



Goddard Mission Services Evolution Center “GMSEC”

Overview

July 2024

NASA Goddard Space Flight Center
Software Engineering Division
GMSEC-support@lists.nasa.gov

GMSEC Background and Introduction



GMSEC was established in 2001 to coordinate ground and flight data systems development and services at GSFC. It has been operational since 2005.

– Goals

- Simplify development, integration and testing
- Facilitate technology infusion over time
- Support evolving development and operational concepts
- Allow for mix of heritage, COTS and new components while avoiding vendor lock-in

– Concepts

- Standardize interfaces – not components
 - Provide a middleware infrastructure
 - Allow users to choose – GMSEC does not decide which components are best or dictate which components a mission must use. It's the mission/user's choice!
- Some say it is like what Apple has done – created a simple interface standard and communications approach and let others develop compatible tools beyond anyone's expectations.

Other NASA Centers and U.S. government space organizations are now recognizing the benefits of these simple concepts and are each working with NASA/GSFC's GMSEC Team.

GMSEC Introduction



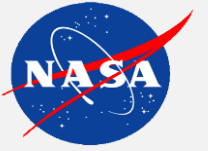
The Goddard Mission Services Evolution Center (GMSEC) is a proven satellite mission operations center open architecture software framework for use at the mission, fleet, or enterprise level.

The GMSEC team does not build ground systems. We build products that enable the mission ground system development teams to build their systems with the best possible mix of available mission support products and added GMSEC components in the areas of automation and situational awareness.

We have had close collaboration with others to ensure its success and increase its value and broad use.

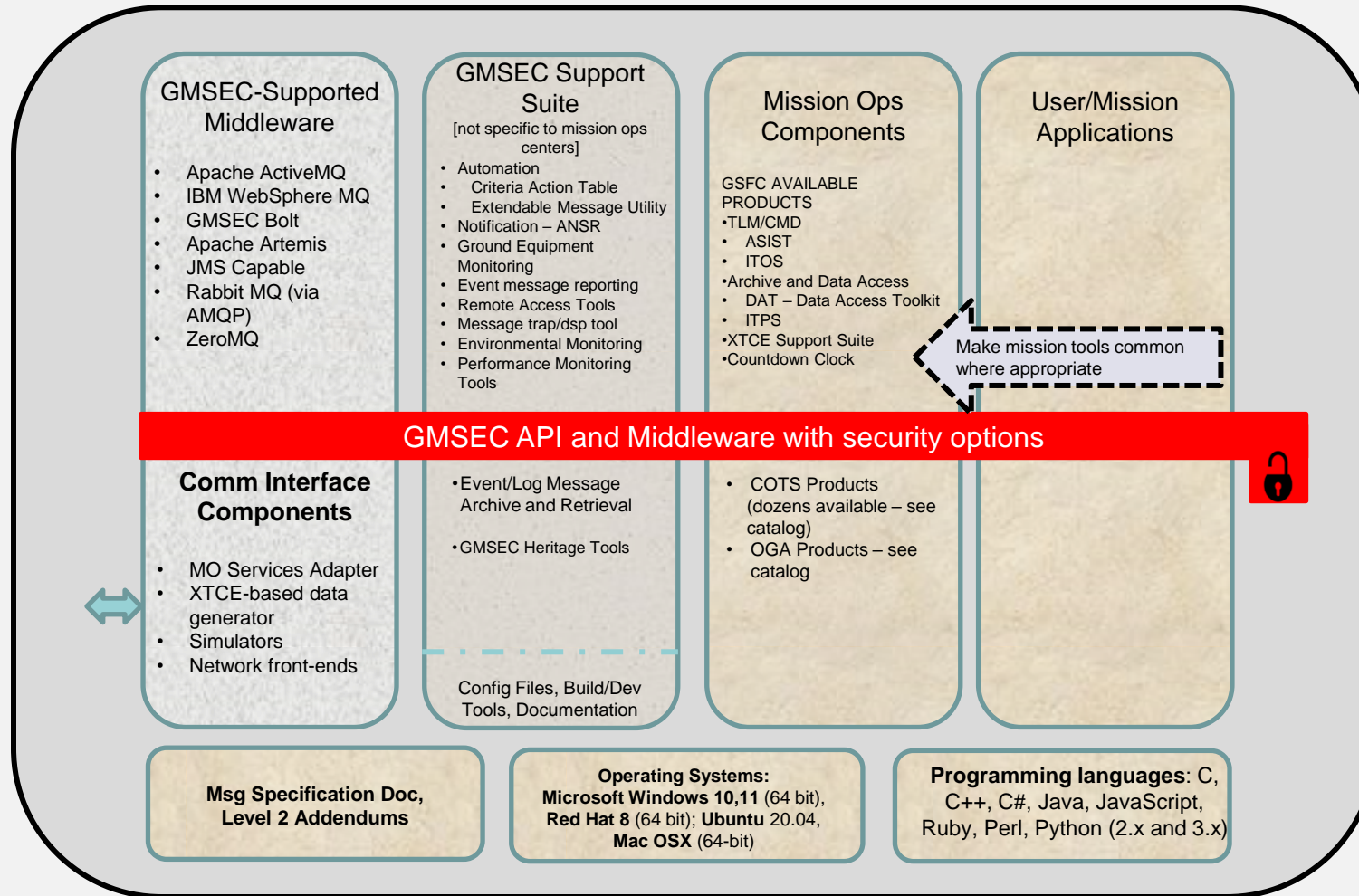
- Command and Control system product vendors
- Major integration contractors
- Other NASA Centers
- Other U.S. government space organizations
- Space standards organizations

