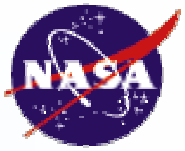
GSAW2003

Ground System Architectures Workshop



Manhattan Beach Marriott - Manhattan Beach, California - March 4-6, 2003

Standards- and Component-Based Mission Operations Architecture

at

NASA's Goddard Space Flight Center

Dan Smith

NASA Goddard Space Flight Center
Code 581.0
Greenbelt, Maryland 20771

Danford.s.smith@nasa.gov
(301) 286-2230

Introduction

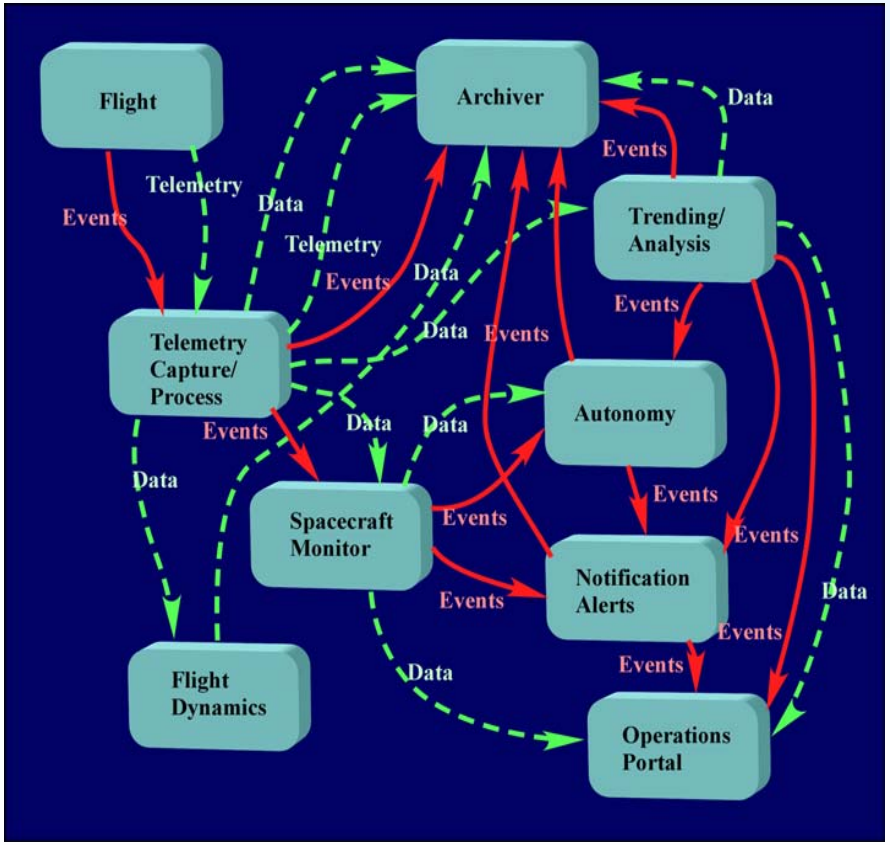
- The GSFC Mission Services Evolution Center (GMSEC) was established in 2001 to coordinate ground and flight data systems development and services at NASA's Goddard Space Flight Center (GSFC).
- The GMSEC system architecture represents a new way to build the next generation systems to be used for many different missions for years to come.
- The old approach was to find or build the best products available and integrate them into a reusable system to meet everyone's needs, but . .
  - Requirements, product offerings, and companies may change tomorrow
  - There is too much variation in mission needs to assume one size can fit all
  - It is often difficult to infuse new technologies into a large, configured system
- The new approach assumes that needs, products, and technology will change.

