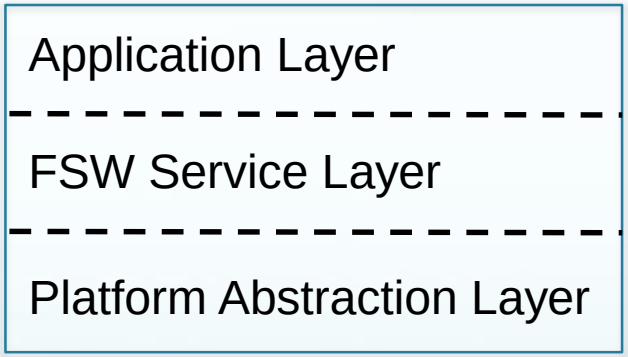
core Flight System (cFS)

A Low Cost Solution for SmallSats

David McComas
Susanne Strege
Jonathan Wilmot
NASA Goddard Space Flight Center
Software Engineering Division Flight
Software Systems Branch

What is the core Flight System?

- The cFS is a re-usable spacecraft flight software architecture and software suite that is both platform and project independent
- Layered architecture and compile-time configuration parameters make it scalable and portable to a wide range of platforms



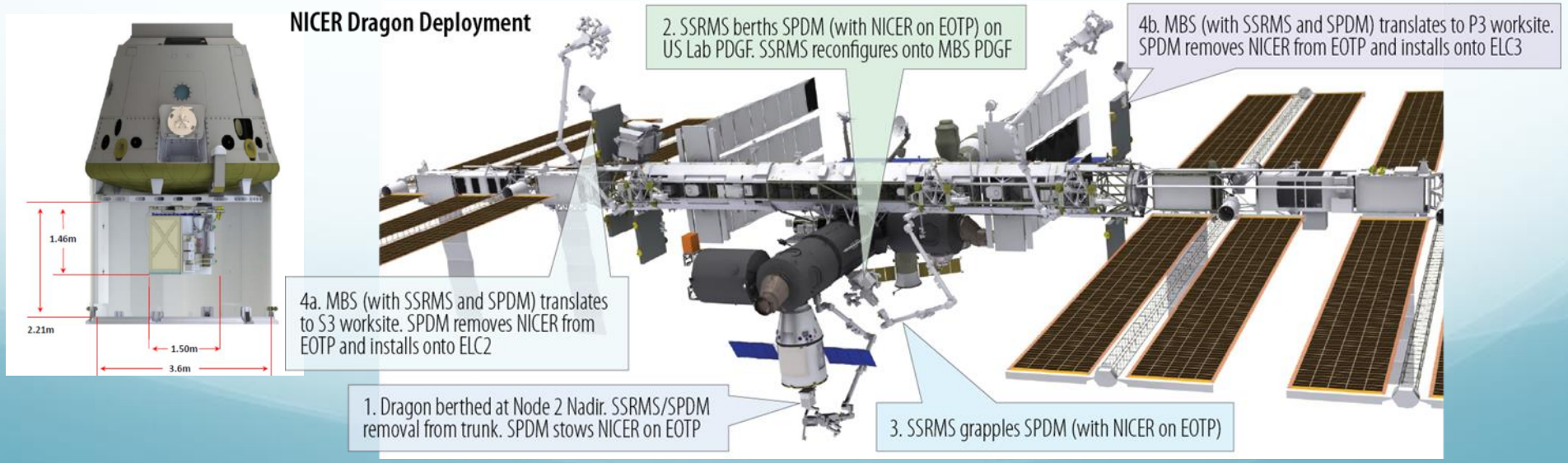
- Original product created by NASA's Goddard Space Flight Center
- The FSW Service and Platform Abstraction layers are now controlled by a NASA multi-center configuration control board

Recent cFS Success Stories

- Johnson's Morpheus: 14 months from concept to flight test in 2010
- Goddard's Class B missions: Global Precipitation Measurement (GPM) launched February 2014 and Magnetospheric Multscale (MMS) launched March 2015
- Goddard's 2014 Class D balloon mission: Observatory for Planetary Investigations from the Stratosphere (OPIS)
  - Baseline command and data handling software was up an running on the target platform (Intel Core Duo/Xenomai) within a month and launched 6 months later
- DARPA's F6 program: Emergent funded (2013-2014) to develop Flight Software to Provide Autonomous Satellite Cluster Services
  - Cluster Flight System applications ported to cFS in less 6 months and formally demonstrated in simulation test bed