Observed GMSEC Benefits/Notes



1. Automation for cost and risk reduction is the #1 selling point
2. Most commercial command and control products are now GMSEC compatible – increasing choices for the missions
3. Significant reduction in integration time
4. Components added/upgraded without impacting existing system
5. Ideal for using multiple small distributed development teams/vendors
6. New concepts emerging for small independent components that integrate with the bus and provide immediate benefits
7. Standard message approach provides collaboration possibilities with other organizations
8. Enables new approach for maintenance of very long-term systems

GMSEC Architecture



General GMSEC Discussion Topics

Why the GMSEC API and Software?



Today's middleware messaging systems work great. In many cases, one can simply select a messaging product and have applications use it directly.

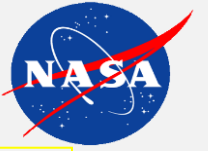
- But NASA teams have been burned by this approach before. We have missions that need to last 20+ years. We have had middleware messaging products be dead-ended due to corporate acquisitions, security failures, new business plans, cost changes, and new attempts at standardization.
- We also have needs that range from critical operations of billion-dollar spacecraft down to desktop product development. A high-end [costly] system may be best for one and a freeware capability may be best for the other.
- We need to be able to safely change middleware products over time.

But the GMSEC API can do a lot more

- Abstraction layer to allow different middleware products to run without changing the applications software.
- Translation to/from more efficient formats for transmission
- Security capabilities kept independent from the applications software
- Message validation at multiple levels

Organized into a single distributable, government-owned, contractor and vendor independent product, the GMSEC API software can be broadly and consistently applied throughout a large spacecraft control community.

GMSEC Framework



- The GMSEC Framework consists of the GMSEC API, standardized messages, and an underlying middleware to interface with other components.
 - Standard API available as NASA Open Source from: https://github.com/nasa/GMSEC_API
 - “Secure API” is also available
 - Note: API Open-Source does not contain modules to build CompatC2 wrappers.
 - GMSEC Architecture Document and Message Specifications available upon request.
- GMSEC supports a number of programming languages, COTS and GOTS middleware products, and operating systems.
 - **Programming languages:** C, C++, C#, Java, NodeJs, Ruby, Perl, and Python (2.x & 3.x)
 - **Middleware Products:** IBM WebSphere MQ, Apache ActiveMQ / Artemis, GMSEC Message Bus and Bolt, JMS Capability, RabbitMQ (via AMQP), ZeroMQ
 - **Operating Systems:** Microsoft Windows 10 / 11 (64 bit), Red Hat 8 (64 bit); Mac OSX (64-bit)
- Framework can be applied for an individual mission, for a constellation, for an enterprise, or for the communications between independent systems.

GMSEC Team's Goals within NASA



1. To be NASA's leader in open system approaches for satellite ground systems.
2. To promote the premise that each Center has their own capable experts and their own special heritage capabilities and operations approaches. Our goal is to enable the use of a mix of FOSS, GOTS, and COTS components from multiple sources to build the most appropriate system for each mission while enabling organizations and missions to utilize the best common capabilities.
3. To support GSFC's goal to be the recognized developer for a select set of ground system components, including low-cost real-time telemetry, command, display, and analysis systems for mission operations.
4. To be a leader in working with the ground system industry and other government space organizations. By "working well with others", GSFC can leverage the broad base of experience across these other groups and add to the advancement of the overall state-of-the-practice.

