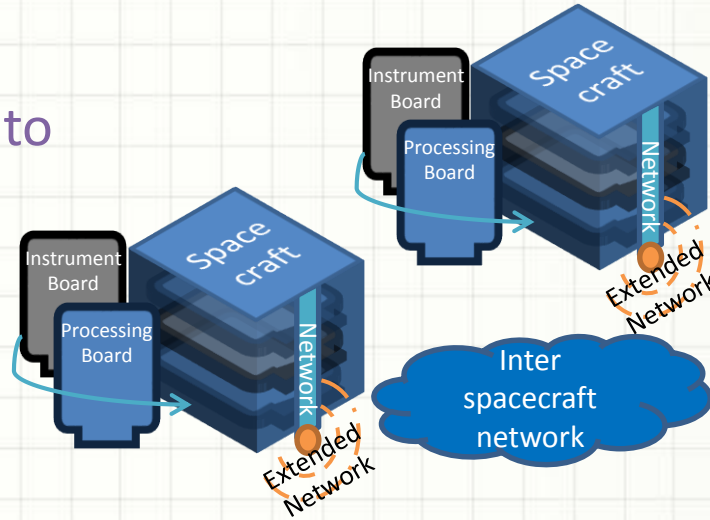
- Aims to be reliable, interoperable, portable and reusable
  - Expedite spacecraft development
    - Facilitate quick prototyping, integration (plug-and-play) and testing
  - Enable in-flight reconfiguration
    - Generalizing device usage to resource provider
- Technologies employed
  - DDS
    - For automatic device discovery and plug and play capability
  - cFE/CFS



