Use of CCSDS File Delivery Protocol (CFDP) in NASA/GSFC's Flight Software Architecture: core Flight Executive (cFE) and Core Flight System (CFS)

Jonathan Wilmot
Software Engineering Division
NASA/Goddard Space Flight Center
Jonathan.J.Wilmot@nasa.gov
301-286-2658

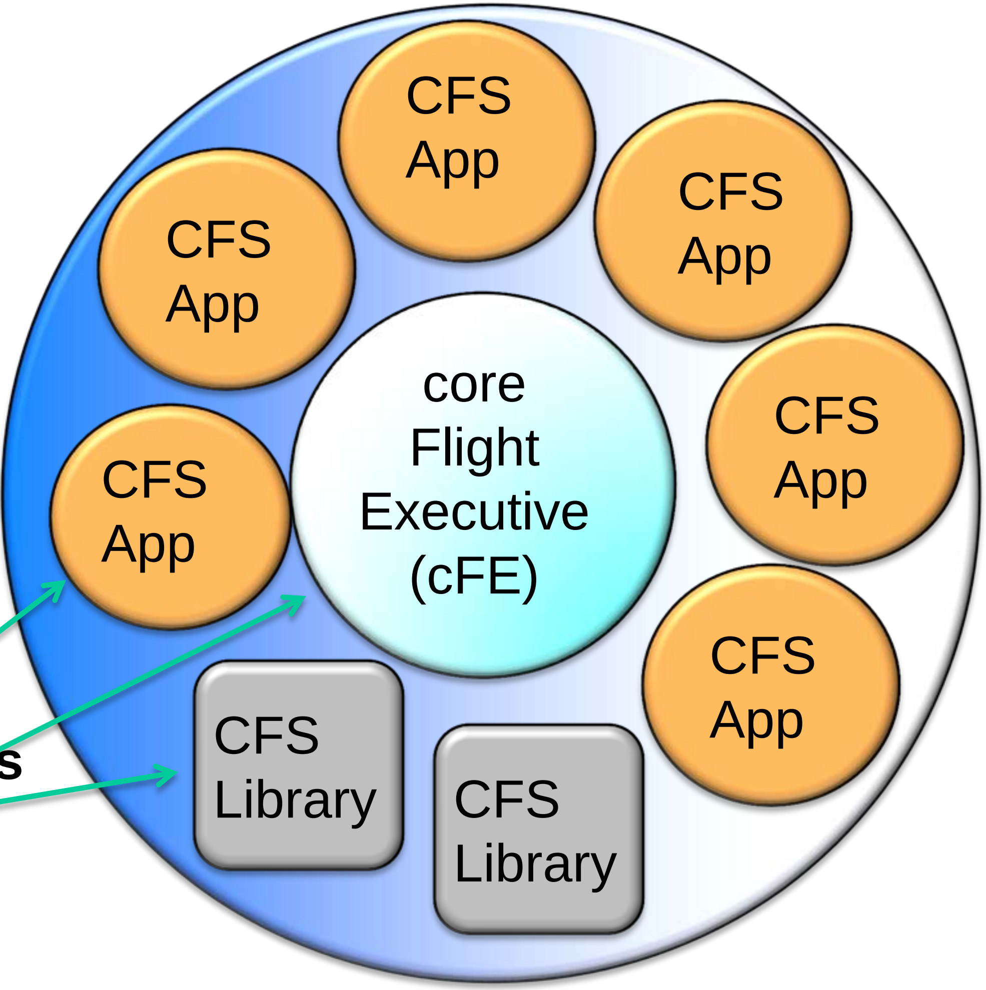
Mission Background

- **cFE/CFS is used on many missions across NASA, Goddard Space Flight Center (GSFC), Johnson Space Center (JSC), Ames Research Center (ARC), Johns Hopkins Applied Physics Laboratory (APL), Glenn Research Center (GRC), and now recently Kennedy Space Center (KSC)**
  - cFE/CFS is baseline for GSFC in-house missions
  - cFE is baseline for APL missions (*Radiation Belt Storm Probes* launched 8/2012)
  - Launched in June 2009, Lunar Reconnaissance Orbiter (LRO) uses cFE/CFS and CFDP
  - Upcoming GSFC missions, Global Precipitation Measurement (GPM) and Magnetospheric Multiscale (MMS) use cFE/CFS and CFDP
    - Currently in environmental testing
- **File systems are required on missions using cFE**
- **All cFE missions have selected the option of also storing science data in a file system**
- **CFDP was selected as the standard on-orbit spacecraft file transfer mechanism**
- ***James Webb Space Telescope (JWST) uses files and CFDP***
  - Architecture predates cFE/CFS