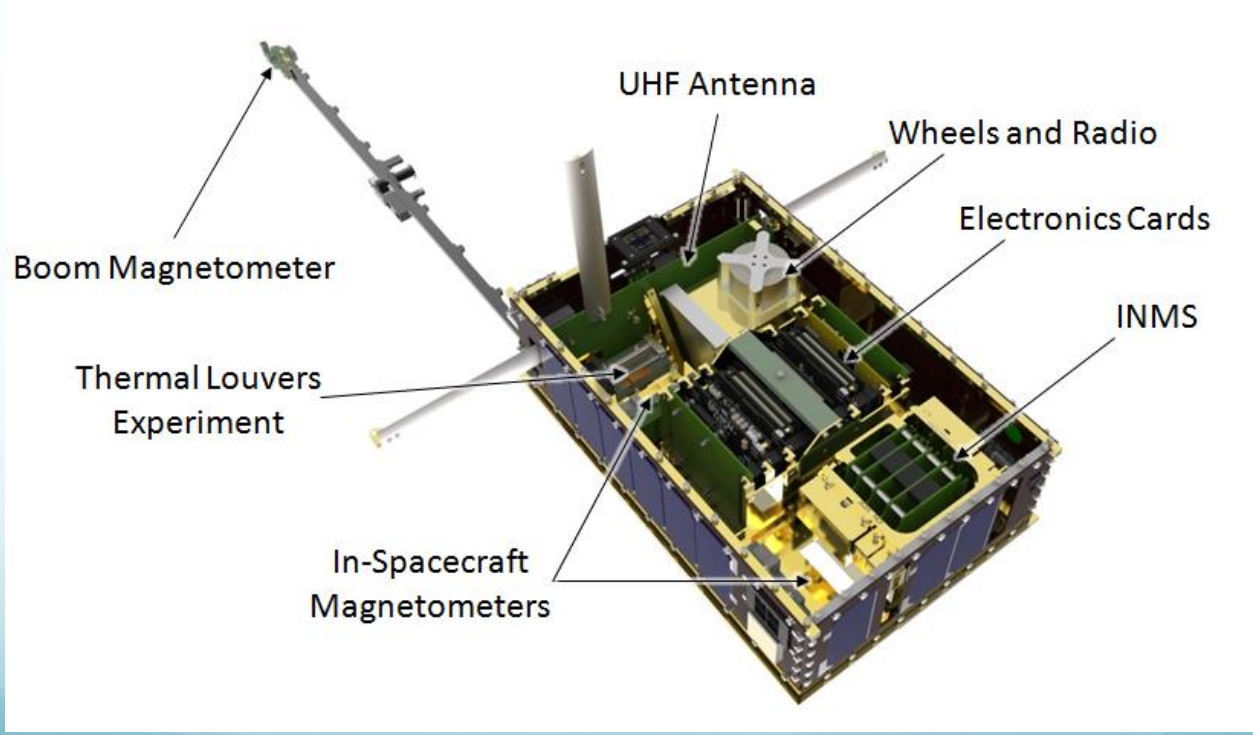
In Development - NICER

- Objectives
  - Reveal the nature of matter in the interiors of neutron stars
  - Uncover the physics of dynamic phenomena associated with neutron stars
  - Determine how energy is extracted from neutron stars
- On-board Processor
  - Broad Reach Engineering Radiation Hardened BRE440 PowerPC
    - 32 Bit RISC embedded processor
    - 83 MHz OSC (2 MIPS / MHz = ~166 MIPS)
  - VxWorks 6.7



In Development - Dellingr

- Objectives
  - Low cost science and technology demonstration
- On-board Processor
  - ARM7
    - 40 Mhz, 2Mb RAM
  - FreeRTOS



