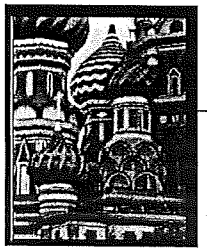
7th International Symposium Reducing the Costs of Spacecraft Ground Systems and Operations (RCSGSO)
11 - 15 June 2007
Moscow, Russia

Reducing Development and Operations Costs Using NASA's "GMSEC" Systems Architecture

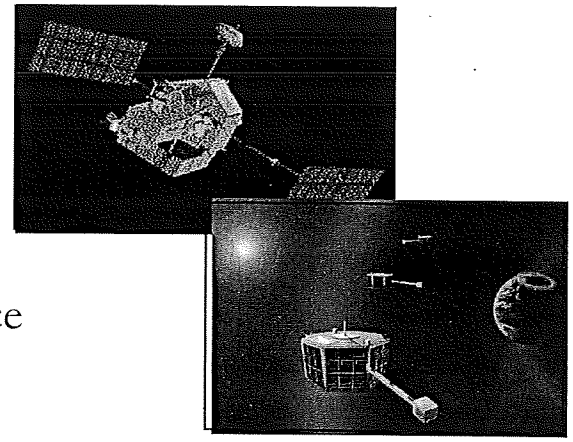
Dan Smith
John Bristow
Patrick Crouse

NASA Goddard Space Flight Center
Greenbelt, Maryland USA 20771



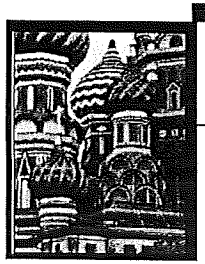
NASA/GSFC Mission Background

- NASA/GSFC manages about 30 spacecraft
  - ½ at the NASA campus near Washington, DC
  - ½ at Universities around the United States

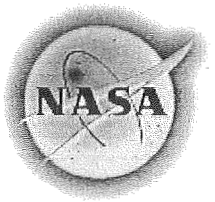


- Typical characteristics . . .
  - Scientific missions in low-earth orbit, a few in deep space
  - Each mission has its own control center and ops team
  - Mission durations of 6 months to 20+ years

- Primary issues
  - Cost of development, ops and maintenance
  - Slow advancement of new capabilities and technologies
  - Little use of commercial software (COTS)

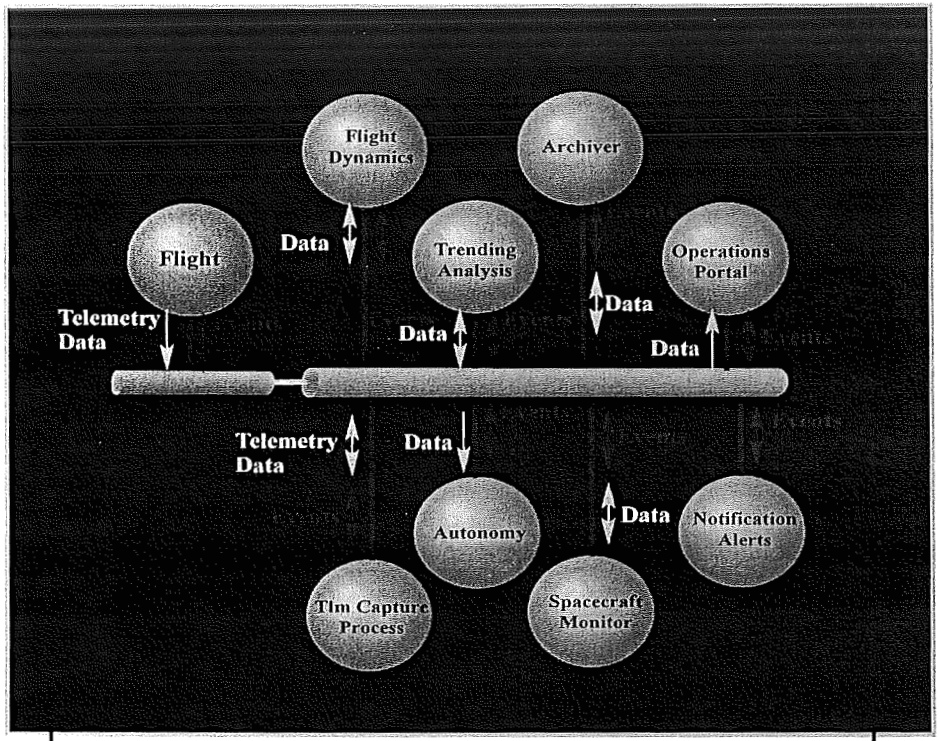


GMSEC Architecture Approach (Goddard Mission Services Evolution Center)