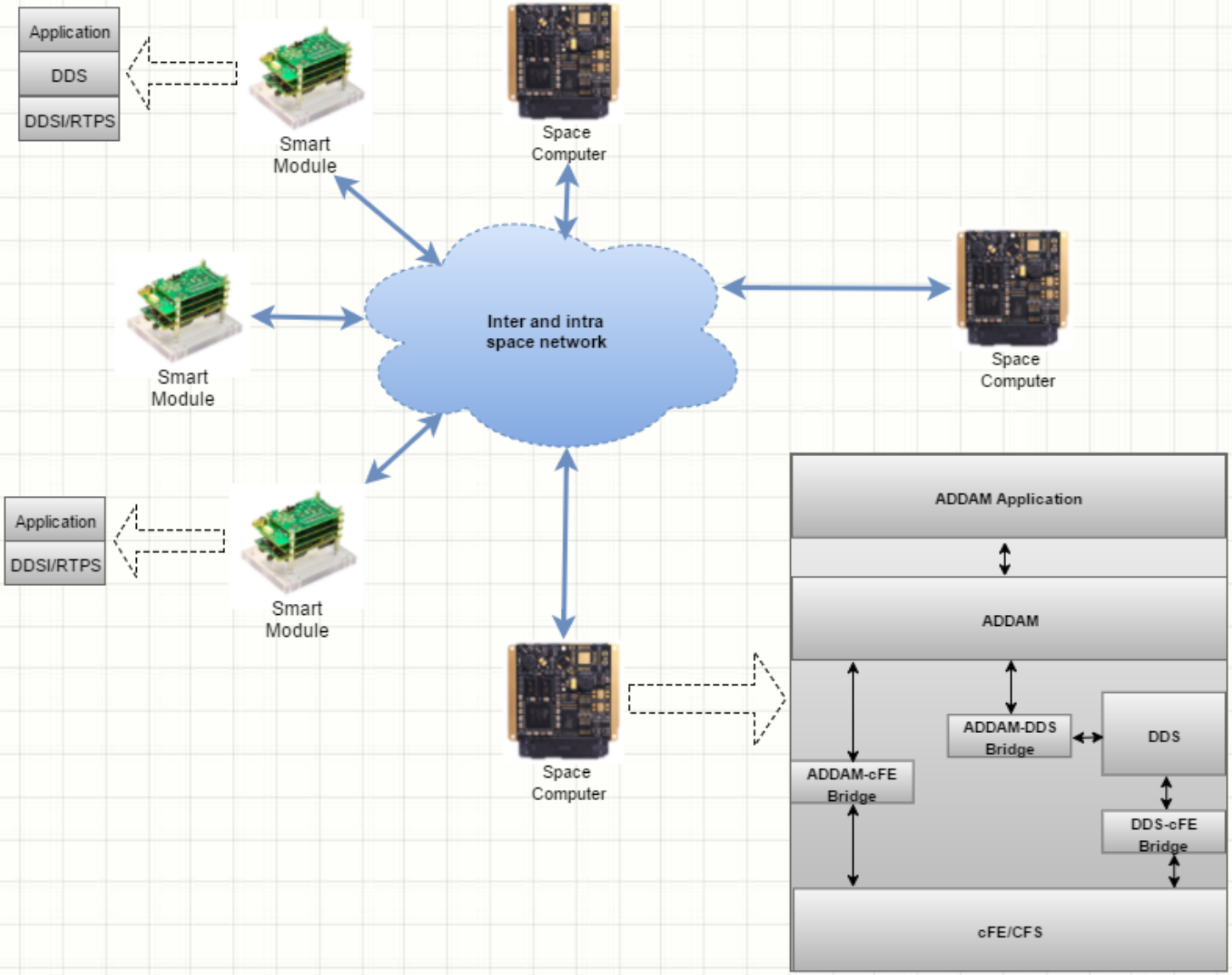
Goal: Enhanced Spacecraft Capability



- Aims to be fast, efficient, reliable, and scalable
  - Enhanced dependability
    - Make use of redundant resources to improve reliability and availability
  - Distributed resources
    - Producer-consumer relationships for advanced functionality
  - Parallel computing
    - Coordinated resources improving performance, scalability
- Technologies employed
  - ADDAM
    - For enhanced dependability and parallel computing

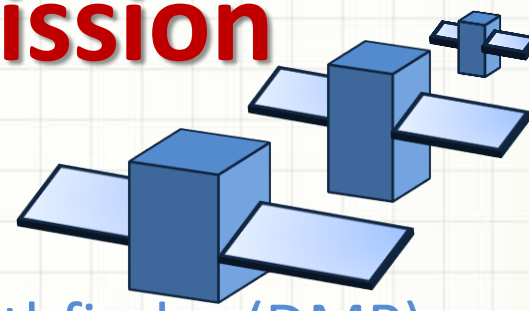


Architectural Overview



PROTOTYPE: NETWORKED STEREOSCOPIC IMAGING

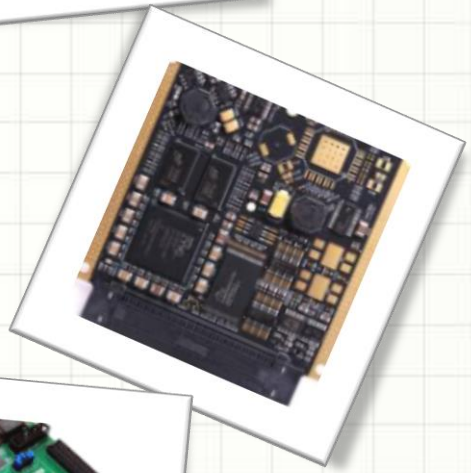
Prototype: Motivating Mission

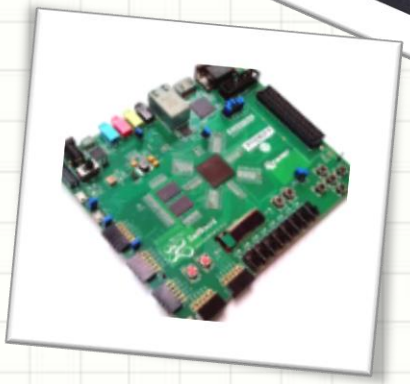


- Platform
  - Emulate Distributed Measurement Pathfinder (DMP)
    - 3 small satellites in formation, each with 1 camera and 2 space processors (DMP is a proposed mission by GSFC)
  - Intra-satellite network unified with inter-satellite network for simplicity
    - Network bridge is assumed for communications across satellites
    - Testing performed with Ethernet switch
- Test bed components mimicking target platform
  - Featuring smart modules (processors, sensors)

