Evolving Flight and Ground Data Systems for Low-Cost Development and Mission Automation

Dan Smith
Dan.Smith@nasa.gov

Jonathan Wilmot
Jonathan.J.Wilmot@nasa.gov

GMSEC Public Website: http://gmsec.gsfc.nasa.gov

Topic Summary

- This presentation covers the work done since 2001 by 580/595 to introduce new system architectures to the domain of mission flight and ground systems.

- The new approaches are already saving money for existing missions and have the potential to enable new mission designs and operations approaches.

Introduction

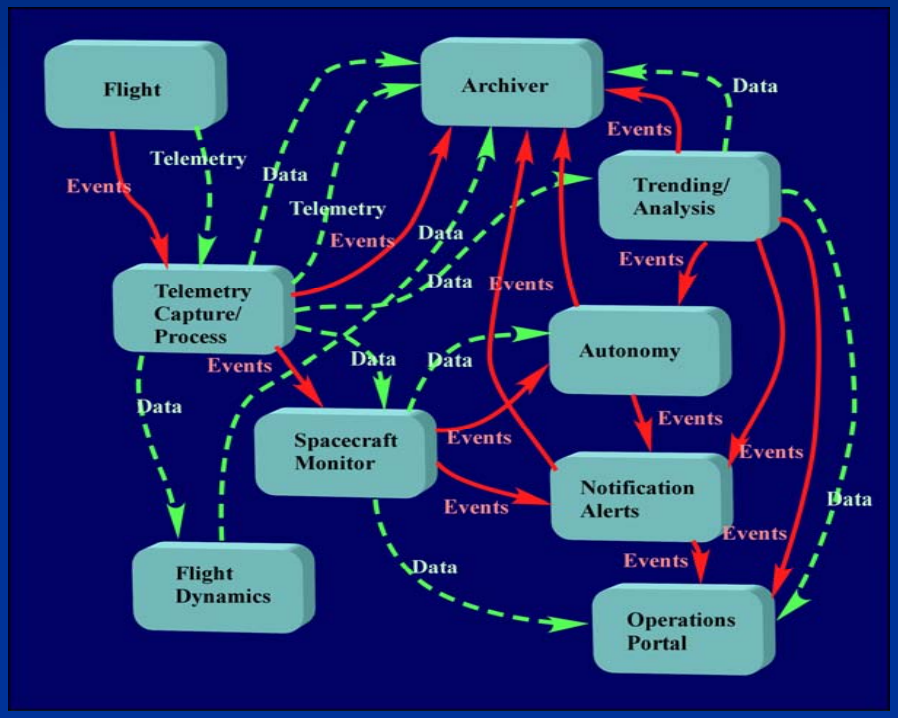
- Mission data system approaches at GSFC have changed slowly at GSFC
  - Generally based on the previous system approach
  - Not a lot of functional enhancement
  - Sharing of capabilities between missions is limited
- Significant Mission-Level Impacts
  - High cost of operations due to lack of automation
  - Very conservative mission design proposals
- GSFC mission systems have not kept pace with the commercial space sector

GMSEC Approach
(Goddard Mission Services Evolution Center)

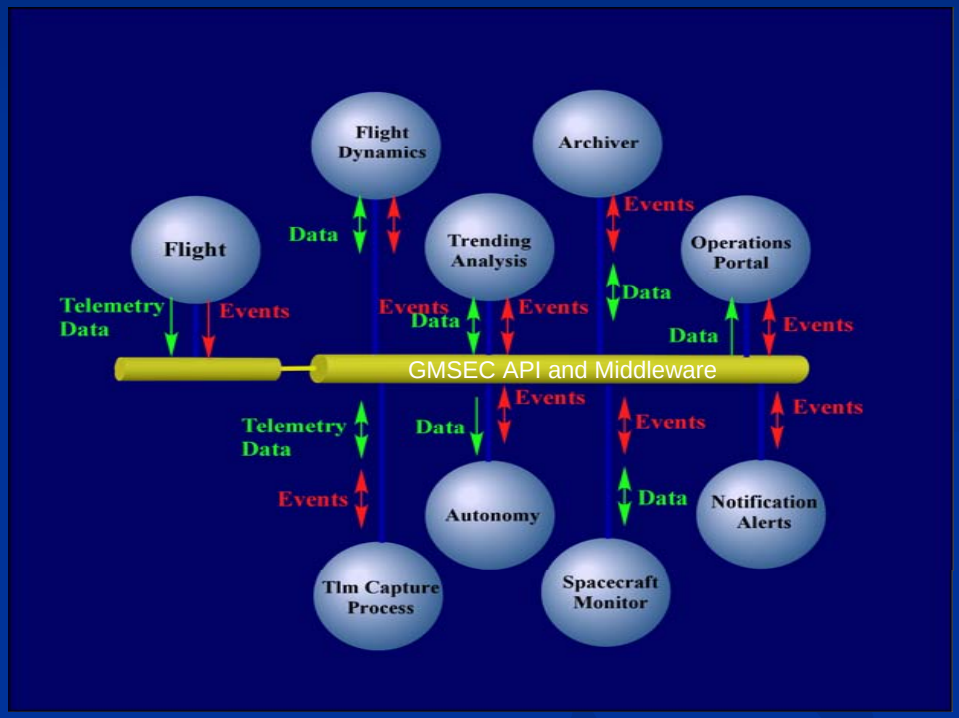
- Goals
  - Simplify integration and development
  - Facilitate technology infusion over time
  - Support evolving operational concepts
  - Avoid vendor lock-in
  - Allow for mix of heritage, COTS and new components
- Concepts
  - Standardize interfaces – not components
  - Provide a middleware infrastructure
  - Allow users to choose – GMSEC doesn't decide which components are best or dictate which components a mission must use. It's the mission/user's choice!