- Goals
  - Simplify integration and development
  - Facilitate technology infusion over time
  - Support evolving operational concepts
  - Allow for mix of heritage, COTS and new components
- Concepts
  - Standardize interfaces – not components
  - Provide a middleware infrastructure
  - Allow users to choose their products (no single answer)
  - Create a general purpose approach with broad applicability

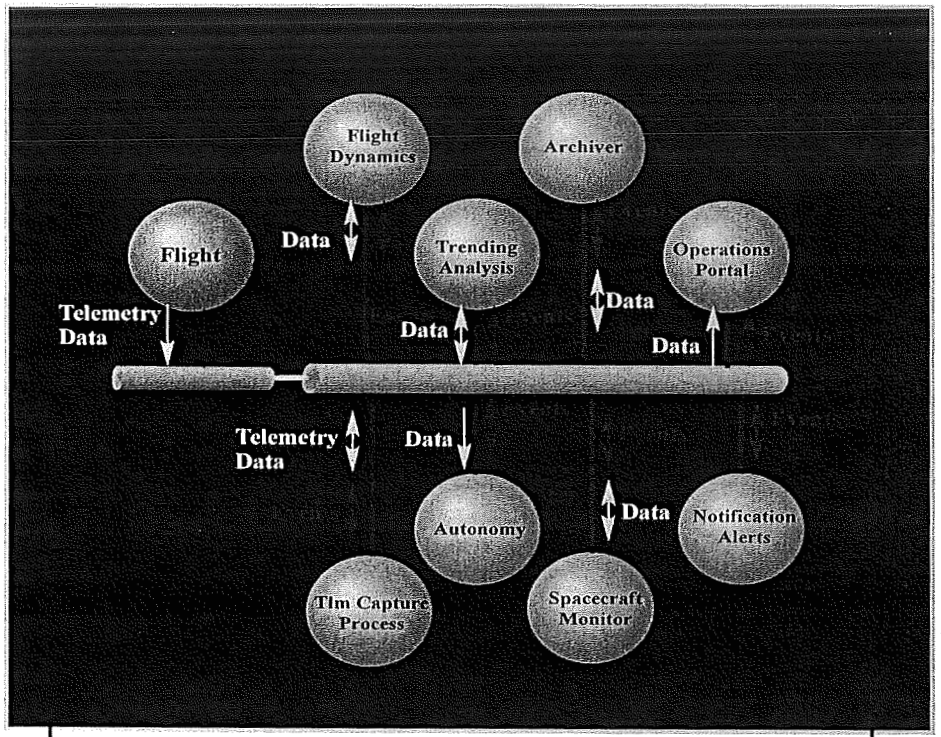
GMSEC Publish/Subscribe Communications



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

- Applications "publish" their data onto the bus
- Other applications "subscribe" to the types of data they are interested in
- The message bus (middleware) routes the data to the requested applications

