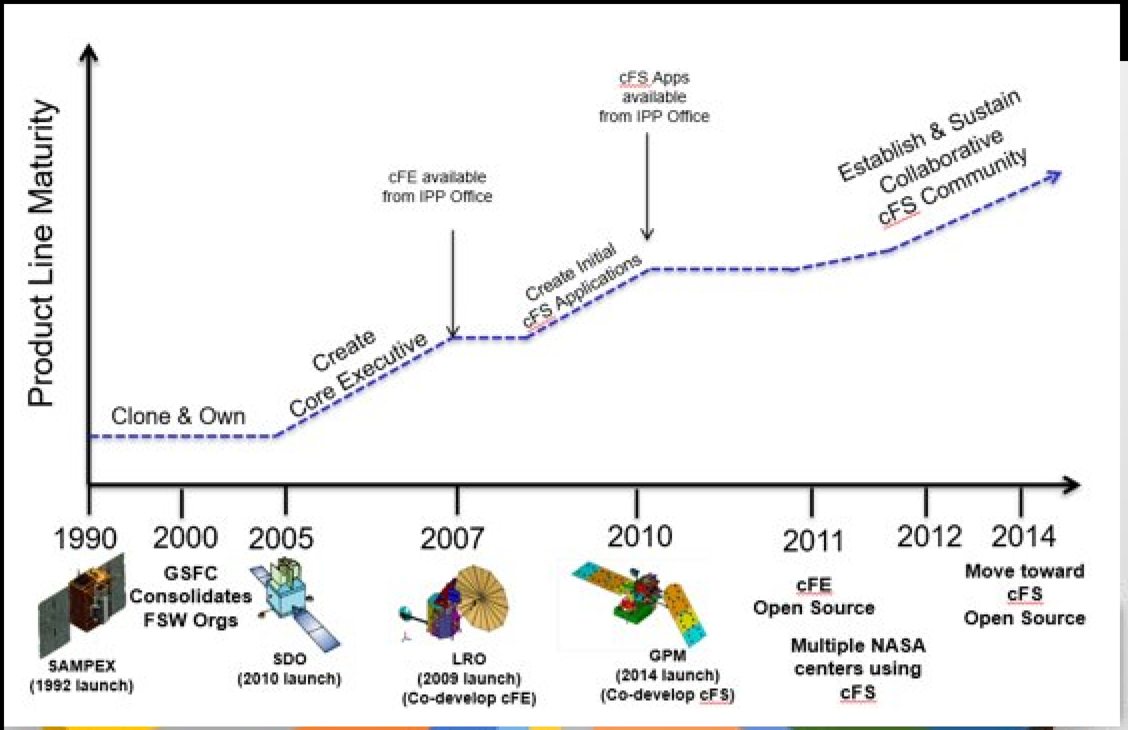
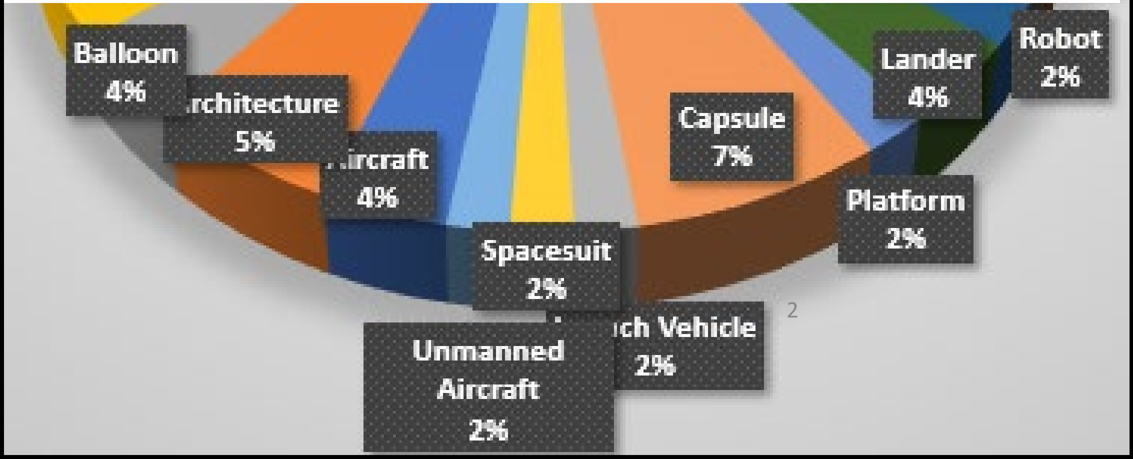
Verbal Introduction