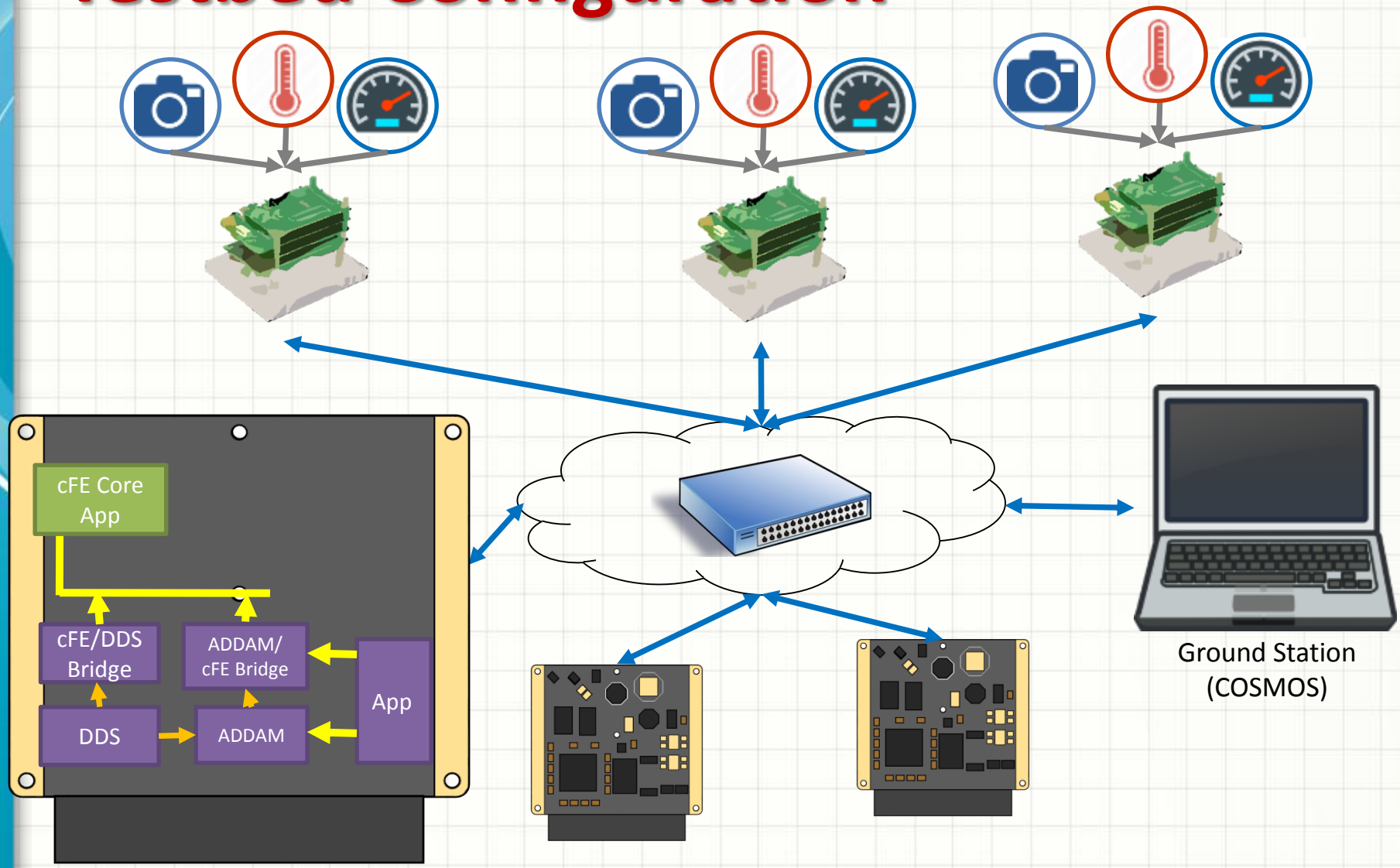
Prototype Hardware Components

- CSPv1 computers
- MISL RM48 modules
- Sensors
  - OV7670 imaging sensor
  - Thermocouple
  - Accelerometer
- Ethernet switch
- Ground station computer





Testbed Configuration



Planned Prototype Use Cases

- Communication between CSPs and smart modules
  - Verify communication between CSPs and smart modules
  - Performed by reading telemetry data sent to ground station computer
- Plug-and-play functionality
  - Add and remove smart modules and CSPs arbitrarily to verify plug-and-play capability
- Image stereoscopy
  - Perform image stereoscopy on a single node and on distributed nodes using ADDAM
  - Observe performance improvements when using ADDAM on distributed nodes

Prototype Use Cases (continued)

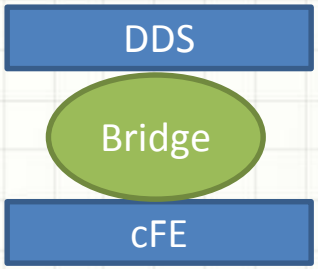
- Parallel processing