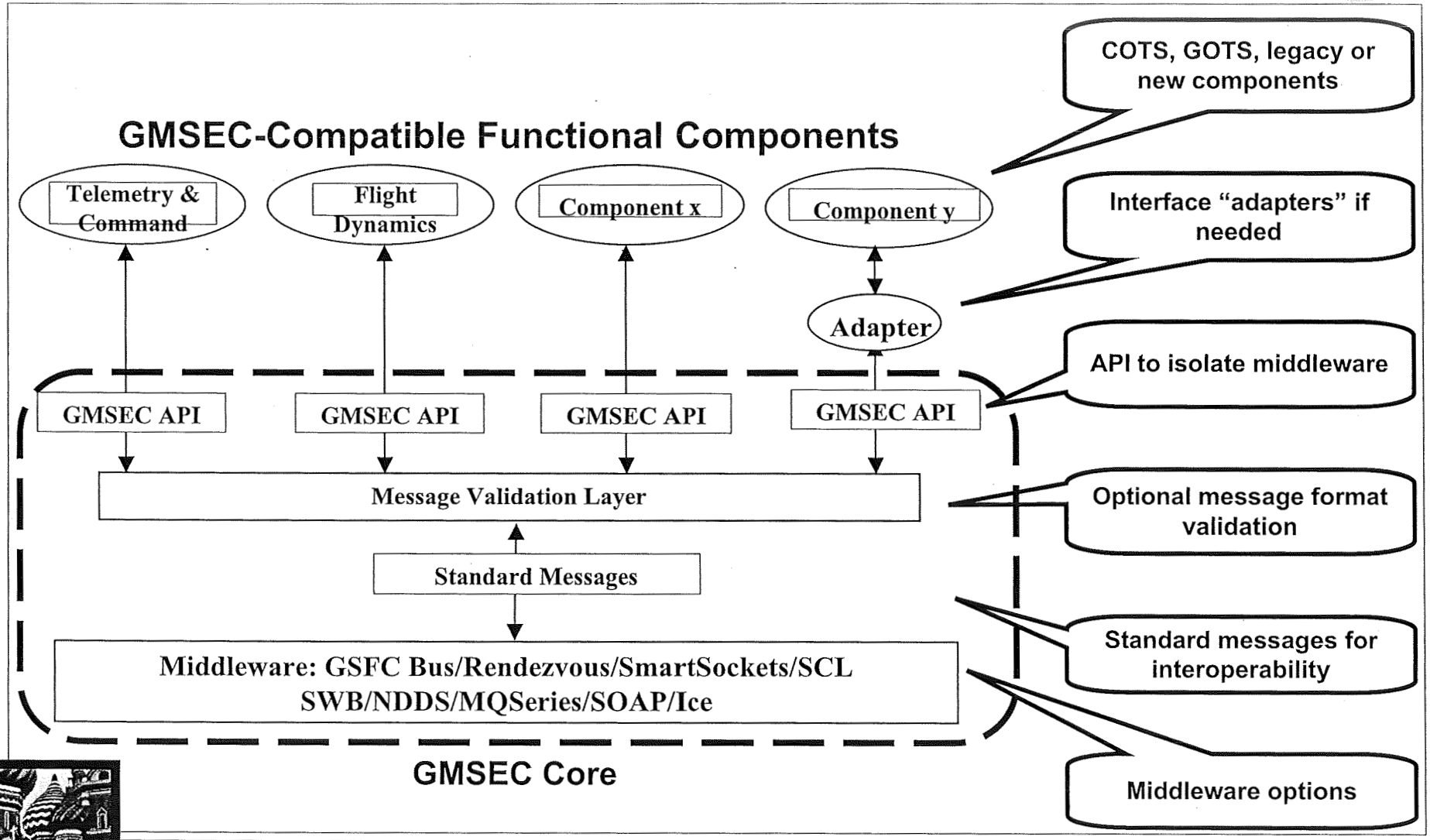
GMSEC Publish/Subscribe Communications



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

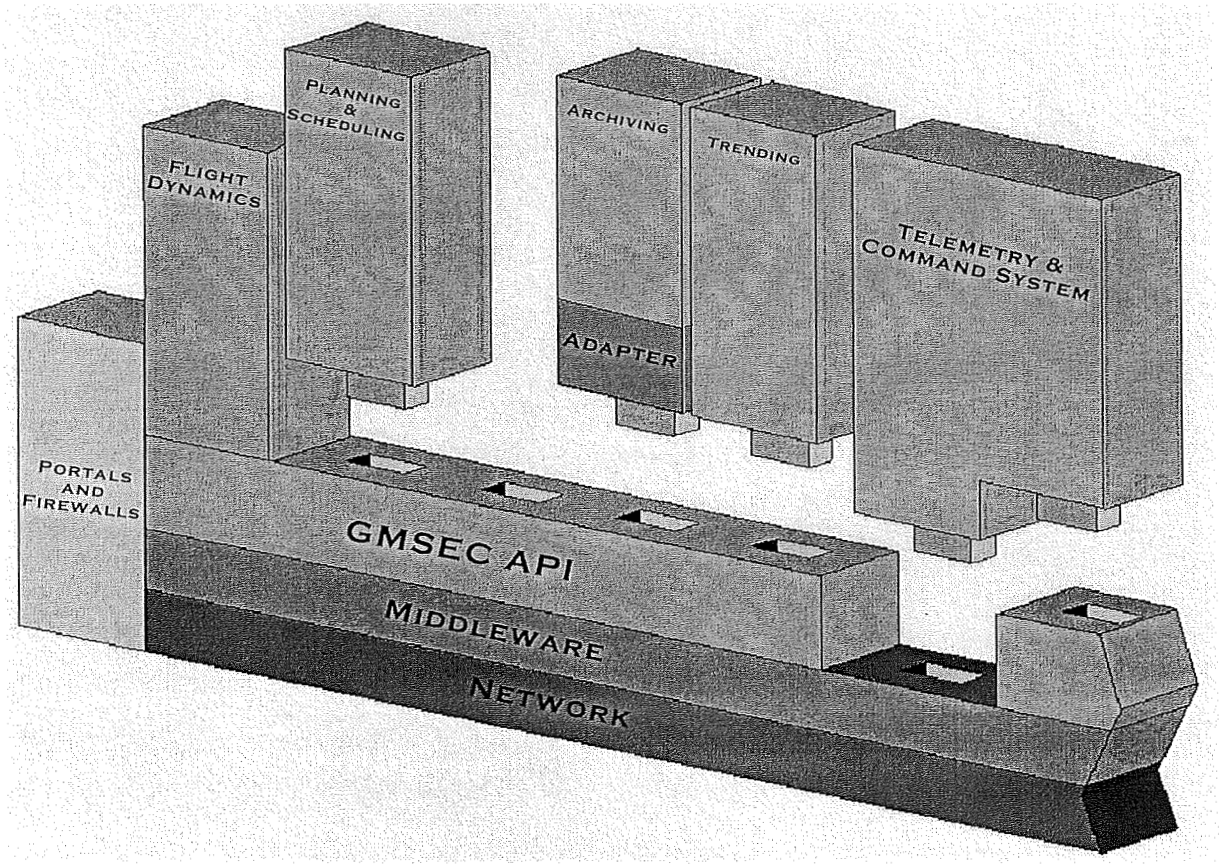
- Applications "publish" their data onto the bus
- Other applications "subscribe" to the types of data they are interested in
- The message bus (middleware) routes the data to the requested applications