GMSEC Concepts

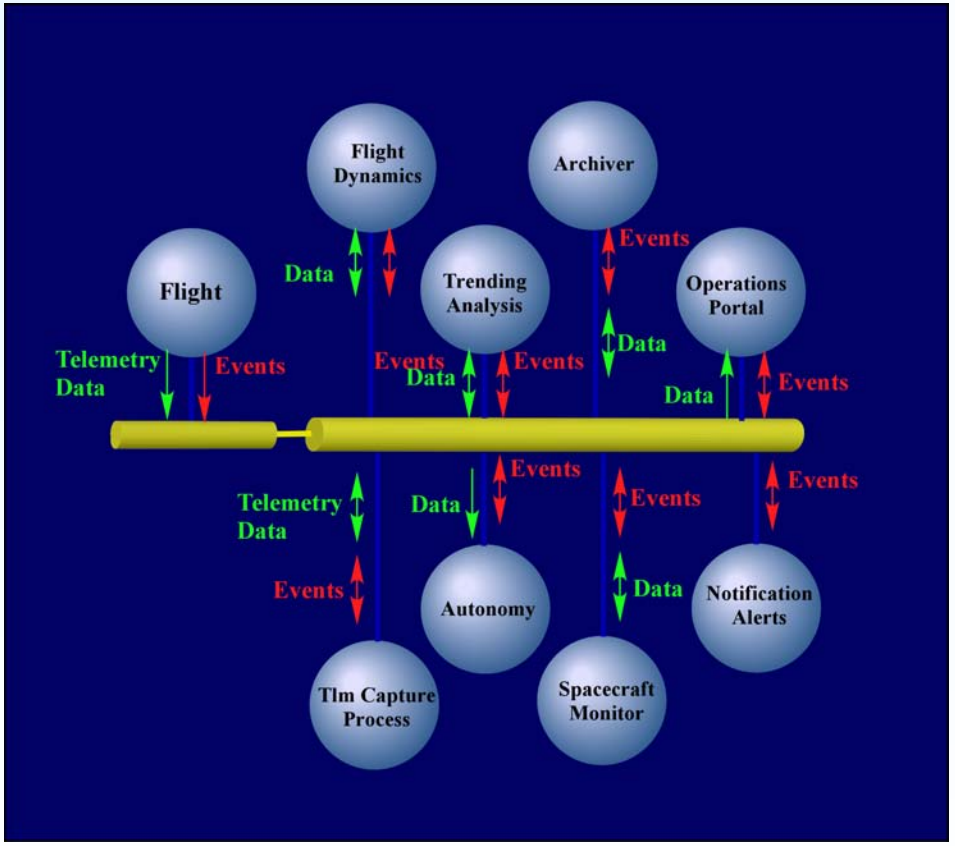
- Standardized Interfaces
  - COTS or in-house tools should have the same key interface definitions (or functionally similar)
  - Use XML where appropriate
  - Goal is to allow for plug-and-play modules that can be integrated quickly and to allow the "trading" of components with other organizations
- Middleware
  - Provides message-based communications services on a GMSEC "software bus"
    - Publish/subscribe, point-to-point, file transfer
  - Can support many higher-level functions including message adaptation, failovers, load balancing, and data portals.
  - Makes it much easier to add new tools
- User Choices
  - We are not comparing available tools and declaring one to be the best for all missions.
  - Want to give user a choice of T&C systems, flight dynamics systems, etc.
  - Choice based on system's merits for a given mission
- GMSEC "Owns" the Architecture and Interfaces
  - The traditional development organizations still own their domain areas
  - A contractor or in-house team creates the mission's system from the GMSEC offerings, populates the databases, adds mission unique features, etc.

