National Aeronautics and
Space Administration

High-Performance Spaceflight Computing (HPSC) Middleware Update

Alan Cudmore

NASA Goddard Space Flight Center
Flight Software Systems Branch (Code 582)

Alan.P.Cudmore@nasa.gov

This is a non-ITAR presentation, for public release and reproduction from FSW website.

www.nasa.gov/spacetech

Agenda

- **High Performance Spaceflight Computing (HPSC) Overview**
  - HPSC Project Overview
  - HPSC Chiplet Architecture
  - HPSC Software Architecture
- **Middleware Overview**
  - Middleware Project Overview
  - Middleware Architecture
  - Middleware Schedule/Milestones
- **Middleware Release 2**
  - Middleware Release 2 Overview
  - High Performance Fault-Tolerant Embedded Computing (HPFEC) Mission Use Case
  - Core Flight System (cFS) Mission Use Case
- **Future Middleware Releases**
  - Middleware Release 3
  - Middleware Release 4 and Beyond

This is a non-ITAR presentation, for public release and reproduction from FSW website.

High Performance Spaceflight Computing (HPSC) Overview

HPSC Program Goals

- Dramatically advance the state of the art for spaceflight computing
- HPSC will provide a **nearly two orders-of-magnitude improvement** above the current state of the art for radiation hardened spaceflight processors, while also providing an unprecedented **flexibility to tailor performance, power consumption, and fault tolerance** to meet widely varying mission needs
- These advancements will provide **game changing improvements in computing performance, power efficiency, and flexibility**, which will significantly improve the onboard processing capabilities of future NASA and Air Force space missions

HPSC Program Funding/Management

- HPSC is **funded** by NASA's Space Technology Mission Directorate (STMD), Science Mission Directorate (SMD), and the United States Air Force
- The HPSC project is **managed** by Jet Propulsion Laboratory (JPL), and the HPSC contract is managed by NASA Goddard Space Flight Center (GSFC)

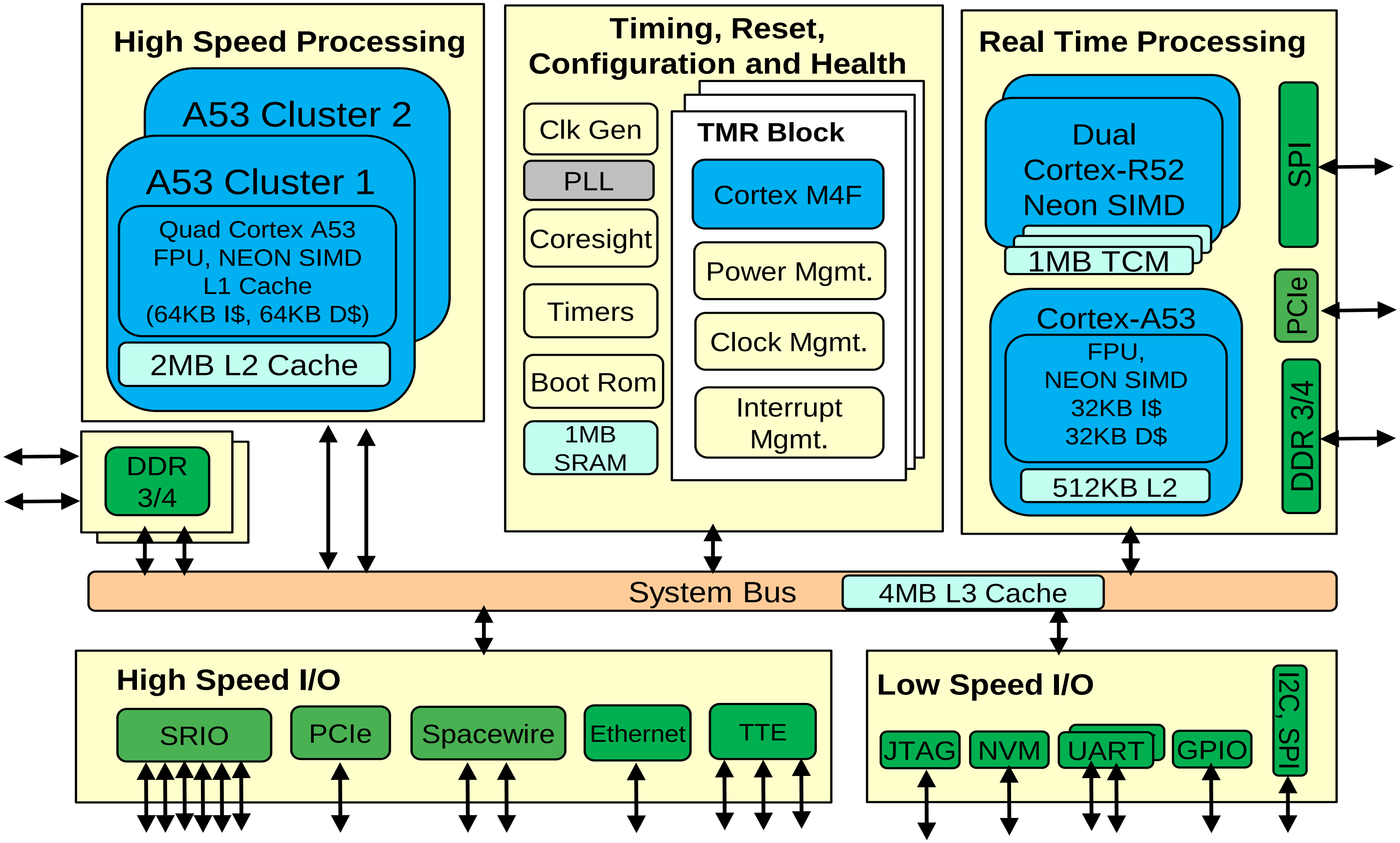
HPSC Program Deliverables

Under the base contract, Boeing will provide:

- Prototype radiation hardened multi-core computing processors (**Chiplets**), both as bare die and as packaged parts
- Prototype **system software** which will operate on the Chiplets
- **Evaluation boards** to allow Chiplet test and characterization
- **Chiplet emulators** to enable early software development