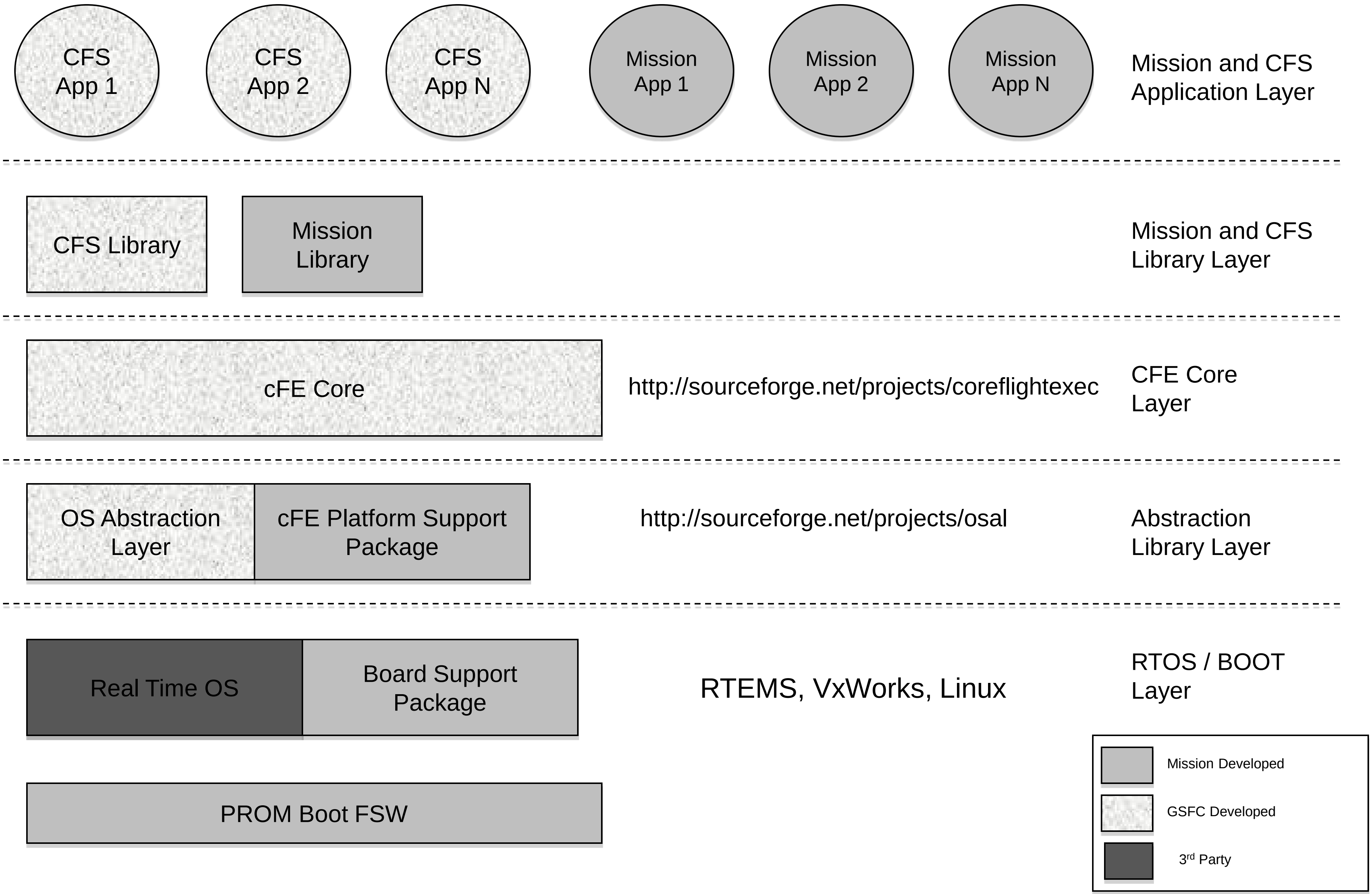
cFE/CFS Introduction

- **Core Flight System (CFS)**
  - A Flight Software Architecture consisting of the cFE Core, CFS Libraries, and CFS Applications
- **core Flight Executive (cFE)**