Interface Standards and Middleware Simplifies Architecture

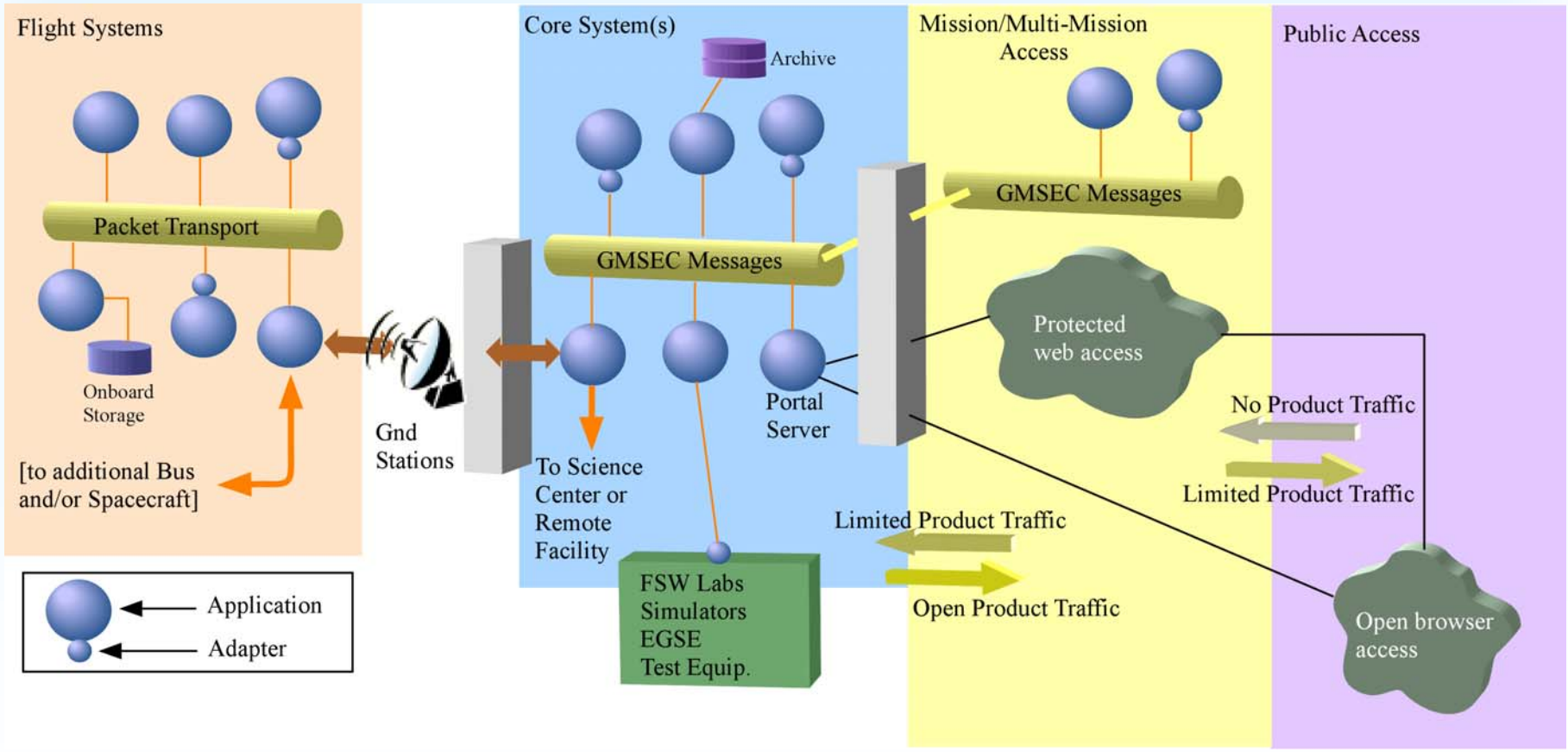
Socket Connections



Middleware Connections



Architecture Overview



"GMSEC standards manage the pipes"

Message Standardization

- Several key messages drive most of the applications

  - **TLM/CMD data base**
  - **Telemetry packet**
  - **Command packet**
  - **Event message**
  - **Data value**
  - **Directive, Response**
  - **Archive request/retrieval**
  - **Schedule request**
  - **Schedule**
  - **Flight dynamics information**

- Collection of contractors, civil servants, and vendors working on most message definitions
- Once prototyped, definitions provided to the Object Management Group (OMG) Space Domain Group Task Force
- OMG is working with industry on XML-based TLM/CMD data base. Effort should be completed this year.

- In-house tools are being modified to match standards as a validation exercise
- Contractors are working on several <u>new</u> tools matched to the standards
- Vendors may either match standards or support creation of "adapters"
  - "Adapters" convert between a tool's format and the GMSEC format

