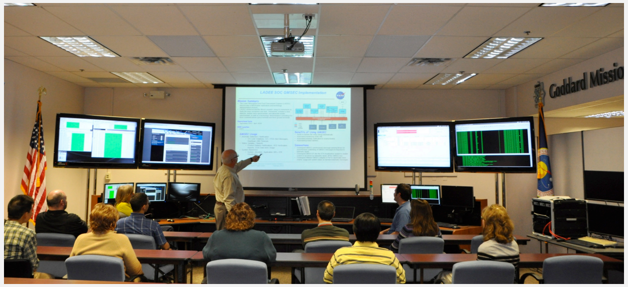
Goddard Mission Services Evolution Center "GMSEC"

Overview

November 2019. updated June 1, 2022

Jay Bugenhagen
NASA Goddard Space Flight Center
Software Engineering Division
john.l.bugenhagen@nasa.gov



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

- Mix of customer support, SW maintenance, new development, and research-like development. Staff of 15.
  - 1 major release, 2-4 minor releases, multiple patches / year
  - Using a modified agile SW development process:
    - Automated unit & component testing, automated document generation (RTM, test reports, test procs, VDD, readme files…)

GMSEC Background

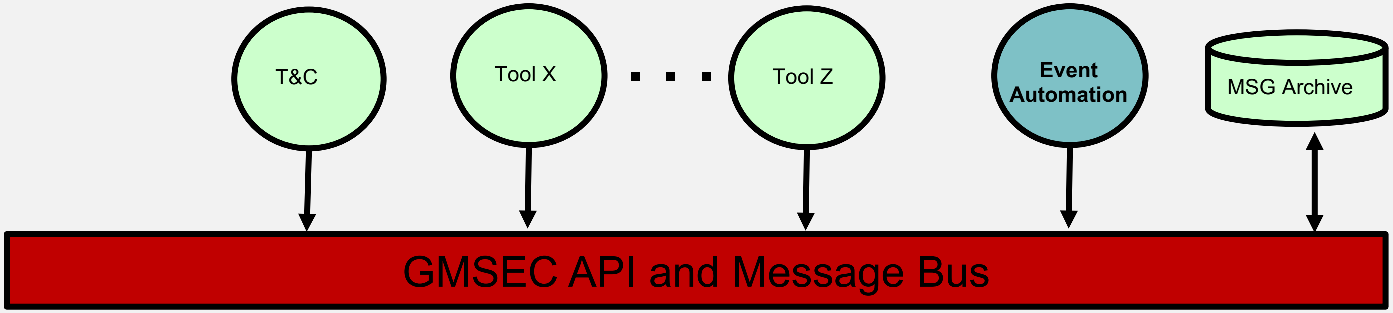
GMSEC was established in 2001 to coordinate ground systems development and services at GSFC.

- 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 doesn't 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 – create 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 working with NASA/GSFC's GMSEC Team. Many COTS products are "GMSEC Compliant."