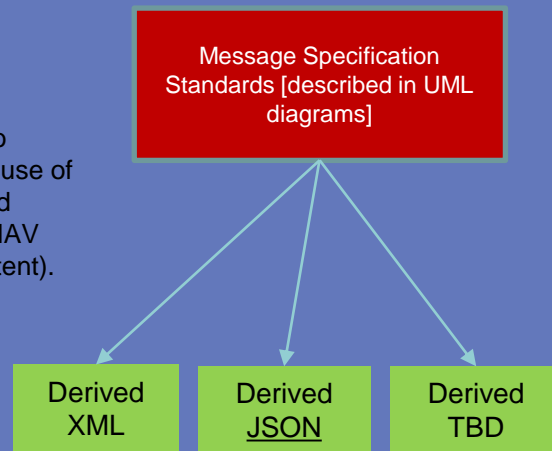


GMSEC as an OMG Standard



OMG Space Domain Task Force’s evolving model for mission operations data systems standardization that they call “architecture agnostic” or “platform independent”.

OMG wants to maximize the use of CCSDS-based formats (i.e. NAV message content).



OMG-Hosted Architecture/Component Catalog – all entries are compliant with the message standards

| | Reference Architecture A "GMSEC" | Reference Architecture B | Reference Architecture C |
|----------------------|----------------------------------|--------------------------|--------------------------|
| Infrastructure | | | |
| Core S/W | | | |
| Documentation | | | |
| Government Products | | | |
| Commercial Products | | | |
| Open source products | | | |

Reference Architectures and Components are not part of formal standards.

Reference Architecture examples could include NASA’s GMSEC or CCSDS’s MO-Services or some yet-to-be-created framework.

GMSEC Component overview



GMSEC provides a suite of general-purpose applications/components

- GMSEC-Compliant
 - Use the GMSEC API to connect to the middleware bus and publish/subscribe to messages
 - Messages adhere to the C2MS specification
 - Publish regular heartbeat messages to indicate health
 - Log significant Events to the bus

Provide Various Functionalities

- Inspect messages
- Send email/text notifications
- React to events
- Monitor component health
- Validate message compliance
- 12+ Legacy components that are primarily in maintenance mode
 - Bug Fixes
 - Minor enhancements
- New development
 - GEMU – Allow users to create their own ‘applications’
 - GSS OpenMCT – Access to messages from a web browser using OpenMCT framework
 - GPTU – Provides Performance Assessment