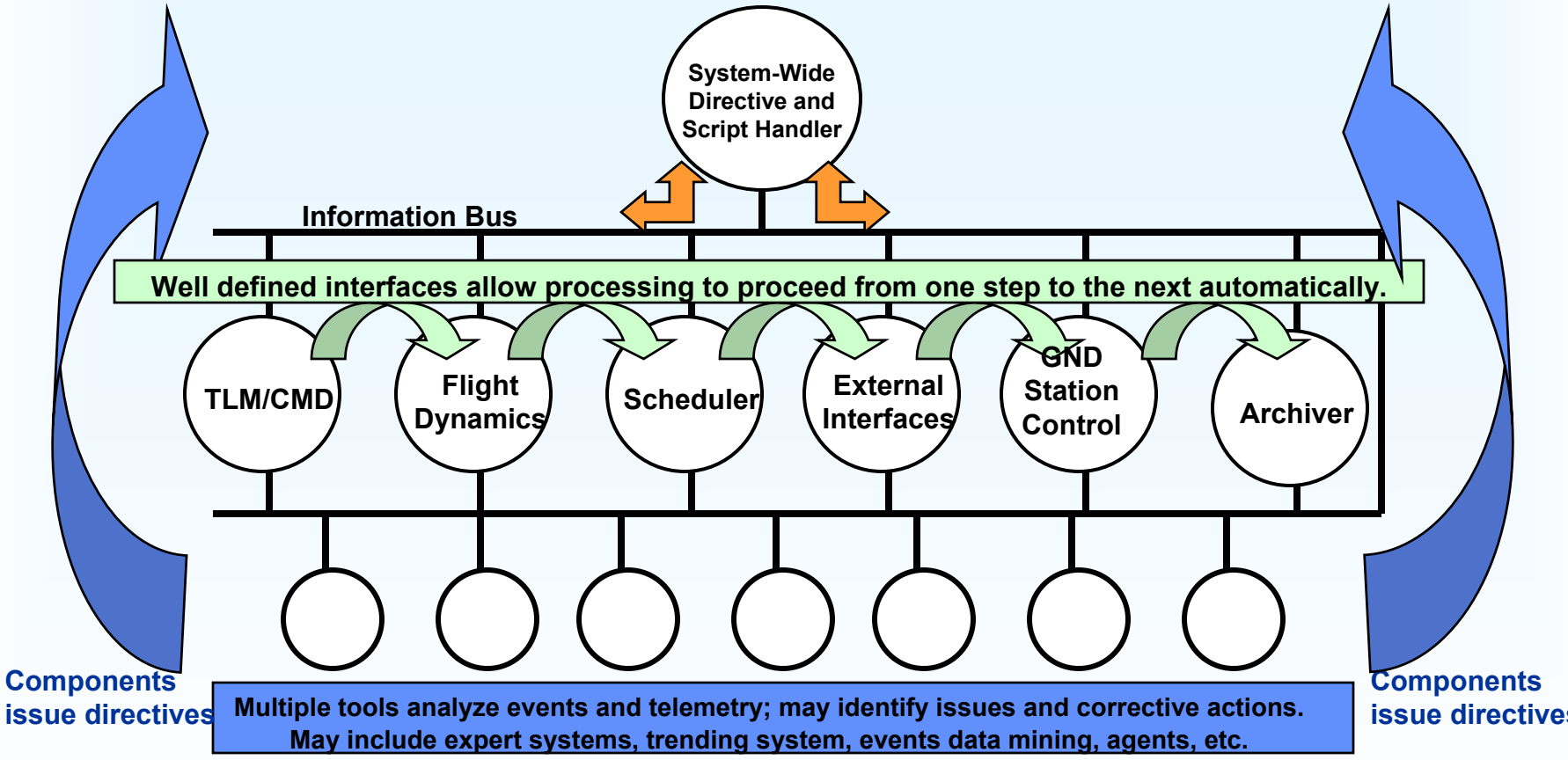
Event Message Example

- GMSEC is creating an system-wide event message handler
  - All applications will publish their events
  - The events handler will subscribe to all events
  - XML message standard adaptable to most systems we've evaluated
- Limited data mining increases analysis value
  - Associate one message with another. Examples:
    - Time between AOS and LOS
    - Difference between scheduled and actual pass times
  - Simple counting, searches, etc.
    - Command or alarm activity level by shift
- Looking at analogies to e-mail systems
  - Tremendous potential value for event messages to carry body text and attachments
  - User-defined filtering and action macros

Benefits of the Approach

1. Developers see a great reduction in integration time
2. New components added and upgrades made without impacting existing system
3. Many suggestions are being made for small independent components that simply integrate to the bus
4. Missions seem more willing to move to several new components (don't need an "all or nothing" approach)
5. The approach works well for our environment where many contractors each contribute a small piece of the total MOC (cross-team coordination/overhead is greatly reduced)
6. Some vendors see GMSEC-compliance as a way to finally enter the NASA marketplace
7. Other organizations beginning to talk to us about trading components that use standardized interfaces

Automation and Autonomy



***Interface standards** are required for **automation** advances which then allow for **autonomy**.*

System Status

- Currently comparing JMS, Smart Sockets, and TIBCO/Rendezvous messaging systems
- Several message definitions in draft form, plan on publishing initial Message Specification this summer
- February 2003 demo included simulators, telemetry and command system, flight dynamics tools and an separate event message system
- By September 2003 should have a fully functional mission operations control center prototype
- By September 2004 should have operational system available with choices for major components
- Working with 5 future NASA missions; 4 of which are expecting to use the GMSEC architecture

- Establishing a public web site – URL is TBD