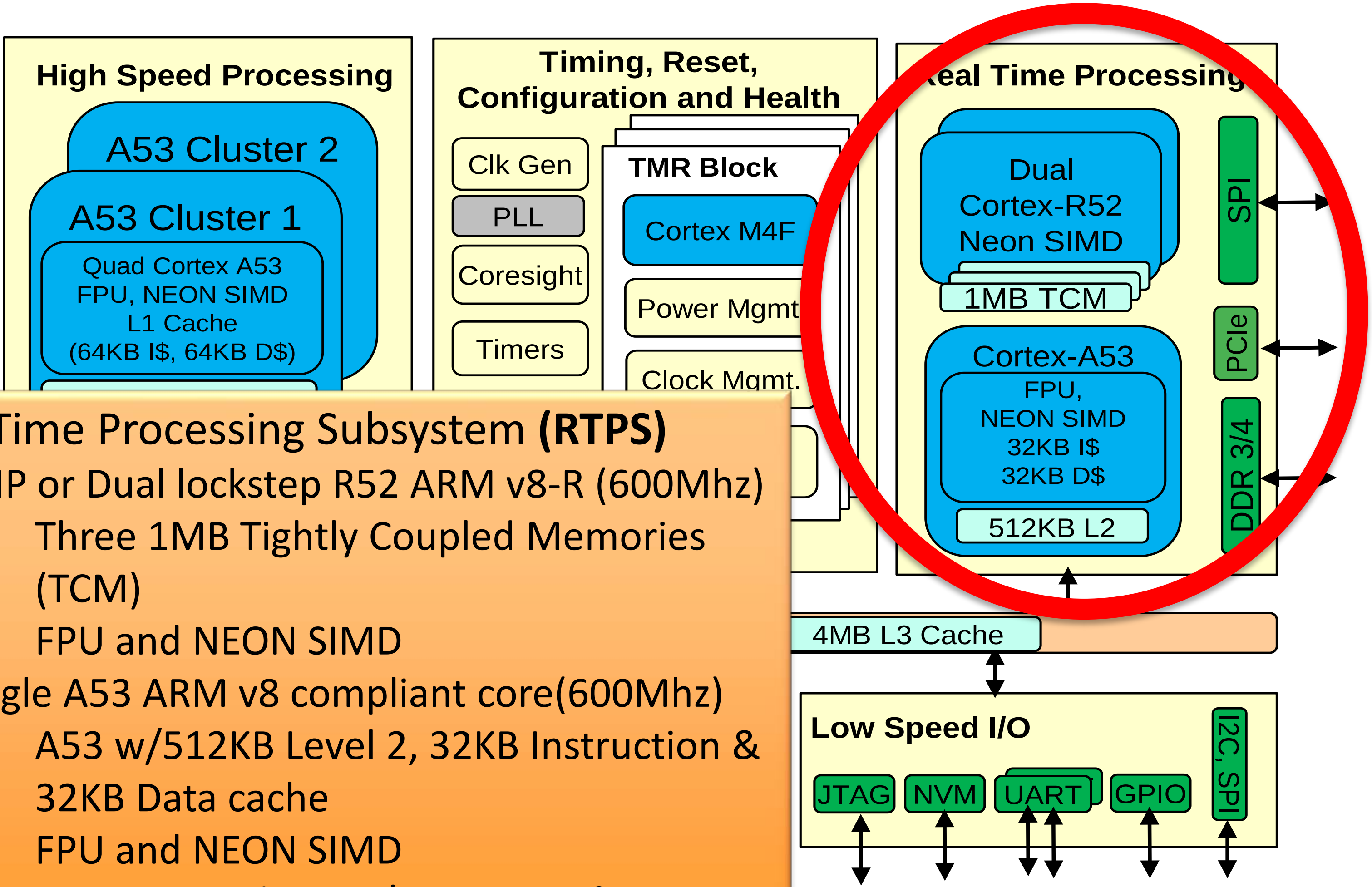
This is a non-ITAR presentation, for public release and reproduction from FSW website.

HPSC Chiplet Architecture



This is a non-ITAR presentation, for public release and reproduction from FSW website.