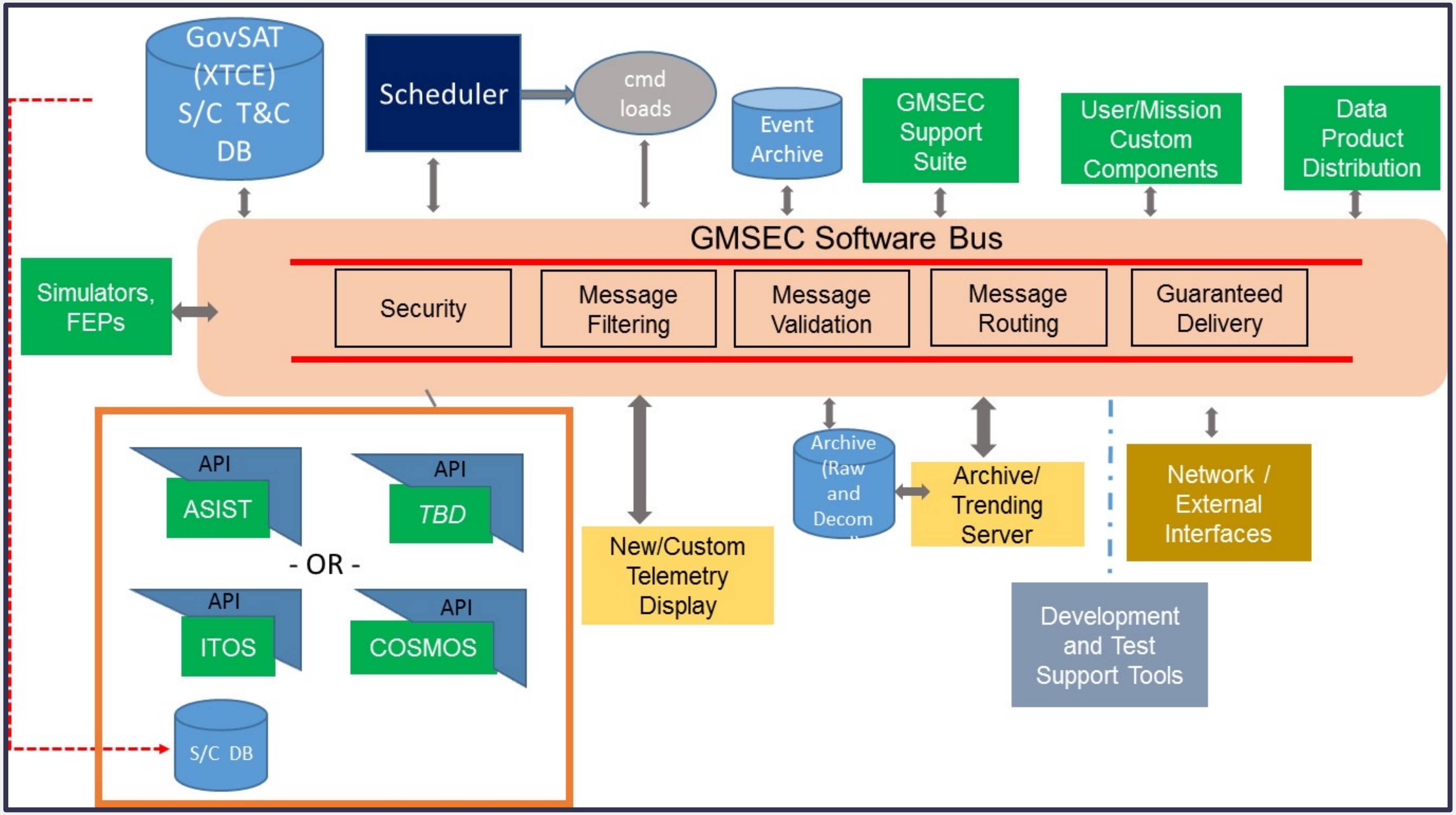
Basic GMSEC Concept



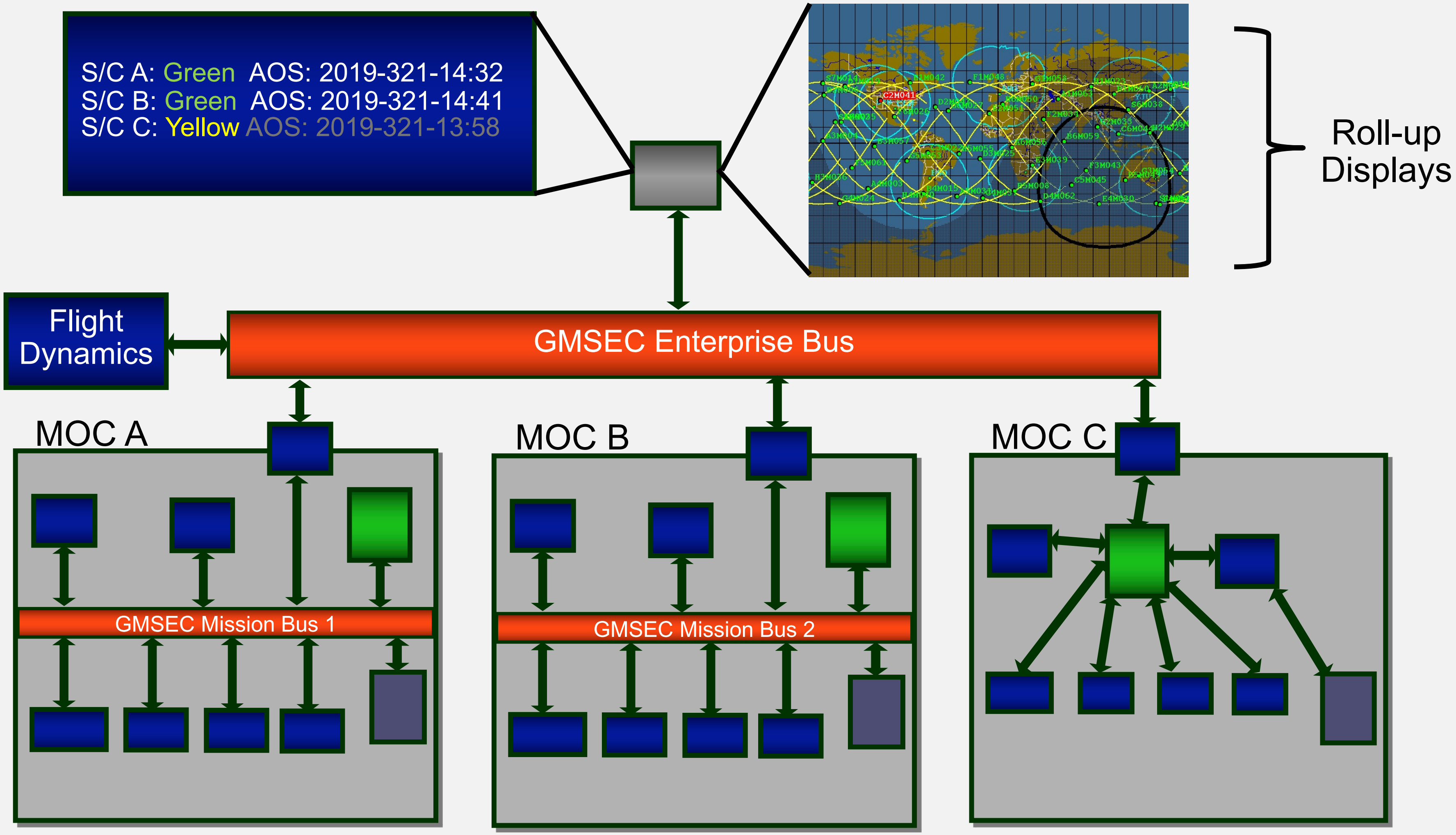
- Message bus and standard messages.
- Publish/Subscribe messaging pattern.
- API supports multiple middleware products and provides the communications connections for the applications. Built-in security, routing, etc.
- Common message formats now a formal industry standard through the Object Management Group (OMG). Rebranded as “Command and Control Message Specification” (C2MS).