GMSEC Facilitates Plug-and-Play

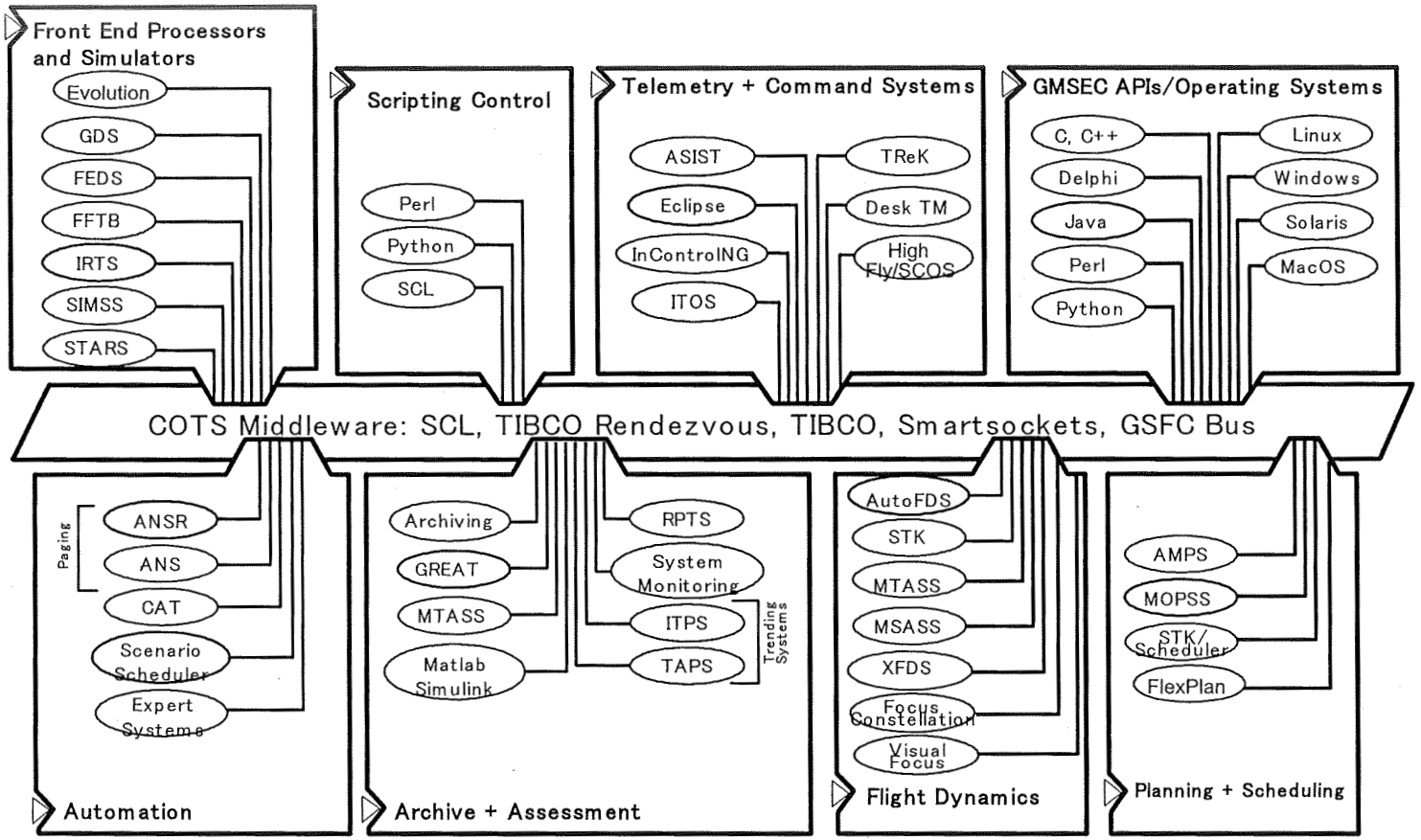


GMSEC Plug-and-Play Concept

By creating a "framework", individual applications can be easily integrated into an existing system.



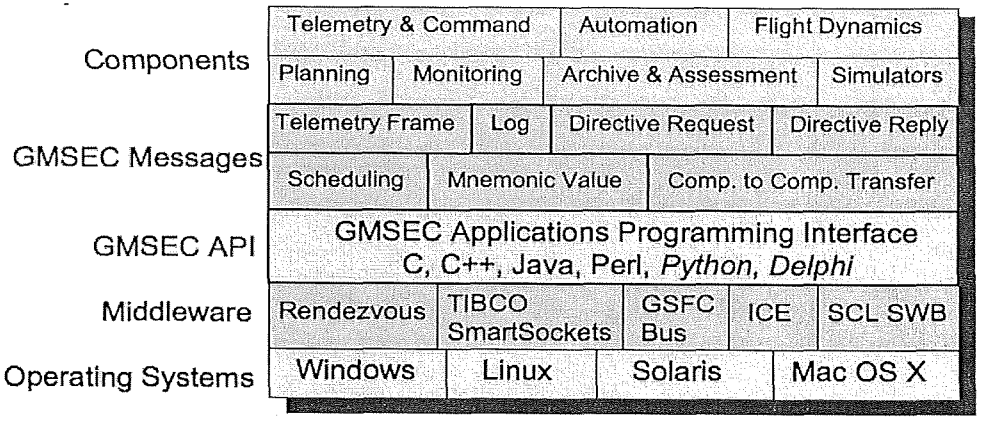
GMSEC Component Catalog

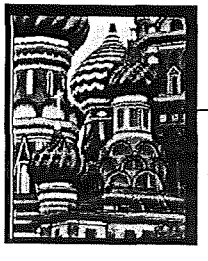


Choices are available for many subsystems. The TRMM mission selected catalog components to best meet their reengineering needs. [component names not important]

GMSEC System Technical Status

- Began in 2001
- Over 50 components available
- Mature Application Programming Interface (API)
  - Multiple middleware choices
  - Multiple operating systems
  - Multiple programming languages
- Architecture, API and GSFC Bus OPEN SOURCE since April 2006
- Automated test package - 24,000+ combinations of middleware, languages, platforms, operating systems