HPSC Chiplet Architecture

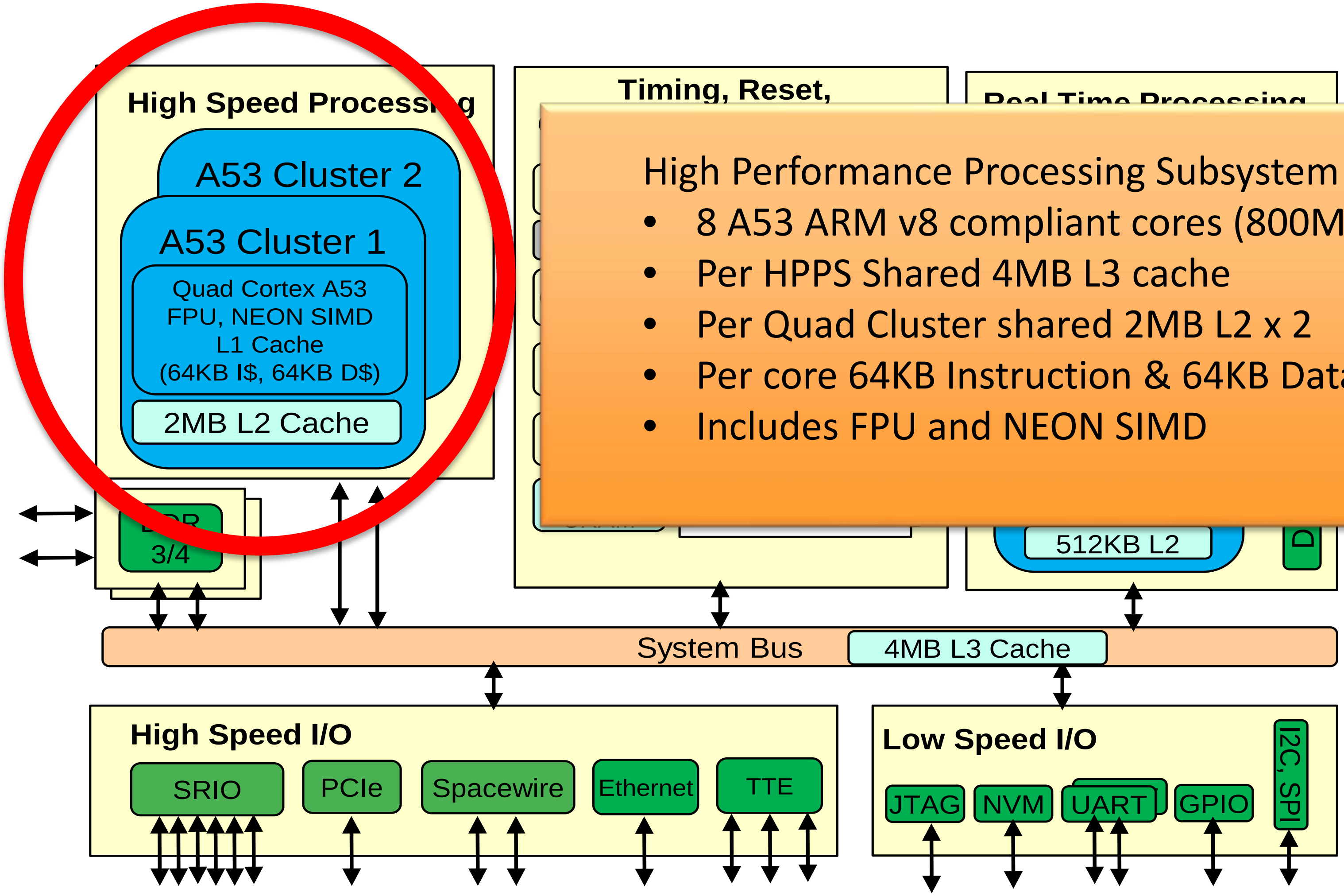


Real Time Processing Subsystem **(RTPS)**

- SMP or Dual lockstep R52 ARM v8-R (600Mhz)
  - Three 1MB Tightly Coupled Memories (TCM)
  - FPU and NEON SIMD
- Single A53 ARM v8 compliant core(600Mhz)
  - A53 w/512KB Level 2, 32KB Instruction & 32KB Data cache
  - FPU and NEON SIMD
- Separate PCIe and DDR3/DDR4 interfaces

This is a non-ITAR presentation, for public release and reproduction from FSW website.

HPSC Chiplet Architecture



High Performance Processing Subsystem **(HPPS)**

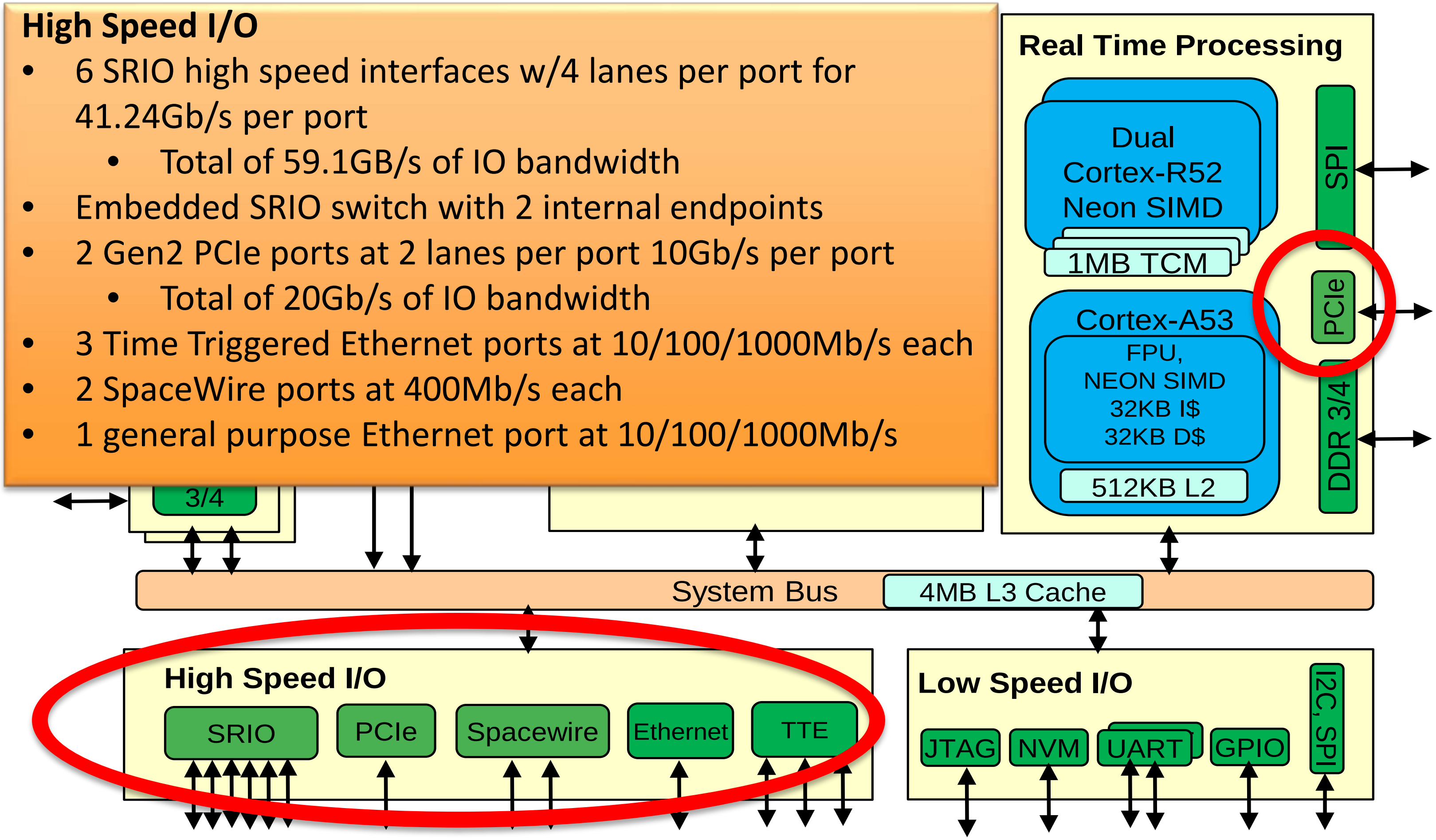
- 8 A53 ARM v8 compliant cores (800Mhz)
- Per HPPS Shared 4MB L3 cache
- Per Quad Cluster shared 2MB L2 x 2
- Per core 64KB Instruction & 64KB Data cache
- Includes FPU and NEON SIMD

This is a non-ITAR presentation, for public release and reproduction from FSW website.