GMSEC “GOTS” Components



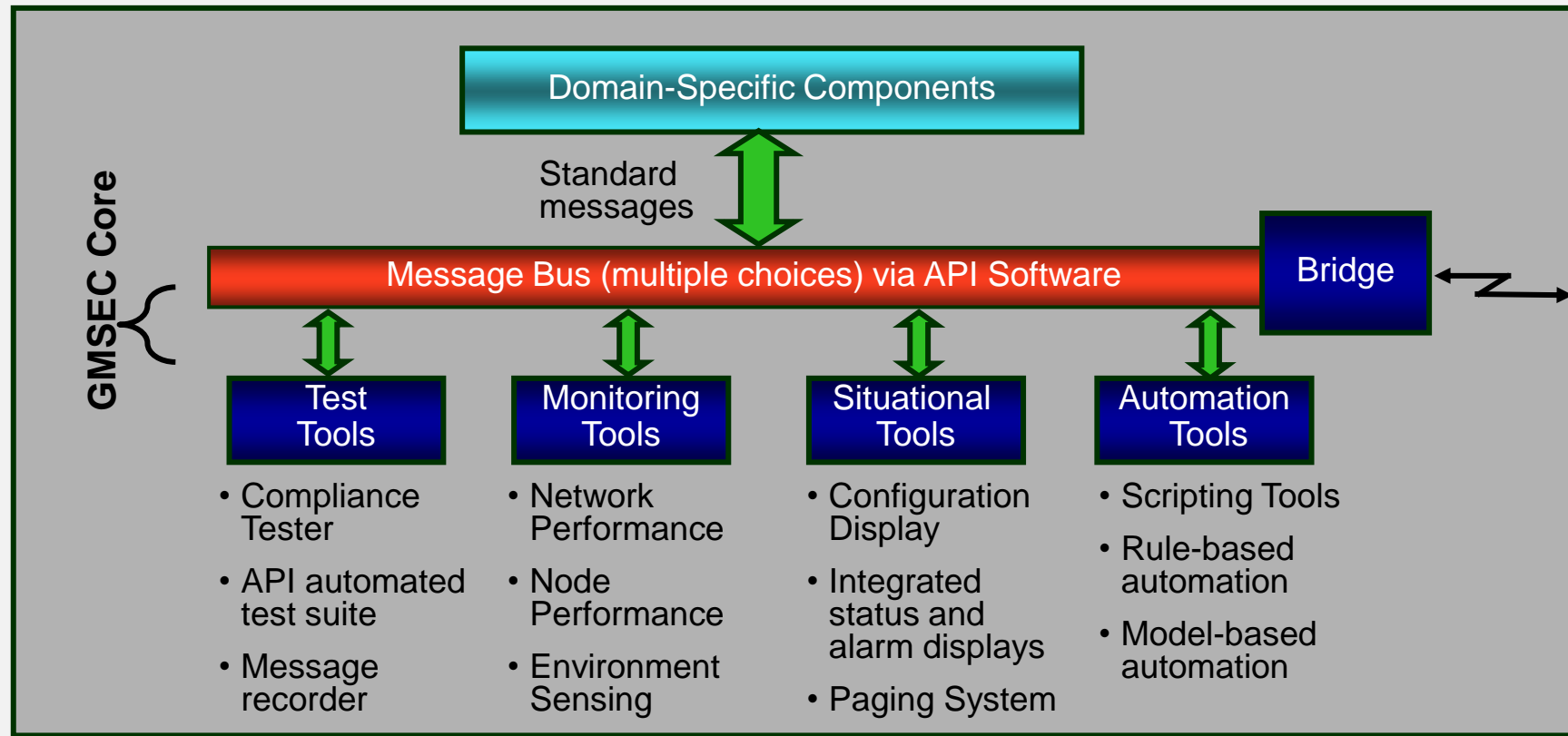
Several years of operational use

- **GREAT** **GMSEC Reusable Event Analysis Toolkit**
- **CAT** **GMSEC Automation. “Criteria Action Tool”**
- **ANSR** **GMSEC Paging Tool**
- **GEDAT** **GMSEC System Display**
- **SA** **GMSEC Node Interface. “System Agent”**
- **RAA** **GMSEC Room Alert Adapter**
- **GRASP** **GMSEC Remote Data Access Tool**
- **GSS** **GMSEC Services Suite (Web)**