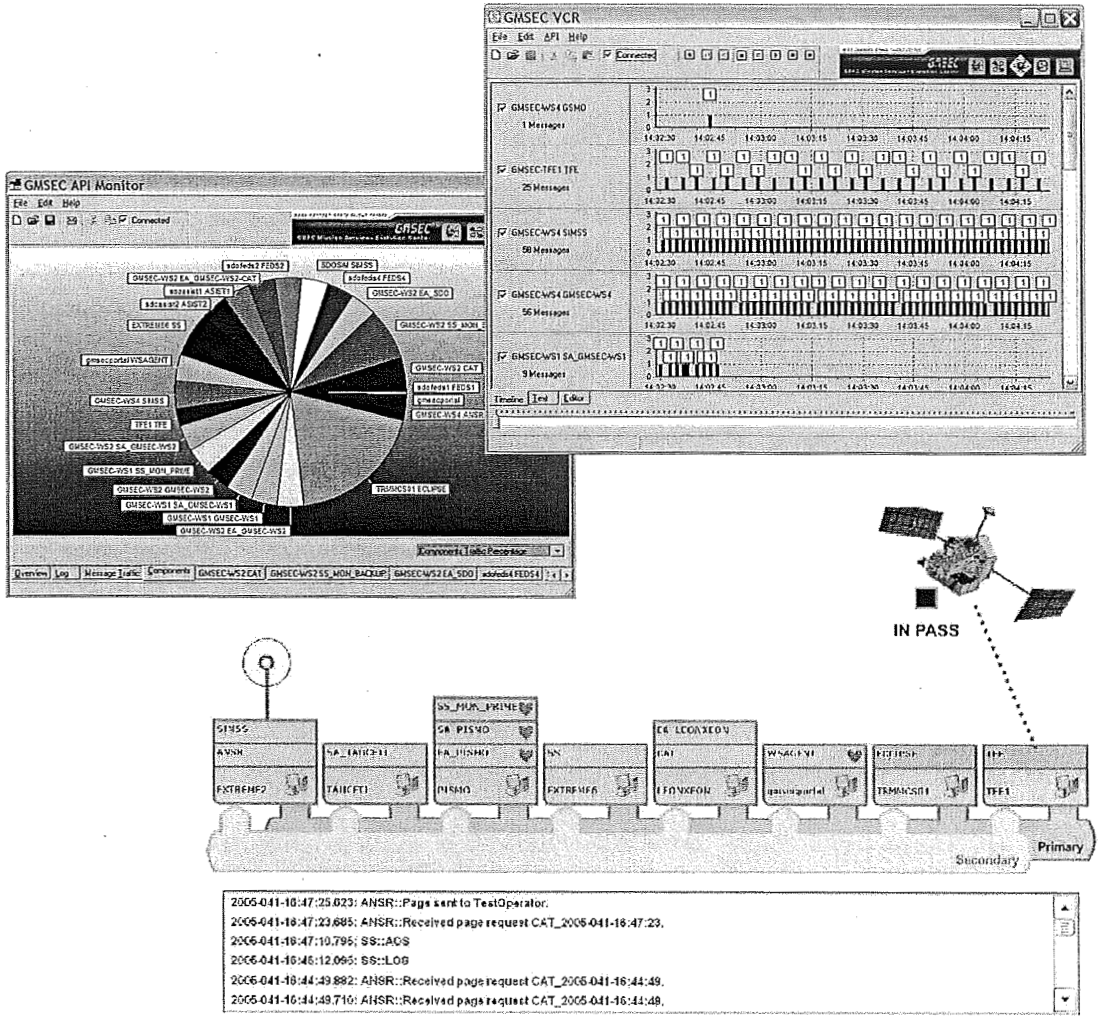


Automation Concepts

- Each component publishes status messages and accepts control directives
- 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**
- **Observation: As users begin to automate, they realize there is even more they can have the system and tools do for them**