- The architecture enables new approach 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 Architecture



Example GMSEC System (Enterprise version)



S/C A: Green AOS: 2019-321-14:32
S/C B: Green AOS: 2019-321-14:41
S/C C: Yellow AOS: 2019-321-13:58

Subject Name (routing header) Details

Subject Standard	Domain Elements		Mission Elements			Message Elements		Miscellaneous Elements					
Speci-fication	DOMAIN 1	DOMAIN 2	MISSION	CONST	SAT	TYP	SUBTYP	ME 1	ME 2	ME 3	ME 4	ME 5	ME 6

Fixed portion

All elements required

Variable portion

Each message definition determines whether a Miscellaneous Element is required or optional

Example:

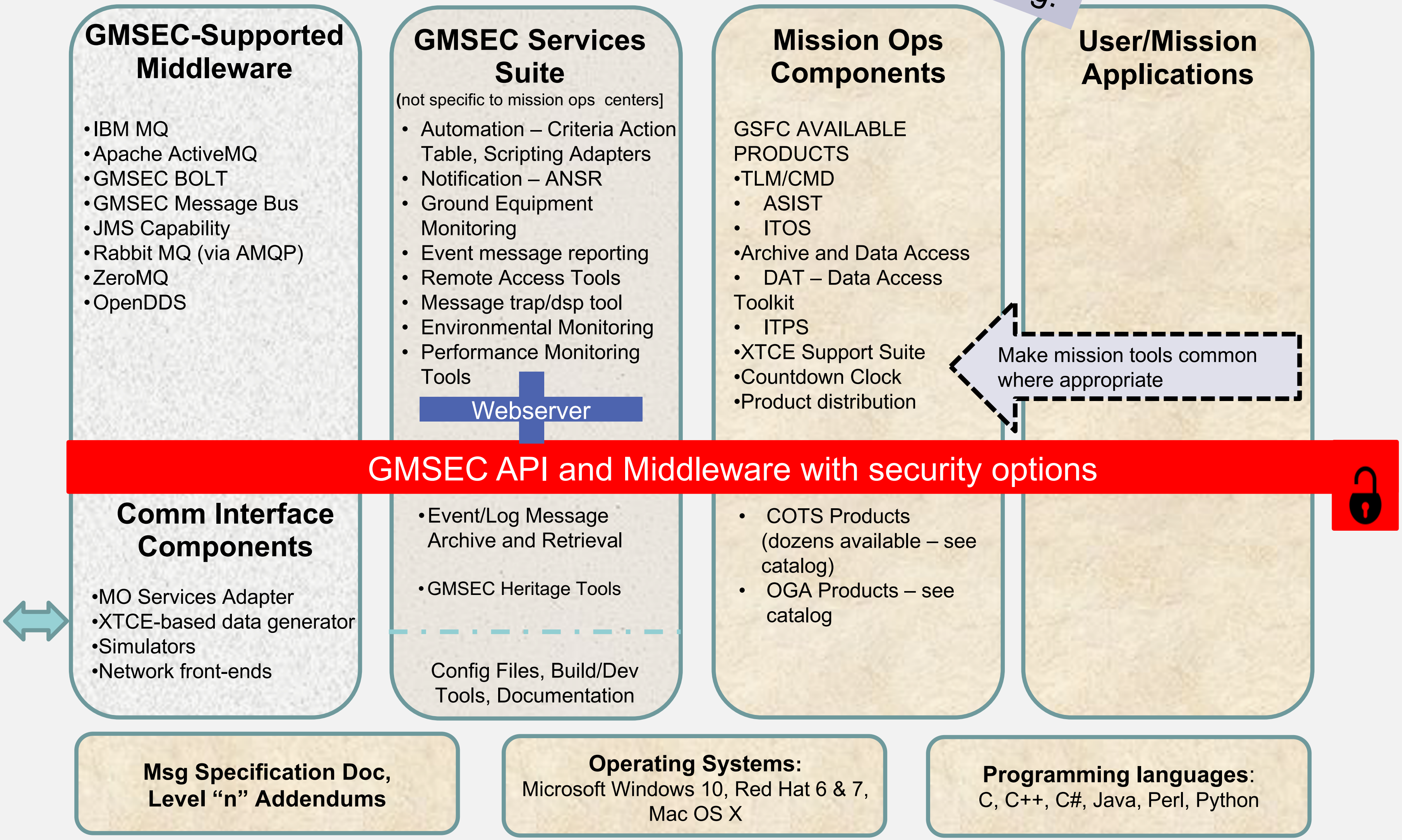
C2MS.GSFC.BLDG_32.ESMO.EOS.AQUA.MSG.LOG.TLM_PROGRAM_3.TAC.RED.3

Software can "subscribe" to filtered sets of messages by specifying or wildcarding fields

C2MS.GSFC.BLDG_32.ESMO.EOS.*.MSG.LOG.*.TLM.RED.3 (all spacecraft, all RED alarms)

GMSEC Architecture

May need updating.