New Initiatives

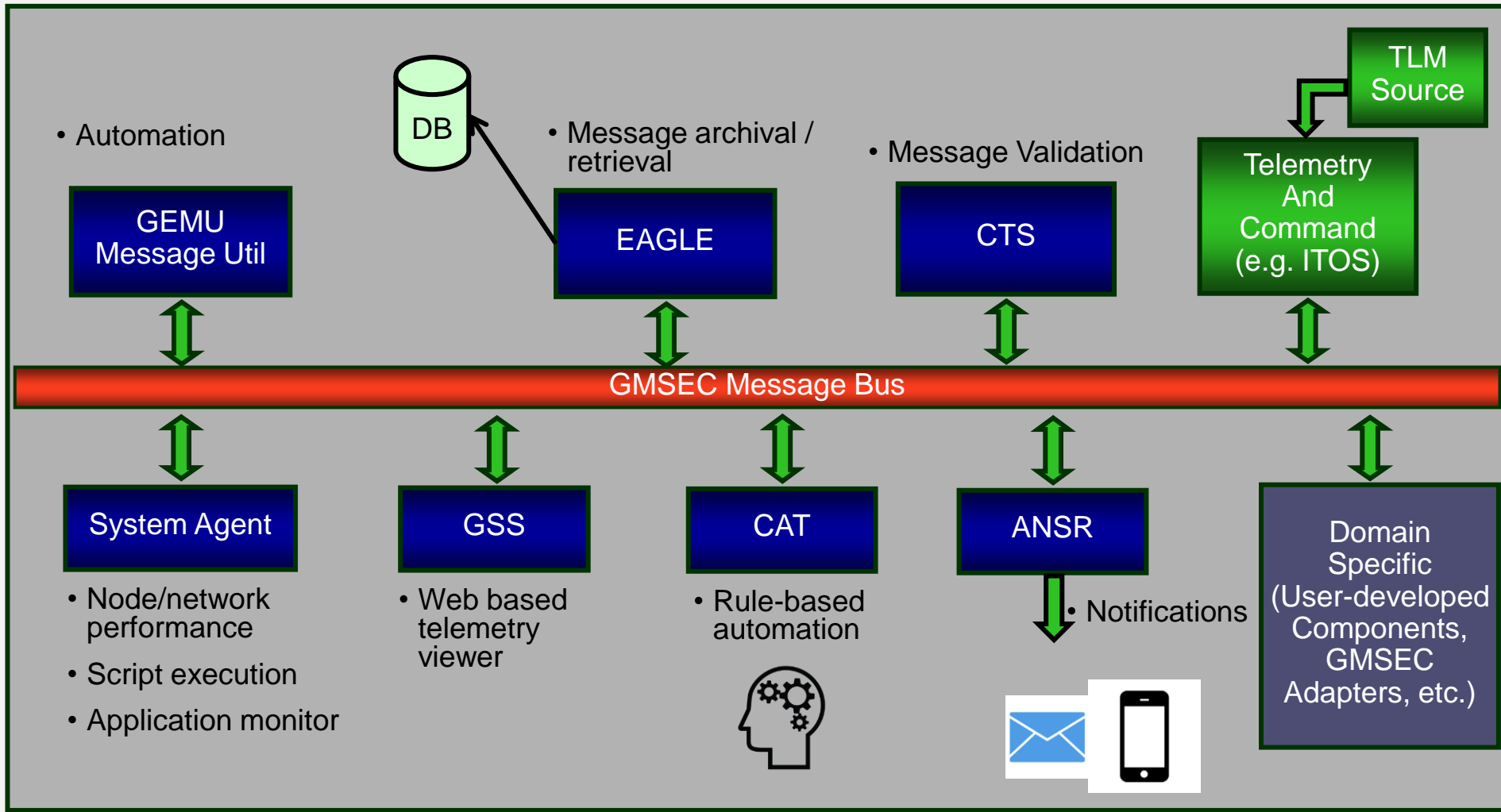
- **GEMU** **GMSEC Extendable Message Utility**
- **GSS 4.0** **Utilizes NASA/Ames OpenMCT Framework**
- **XTCE telemetry and command database tools**

Core GMSEC System Supports any Domain



GMSEC's common service tools bring immediate value to any system.

Example GMSEC System

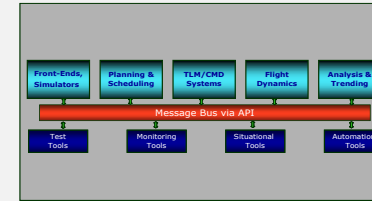


Examples of GMSEC's Mission Benefits

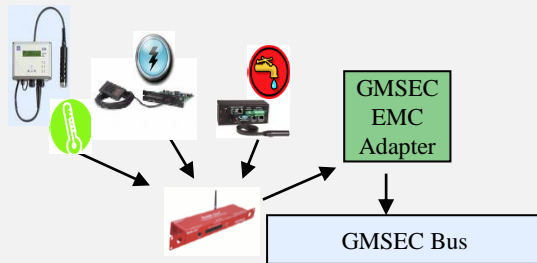


The architecture enables new approaches for automation

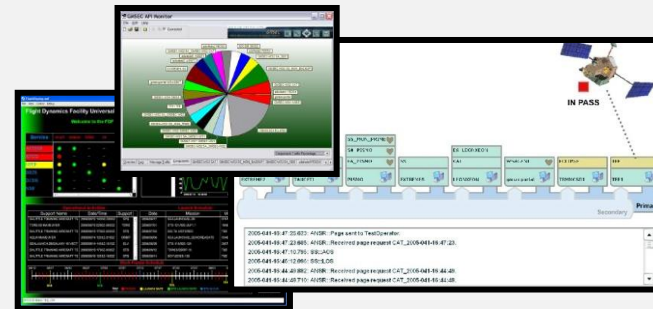
- Can “listen” for status from all components → situational awareness
- Can direct actions of component → system-wide control
- Recognize status and respond → event-driven automation



GMSEC allows for monitoring of temperature, humidity, disk usage, etc. for GSFC control centers.