GMSEC Publish Subscribe Communication

Traditional Design
Socket Connections



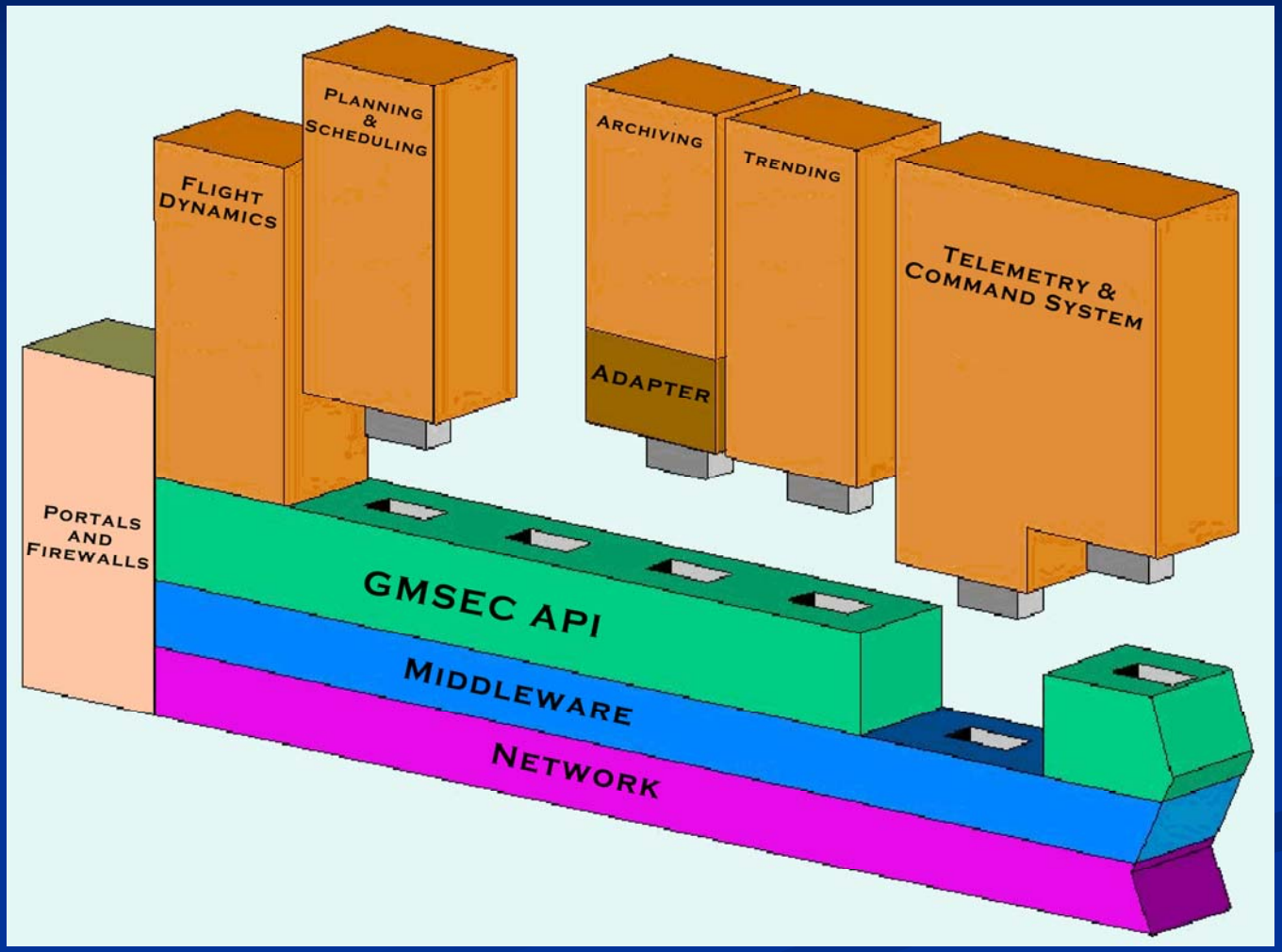
GMSEC Design
Middleware Connections



Middleware simplifies integration by having components interface to a bus and not to each other.