GMSEC-Supported Middleware

- IBM MQ
- Apache ActiveMQ
- GMSEC BOLT
- GMSEC Message Bus
- JMS Capability
- Rabbit MQ (via AMQP)
- ZeroMQ
- OpenDDS

GMSEC Services Suite

(not specific to mission ops centers]

- Automation – Criteria Action Table, Scripting Adapters
- Notification – ANSR
- Ground Equipment Monitoring
- Event message reporting
- Remote Access Tools
- Message trap/dsp tool
- Environmental Monitoring
- Performance Monitoring Tools

Webserver

Mission Ops Components

GSFC AVAILABLE PRODUCTS

- TLM/CMD
- ASIST
- ITOS
- Archive and Data Access
- DAT – Data Access Toolkit
- ITPS
- XTCE Support Suite
- Countdown Clock
- Product distribution

User/Mission Applications

Make mission tools common where appropriate

GMSEC API and Middleware with security options

Comm Interface Components

- MO Services Adapter
- XTCE-based data generator
- Simulators
- Network front-ends

- Event/Log Message Archive and Retrieval
- GMSEC Heritage Tools

Config Files, Build/Dev Tools, Documentation

- COTS Products (dozens available – see catalog)
- OGA Products – see catalog

Msg Specification Doc, Level “n” Addendums

Operating Systems: Microsoft Windows 10, Red Hat 6 & 7, Mac OS X

Programming languages: C, C++, C#, Java, Perl, Python

GMSEC Framework

- The GMSEC Framework consists of the GMSEC API, standardized (yet tailorable) messages, and an underlying middleware to interface with other components.
  - Standard API available as NASA Open Source.
  - “CompactC2 API” available for government use (includes security plug-ins).
  - GMSEC Architecture Document available upon request.
- GMSEC supports a number of programming languages, COTS and GOTS middleware products, and operating systems.
  - **Programming languages**: C, C++, C#, Java, Perl, Python
  - **Middleware Products:** IBM MQ, Apache ActiveMQ, GMSEC Bolt, GMSEC Message Bus, JMS Capability, RabbitMQ (via AMQP), ZeroMQ, OpenDDS
  - **Operating Systems:** Microsoft Windows 10; Red Hat 6, & 7; Mac OS X
- Framework can be applied for an individual mission, for a constellation, for an enterprise, or for the communications between independent systems.

GMSEC Emphasis – Balancing User Needs

Software Development

- Simplified msg generation (MIST)
- Sample programs
- Lightweight Middleware

System Integration and Test

- Install scripts
- Performance Test Utility (PTU)
- Detailed msg display/archive (GREAT)
- Message Validation
- TLM Data Generator
- Message Generator

Sys Admin, Ground Sys. Monitoring

- Node and software status (GEDAT)
- Detailed msg display/archive (GREAT)
- Web Services framework (GSS)

Satellite Operations

- Automation (CAT)
- Alert Notification (ANSR)
- Scripting
- Event msg display/archive (GREAT and GSS)
- Data Analytics
- Many COTS products

Different customers have their focus on different blocks of the above diagram

GMSEC “GOTS” Components

Several years of operational use

– GREAT	GMSEC Reusable Event Analysis Toolkit
– CAT	GMSEC Automation. “Criteria Action Table”
– ANSR	GMSEC Paging Tool
– GEDAT	GMSEC System Display
– SA	GMSEC Node Interface. “System Agent”
– RAA	GMSEC Environmental Data Tool
– GSS	Web Services Suite of Tools (evolving)
– GPD	GMSEC Parameter Display
– GRASP	GMSEC Remote Data Access Tool
– GenSim	Simple data generator (XTCE-based)

New Initiatives

- Events analysis and data mining tools
- XTCE telemetry and command database tools
- Space data link encryption service (starting planning phase)

Primary Development Areas – GSS, CAT, Events/Log Analysis

- GMSEC Services Suite (GSS)
  - Web-enabled GMSEC dashboard
  - February 2019 was first time included with GMSEC full release
  - Team has worked closely with MMS engineering and ops team
  - JPSS is using it in the CGS – common ground system
  - Working on security and some refresh for major release in 2020