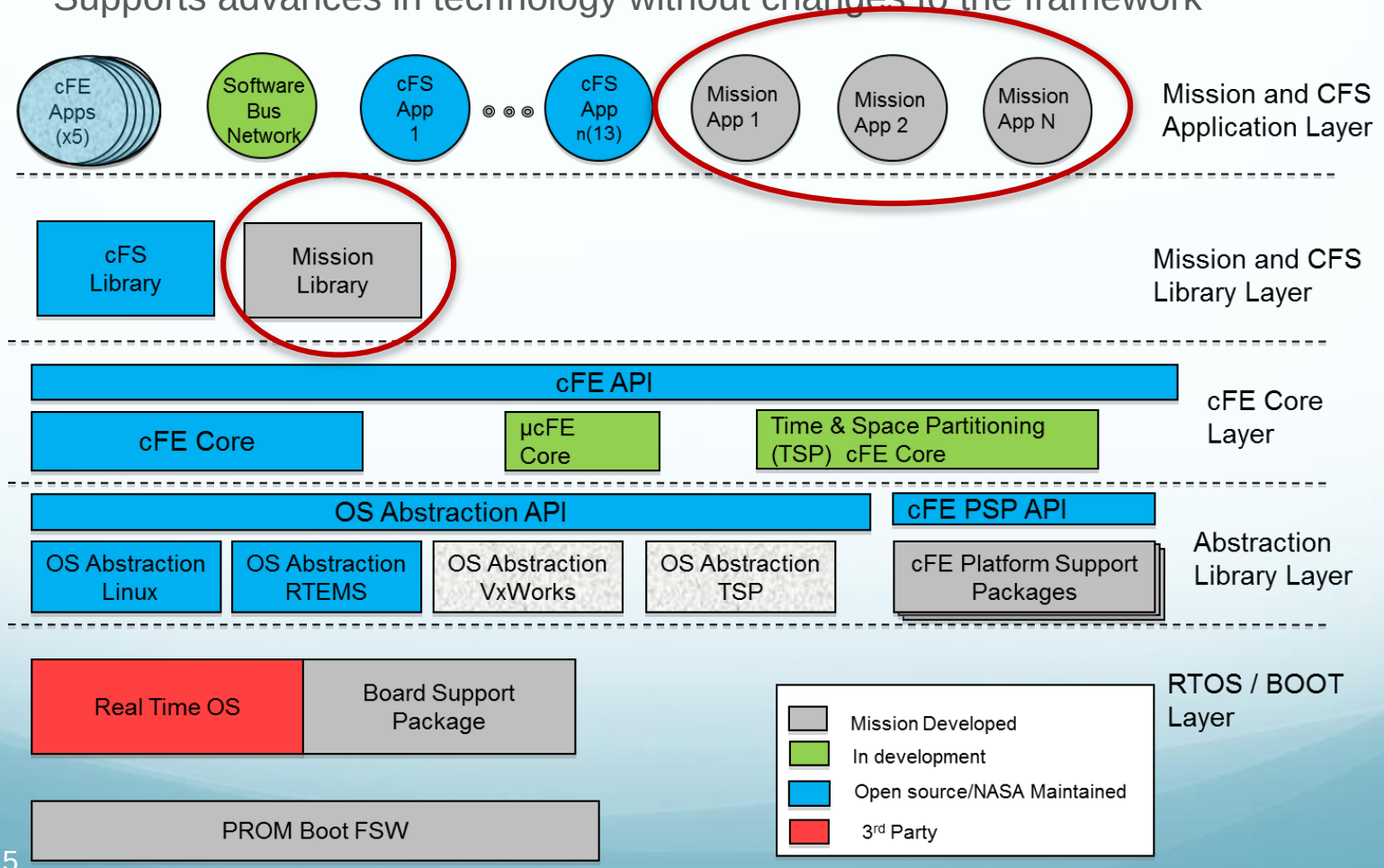
In Development - PiSat

- Objectives
  - Low cost test bed
- On-board Processor
  - Raspberry Pi
  - Raspberry Pi OS (DEBIAN/Linux)

cFS Architecture Highlights

cFS Key Features

- Layered architecture
  - Reusable components
  - Platform Independent
  - Supports advances in technology without changes to the framework



cFS Core Services

Executive Services

- Manages the software system

Software Bus Services

- Provides publish/subscribe software bus messaging interface

Time Services

- Provides spacecraft time

Event Services

- Provides interface for sending, filtering, and logging event messages

Table Services

- Provides interface to manage table images

The cFS core layer is the system glue. It provides the common software functions that are needed by all missions.