Plug-and-Play Concept

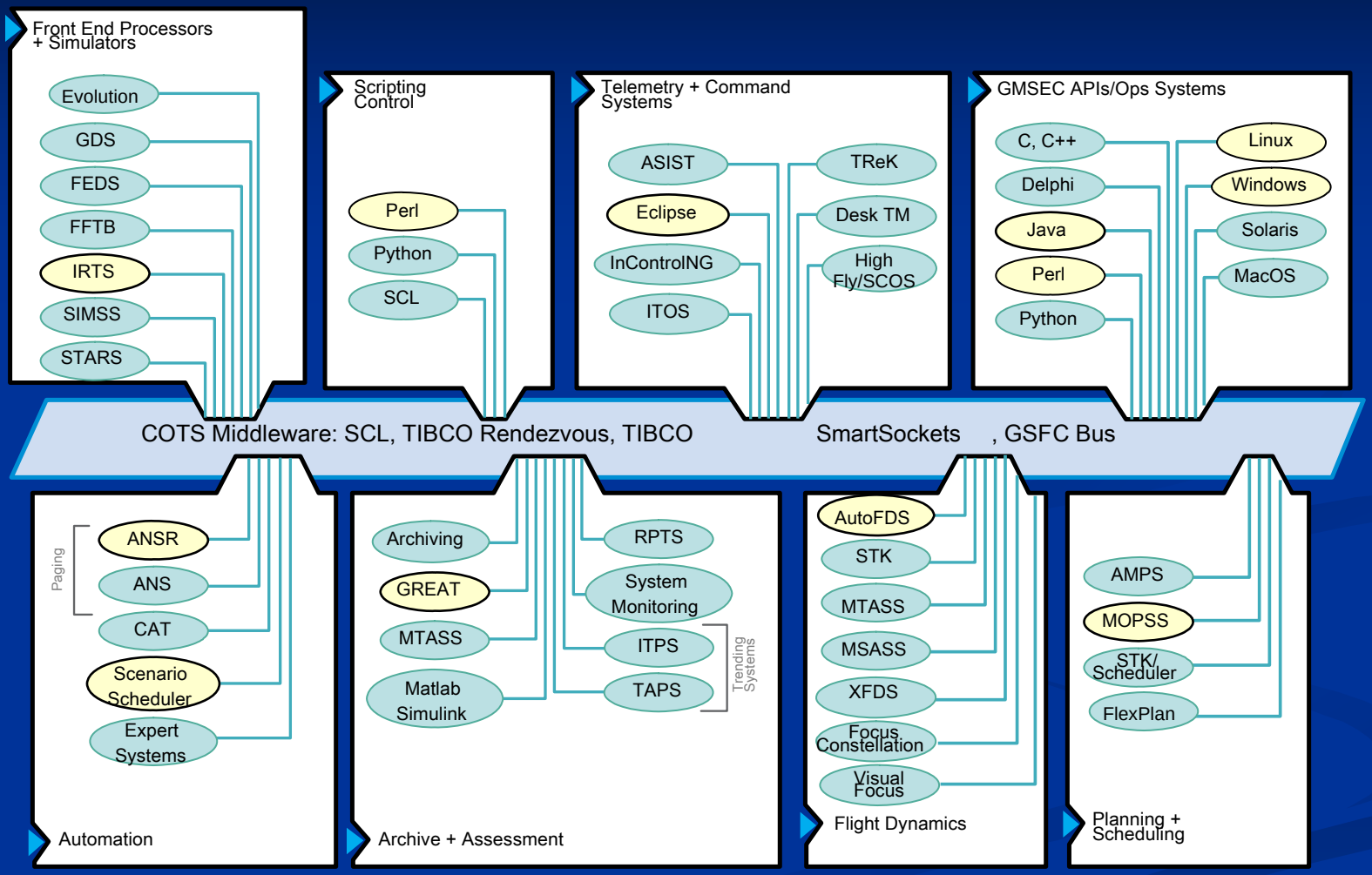
By creating a "framework", individual applications can be easily integrated into an existing system without regard to many underlying implementation details.



GMSEC System Status

- Dozens of components now available
- GMSEC Architecture Application Programming Interface (API)
  - Available for download from protected site
  - Middleware Independent – no vendor lock-in of critical framework
    - Middleware options including: TIBCO SmartSockets, TIBCO Rendezvous, ICS SWB, GSFC Bus, IBM WebSphere, SOAP
  - Cross Platform API: Linux RH (7.1-7.3,9, + Enterprise), Suse (8.1)
    - Windows NT/2000/XP; Solaris (gcc + cc)
  - Multiple Languages: C, C++, Java, Perl, Python, Delphi
- Automated test package for over 8,000 combinations of features, languages, platforms and operating systems
- Working on patent applications and OPEN SOURCE process

GMSEC Component Catalog



Choices are available for many subsystems. The TRMM mission selected catalog components to best meet their reengineering needs.

GMSEC Operational Status

Existing-mission reengineering efforts

- Tropical Rainforest Measuring Mission (TRMM)
  - Goal: reduce operations cost by 50%
  - Operational with GMSEC architecture since Fall 2005
  - Pathfinder for Terra, Aqua, Aura automation (2005-2007)
- Small Explorer (SMEX) missions – SWAS, WIRE, TRACE, SAMPEX
  - Operational on SWAS, preparing for others
  - Conducted a successful 2-week lights-out operation
  - Pathfinder for low-cost fleet operations & updating existing space science missions

New GSFC missions

- ST5 Constellation system - ready for launch
- GLAST, SDO in development, discussions underway with others
- Future: MMS, GPM (long-range schedules)

Other applications

- Other NASA Centers and related facilities: GSFC/WFF, JHU/APL, MSFC, KSC
- GSFC Flight Dynamics Facility (FDF)
- Commercialization interest: Several major commercial satellite operators

GSFC Missions

Organization	Purpose	Notes
TRMM	Lights out setup and acquisition of telemetry data under control of GMSEC automation.	ORR held August 9, 2005. Now operational.
SMEX	Constellation Ops with INcontrol-NG	SWAS in parallel operations phase since June 2005, others being worked. Two weeks lights out ops test completed successfully.
TERRA, AQUA, AURA	Constellation Ops with Eclipse and GMSEC architecture. Firewatch ops con.	GMSEC automation will assume role of operator for routine Proc execution and monitoring. Terra SRR was in May 2005.
ST5	3-satellite Constellation Ops with ASIST, AMPS, Simulink.	Adaptive scheduling and modeling. Two weeks lights out at end of mission.
GLAST	Ground system automation.	Plans in place for GMSEC-based approach for MOC components.
SDO	24/7 GeoSynch mission with dual redundant antenna and data feeds.	System design being updated in early 2006 for GMSEC-based bus and automation
Code S Missions – Wind, Polar, SOHO, ACE, XTE.	MOMs task underway to determine best use of GMSEC architecture and trade of T&C systems to replace TPOCC.	Could show use of multiple components on the same GMSEC bus supporting different missions.
Wallops/Code 589	Developed proposal for using GMSEC for range operations	Wallops team investigating design options
MMS	Full ground system.	Provided FY05 and FY06 funding to GMSEC.
GPM	Full ground system.	Support planned for PDR. Good advocate for GMSEC technology.
JWST	TBD	Have viewed GMSEC demos, involved in common DB talks.
LRO	Full ground system.	Planning meeting scheduled for February 2006