HPSC Chiplet Architecture

High Speed I/O

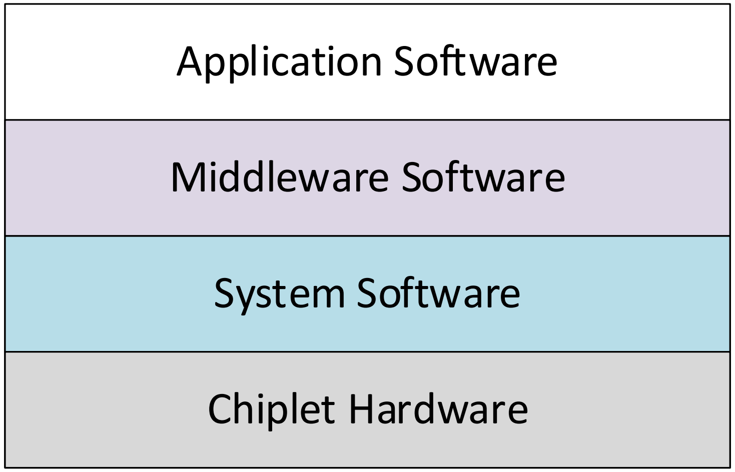
- 6 SRIO high speed interfaces w/4 lanes per port for 41.24Gb/s per port
  - Total of 59.1GB/s of IO bandwidth
- Embedded SRIO switch with 2 internal endpoints
- 2 Gen2 PCIe ports at 2 lanes per port 10Gb/s per port
  - Total of 20Gb/s of IO bandwidth
- 3 Time Triggered Ethernet ports at 10/100/1000Mb/s each
- 2 SpaceWire ports at 400Mb/s each
- 1 general purpose Ethernet port at 10/100/1000Mb/s



This is a non-ITAR presentation, for public release and reproduction from FSW website.

HPSC Software Architecture

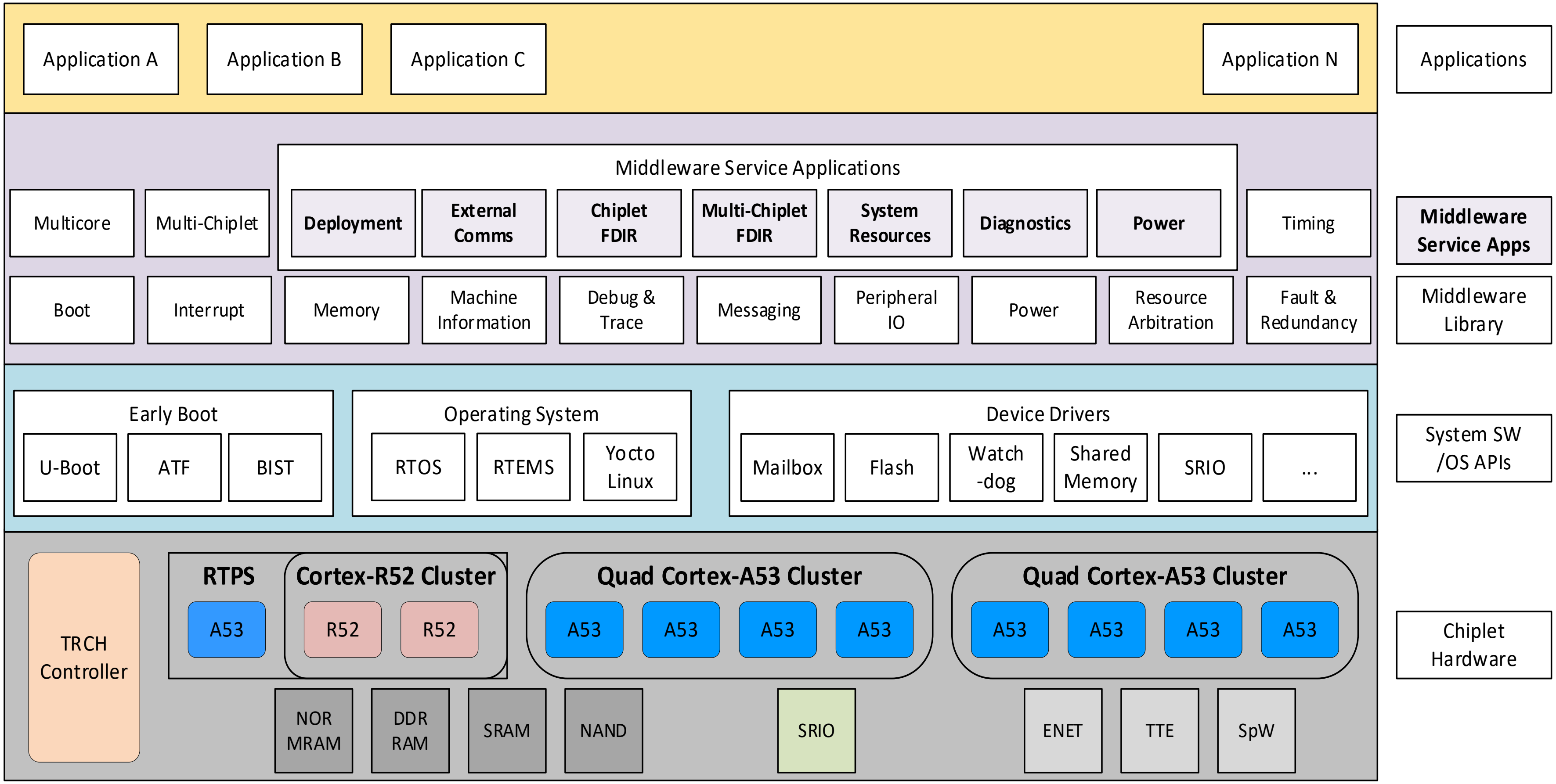
- **Layered Architecture**
  - Provides separation of concerns with well-defined interfaces
  - Provides application developers with tools, abstractions and methodologies to make use of lower level HPSC SW services and Chiplet HW in a consistent manner



- **Middleware Layer**
  - Allocate and manage processor cores, devices and resources to dynamically trade computational performance, energy management and fault tolerance objectives of the mission
  - Provides for mission and platform configurability
  - Allows Mission Integration Engineers (MIE) to reason about interactions between applications and the system
- **System Software Layer**
  - RTOS/OS, hypervisor and low level device drivers for the Chiplet are provided by HPSC vendor University of Southern California Information Sciences Institute (USC-ISI)

This is a non-ITAR presentation, for public release and reproduction from FSW website.