cFS Applications

Application	Function
CFDP	Transfers/receives file data to/from the ground
Checksum	Performs data integrity checking of memory, tables and files
Command Ingest Lab	Accepts CCSDS telecommand packets over a UDP/IP port
Data Storage	Records housekeeping, engineering and science data onboard for downlink
File Manager	Interfaces to the ground for managing files
Housekeeping	Collects and re-packages telemetry from other applications.
Health and Safety	Ensures that critical tasks check-in, services watchdog, detects CPU hogging, and calculates CPU utilization
Limit Checker	Provides the capability to monitor values and take action when exceed threshold
Memory Dwell	Allows ground to telemeter the contents of memory locations. Useful for debugging
Memory Manager	Provides the ability to load and dump memory.
Software Bus Network	Passes Software Bus messages over Ethernet
Scheduler	Schedules onboard activities via (e.g. HK requests)
Scheduler Lab	Simple activity scheduler with a one second resolution
Stored Command	Onboard Commands Sequencer (absolute and relative).
Telemetry Output Lab	Sends CCSDS telemetry packets over a UDP/IP port

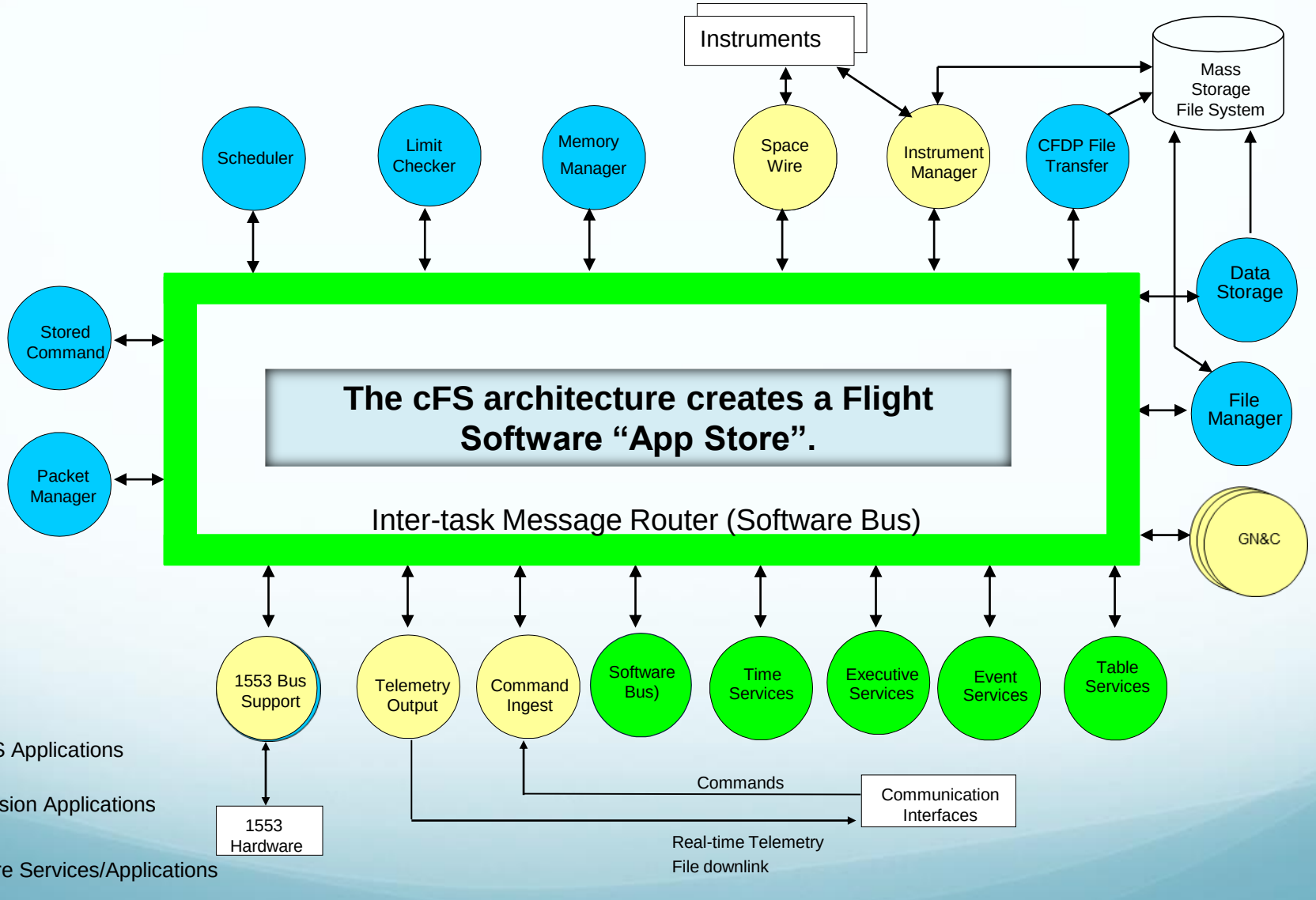
A Complete Engineering Solution

Includes reusable:

- Requirements
- Source Code
- Design Documentation
- Development Standards
- Test Artifacts
- Tools
  - Unit Test Framework
  - Software Timing Analyzer
- User's Guides
  - Application Developers Guide
  - API Reference Guides
  - Deployment Guides
  - Flight Operations Guides
- Command & Telemetry GUI

The CFS architecture reduces Non-Recurring Engineering (NRE) up to 90%

Component Based Architecture



CFS Component Metrics

Component	Version	Logical Lines of Code	Configuration Parameters
Core Flight Executive	6.3.2	12930	General: 17, Executive Service: 46 Event Service: 5, Software Bus: 29 Table Service: 10, Time Service: 32
CFDP	2.2.1	8559	33
Checksum	2.2.0	2873	15
Data Storage	2.3.0	2429	27
File Manager	2.3.1	1853	22
Health & safety	2.2.0	1531	45
Housekeeping	2.4.0	575	8
Limit Checker	2.2.1	2074	13
Memory Dwell	2.3.0	1035	8