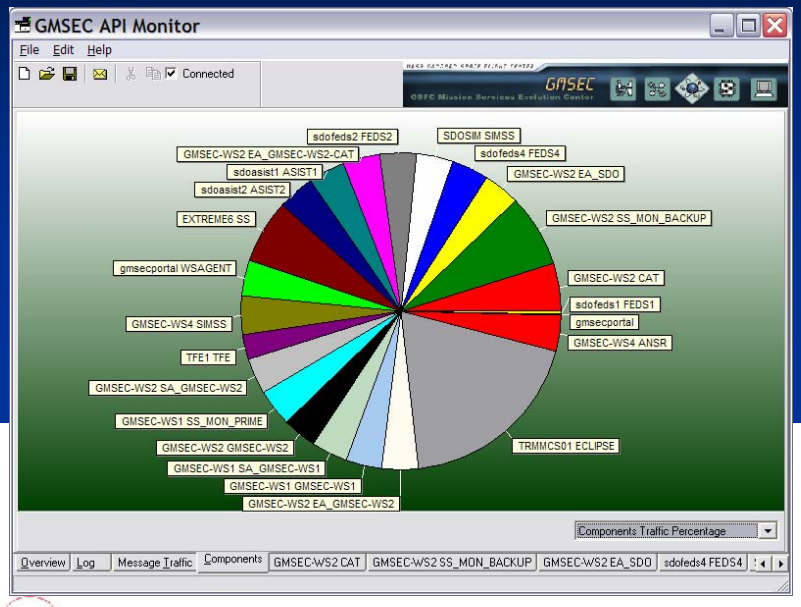
Automation Concepts

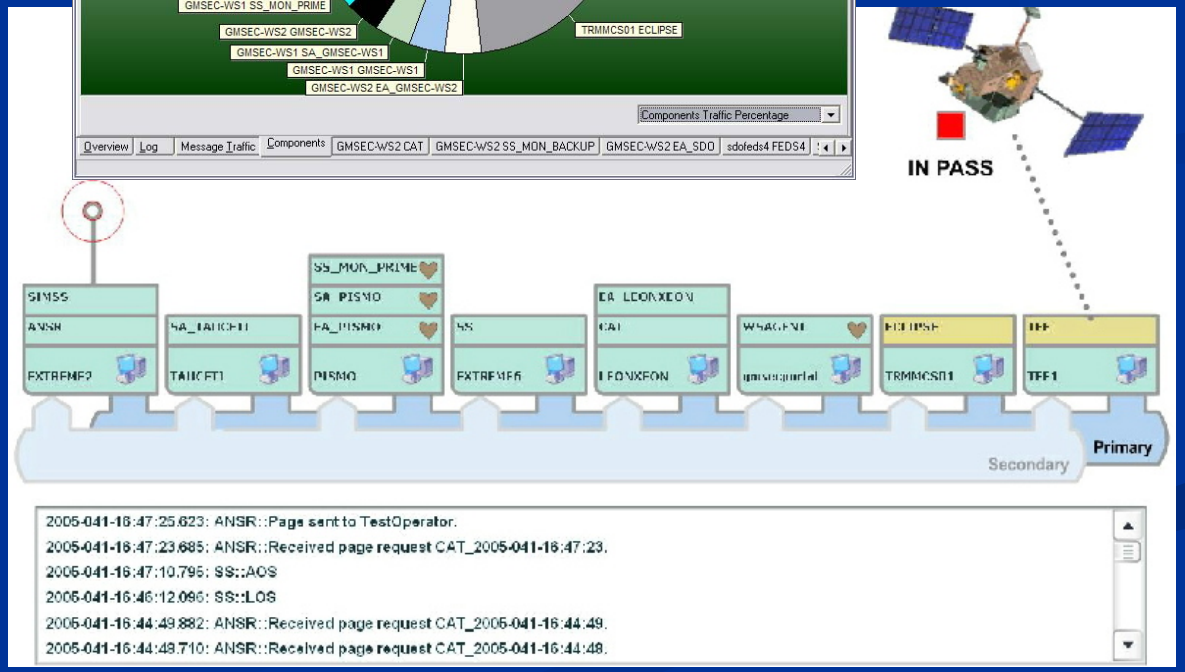
- Each Component Must Meet Certain Standards
  - Meet its functional interfaces
  - Publish keep-alive and status messages to the bus
  - Accept control directives over the bus

- Common Tools Cross Domain Boundaries
  - Tools can "listen" for status from all components
    - Provides system-wide situational awareness
  - Single tools can direct actions of any number of components
    - Provides system-wide control ability
  - "Criteria-Action Tool" provides ability to define situational awareness rules and corresponding actions
    - Allows for event-driven automation
    - Complex temporal and cross-domain rules and actions defined by the Mission Operations Team
      - Allows for lights-out operations
      - Automates many routine flight operations team functions
      - Properly expanded, could enable cross-mission collaboration and automation

Tool Development is Simplified

- Support tools are easy to develop
- May not require any integration with other components
- Simply monitors messages on the bus
- Examples
  - Performance tool
  - Configuration display