  - Demo multinode, multicore, & SIMD processing of images
  - Between and within CSPs (MPI, OpenMP, NEON)
- Coordinator failover
  - Verify election of new coordinator on failure of coordinator node
  - Performed by unplugging coordinator node and observing telemetry data on ground station
- Process failover
  - Verify ADDAM processes are respawned when they are killed
  - Performed by sending kill command from ground station and observing telemetry data on ground station to confirm election of new coordinator

Project Progress

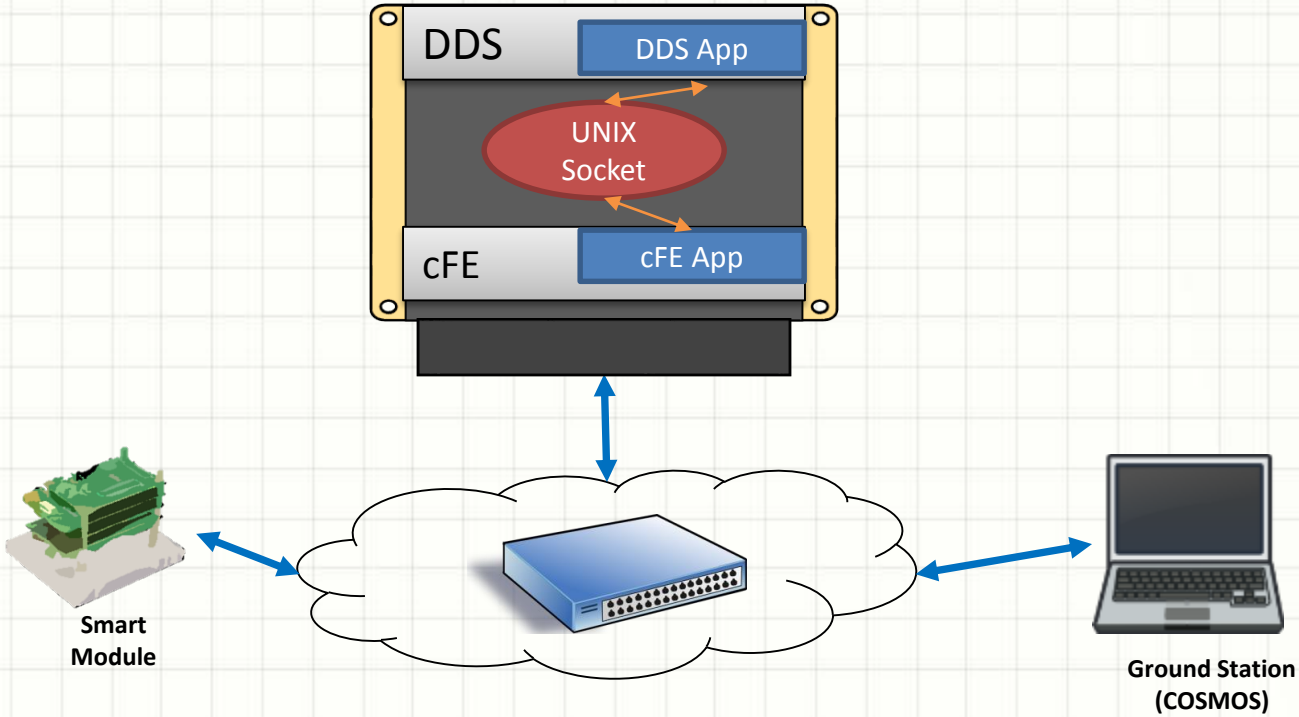
- Data Distribution Service
  - Chose OpenDDS[1] for use in space computer
  - Chose FreeRTPS[2] implementation for use in smart modules
  - Defined IDL[3] message types to be exchanged between space computers and smart modules
- cFE
  - Built cFE-DDS bridge
  - Built ADDAM-DDS bridge