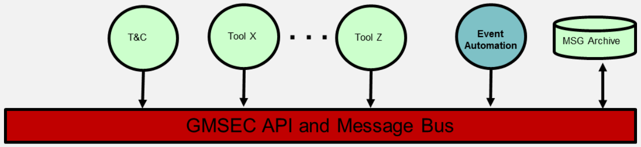
- Criteria Action Table (CAT)
  - GMSEC rules-based automation tool. One of the 1st GMSEC components. 15 years old.
  - Have been working with many missions on what a replacement should look like.
    - Comments have ranged from “don’t change anything” to “you better just start over”.
    - The ops-driven approach has really helped the GMSEC team’s awareness of the breadth of needs
  - Still forming the vision prior to starting system design. Will be considering a number of COTS tools that were not available when CAT was first implemented.
  - Will be renamed to SMART – Situation Monitor And Reaction Tool

New Application Area: Events/Log Analysis

- The challenge
  - Missions have always generated thousands of event/log messages. Recently, system admin messages, system logs, multi-mission ops and more have increased the number of messages.
  - NOAA/JPSS recently estimated that they will generate 3 million messages/day, one billion per year!
  - We don’t have good tools to analyze these logs. Specific queries are often handled with specific scripts.

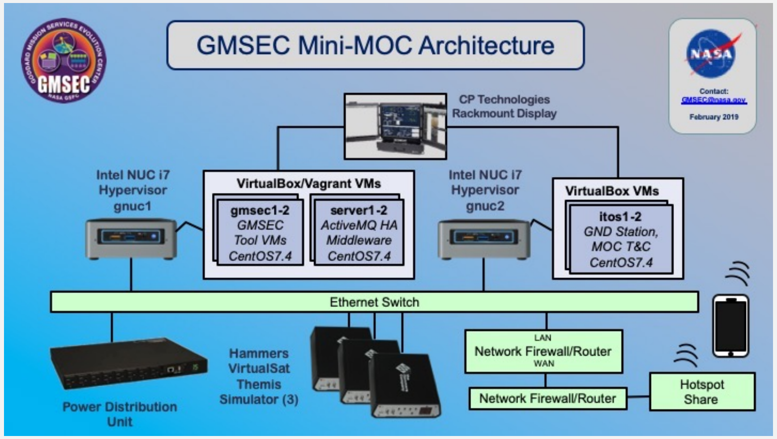
- The opportunity
  - The GMSEC architecture provides GMSEC-C2MS standardization and centralized event logging
  - Recent technology advances in large databases, data analytics tools, and the cloud enable very sophisticated capabilities. Elastic Search and Kibana are recognized leading products.

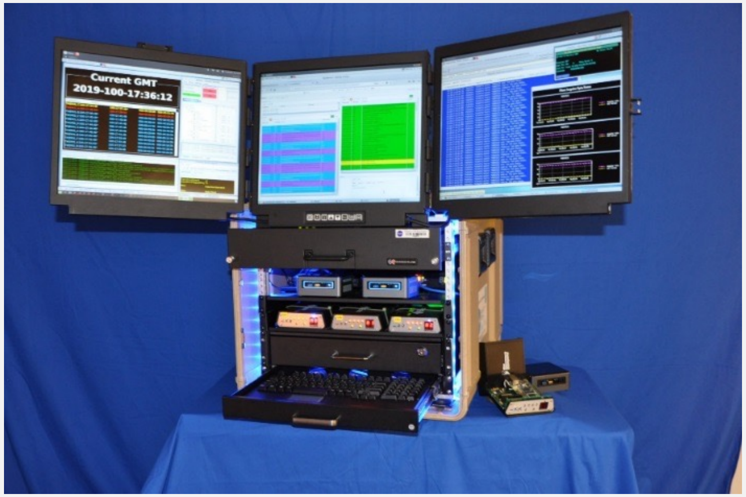
- Recent progress
  - Received archive message sets from SSMO, EOS and JPSS for testing – able to rapidly ingest and analyze the msgs.
  - Lots of mission interest: vMMOC, ESMO, USAF, JPSS
  - Would like to move towards cloud-based multi-mission, multi-purpose events-as-a-service.



The GMSEC Mini-MOC, aka Big Bertha, aka MOC 165

- Portable, scalable, and customizable mission ops system
- Uses include demonstrations, conferences, training, integration and test, possibly even for cubesat or small mission ops
- Showcases use of GSFC components and industry standards
  - ITOS telemetry and command system
  - GMSEC and the C2MS messaging standard
  - CCSDS space communication protocol for command and telemetry.