- Welcome to another video in my “software short” series
- Todays topic is called “What is cFS?” “Core Flight System”
- By way of some animated PowerPoint I’m gonna do a breief
  - Intro
  - Go over the CFS layers and components
  - Give Give some exampes on use and architecting software with CFS
  - And provide a ton of links for the code

Introduction – What is cFS

- **Core Flight System** - An Open-Source Reusable Embedded Real-time <Flight> Software Framework
  - Developed by Goddard Spaceflight Center
    - First real-time flight software framework *built specifically for reuse*
    - First Used on LRO Spacecraft (2007), Proliferated Since
  - Why use it? - Save time/$$, reliability, commonality, support tools
  - NASA Agency Asset for Flight Software Reuse
    - Used across NASA on multiple projects (40 here)
      - NASA Projects: BioSentinel, DSA, LADEE, Starling, VIPER, RFC, MPCB AA-2, ASPECT, Orion MPCV, Gateway, Kenobi, Morpheus, Seeker, SPLICE, xEMU, ATLAS, autoNGC, BITSE, CCRS, CeREs, Dellingr, GEDI, GPM, LRO, MMS, Navcube, NICER, OCI, OPIS, OSAM-1, PACE, RST, STF-1, WFI, ICAROUS, LOFTID, SWS, MAV
    - Outside of NASA Use – Open Source
      - DoD and other US Agencies (USSF, AFRL, US Navy, NRL, DARPA)
      - Academia – PennState, Capitol Tech, Embry-Riddle
      - Private Companies – Emergent, Astrobotic, Ten-One, AiTech, OpenSatKit
      - International – ESA, TASA, JAXA, KASI, UAE
  - Key points:
    - Broadly Applicable - Class A Safety-critical across Entire Spacecraft to single embedded science instrument
    - Lightweight and Compatible - cFE ~10KSLOC, Written in C
    - Data-driven, Configurable, Reusable apps, Support tools
    - Extensible – plug-in user apps
    - Multi-platform and operating system support