Memory Manager	2.3.0	1958	25
Stored Commanding	2.3.0	2314	26
Scheduler	2.2.0	1164	19

- Two scopes of configuration parameters: mission or processor
- Configuration parameters span a large functional range from a simple default file name to a system behavioral definition like the time client/server configuration

Example Mission Code Metrics

Global Precipitation Measurement (GPM)

- Noteworthy items
  - \+ cFE was very reliable and stable
  - \+ Easy rapid prototyping with heritage code that was cFE compliant
  - \+ Layered architecture has allowed COTS lab to be maintained through all builds
  - \- Addition of PSP changed build infrastructure midstream

- Lines of Code Percentages:

Source	Percentage
BAE	0.3
EEFS	1.7
OSAL	2.1
PSP	1.0
cFE	12.4
GNC Library	1.6
CFS Applications	23.5
Heritage Clone & Own	38.9
New Source	18.5

Building a cFS Community

cFS Contributions From Other Organizations

Organization	Contribution	Notes
Johnson Space Center	Trick Simulator integration, Enhanced Build environment, Training materials, ITOS integration, multiple new platforms	
Johnson Space Center	Class A certification of OSAL, cFE and selected cFS applications	Use in Orion Backup flight computer, video processing unit, and Advanced Space Suit
Johnson Space Center	Enhanced Unit tests and increased code coverage, new performance analysis tool	
Glenn Research Center	Code Improvements, modern build environment (cmake), Electronic Data Sheet integration	
Ames Research Center	cFS community configuration management services, continuous integration build services	
Ames Research Center	Simulink Interface Layer for auto-coding cFS applications	
JHU/APL	Multi-Core cFE/OSAL port	Joint IRAD with GSFC, will be used for GSFC MUSTANG flight processor card
DARPA/Emergent	Fractionated Spacecraft / Distributed Mission cFS applications Formation Flying	Part of DARPA F6 project, they hope to make the apps available as open source
Interns and misc contributors	cFS development tools are being created and shared by many organizations Miscellaneous bug fixes reported via open source sites.	