HPSC Software Architecture



This is a non-ITAR presentation, for public release and reproduction from FSW website.

System Software Layer

- **Provided by vendor USC - Information Sciences Institute (USC-ISI)**
  - **Runtime Components:**
    - **Bootloaders**:
      - Custom TRCH Bootloader
      - U-boot for RTPS and HPPS
    - **RTOS**:
      - RTEMS for TRCH and RTPS/R52
    - **Linux:**
      - Yocto based Linux with KVM hypervisor for HPPS
    - **Chiplet Device drivers**
      - Memory controllers, DMA, peripheral I/O devices, GIC…
    - **Software Build-In System Tests** (BIST(s))
      - Memory test, core-cache path, cores, …
    - **Software-Implemented fault tolerance framework**
  - **Development Tools:**
    - A software Chiplet emulator: based on QEMU
    - Eclipse-based IDE with advanced performance profiling

Middleware Overview

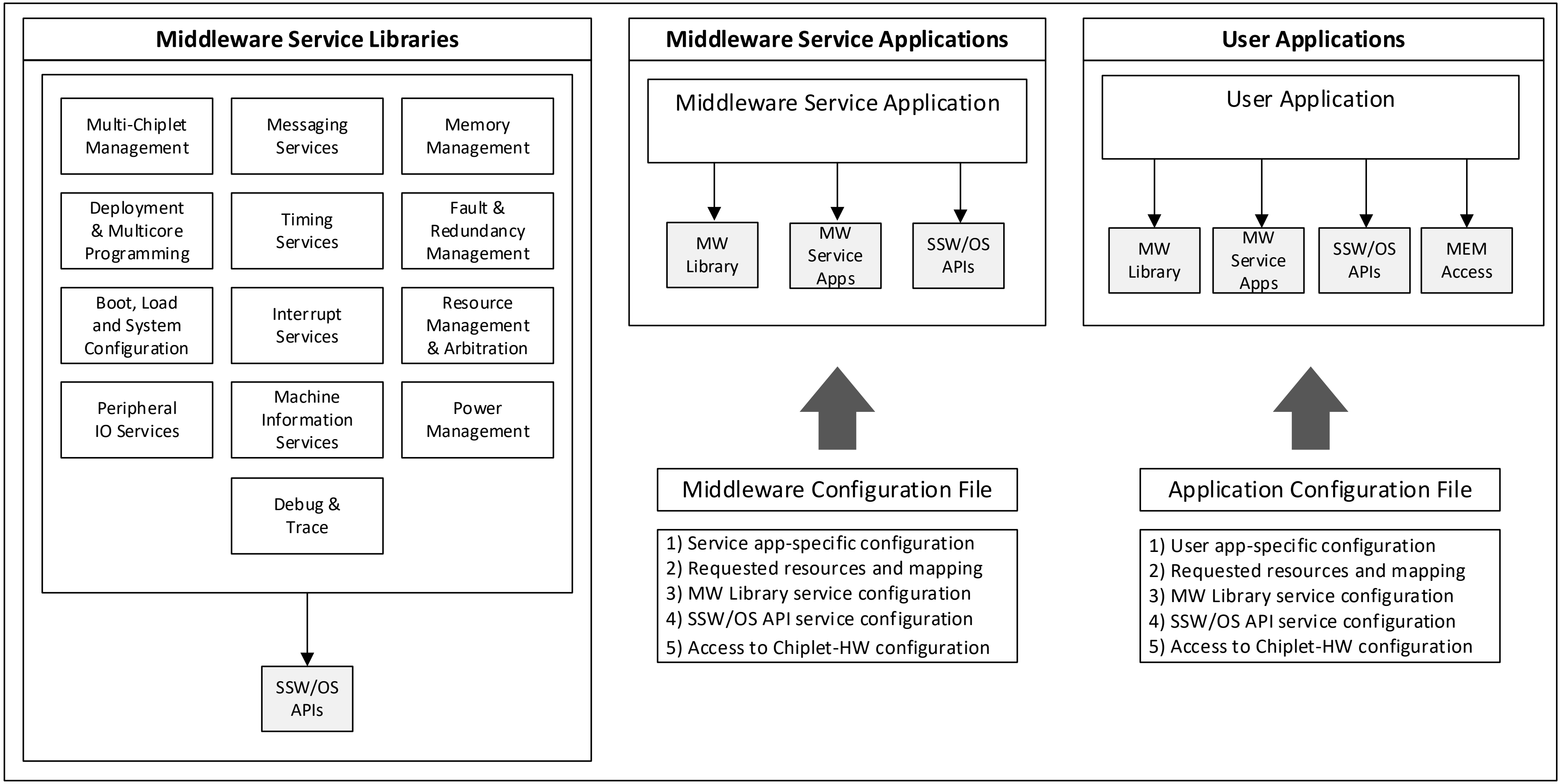
- **Middleware Project Information**
  - A joint project between NASA Jet Propulsion Lab (JPL) and NASA Goddard Space Flight Center (GSFC)
  - Funded by the Air Force Research Lab (AFRL)
- **Middleware Objectives**
  - Provide applications with an efficient and consistent functions to utilize features of the Chiplet multicore platform for
    - Resource allocation and management (e.g., cores, memories, peripheral devices)
    - Configuration management
    - Power and Performance management
    - Fault tolerance and redundancy management
    - Sharing hardware resources using a common middleware library and services
    - Messaging Services
  - Facilitate deployment of applications for execution with different characteristics and resource needs (e.g., computational needs, mission criticality, timing criticality, …)

This is a non-ITAR presentation, for public release and reproduction from FSW website.

Middleware Architecture

- **Middleware Service Libraries**
  - Middleware service library consists of building block functions with APIs that provide a set of application services.
  - MW service library has no runtime thread context on its own.
  - MW service library is called in user application context for invoking MW services that interacts with Chiplet devices, RTOS, OS and other services.
- **Middleware Service Applications**
  - MW Service Applications provide a set of MW services to meet specific system needs once deployed.
  - MW Service Applications are executable programs with runtime context. Each service program is configured to a specific deployment need / environment via a user specified configuration file at boot time.