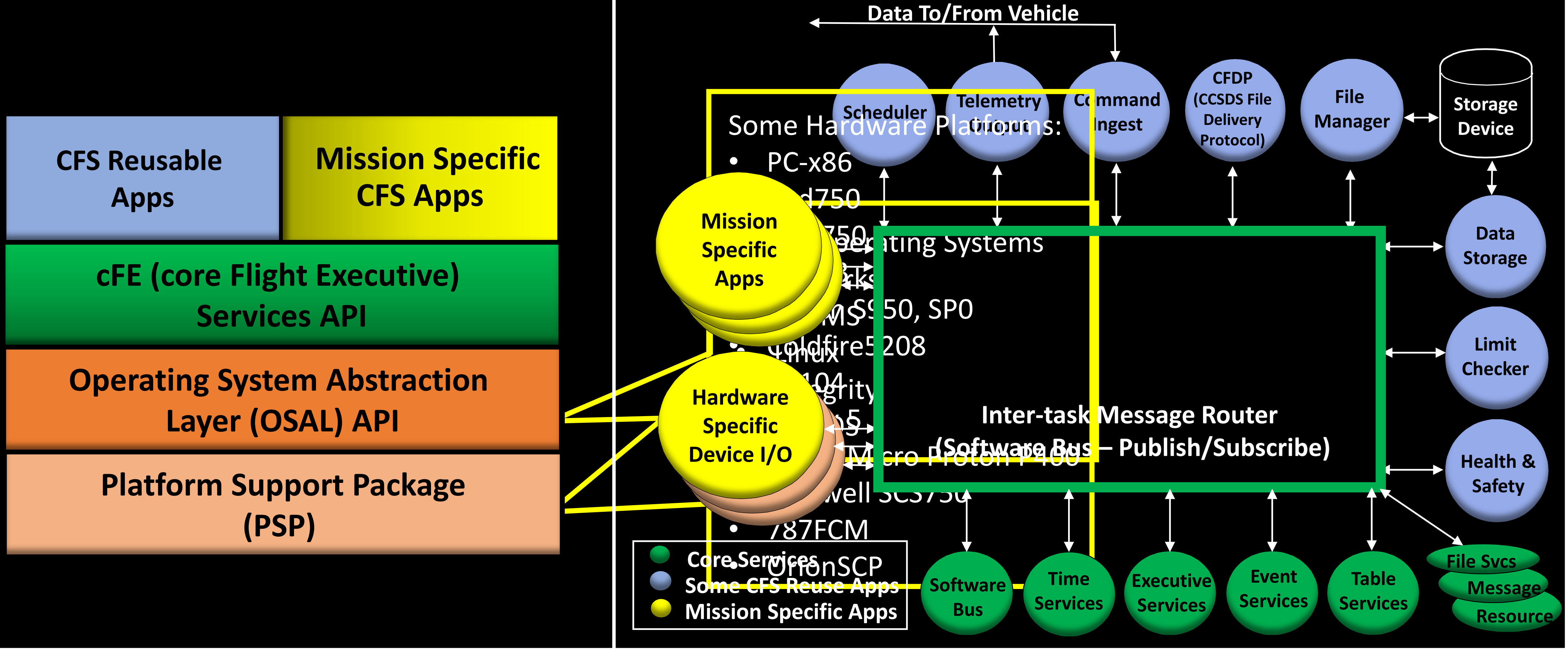




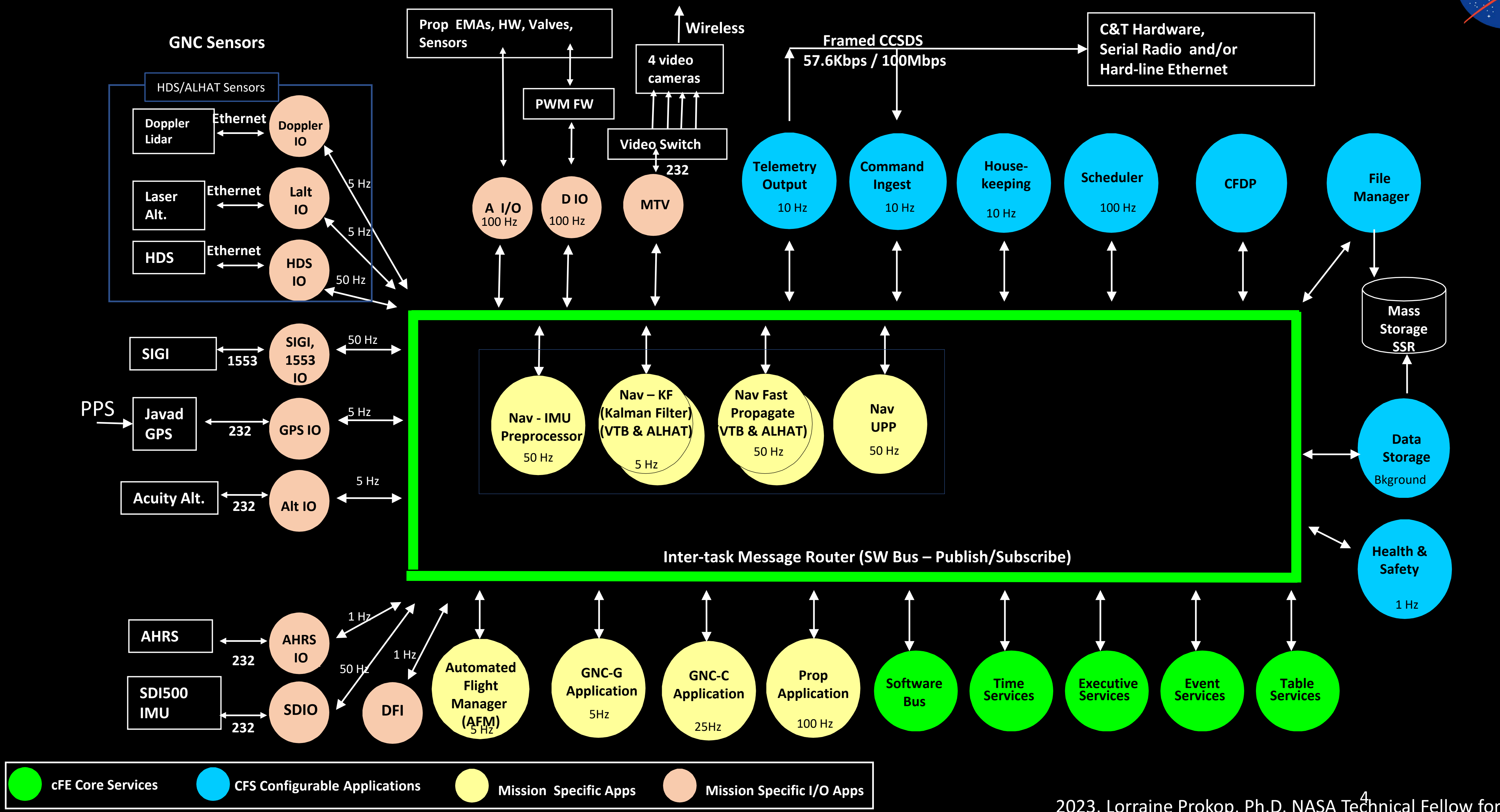
Core Flight System (CFS) Architecture Overview