  - A framework of *mission independent, re-usable, core* flight software services and operating environment
- **For cFE/CFS, each element is a separate loadable file**

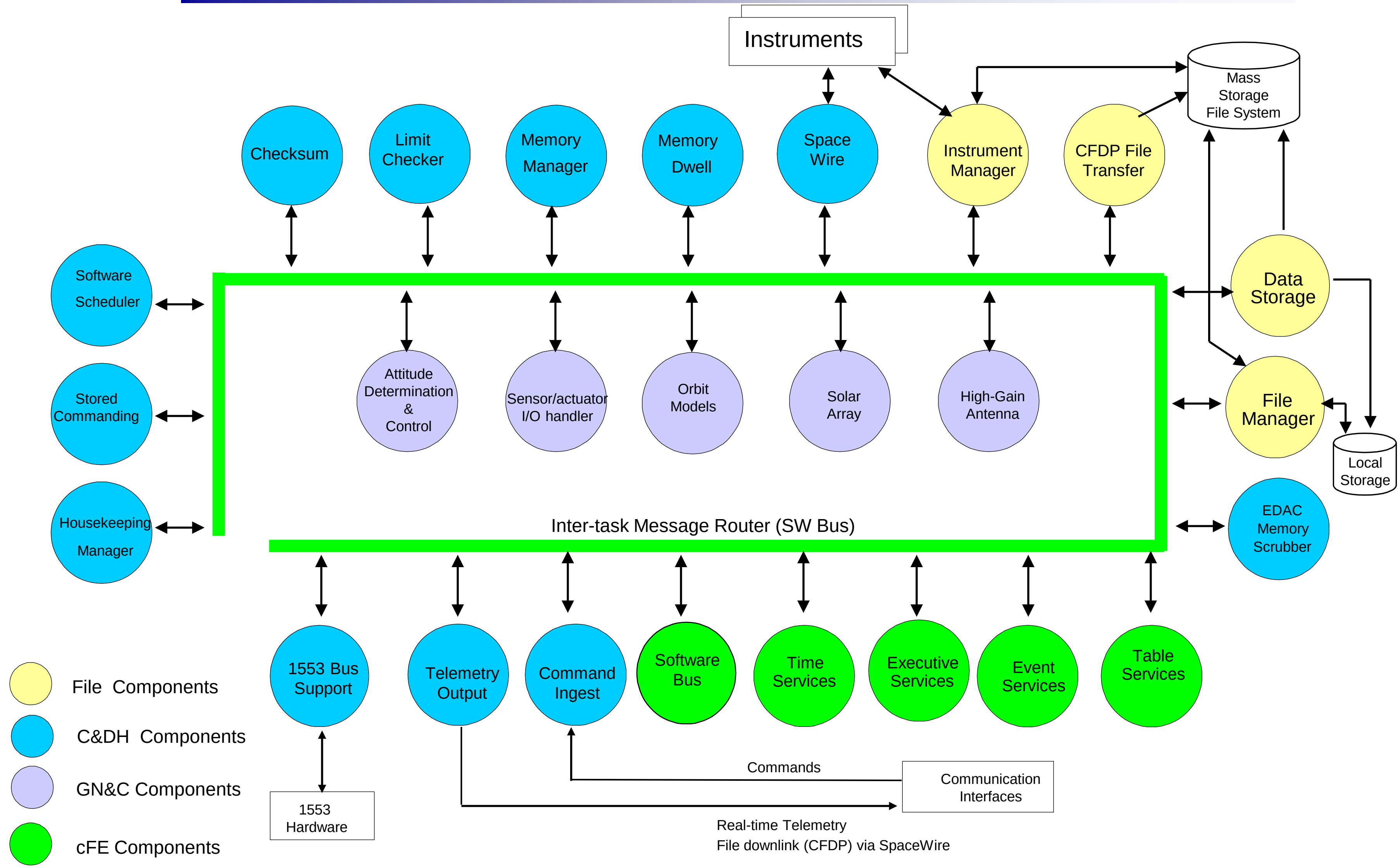
Core Flight System (CFS)