Middleware Service Conceptual Patterns



Middleware Service Libraries

Multi-Chiplet Management

Messaging Services

Memory Management

Deployment & Multicore Programming

Timing Services

Fault & Redundancy Management

Boot, Load and System Configuration

Interrupt Services

Resource Management & Arbitration

Peripheral IO Services

Machine Information Services

Power Management

Debug & Trace

SSW/OS APIs

Middleware Service Applications

Middleware Service Application

MW Library

MW Service Apps

SSW/OS APIs

Middleware Configuration File

1) Service app-specific configuration
2) Requested resources and mapping
3) MW Library service configuration
4) SSW/OS API service configuration
5) Access to Chiplet-HW configuration

User Applications

User Application

MW Library

MW Service Apps

SSW/OS APIs

MEM Access

Application Configuration File

1) User app-specific configuration
2) Requested resources and mapping
3) MW Library service configuration
4) SSW/OS API service configuration
5) Access to Chiplet-HW configuration

This is a non-ITAR presentation, for public release and reproduction from FSW website.

Middleware Service Libraries

Service Name	Service Function
Deployment and Multicore Programming	Support different deployment configurations on multi-core systems
Machine Information	Get information about cores, clusters, Chiplets, and system
Resource Management/Arbitration	Manage configuration files and assign available resources to software clients
Memory Management	MMU configuration, memory allocation to applications
Interrupt	Configure Interrupt controller to route interrupts to specific cores at runtime
Timing	Establish, synchronize, and distribute time
Messaging	Send/receive messages between tasks on the same or different cores and Chiplets
Power Management	Power on/off setting for cores/clusters/Chiplets, and set clock rates
Debug and Trace	Support software debugging and performance metrics
Boot and Load Process	Provide an Initial Program Loader (IPL) and support various boot types/processes
Peripheral IO	Allocate and configure Chiplet Peripheral I/O to software clients
Multi-Chiplet Management	Configure and manage Inter-Chiplet interfaces (e.g., boot, messaging, resources)
Fault and Redundancy Management	Fault detection (HW/SW faults) and recovery

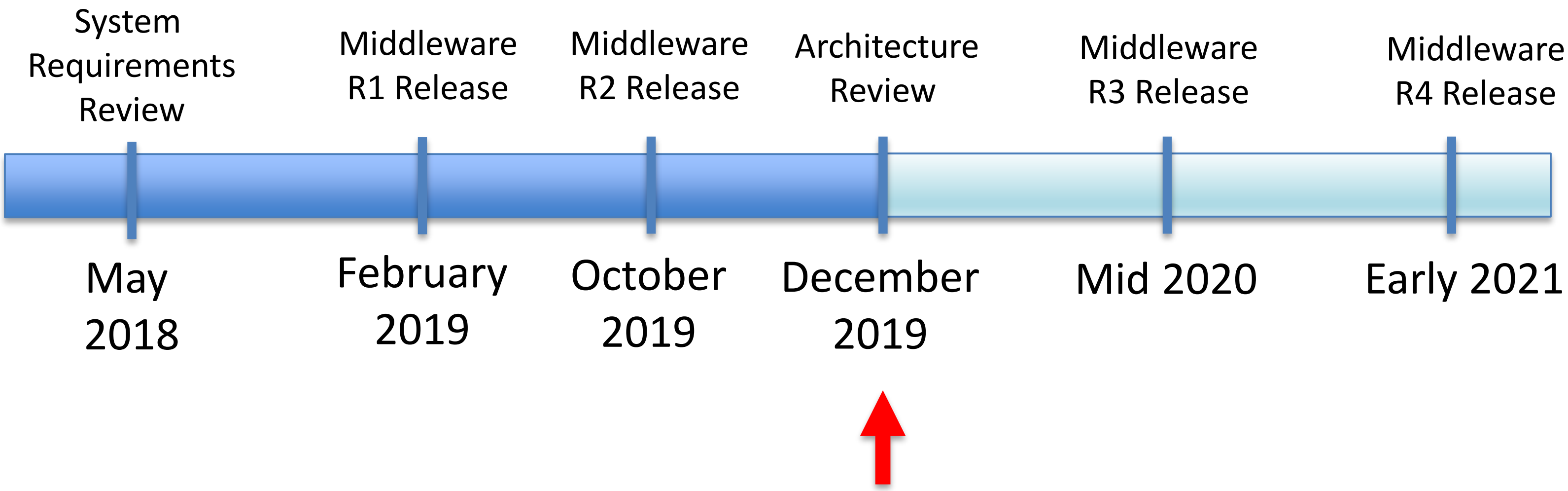
This is a non-ITAR presentation, for public release and reproduction from FSW website.

Middleware Service Applications

MW Service Application	System-Level Function
Deployment	Set up the system and start applications
Hardware Resource	Arbitrate access to common pool of hardware
Power	Apply power management policy
Diagnostics	Centralize information available in system
Communications	Share access to external communications device
Multiple Chiplet FDIR	Coordinate FDIR activity across chiplets
Single Chiplet FDIR	Coordinate FDIR activity within chiplet

This is a non-ITAR presentation, for public release and reproduction from FSW website.