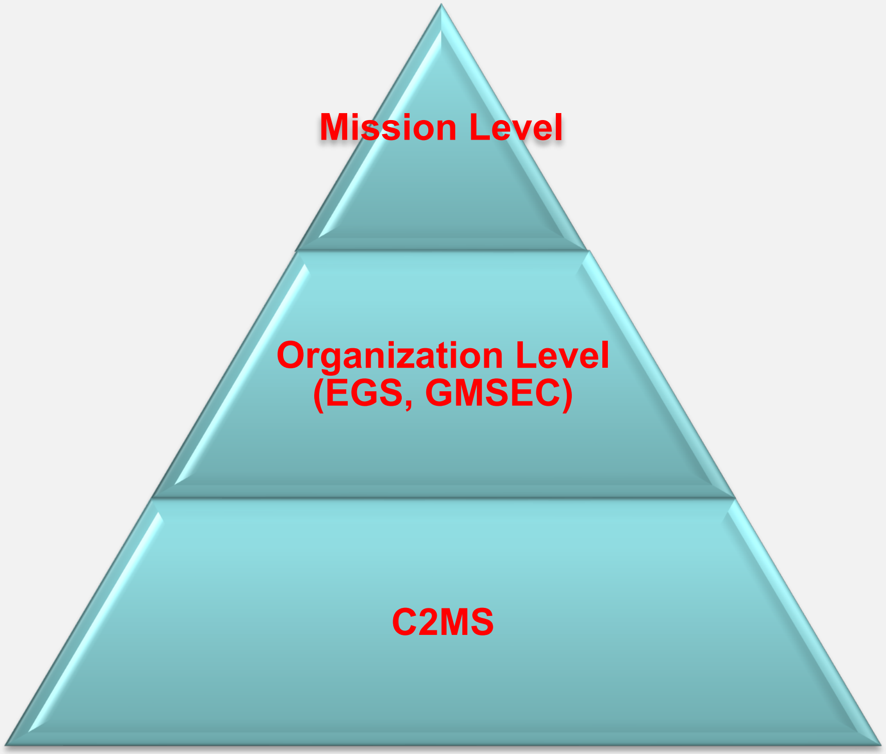




- Hardware simulator configured for Themis constellation of spacecraft using ColdFire processor within Hammers VirtualSat and Core Flight Executive (cFE). May move to software-based simulators.
- Employs application deployment, configuration management, and continuous delivery tools like Vagrant, VirtualBox, Ansible, etc.
- Material cost (without simulators) is about $25K.
- Air Force is very interested in NASA building similar systems for them. Could also serve as a special-purpose self-contained vMMOC package

Governance and Message Validation

- GMSEC supports a multi-level governance and message validation concept.
  - The OMG's C2MS is considered the prime level.
  - At the next level, organizations define their own messages or specific instantiations of the C2MS level.
  - Missions may define some of their own specific messages.
- Message validation files can be loaded for each level.
- If enabled, the API can perform message validation and reject invalid messages.



- EGS has been discussing their validation ops con. Is it a development and integration tool? A product certification tool? An operational run-time security tool? Must consider complexity and performance of validating each field of every message in real-time.

Providing a Framework To a Mission Team

-- NASA GMSEC Example --

It takes a lot more than just good software.

INTRODUCTION	SPECs/STANDARDS	SOFTWARE	TECHNICAL SUPPORT	REFERENCE	DEVELOPMENT/MAINT.
Overview Presentation	Message Spec.	Open Source - API	Help Line (e-mail)	Public Website/Wiki	Software Processes
Architecture Document	XML Schemas	Open Source - Middleware	Getting Started Guide	Conference Papers	Configuration Management
[Video]	Augmentation Specs	NASA S/W (by request)	Developer's Toolkit	Performance Test Report	Maintenance Capability
Strategic Plan/Roadmap/ Sustainable Business Plan	Tailoring Guides	Vendor Catalog of commercial components	Classes Overview API Concepts Security, Info Assurance Components Building a System Testing Automation	Special Reports Cyber Security Prototype Results Use Case Perf. Tests Enterprise Level Arch. Governance Situational Awareness DoD Msg. Spec Suppl.	Enhancement and New Product Development Capability
	Governance Plan	Software Dev & Release Process	Subject Matter Experts	Lessons Learned Documents	Experienced contractors
			Big Bertha, Ad Hoc User Meetings, CCB	Product Demo Lab	

A lot of work is required to provide what systems developers need to accept, implement and deploy a framework system specified by others. Similar to a COTS product model, but even broader and more complex.

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; can support parallel testing
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

Early Air Force Demo Findings and Benefits

As reported to General Sheridan by Aerospace Corporation based on GMSEC demo lessons-learned study.

- **Findings**
  - Interoperability – Easily integrate ground systems using frameworks
  - Adaptability – Not technically difficult to incorporate existing (“legacy”) C2 systems into frameworks
  - SSA – Space-/ground-asset situational awareness a key benefit of this architecture
  - Standards – Must adopt framework & standards to achieve interoperability
- **Benefits**
  - Architecture is flexible to implementation & operations concepts
  - Allows evolutionary acquisition of SATOPS
  - Lowers long-term development/sustainment costs
  - Allows use of products from multiple vendors – eliminates program “vendor lock-in”
  - Improves availability/reliability/survivability of entire space enterprise
    - Supports rapid (hours/days) configuration of ground station to support a different mission
  - Allows programs to share common ground/antenna satellite resources
  - Enables space/ground situational awareness as well as enterprise management

The Air Force has selected GMSEC as the core of their Enterprise Ground Services (EGS) system and is the largest non-NASA GMSEC customer.