CFS Flight Software Layers



Exemplar GSFC Flight Software Architecture



Note - Some connection omitted for simplicity

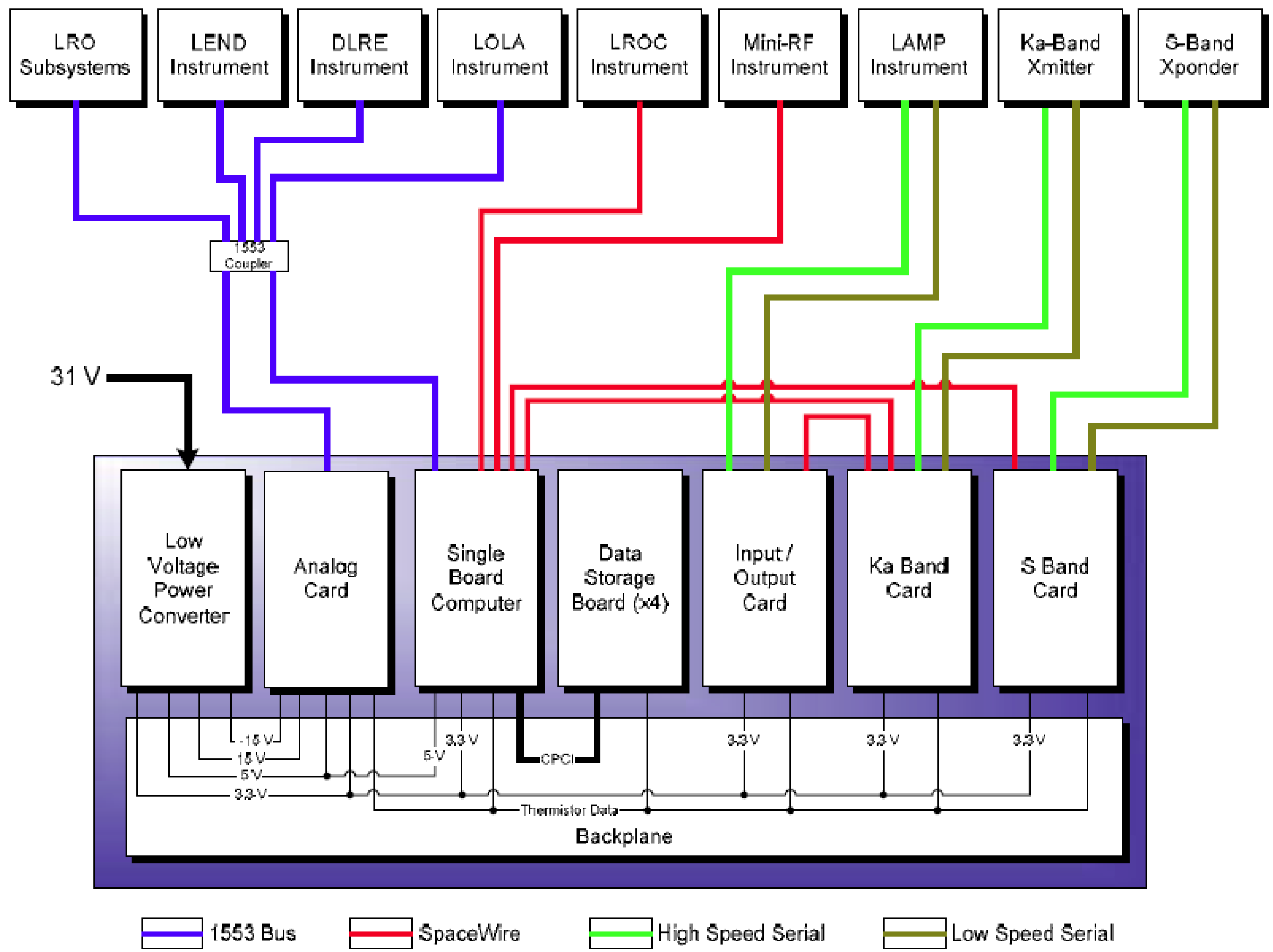
On-Board File Use Cases

- **System Boot**
  - Selects boot image files from a very simple file system
- **Reloadable configuration data (cFE Tables)**
  - Multiple files can be mapped to each table for use during mission modes
- **Engineering data storage**
  - Telemetry, diagnostic data, configuration data, logs
- **Software development and on orbit maintenance**
  - Software component loads
- **Science data storage**
  - High speed bulk data recorders
- **File transfer mechanisms**
  - Trivial File transfer Protocol (TFTP)
    - System booting in labs
  - File Transfer Protocol (FTP)
    - Component loading and development in labs, both uplink and downlink
  - CCSDS File Delivery Protocol (CFDP)
    - Flight system data transfers, both uplink and downlink

Onboard File System Types and Media

- cFE ROM/EEPROM
  - Simple, few hundred lines of code: All cFE file APIs
  - No block device layer
    - All files are contiguous in memory
    - Flat, no sub directories
  - Variable size files only to a maximum predefined size
  - Simple block device layer and RAM driver expected in next version
- RAM
  - RTEMS File System (RFS)
    - Layered on cFE Block Device driver
  - VxWorks DOS (with modifications for Mars Exploration Rover type issues)
    - VxWorks Block Device Driver
  - JPL uses internally developed POSIX RAMFS
- Solid State Recorders (SSR) (both RAM and EEPROM)
  - Requires CFS device specific SSR Block Device driver
    - Hardware typically does not allow direct address mapping
  - RTEMS File System (RFS)
  - VxWorks DOS (with modifications for Mars Exploration Rover type issues)

Lunar Reconnaissance Orbiter (LRO) C&DH Architecture



Public source http://klabs.org/DEI/Processor/PowerPC/rad750/papers/LROCDHAIAA092008.pdf