[1] Open source DDS implementation by Object Management Group (OMG)
[2] Open source RTPS implementation by Open Source Robotics Foundation (OSRF)
[3] Interface Description Language

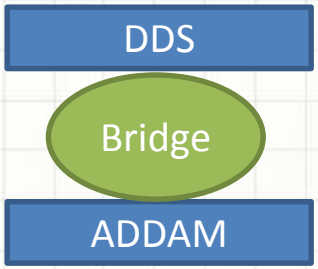
cFE-DDS Bridge



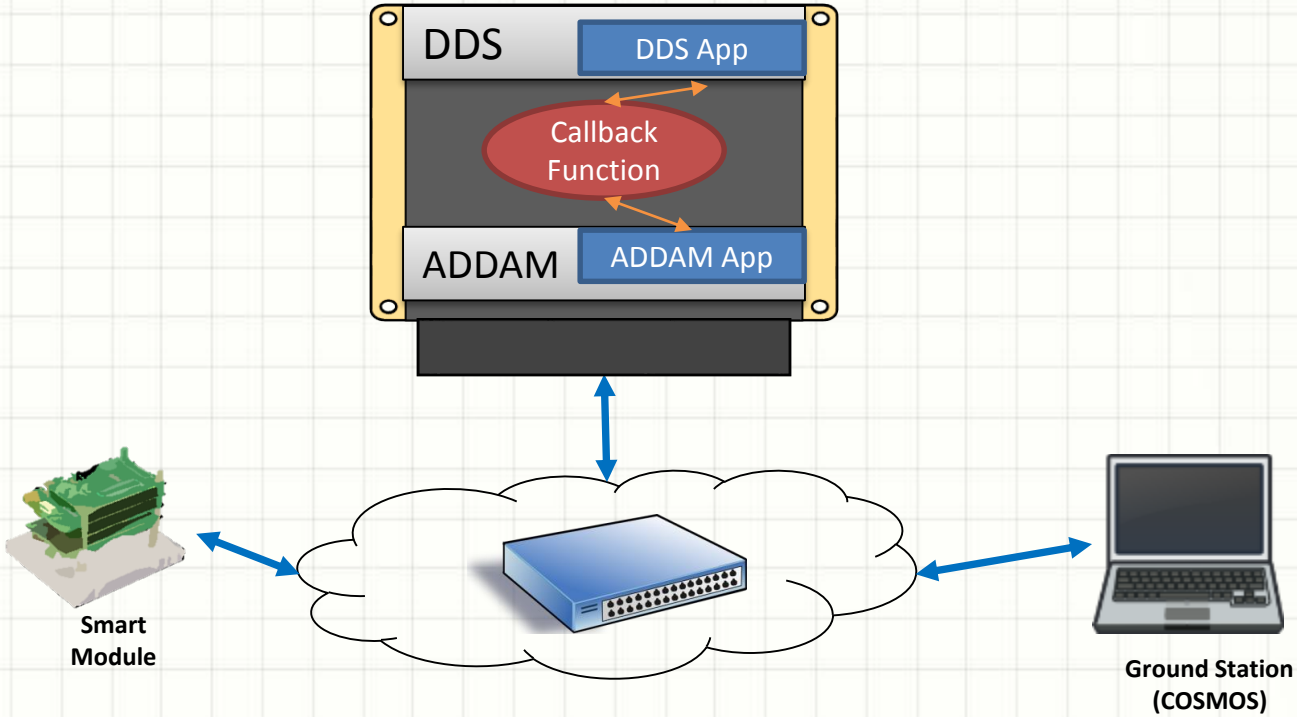
- Forwards ground station commands to smart modules
- How it works
  - cFE app sends data(commands) received from ground station to UNIX socket
  - DDS application listens on UNIX socket and interprets commands and forwards them to relevant smart modules