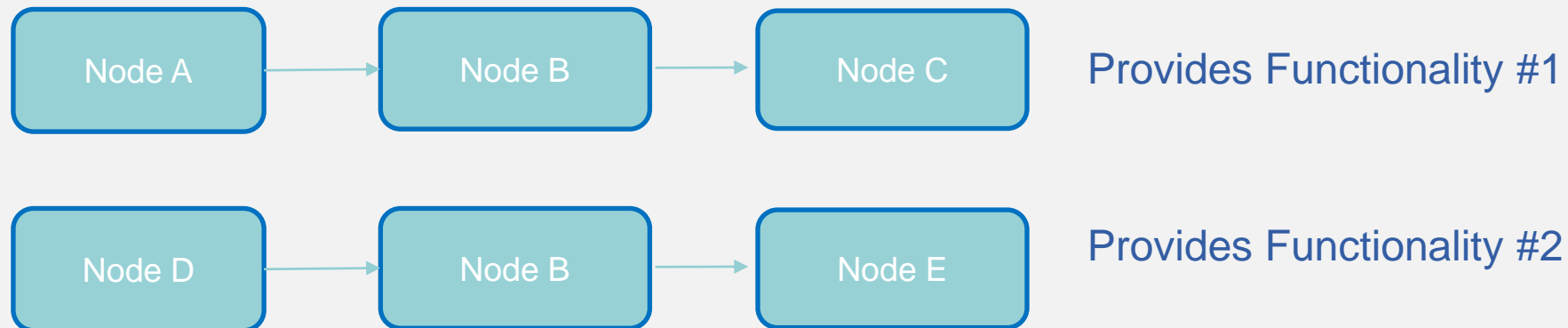
Tools show network performance, system configuration, and processing status.



GEMU – Generic Extendable Messaging Utility



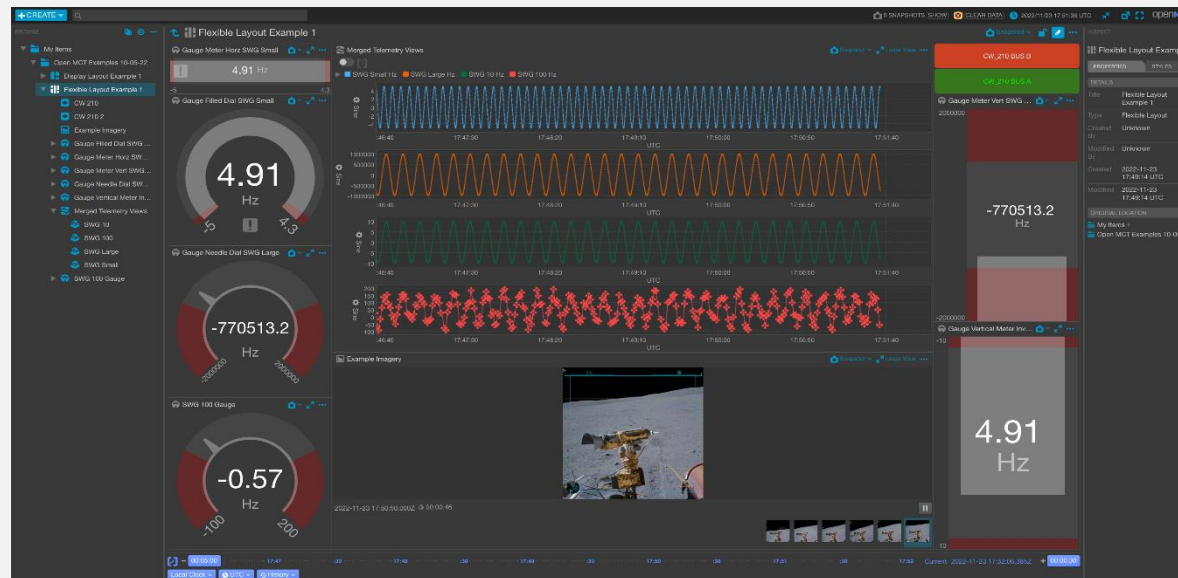
- GEMU is a software component that is designed around the flow-based programming model
- Computer programs are created with “flows” rather than lines of code
- The flows are a network of black box functional nodes, which perform a specific task and have an input or an output or both
- Connections are formed between nodes by connecting the output of one to the input of another
- Different applications can be created by connecting different nodes in various orders to achieve different functionalities