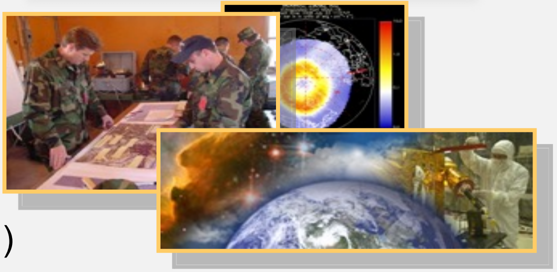
Final GMSEC Technical Notes

- GMSEC is not a radical technology – it is a smart application of key technologies influenced by a deep understanding of the U.S. satellite control vendor industry

- GMSEC follows the approaches outlined in the DoD Open Systems Architecture Guidebook

- The value is that it addresses key issues for a certain class of users
  - Government-owned message specification and governance
  - Allows for use or mix of heritage or COTS software
  - Evolvable over time
  - Does not assume "one size fits all" will work
  - Does not enforce a specific operations concept
  - Simple enough for vendors to invest in
  - Is an enabler for new levels of situational awareness and automation
  - Encourages shared software or capabilities across organizations

Apache ACTIVEMQ
AGI
gmv INNOVATING SOLUTIONS
ATTENTION Software
KRATOS RT LOGIC
LOCKHEED MARTIN
IBM
designAmerica
HORIZON
HARRIS
Raytheon
ECLIPSE
L3
the Hammers Company
a.i. solutions



- Great quotes from the Air Force . . .(one from a Lt. Colonel, one from a General)
  - "Gentlemen, I think I have seen the future. In ten years you will be using this approach in ways you can't even imagine today."
  - "This is the biggest game-changer in how we operate that I have seen in my career."

The GMSEC architecture and software is enabling new levels of collaboration between government and industry to efficiently meet the long-term goals we all share. The benefits of simplified integration, a broader set of available components, increased automation and the enabling of new operations concepts are realized through the open GMSEC architecture.

DoD Open Systems Architecture (1 of 2)

DoD Open Systems Architecture.

Contract Guidebook for Program Managers. December 2011

(updated in August 2013)

“**Background**: An open architecture is defined as a technical architecture that adopts open standards supporting a modular, loosely coupled and highly cohesive system structure that includes publishing of key interfaces within the system and full design disclosure. The key enabler for open architecture is the adoption of an open business model which requires doing business in a transparent way that leverages the collaborative innovation of numerous participants across the enterprise permitting shared risk, maximized asset reuse and reduced total ownership costs. The combination of open architecture and an open business model permits the acquisition of Open Systems Architectures that yield modular, interoperable systems allowing components to be added, modified, replaced, removed and/or supported by different vendors throughout the life cycle in order to drive opportunities for enhanced competition and innovation.

The U.S. Government’s (“Government”) ability to obtain at least Government Purpose Rights (GPR) in technical data and computer software is critical to this effort. Data and design artifacts related to the interfaces between modules are particularly important. One way of measuring the “openness” of a system is how readily a system component can be replaced with one developed by a different vendor, with no loss in overall system effectiveness.”

DoD Open Systems Architecture (2 of 2)

“OSA is composed of five fundamental principles:

1. Modular designs based on standards, with loose coupling and high cohesion, that allow for independent acquisition of system components:
2. Enterprise investment strategies, based on collaboration and trust, that maximize reuse of proven system designs and ensure we spend the least to get the best;
3. Aggressively transform our life-cycle sustainment strategies for software intensive systems through proven technology insertion and product upgrade techniques;
4. Dramatically lower development risk through transparency of system designs, continuous design disclosure, and Government, academia, and industry peer reviews;
5. Strategic use of data rights to ensure a level competitive playing field and access to alternative solutions and sources, across the life cycle.

Achievement of these five principles requires an affirmative answer to a fundamental question:

Can one or more qualified third parties add, modify, replace, remove, or provide support for a component of a system, based on open standards and published interfaces for the component of that system?”

We believe GMSEC to be an excellent (and successful) example demonstrating the benefits of the DoD OSA approach.

Additional GMSEC Information

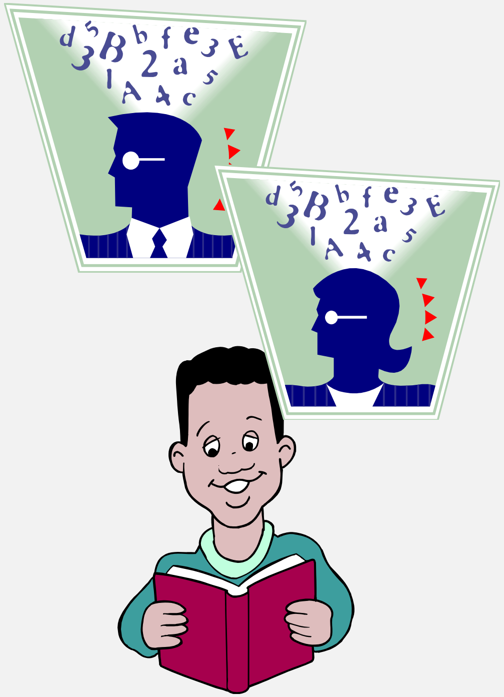
Theresa Beech, Project Manager
theresa.w.beech@nasa.gov