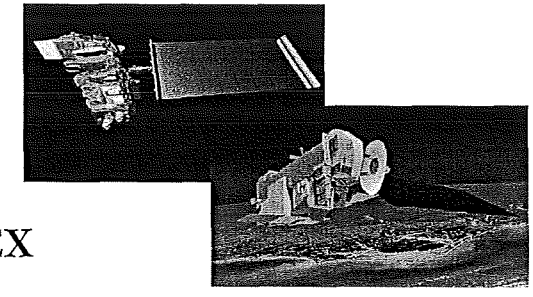
Tool Development is Simplified

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

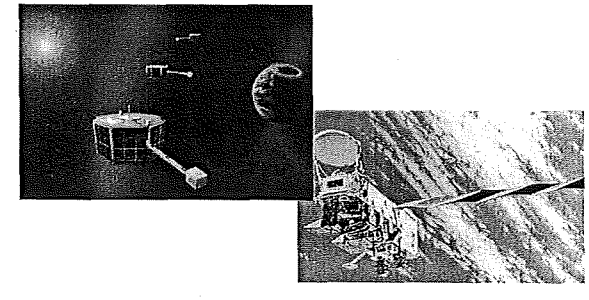


GMSEC Operational Status

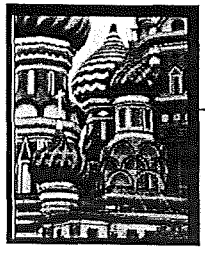
- First 3 missions each selected a different telemetry and command system



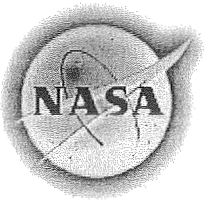
- **Tropical Rainforest Measuring Mission (TRMM)**
  - Automation reduced operations cost by 50%
  - Pathfinder for other Earth Science missions
- **Small Explorer (SMEX) missions – SWAS, TRACE, SAMPEX**
  - Conducted a successful 2-week lights-out operation
  - Pathfinder for low-cost fleet operations & updating existing space science missions
- **ST5 3-Satellite Constellation System - Launched March 2006. 90 day operational period**
  - Demonstrated with subsystem modeling and closed-loop automation
  - Successful 2-week "lights out" operations
- NASA Marshall Space Flight Center using GMSEC operationally for Space Station attached experiments



- **New GSFC missions**
  - Working with 6 future missions



Lessons Learned / GMSEC Benefits



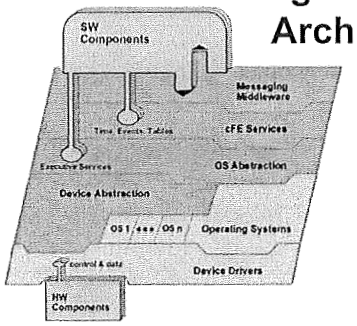
- Significant reduction in integration time
- Components added/upgraded without impacting existing system
- Ideal for using multiple small distributed development teams
- Allows for new ideas for independent tools and capabilities
- Missions more willing to adopt the approach if "old favorite" components can still be used
- Some vendors see message compliance as a way to enter what had appeared to be a closed marketplace
- Standard message approach provides collaboration possibilities with other organizations
- The same concepts can apply to ground, flight, or other domains

Future Directions

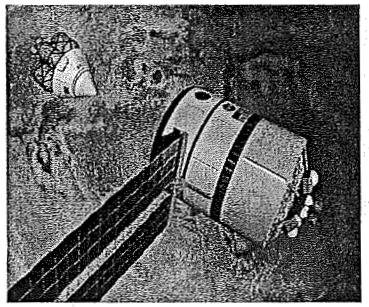
- Similar approach now being applied to flight software
- GMSEC being used for flight dynamics facility re-engineering
- Concepts being adopted for NASA's Exploration Initiative

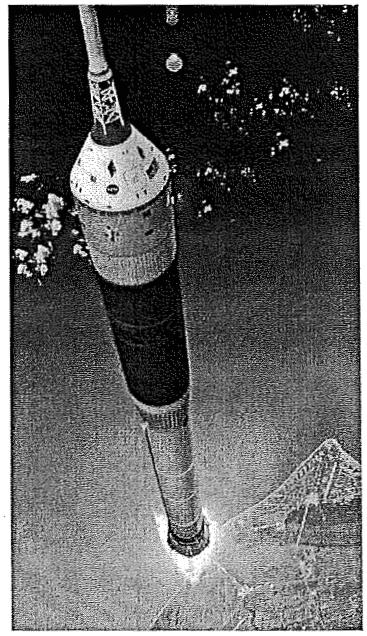
- GMSEC Progress Continues
  - Situational awareness
  - Security
  - Automation/autonomy
  - Data mining
  - Network/system performance tools

Flight Software Architecture



NASA's Exploration Initiative





Conclusions

- GMSEC's message-bus component-based framework architecture is well proven and provides significant benefits over traditional flight and ground data system designs.

- Missions benefit through increased set of product options, enhanced automation, lower cost and new mission-enabling operations concept options.