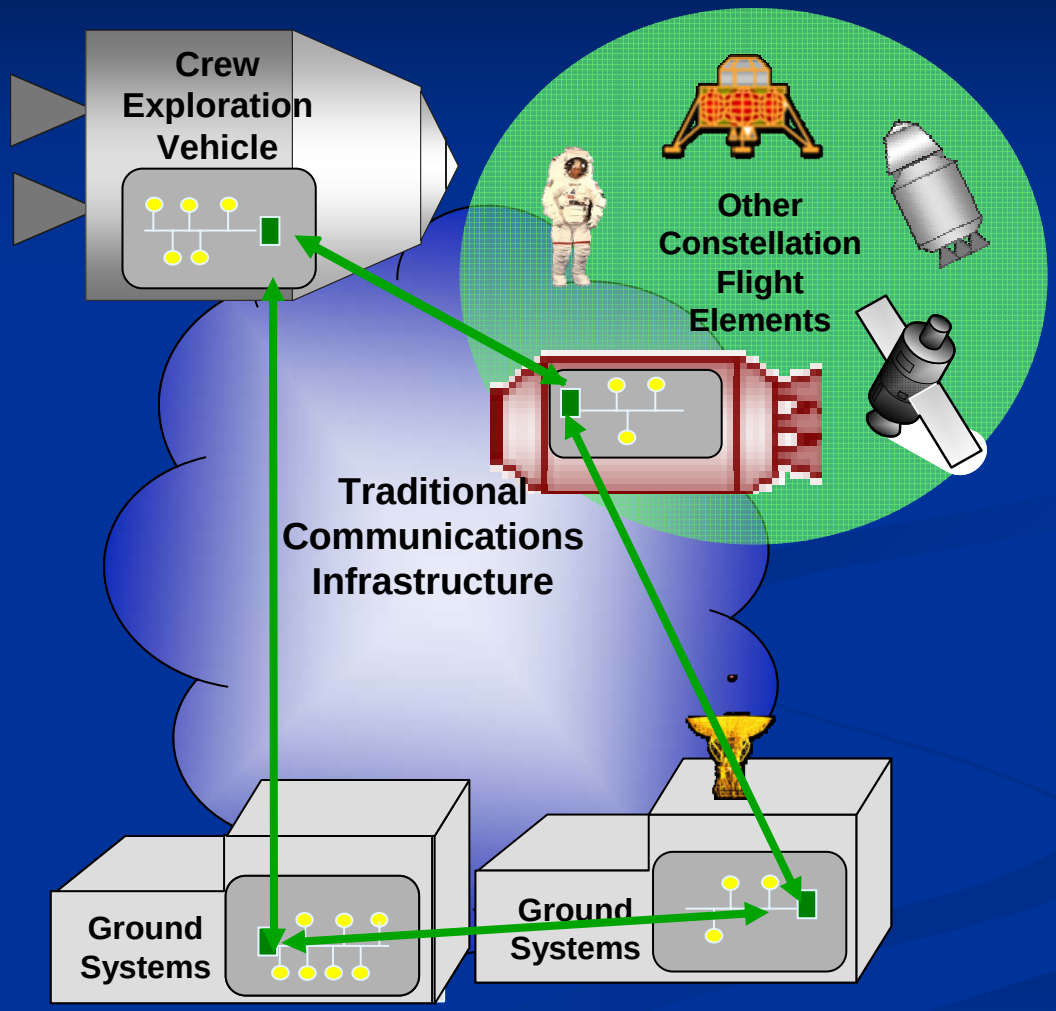




Ties to Exploration

- A NASA multi-Center team was formed in June 2004 to examine approaches to meet the mission data system needs for Exploration. Themes of interoperability, flexibility, commonality, and evolvability
- The GMSEC approach stood out as the clear starting point.
- The Exploration team has continued to expand and the reliance on the GMSEC architecture has continued to grow.
- Considerable GMSEC involvement
  - GMSEC-based labs now at other NASA Centers
  - GMSEC middleware moving messages between Centers
  - GMSEC engineers supporting demo and spec development
- One NASA award issued for first full year of involvement in Agency-wide C3I effort.
- Designing and prototyping GMSEC-like approaches to Exploration ground systems – **official plans are not public**.

Concepts Apply to Flight and Ground



GMSEC Benefits

1. Significant reduction in integration time
2. Components added/upgraded without impacting existing system
3. Ideal for using multiple small distributed development teams and vendors
4. New concepts emerging for small independent components that integrate with the bus and provide immediate benefits
5. Missions more willing to adopt the approach if "old favorite" components can still be used
6. Some vendors see message compliance as a way to enter what had appeared to be a closed marketplace
7. Standard message approach provides collaboration possibilities with other organizations

- **Same concepts can be extended to flight systems!**

GMSEC Lab

GMSEC Development and Demonstration Lab is in Building 23, Room N305.

New Architecture for Flight

- Core Flight System -

"Advantages of the Core Flight System on Mission Development and Deployment"

TOPICS

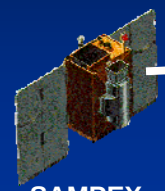
- CFS Definition
- Software Heritage
- Mission Advantages
- Concepts & Standards
- Examples
- Status

Core Flight System Definition

The Core Flight System (CFS) is a platform-independent, Flight Software (FSW) environment integrating a reusable core flight executive, software component library, and a Integrated Development Environment (IDE).

The CFS is the flight system counterpart to the GMSEC ground solution.

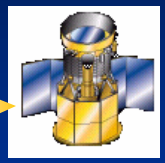
CFS Goddard Heritage



SAMPEX
(launched 8/92)

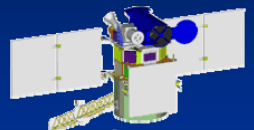
SWAS
(launched 12/98)

TRACE
(launched 3/98)

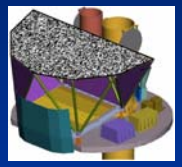


WIRE
(launched 2/99)

SMEX-Lite



Triana
(waiting for launch)



Swift BAT
(12/04)

XTE *(launched 12/95)*

TRMM *(launched 11/97)*

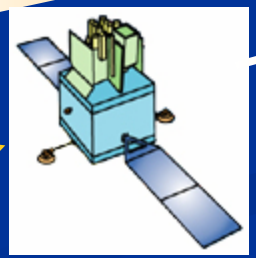
JWST ISIM
(2011)

IceSat GLAS *(01/03)*

MAP *(launched 06/01)*

Core FSW Executive
Core FSW System

ST-5 *(5/06)*



SDO (2007)

LRO (2009)

Future Spacecraft and Instruments

CFS Key Mission Advantages

- Deploy quality flight software faster with a more predictable schedule
- Reduce mission flight software costs by up to one third
- Enables parallel multi-team collaborative development
- Enable seamless system interoperability
- Simplifies technology infusion and system evolution during development and on-orbit