Core Flight Software Framework- Architectural Layers

Notional CFS Application Software Architecture

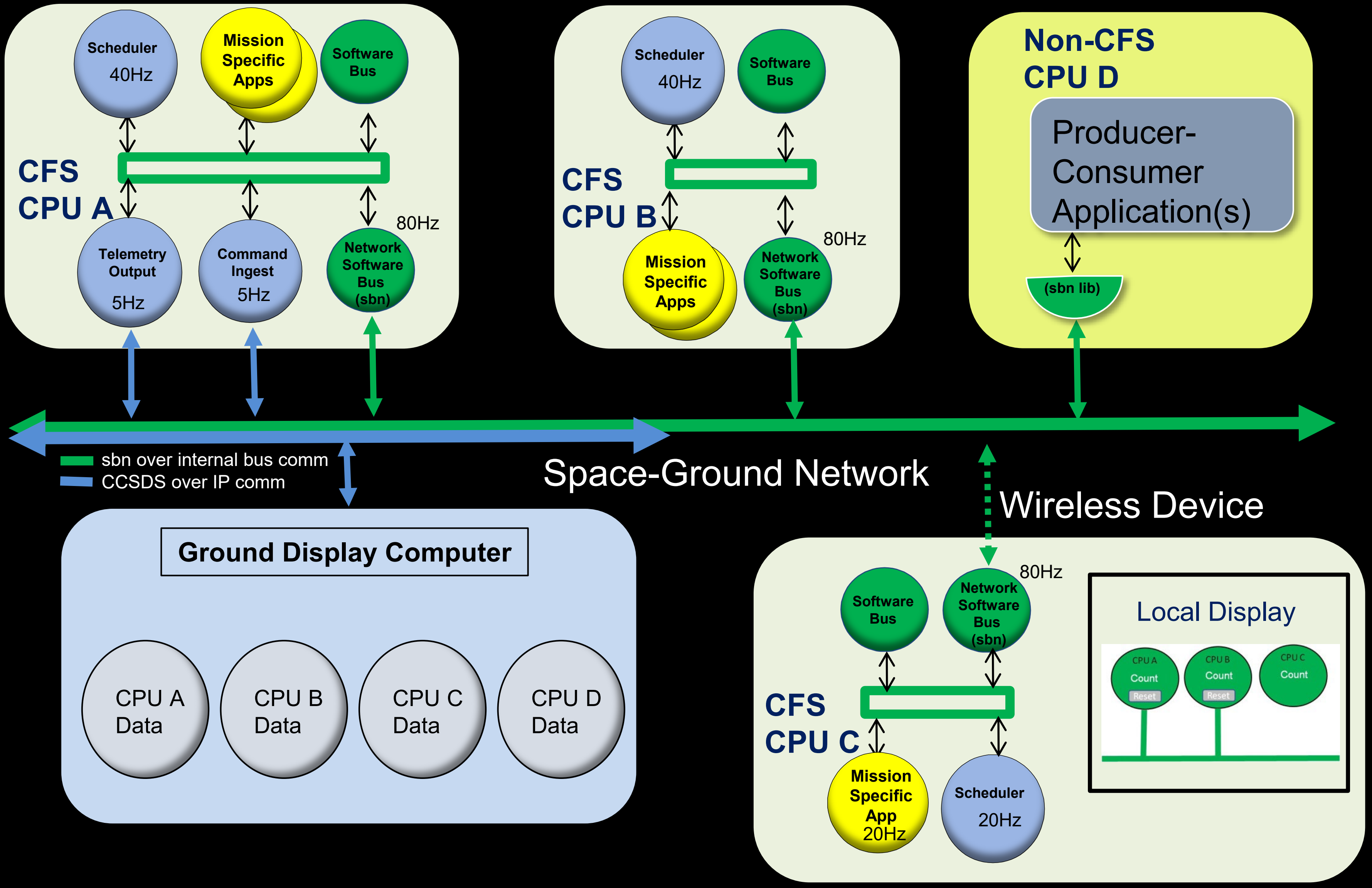


Example - Morpheus Flight Software Architecture *(2013)*



2023, Lorraine Prokop, Ph.D, NASA Technical Fellow for Software

Example - Notional Distributed CFS Configuration



Example - System Software Architecture with CFS and Reusable Components *(Morpheus, 2013)*

MORPHEUS

NASA

Flight Software

C

- CFS Core Apps
- Morpheus Specific Applications
- Custom Sensor/ Effector Apps
- CFS Infrastructure (Goddard)
- VxWorks 6.7 Operating System
- PPC 750GX Processor, cPCI (AiTech)
- I/O Devices (Serial, 1553, A/D)