Middleware Schedule / Milestones



System Requirements Review — May 2018

Middleware R1 Release — February 2019

Middleware R2 Release — October 2019

Architecture Review — December 2019

Middleware R3 Release — Mid 2020

Middleware R4 Release — Early 2021

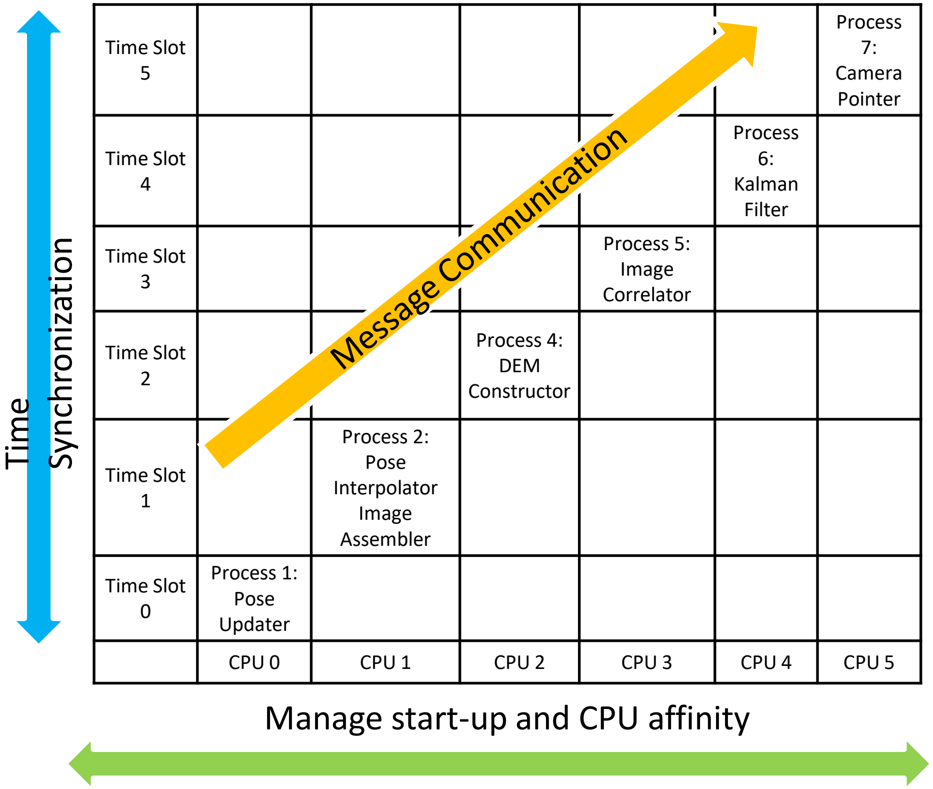
This is a non-ITAR presentation, for public release and reproduction from FSW website.

Middleware Release 2

- Middleware Release 2 was completed in October 2019
- Uses the HPSC System Software Release 4 from ISI:
  - RTPS/R52 (AArch32) and HPPS/A53 (AArch64) tool chains
  - QEMU emulation of HPPS and RTPS to support MW testing and demonstration
  - RTEMS in RTPS/R52 and Yocto Linux for HPPS/A53 to run MW applications
  - System software drivers to implement MW use cases (Watchdog Timer (WDT), mailbox, shared memory)
- Built upon the capabilities from R1
  - Implemented Application Programming Interfaces (APIs) for 10 of 13 defined Middleware (MW) service libraries
    - ➢ Started implementation on 5 new service libraries
    - ➢ Improved on 5 existing service libraries
    - ➢ Not all functions defined for the 10 services are implemented
    - ➢ Not all MW service libraries are ported to R52/RTEMS yet
- Worked on two reference missions using MW functions for execution in QEMU
  - JPL HPFEC application to use additional MW capabilities
  - Ported the GSFC core Flight System (cFS) framework and sample applications to work with MW on HPPS Yocto Linux and RTPS RTEMS

JPL HPFEC Reference Mission

High Performance, Fault-Tolerant Embedded Computing (HPFEC) Synthetic Application



Time Slot 5						Process 7: Camera Pointer
Time Slot 4					Process 6: Kalman Filter	
Time Slot 3				Process 5: Image Correlator		
Time Slot 2			Process 4: DEM Constructor			
Time Slot 1		Process 2: Pose Interpolator Image Assembler				
Time Slot 0	Process 1: Pose Updater					
	CPU 0	CPU 1	CPU 2	CPU 3	CPU 4	CPU 5

This JPL reference mission models elements found in applications for guidance, navigation and control (GNC) of aircraft and spacecraft.

The reference mission is implemented with six processes that use middleware in the following ways:

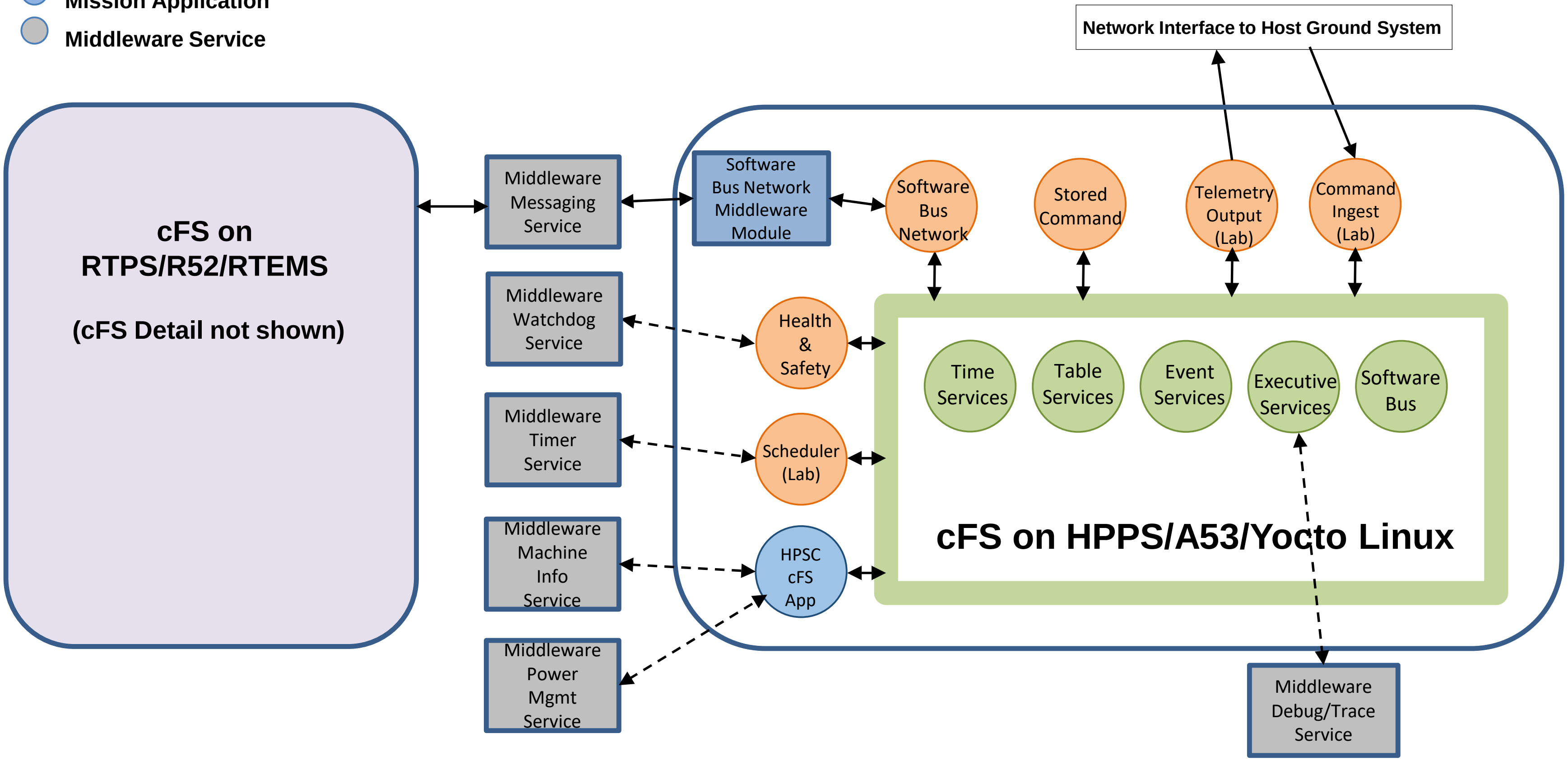
- 🟩 Manage start-up for processes and set CPU affinity
- 🟦 Synchronize process workloads on time slots
- 🟨 Communicate between processes