CFS Key Mission Advantages

- Provides a platform for rapid prototyping
  - Develop applications to the CFS API and quickly integrate and test
  - Applications can be modified and quickly reloaded on the running system
- Messaging middleware has a dynamic network discovery protocol to allow system components to come and go as needed
- Cross platform portability of applications
  - Applications developed on desktops can be recompiled, linked and loaded to the target system
- Applications can be easily added or updated on orbit

CFS Key Mission Advantages

- Library/Catalog of applications will expand allowing more choices and functionality for following missions
- Mission developers can scale targets from low power Coldfire to high end PowerPC flight processor.
- High visibility into software system performance and configuration at run-time
- Local event logging services for fault analysis (Black Box)
- Ease of command and telemetry re-routing
  - Authenticated commands can be from any system
  - Backup systems can be powered on and quickly have command and telemetry interfaces.
  - Very powerful model for contingencies

CFS Concepts and Standards

- Layered Architecture
- Standard Middleware/Bus
- Standard Application Programmer Interface

} Core Flight Executive

- Plug and Play
- Reusable Components

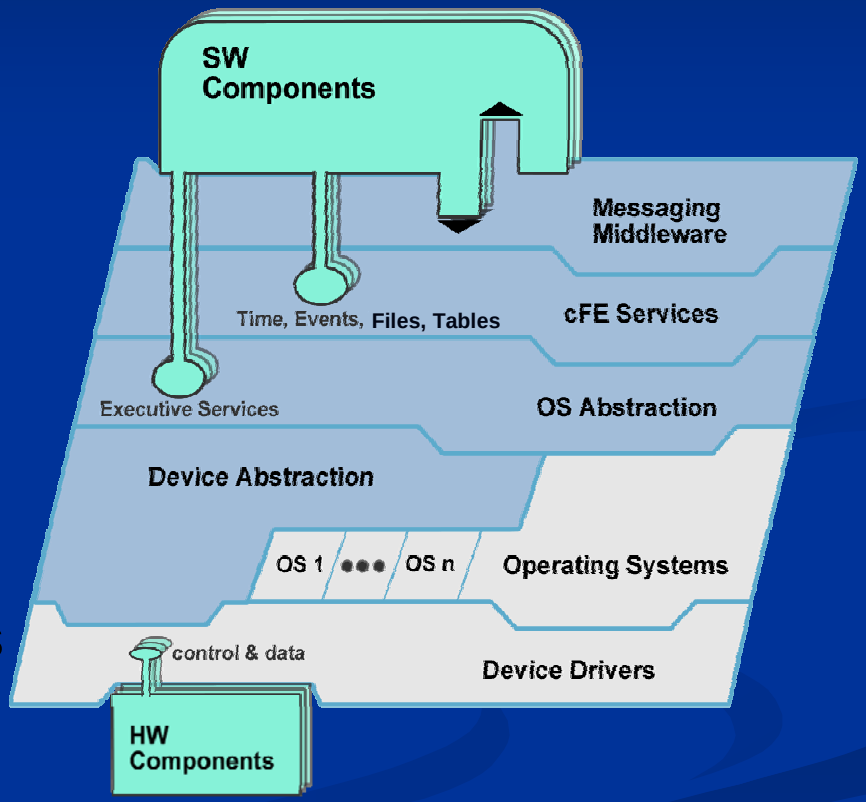
} Component Library

- Configuration Management
- Requirements Tracking
- Development Standards
- Development Tools

} Integrated Development Environment (IDE)

Core Flight Software Executive (cFE)

- Strategic Software Layering
  - Software of a layer can be changed without affecting the software of other layers
- Advanced Message Handling
  - Eliminates manual configuration of FSW
  - Automates integration of FSW with applications and hardware components (Publish/Subscribe model
- Standardized, Abstracted Interfaces
  - Minimizes software impacts from flight hardware, RTOS(*), and application changes

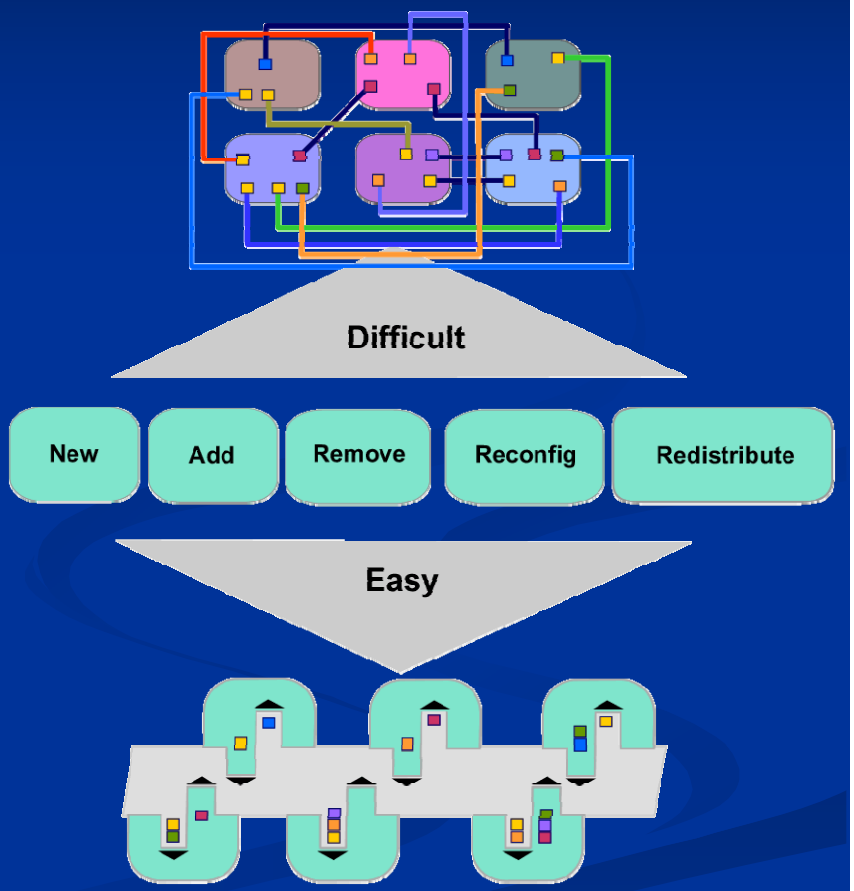


Plug and Play

- cFE API's support add and remove functions
- SW **components can be switched in and out at runtime**, without rebooting or rebuilding the system SW.
- **Qualified Hardware and CFS-compatible software both "plug and play."**