Reuse Software

New Software

Tool Chain

Open Source

- **VMWare** (local PC/Mac)
- **Eclipse** (local IDE)
- **CentOS/Linux** (local VM OS)
- **GNU C/C++, Java** (compile/Xlate)
- **Subversion** (CM)
- **Redmine** (change tracker)
- **Hudson** (build checker)
- **UCC** (code count metrics)
- **Windriver Workbench** (target IDE/OS)
- **Parasoft C++test** (standards checker)

Simulation Software

C++

- EDGE Visualization
- Morpheus Specific System & I/O Models
- Dynamics, Time, Environment Models
- JEOD
- Generic Systems Models
- Valkyrie
- Trick Simulation Core (JSC)
- Linux OS

Ground Software

Java

- Displays & Controls
- scripts
- Database (postgreSQL)
- CFS Command & Data Dictionary
- ITOS Infrastructure (Goddard) (Data Com/Decom, Recon, Distribution, Display, Scripting, Recording, Post processing)
- Linux OS

2023, Lorraine Prokop, Ph.D, NASA Technical Fellow for Software

CFS Ecosystem - Supporting Tools, Apps, Libs, Systems

- Main Source Repo: https://github.com/nasa/cfs
- Community of Practice and Wiki: https://github.com/nasa/cFS/wiki
- Distributions:
  - cFS-101: Virtual machine distribution https://github.com/nasa/CFS-101
  - OpenSatKit: Open source distribution kit for satellite software https://github.com/OpenSatKit/OpenSatKit
- Ground Systems and Simulations:
  - cFS-EDS-GroundStation: Ground station implemented via EDS https://github.com/nasa/cFS-EDS-GroundStation
  - ASIST, Advanced System for Integration and Spacecraft Test http://www.asist-dai.com
  - COSMOS http://cosmosrb.com/
  - ITOS, Integrated Test and Operations System http://www.hammers.com
  - YAMCS, Yet Another Ground system, http://www.yamcs.org
  - Simulation: TRICK https://nasa.github.io/trick/
  - Simulation: 42 https://software.nasa.gov/software/GSC-16720-1
- Interfaces and Libraries:
  - SIL: Simulink Interface Layer at https://github.com/nasa/SIL
  - ECI: External Code Interface at https://github.com/nasa/ECI
  - SBN-Client: External code interface to SBN at https://github.com/nasa/SBN-Client
  - cFS_IO_LIB: IO library at https://github.com/nasa/CFS_IO_LIB
  - cFS_LIB: at https://github.com/nasa/cfs_lib
  - EdsLib: CCSDS SOIS Electronic Data Sheet Tool and Library at https://github.com/nasa/EdsLib
  - fs_lib: File services library at https://github.com/nasa/fs_lib
- Supporting Products and Tools:
  - CTF: cFS Test Framework (https://github.com/nasa/CTF)
  - CCDD: Command and Data Dictionary Tool (https://github.com/nasa/CCDD)
  - Perfutils-java: Java based performance analyzer for cFS https://github.com/nasa/perfutils-java
  - gen_sch_tbl: Tool to generated SCH app tables https://github.com/nasa/gen_sch_tbl
  - CryptoLib: Software-only CCSDS Space Data Link Security Protocol - Extended Procedures (SDLS-EP) https://github.com/nasa/CryptoLib
- Reusable Applications
  - CS: Checksum https://github.com/nasa/CS
  - CF: CFDP https://github.com/nasa/CF
  - CI: Command Ingest https://github.com/nasa/CFS_CI
  - DS: Data Store https://github.com/nasa/DS
  - FM: File Manager https://github.com/nasa/FM
  - HK: Housekeeping https://github.com/nasa/HK
  - HS: Health and Safety https://github.com/nasa/HS
  - LC: Limit Checker https://github.com/nasa/LC
  - MD: Memory Dwell https://github.com/nasa/MD
  - MM: Memory Manager https://github.com/nasa/MM
  - SBN: Software Bus Network https://github.com/nasa/SBN
  - SC: Stored Commands https://github.com/nasa/SC
  - SCA: Stored Command Absolute https://github.com/nasa/SCA
  - SCH: Scheduler application https://github.com/nasa/SCH
  - TO: Telemetry Output https://github.com/nasa/CFS_TO
  - Skeleton App: https://github.com/nasa/skeleton_app

Conclusion

- Hope this was informative
- Ton of information online
- As always, open to questions, comments, suggestions for more of these videos!