This is a non-ITAR presentation, for public release and reproduction from FSW website.

GSFC cFS Reference Mission

- 🟩 cFE Core Service
- 🟧 cFS Reuse Application
- 🟦 Mission Application
- ⬜ Middleware Service

Reduced Set of cFS Applications for initial HPSC Integration and Demo



This is a non-ITAR presentation, for public release and reproduction from FSW website.

Middleware Release 3

- Middleware Release 3 planning will be influenced by HPSC Re-plan (Fall 2019)
- Middleware R3 activity will include:
  - Scrub and enhance MW requirements for future releases
  - Specify updated requirements for R3 implementation
  - Design and implement R3 release requirements
    - Fault Tolerance and Redundancy Service
    - Multi-Chiplet communication, timing & synchronization, boot up and initialization
    - API for Deployment Configuration
    - Enhance services from R1, R2
  - Install and exercise delivered capabilities of SSW R5+ releases
    - RTEMS
    - Eclipse IDE

Middleware Release 4 and Beyond

- Middleware Release 4 work to complete
  - Complete all 13 Middleware Service Libraries
  - Complete all 6 Middleware Service Applications
  - Continue to test and integrate new System Software features
  - Continue refining requirements
  - Improve and complete documentation

- Potential deployment Tools*
  - Application Characteristics Template
  - Configuration Checker
  - Interaction Modeling
  - Schedule Generator
- Potential diagnostic Tools*
  - Improved Event Visualization
  - View Live Statistics
  - Blackbox Recorder

*may depend on schedule and/or additional funding

Acknowledgements

- **HPSC Middleware Team:**
  - John Lai/JPL, Brent Morin/JPL, Dennis Kou/JPL, Joseph Kochocki/JPL, Sergio Maldonado/GSFC, Jonathan Wilmot/GSFC
- **HPSC Project Team:**
  - Rich Doyle/JPL, Rafi Some/JPL, Jim Butler/JPL, Irene Bibyk/GSFC, Wes Powell/GSFC
- **Boeing / ISI:**
  - Jon Ballast/Boeing, J.P. Walters/USC-ISI

Acronyms (1)

Acronym	Meaning	Acronym	Meaning
AFRL	Air Force Research Laboratory	**DRAM**	Dynamic Random Access Memory
AMBA	Advanced Microcontroller Bus Architecture	**FCR**	Fault Containment Region
API	Application Programmer Interface	**FDIR**	Fault Detection, Isolation, and Recovery
ARM	Advanced RISC Machines	**FPGA**	Field programmable Gate Array
ATF	Arm Trusted Firmware	**FPU**	Floating Point Unit
BIST	Built In Self Test	**GNC**	Guidance Navigation and Control
BSP	Board Support Package	**GOPS**	Giga Operations Per Second
C&DH	Command and Data Handling	**GPIO**	General Purpose Input Output
cFS	core Flight System	**GPU**	Graphics Processing Unit
CPU	Central Processing Unit	**GSFC**	Goddard Space Flight Center
DDR	Double Data Rate	**HEO**	Human Exploration and Operations
DMIPS	Dhrystone Millions of Instructions per Second	**HPFEC**	High Performance Fault-tolerant Embedded Computing
DMR	Dual Modular Redundancy	**HPPS**	High Performance Processing Subsystem

Acronyms (2)

Acronym	Meaning	Acronym	Meaning
HPSC	High Performance Spacecraft Computing	**MIE**	Mission Integration Engineer
HW	Hardware	**MIPS**	Millions of Instructions per Second
I/O	Input / Output	**MMU**	Memory Management Unit
I2C	Inter-Integrated Circuit	**MPI**	Message Passing Interface
IDE	Integrated Development Environment	**MRAM**	Magnetoresistive Random Access Memory
IPL	Initial Program Loader	**MW**	Middleware
ISA	Instruction Set Architecture	**NAND**	NOT-AND logic
ISI	Information Sciences Institute	**NASA**	National Aeronautics and Space Administration
ITAR	International Traffic In Arms Regulations	**NEON**	Single Instruction Multiple Data architecture
JPL	Jet Propulsion Laboratory	**NVRAM**	Non Volatile Random Access Memory
JTAG	Joint Test Action Group (Debug interface)	**PCIe**	Peripheral Component Interconnect express
KVM	Kernel Based Virtual Machine	**QEMU**	Quick Emulator
MB	Megabyte	**R1, R2, R3**	Release 1, Release 2, Release 3

This is a non-ITAR presentation, for public release and reproduction from FSW website.

Acronyms (3)

Acronym	Meaning	Acronym	Meaning
RTEMS	Real Time Executive for Multiprocessor Systems	**SW**	Software
RTOS	Real Time Operating System	**TBD**	To Be Determined
RTPS	Real Time Processing Subsystem	**TMR**	Triple Modular Redundancy
SCP	Self Checking Pair	**TRCH**	Timing, Reset, Configuration, and Health
SIMD	Single Instruction Multiple Data	**TTE**	Time Triggered Ethernet
SMD	Science Mission Directorate	**UART**	Universal Asynchronous Receiver Transmitter
SPI	Serial Peripheral Interface	**USC**	University of Southern California
SPW	Spacewire	**VMC**	Vehicle Management Computer
SRAM	Static Random Access Memory		
SRIO	Serial Rapid Input Output		
SSR	Solid State Recorder		
SSW	System Software		
STMD	Space Technology Mission Directorate		

This is a non-ITAR presentation, for public release and reproduction from FSW website.