Benefits:

- **Changes can be made dynamically** during development, test and on-orbit even as part of contingency management
- **Technology evolution/change** can be taken advantage of later in the development cycle.
- **Testing flexibility (GSE, test apps, simulators)**

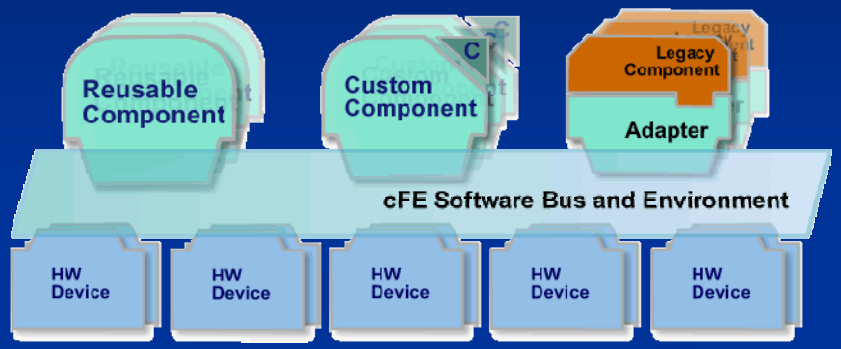


This powerful paradigm allows SW components to be switched in and out at runtime, without rebooting or rebuilding the system SW.

Reusable Components
Catalog & Library

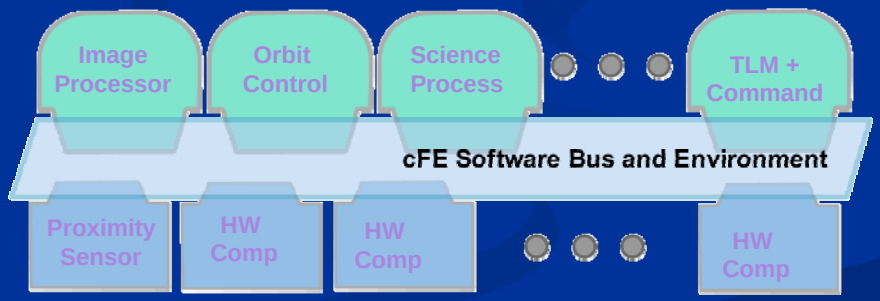
Reusable Components

- Common FSW functionality has been abstracted into a library of **reusable components and services**.
- **Tested, Certified, Documented**
- **A system is built from:**
  - Core services
  - Reusable components
  - Custom mission specific components
  - Adapted legacy components



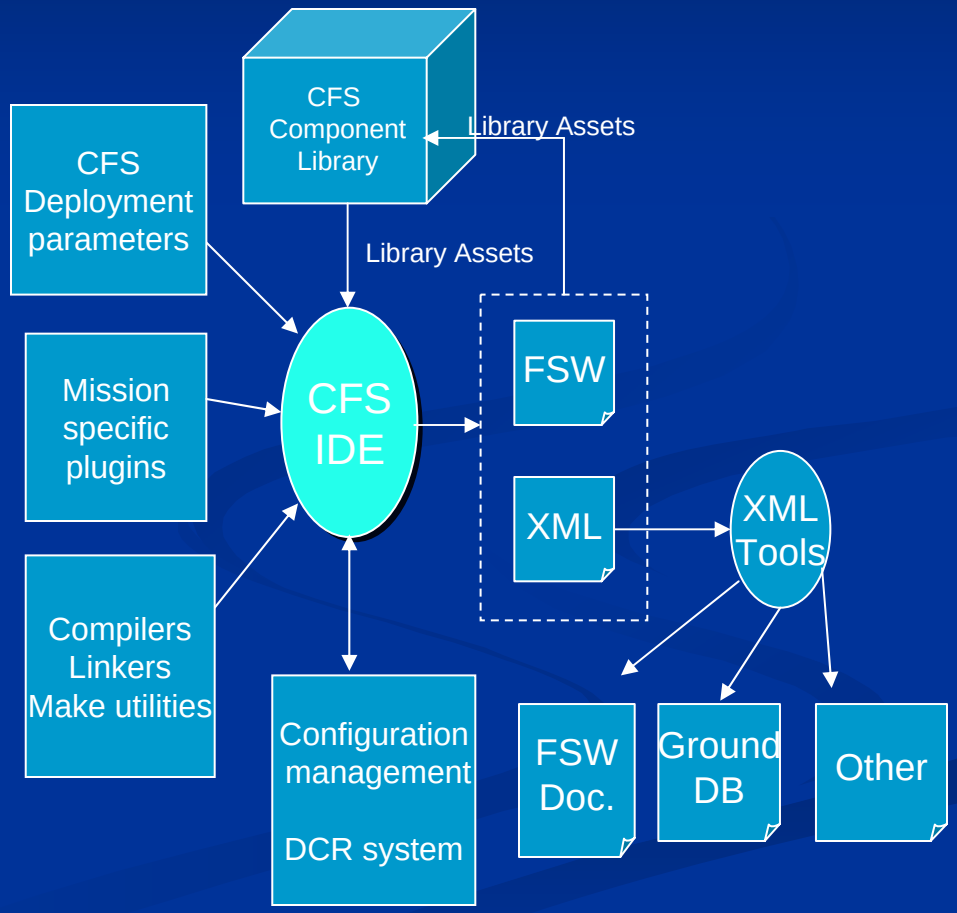
Benefits:

- Reuse of tested, certified components supplies savings in each phase of the software development cycle
- Reduces risk
- **Teams focus on the custom aspects** of their project and don't "reinvent the wheel."