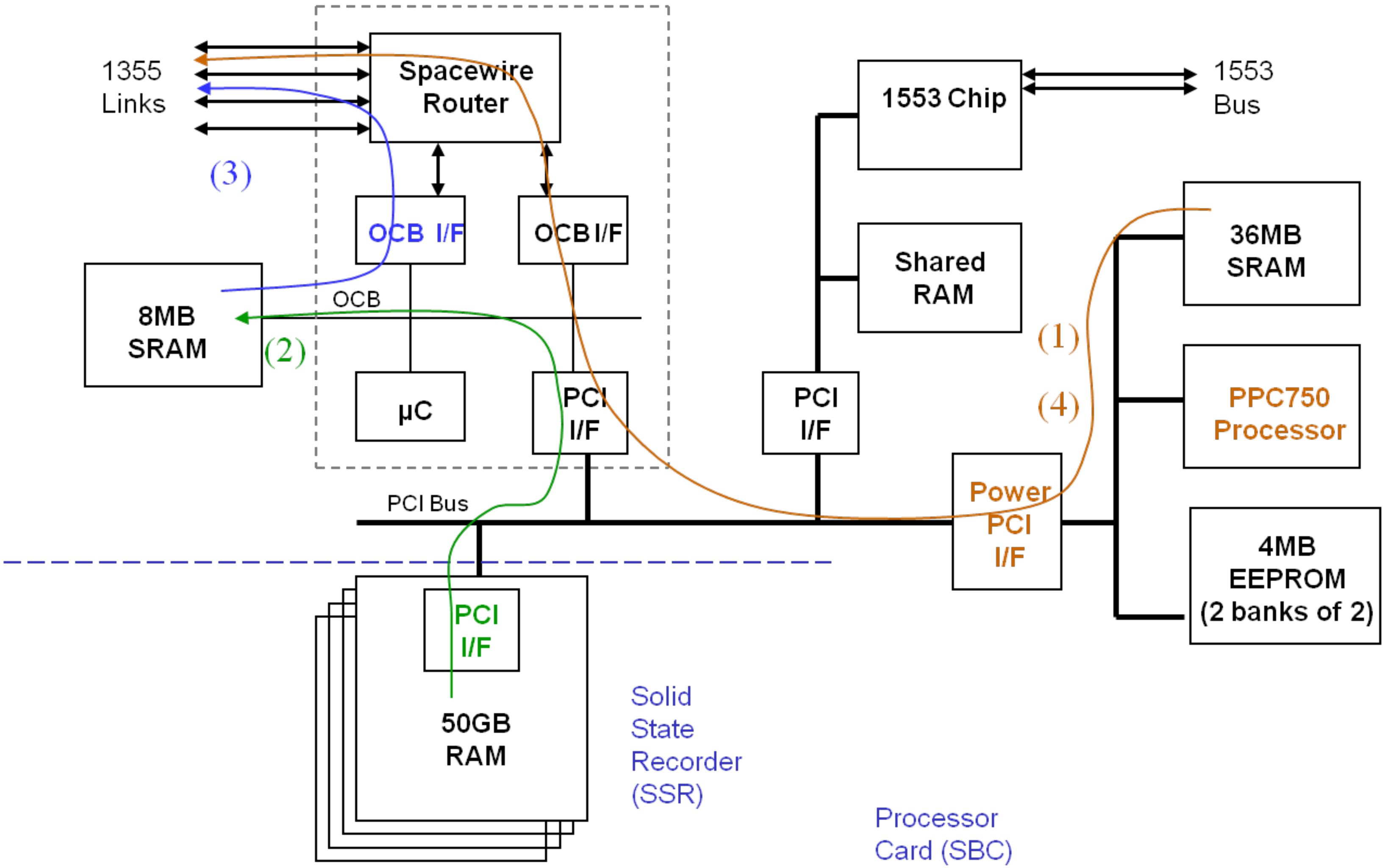
LRO CFDP Implementation

- **All nominal file transfers use CFDP Class 2: reliable**
- **Science file transfers at 100Mbps handled by DMA engine from the Solid State Recorder memory to SpaceWire memory**
  - Software manages a Protocol Data Unit (PDU) send queue for DMA engine
  - Science file PDU retries handled in software
  - Other data files can be interleaved
- **Software on main processor implements CFDP protocol state machines and sends meta data, FIN, EOF, NAC, and ACK PDUs**
- **Slower engineering data file transfers done entirely in software**
- **File manager application determines file transfer priority**
- **This approach has been used for the *Global Precipitation Measurement and Magnetospheric Multiscale missions***

LRO Implementation: CFDP File Downlink

- **(1) CFDP application code begins file transfer**
  - Opens next file from downlink queue
  - Creates and sends CFDP meta-data PDU
  - Call disk driver read() for first 1MB of file data
- **(2) Disk device driver copies data to SpaceWire RAM**
  - Uses PCI DMA for the transfer
  - Returns control to CFDP app. after transfer is complete
- **(3) CFDP app signals µC with header info and data address**
  - µC formats and sends CFDP data PDUs
  - Uses SpaceWire flow control to keep downlink card full
  - Signals CFDP application when the full 1MB has been sent
  - In parallel, CFDP app has already read the next 1MB from the file
- **(4) CFDP app completes file transfer**
  - Creates and sends CFDP EOF PDU and any NAK responses