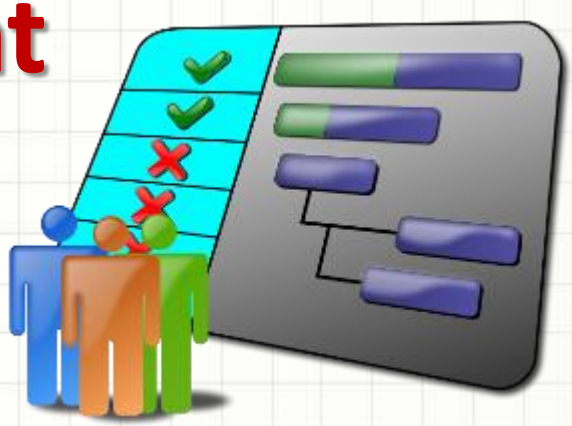
ADDAM-DDS Bridge



- Forwards DDS data from smart modules to applications running on ADDAM
- Application running on ADDAM specifies:
  - Topic name it wishes to receive data from
  - Callback function to be called whenever data samples for topic are available



Upcoming Development

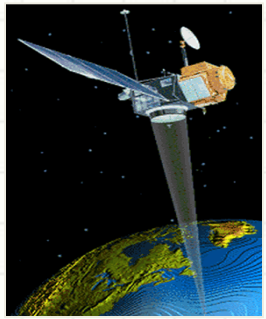


- Data Distribution Service
  - Port FreeRTPS to RM48 (MISL)
- ADDAM
  - Finish coordinator failover implementation
  - Add per-device process failover capability
    - By use of an external daemon
  - Add MPI-like API
    - addam_send, addam_receive, addam_barrier
  - Add logging framework with log replication capability across nodes

MPI = Message Passing Interface
API = Application Programming Interface

Conclusions

- R&D underway on CHREC Space Middleware (CSM)
  - Goals defined by group of CHREC members & sites
  - Focus: **Enhanced spacecraft configuration**
    - Smart modules, plug-and-play, smart spacecraft
  - Focus: **Enhanced spacecraft capability**
    - Dependable, distributed, & parallel computing



- Initial CSM prototype in development
  - Novel mix of existing & emerging technologies
    - cFE/CFS, DDS, CSP, MISL, ADDAM, et al.
  - Initial prototype: CubeSats (3) formation mockup
    - Space computers & smart sensor modules, stereoscopic imaging
  - Initial demo @ CHREC Annual Workshop (CAW15)
    - December 2-3, hosted by NASA Kennedy Space Center

QUESTIONS?