Integrated Development Environment (IDE)

- The CFS IDE is a growing set of integrated tools that support the management, development and configuration of a CFS deployment
- Tools are hosted on the open source Eclipse platform being adopted by many tool vendors, including Wind Rivers' VxWorks
- IDE supports rapid deployment using pick and click graphical user interface for system configuration
- CFS Component consists of: Requirements, Design, Code, unit & build tests, User's guide, and XML description of cmd/tlm database

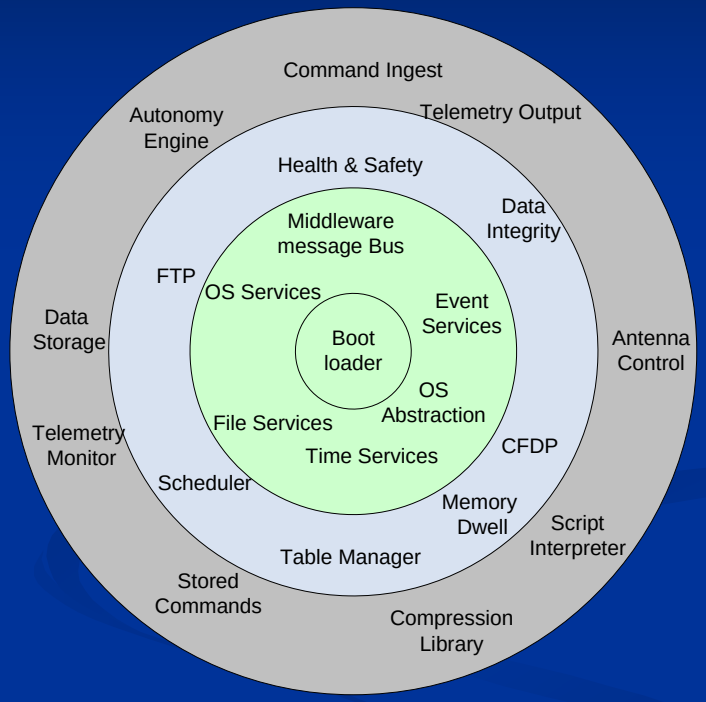


Display Example

- Dynamic Crew Displays or health monitoring functions
  - System request, *Give me all temperature data for CEV fuel tanks.*
  - Display application requests data with some QoS
    - Data management component handles data requests
    - Search engine scans XML files for requested data.
    - Responds to app with "requested data is in packet x with this meta data description"
  - Request data can have several seconds to find and respond to first request, but then requested QoS would be used.
  - Display Applications can then subscribe to packet and extract data as described in XML.
  - Data shows up on Crew Display in proper engineering units
  - Data can come from local or external subsystems/elements

Reusable Platform Example

- Interchangeable avionics boxes have a core ring of standard services and functions.
  - Actual code and implementation can be different for each vendor, but need to meet the API's and ICD's
  - Messages to alert operators or other components of asynchronous events
  - Common software load/dump interfaces
  - Common set of file service interfaces
  - Core ring can use memory management unit (MMU) for self protection
  - Network enabled Boot loader in hardened ROM
- Benefits
  - Speed prototyping and integration times
  - Common training for maintenance teams.
  - Reduced cost for FSW development and maintenance
  - Boxes can take on "personalities" as needed
  - Potential for Fewer flight spares



Core Ring
System Ring
Personality Ring

Target Platforms

- **Embedded Systems**
  - PPC/mcp750 running VxWorks 5.5, VxWorks 6.x, Linux
  - PPC/mcp405 running Linux (imminent)
- **Desktop Systems**
  - PPC/Macintosh running OS X
  - X86/PC running Linux
  - X86/PC running Cygwin under Windows

Future targets: Coldfire running RTEMS, PPC/Rad750 running VxWorks 6.x, PPC/mcp750 running RTEMS and LynxOS

Status

- **2004 multi-CPU/Box prototype demonstrated**
  - Core, generic FSW services and Software components
  - Dynamic application load and startup
  - Dynamic message bus reconfiguration for "Box faults"
- **Version 3.1 cFE, delivered to Autonomy Test Bed (ATB), LRO Nov 16 2005**
  - VxWorks 5.4, Static linking, S-rec file startup
  - For ATB effort, cfE has worked with Vxworks, OSX, and Linux
  - Unit Test Framework (UTF) developed in order to unit test cFE Applications
- **LRO configuration baseline cFE 3.1**
  - VxWorks 5.5, Static linking, S-rec file startup
  - IP/UDP command and telemetry
- **CHIPS on orbit demonstration, Dec 05**
  - VxWorks 5.4
  - IP/UDP command and telemetry
  - Modified cFE to be loaded onto CHIPS as a standalone application

Future work

- Ongoing work
  - IDE development
  - cFE and component performance analysis, optimization, scaling. …
  - Component porting
  - Ground system support tools
  - Incorporate middleware needs for Exploration Initiative.
  - XML definition and exchange protocols
- Version 4 cFE April 06
  - Executive Services, reset/restart logic, application cleanup, suspend/resume, exception handlers
- Version 5 cFE Dec 06
  - VxWorks 6.2 process model integration to start Q1 2006
  - Network support, discovery, fault management, data exchange protocols

Overall Conclusions

- Message-bus component-based architectures are well proven and provide significant benefits over traditional flight and ground data system designs.

- Mission benefit through enabled automation, lower cost and new mission-enabling operations concept options.

☑ Better
☑ Faster
☑ Cheaper

Possible through innovative application of advanced technologies