LRO Implementation: CFDP File Downlink



Ground System Perspectives (1)

- **CFDP can be terminated at the ground station, MOC, or both**
  - LRO ground station passes PDU ack/naks back to the MOC for uplink and bandwidth management
- **CFDP use on GPM is more complex due to periods of no-contact, bi-directional contact, and uni-directional contact**
  - Unlike LRO, GPM has had to automate the dynamic management of CFDP timers to accommodate the asymmetry and variable data rates
- **File based checksum provides an end to end data check that was not available in the previous data systems**
- **LRO currently discards incomplete files in the MOC that are still missing PDUs after timers have expired.**
  - This is a policy based on the mission science return requirements.
- **Having a dynamic file system on GPM instead of a Tropical Rainfall Measuring Mission (TRMM) type partitioned recorder, has allowed for more granular quality of service recorder management**
  - A single file can be kept instead of retaining a whole partition

Ground System Perspectives (2)

- **Previous systems would dump the data recorder partitions twice, or more, to improve the odds of getting complete data sets**
  - Impacts bandwidth costs
  - Increases contact times
  - Increases operations costs
- **Science data files and CFDP transfers reduced the complexity of the GPM Mission Operations Center (MOC). It removed 2 historical MOC subsystems in older missions:**
  1. Packet processing (PACOR) subsystem analogue like in TRMM performing Level 0 packet processing, sorting, ordering, etc
  2. Mission specific automation components to track missing packets and potentially send retransmits
- **Mission specific analysis must be done to balance file size, bit error rate, number of open files, link asymmetry**
  - CFDP and the onboard file manager support the variability

Observations/Conclusions (1)

- **File systems are used extensively in modern flight data systems and provide familiar developer and operational views**
- **There are benefits still to be realized with onboard file system implementations**
  - Bad block re-mapping, device wear leveling
- **File based data recorders offer significant flexibility for operations and science data centers**
- **Current OS file system implementations contain significant overhead and layers for buffering, virtual memory management, and rotational media**
  - GSFC, JPL, and others now have simpler and more robust implementations to target RAM and EEPROM/Flash
- **There are still questions to answer with file system use on partitioned and/or distributed systems as well as NASA Class A safety critical systems**
  - Study projects are underway at GSFC, JSC, and APL
- **To meet high data rate requirements CFDP must be tightly coupled to the on-board file system implementation**
  - High data rates need FPGA and/or micro controller support
  - This is not just CFDP but any high speed data transfer protocol

Observations/Conclusions (2)

- **CFDP provides a standard protocol to reliably transfer files over higher bit-error links**
  - Ground systems have one implementation that is reused mission to mission
- **CFDP works well over highly asymmetric links**
- **CFDP is only the file transfer part of file system operations and management**
  - Other tools are used to manage recorder allocations, data priorities, and interleaving
  - GSFC has standardized these tools for multi-mission use
- **File based data recorders offer significant flexibility for operations and science data centers**
- **CFDP has been shown to deliver more complete science data over a shorter contact time**
  - This is mission dependant based on the mission requirement for science data return (90% to 99%)
- **File systems and CFDP have reduced operations cost:**
  - On-board “level 0” processing (packet processing, sorting, ordering, etc)
  - lower operations complexity
  - Increase automation (Eliminated an 8 hour 7day a week shift, TRMM - GPM)

Backup

cFE Core - Overview

- **A set of** *mission independent, re-usable, core* **flight software services and operating environment**
  - Provides standardized Application Programmer Interfaces (API)
  - Supports and hosts flight software applications
  - Applications can be added and removed at run-time (eases system integration and FSW maintenance)
  - Supports software development for on-board FSW, desktop FSW development and simulators
  - Supports a variety of hardware platforms
  - Contains platform and mission configuration parameters that are used to tailor the cFE for a specific platform and mission.