GSS OpenMCT Migration



- The legacy GMSEC Services Suite (GSS) Web application is based on old AngularJS technology. When it came time to upgrade to a newer version, it was decided that rather than upgrading to a newer version of Angular, we would migrate to a whole new framework designed specifically for visualization of satellite telemetry.
- Open MCT is a next-generation mission operations data visualization framework. It is based on the Vue 3 JavaScript framework and is open source. It is developed at NASA's Ames Research Center in collaboration with the Jet Propulsion Laboratory, and is being used by NASA for data analysis of spacecraft missions, as well as planning and operation of experimental rover systems



Additional GMSEC Information

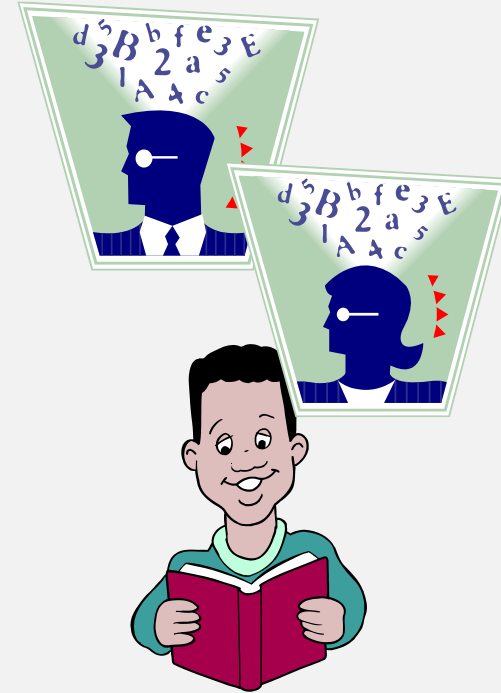


NASA Goddard Space Flight Center
Code 583
Greenbelt, Maryland 20771

GMSEC tech support email: GMSEC-support@lists.nasa.gov

GMSEC Public Website: <https://www.nasa.gov/goddard/gmsec/>
General, high level, copies of component fact sheets

GMSEC Developers Website: limited to NASA development team
System documentation, developer's toolkit, APIs, middleware, etc.
Specific materials can be requested through e-mail requests.



Acronym List



| | | | |
|-------|---|--------|---|
| API | Application Programming Interface | NASA | National Aeronautics and Space Administration |
| | | NOAA | National Oceanic and Atmospheric Administration |
| CCB | Configuration Control Board | XXX | Other government agency |
| CMMI | Capability Maturity Model Integrated | OGA | Other Government Agencies |
| COTS | Commercial Off The Shelf | OPS | Operations |
| | Communications, Standards, and Technology | | |
| CSTL | Laboratory | ORS | Operationally Responsive Space |
| Cx | Constellation | OS | Operating System |
| ESA | European Space Agency | OTF | Operations Technology Facility |
| ESTO | Earth Science Technology Office | PDL | Product Development Lead |
| FDL | Flight Dynamics Facility | RBSP | Radiation Belt Storm Probes |
| GLAST | Gamma-ray Large Area Space Telescope | RFP | Request for Proposal |
| | | | Solar Anomalous and Magnetospheric Particle |
| GMSEC | Goddard Mission Services Evolution Center | SAMPEX | Explorer |
| | | SDO | Solar Dynamics Observatory |
| GOTS | Government Off the Shelf | SMEX | Small Explorer |
| GPM | Global Precipitation Measurement | SOA | Service Oriented Architecture |
| GSFC | Goddard Space Flight Center | ST-5 | Space Technology 5 |
| ISS | International Space Station | SWAS | Submillimeter Wave Astronomy Satellite |
| | | TLM/CM | |
| JSC | Johnson Space Center | D | Telemetry and Command |
| LRO | Lunar Reconnaissance Orbiter | TRACE | Transition Region and Coronal Explorer |
| MMOC | Multi-Mission Operations Center | TRL | Technology Readiness Level |
| MMS | Magnetospheric MultiScale | TRMM | Tropical Rainfall Mapping Mission |
| MOC | Mission Operation Center | USGS | United States Geological Survey |
| MSFC | Marshall Space Flight Center | XSD | XML Schema Definition |