Ongoings

Technical Enhancements

- Integrated Development Environment (IDE)
- Automated tests (unit, functional, build…)
- CCSDS EDS specifications for cFS components
- Integrate Multi-core support into OSAL and cFE
- Integrate/Merge ARINC653 port into OSAL and cFE
- Integrate Dellingr Cubesat FreeRTOS OSAL Port
- Improve scheduler time synchronization
- Expand SB namespace beyond 2^{11}
- Lab upgrades
  - RTEMS 4.11 updates
  - VxWorks 6.9 updates
  - RAD750 simulator
  - MPC8377E: PowerQUICC II Pro Processor test beds
  - LEON3 test bed
  - MCP750 test bed

Operational Enhancements

- Formalize cFS user community
- Web based app store

Back Up

Acronyms

API	Application Programmer Interface
ARC	Ames Research Center
BAT	Burst Alert Telescope
CCSDS	Consultative Committee for Space Data Systems
CDH	Command Data Handling
CFDP	CCSDS File Delivery Protocol
cFE	core Flight Executive
CFS	Core Flight System
CMMI	Capability Maturity Model Integrated
FSW	Flight Software
GLAS	Geoscience Laser Altimeter System
GN&C	Guidance, Navigation, and Control
GPM	Global Precipitation Measurement
GSFC	Goddard Space Flight Center
JSC	Johnson Space Center
LADEE	Lunar Atmosphere and Dust Environment Explorer
LOC	Lines of Code
LRD	Launch Readiness Date
LRO	Lunar Robotic Orbiter
MAP	Microwave Anisotropy Probe
MMS	Magnetic Multiscale Mission
NRE	Non-Recurring Engineering
OSAL	Operating System Abstraction Layer
RBSP	Radiation Belt Storm Probe
RTEMS	Real-Time Executive for Multiprocessor Systems
SAMPEX	Solar Anomalous and Magnetospheric Particle Explorer
SARB	Software Architecture Review Board
SDO	Solar Dynamics Observatory
SMEX	Small Explorer
ST-5	Space Technology 5
SWAS	Submillimeter Wave Astronomy Satellite
TRACE	Transition Region and Coronal Explorer
TRL	Technology Readiness Level
TRMM	Tropical Rainfall Measuring Mission
WIRE	Widearea Infrared Explorer
XTE	X-Ray Timing Explorer

Where is the cFS?

- cFE open Internet access at http://sourceforge.net/projects/coreflightexec/
  - Source code
  - Requirements and user guides
  - Tools
- OSAL open Internet access at http://sourceforge.net/projects/osal/
  - Source code
  - Requirements and user guides
  - Tools
- cFS application suite is also available on sourceforge

Questions? Contact:

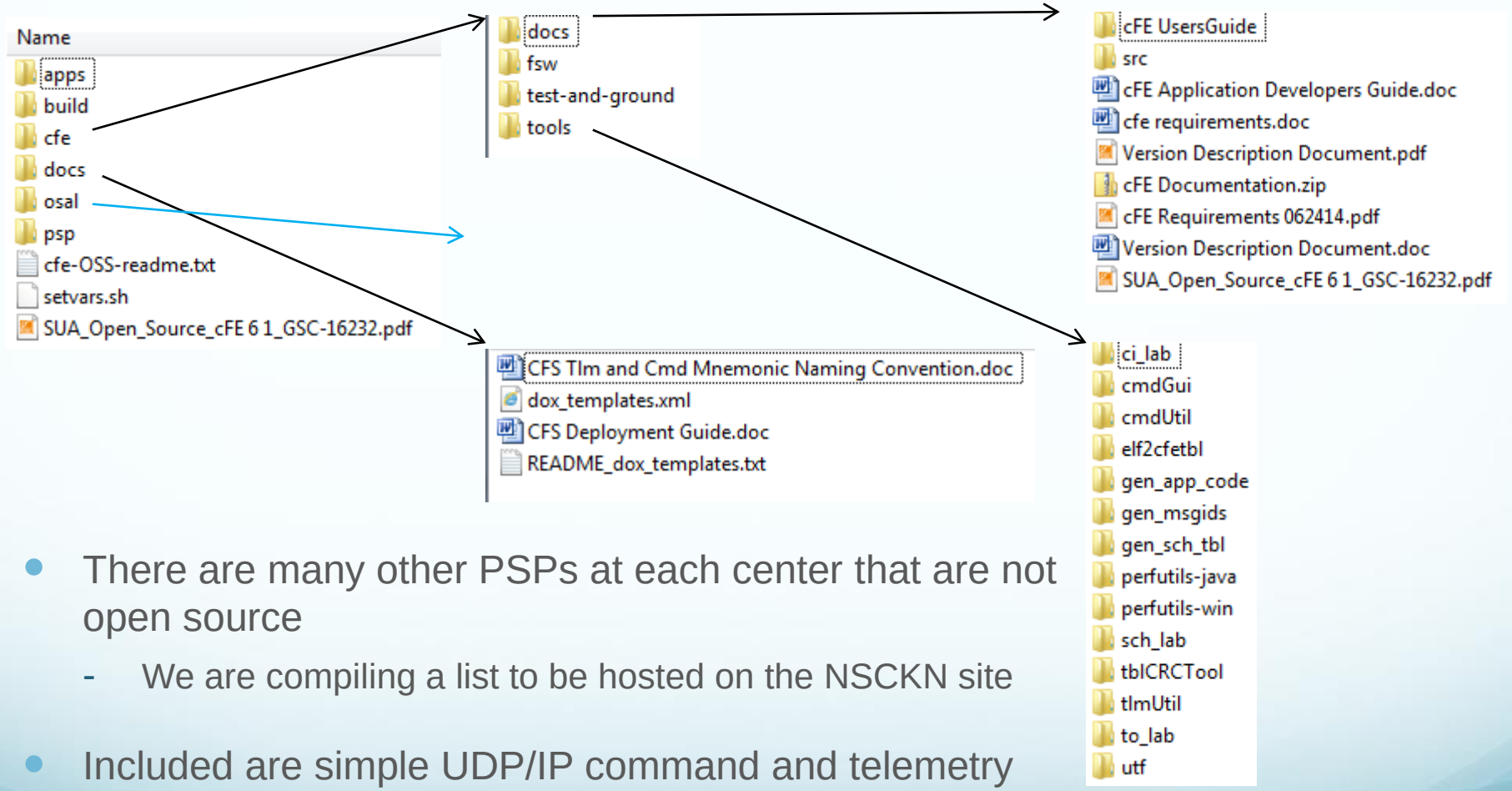
- CFS 582 Branch Management
  - David McComas - NASA GSFC/Code 582 Flight Software Branch
    david.c.mccomas@nasa.gov
    301.286.9038
- CFS Lead Architect
  - Jonathan Wilmot - NASA GSFC/Code 582 Flight Software Branch
    jonathan.j.wilmot@nasa.gov
    301.286.2623
- CFS Product Development Lead
  - Susie Strege - NASA GSFC/Code 582 Flight Software Branch
    susie.strege@nasa.gov
    301.286.3829
- OSAL Product Development Lead
  - Alan Cudmore - NASA GSFC/Code 582 Flight Software Branch
    alan.p.cudmore@nasa.gov
    301.286.5809

Software Facts

Class	Class A, B and lower instantiations
TRL	OSAL & cFE TRL 9, selected cFS Apps TRL 9
CMMI	Certified level 2 for Class B (GSFC) Certified level 3 for Class A (JSC)
Operating Systems	VxWorks, RTEMS, Linux, ARINC 653
Hardware Supported	MCP750, BAE RAD750, Coldfire, LEON3, MCP405, BRE440, and many more at JSC, GRC, ARC, MSFC, and APL
Lines of Code	45K (LOC)
Components available	13
Documentation Available	Requirements, User's Guides, Deployment Guides, Design Documents, Test Plans, Test Reports

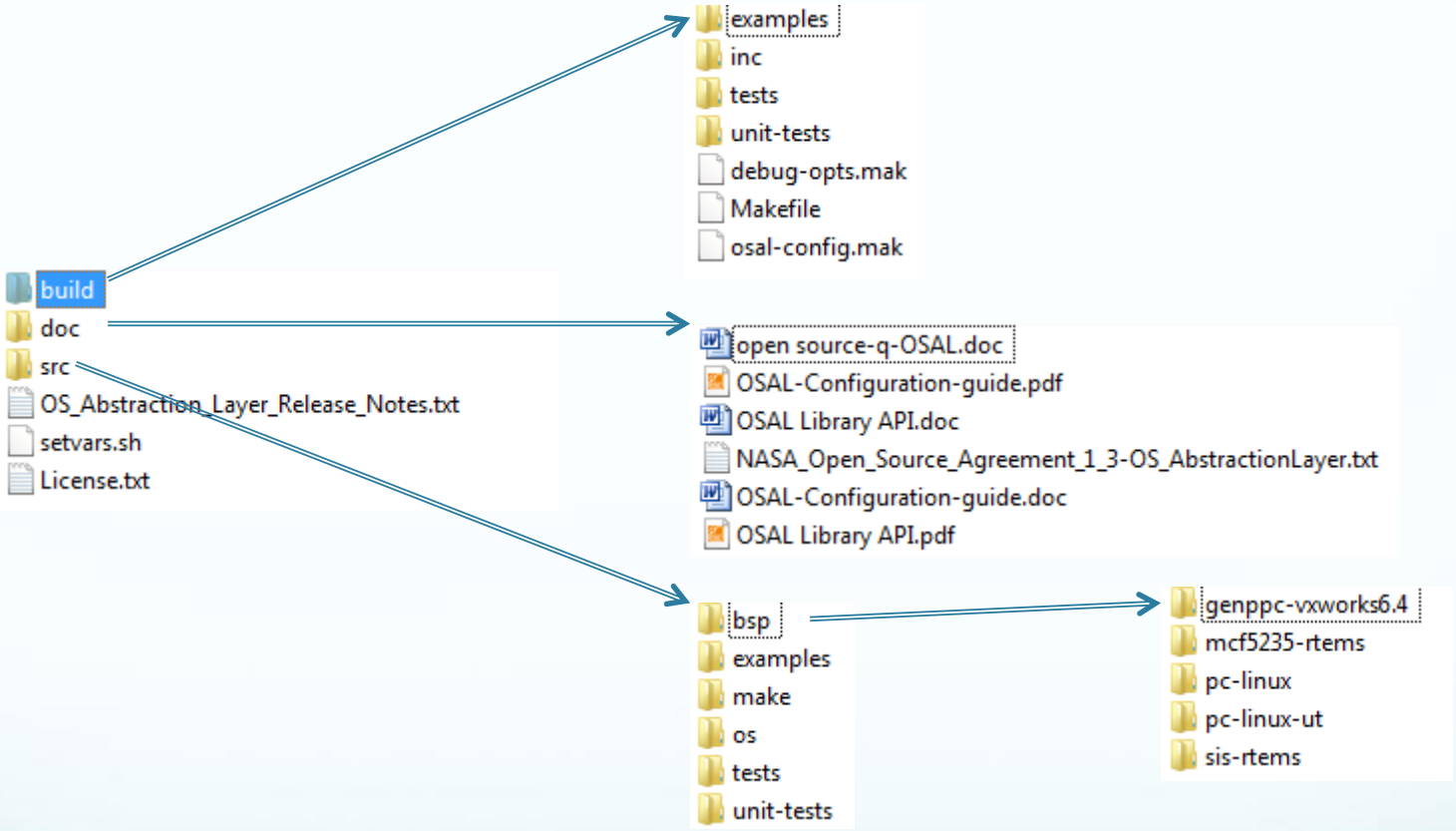
cFS is a software system designed to address software quality and usability issues of performance, reliability, reuse, maintainability, and lifecycle cost.

What's in the in the cFE open source tarball



- There are many other PSPs at each center that are not open source
  - We are compiling a list to be hosted on the NSCKN site
- Included are simple UDP/IP command and telemetry GUIs

What's in the in the OSAL open source tarball



- There are other BSPs at each center that are not open source
  - We are compiling a list to be hosted on the NSCKN site