Jay Bugenhagen**, PDL**
john.l.bugenhagen@nasa.gov

Sharon Orsborne, Deputy PDL
sharon.a.orsborne@nasa.gov

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

GMSEC Public Website: http://gmsec.gsfc.nasa.gov
General, high level, copies of component fact sheets



BACKUP

Acronym List

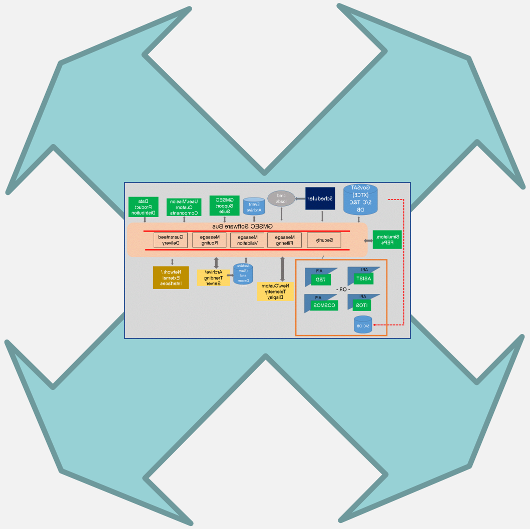
API	Application Programming Interface		MSFC	Marshall Space Flight Center
APL	Applied Physics Laboratory		NASA	National Aeronautics and Space Administration
CCB	Configuration Control Board		NOAA	National Oceanic and Atmospheric Administration
CMMI	Capability Maturity Model Integrated		NRO	National Reconnaissance Office
COTS	Commercial Off The Shelf		OGA	Other Government Agencies
CSTL	Communications, Standards, and Technology Laboratory		OPS	Operations
Cx	Constellation		ORS	Operationally Responsive Space
ESA	European Space Agency		OS	Operating System
ESTO	Earth Science Technology Office		OTF	Operations Technology Facility
FDF	Flight Dynamics Facility		RBSP	Radiation Belt Storm Probes
GLAST	Gamma-ray Large Area Space Telescope		RFP	Request For Proposal
GMSEC	Goddard Mission Services Evolution Center		SAMPEX	Solar Anomalous and Magnetospheric Particle Explorer
GOES	Geostationary Operational Environmental Satellites		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
JSC	Johnson Space Center		TLM/CMD	Telemetry and Command
LDCM	Landsat Data Continuity Mission		TRACE	Transition Region and Coronal Explorer
LRO	Lunar Reconnaissance Orbiter		TRL	Technology Readiness Level
MMOC	Multi-mission Operations Center		TRMM	Tropical Rainfall Mapping Mission
MMS	Magnetospheric MultiScale		USGS	United States Geological Survey
MOC	Mission Operations Center		WIRE	Wide-Field Infrared Explorer

Maturity and Readiness: Yearly Progress

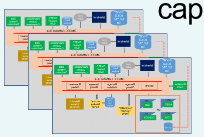
- FY02 – Architecture definition (paper studies)
- FY03 – Lab Created
  - Proof of concept prototypes; Initial message standards
- FY04 – Software development started
  - Development of API, test environment & operational tools
- FY05 – **First operational missions**
  - Labs established at other NASA Centers
  - Exploration Initiative moves towards GMSEC concepts
- FY06 – Expanded operational use. First new mission launch
  - Exploration prototyping across other NASA Centers
  - Made available through NASA Open Source
- FY07 – Stable Operational Use
  - Increased mission participation
  - Spinoff initiatives started – FDF reengineering, Cx Labs/interfaces
- FY08 – New Maturity; CMMI Level 2 certification
  - Expansion to Other Government Agencies (OGA's)
- FY09 – Outside interest grows
  - Collaborations with government agencies, vendors, contractors
- FY10 – Cross-Agency demonstrations;
  - Involvement with Joint SatOPS Compatibility Committee
- FY11 – Security enhancements;
  - Vendor scenario demonstrations
- FY12 – Enterprise and Security work.
  - AF begins development for operational use.
- FY13 – Security Accreditation Process;
  - Automation and enterprise studies
- FY14 – Expanded non-NASA use;
  - mission readiness testing by Air Force;
  - NASA automation efforts
- FY15 – Major agreement signed with Air Force;
  - Began significant upgrades of API and original components
- FY16 – API 4.0 Refresh;
  - Start of Web Services tool suite
- FY17 – Working to make GMSEC a formal industry standard of the Object Management Group
- FY18 – GMSEC submitted to OMG for review and voting
  - Became key partner of SSMO vMMOC effort
- FY19 – OMG publishesGMSEC messages as C2MS standard.

Common Architecture Applications

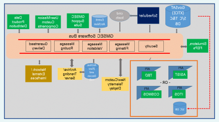
The common architecture and components support different deployment strategies.

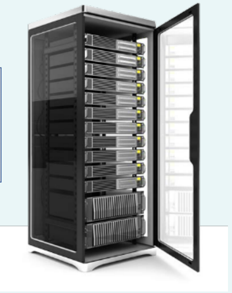


In labs for I&T, and for continued development of evolving ops concepts and capabilities.

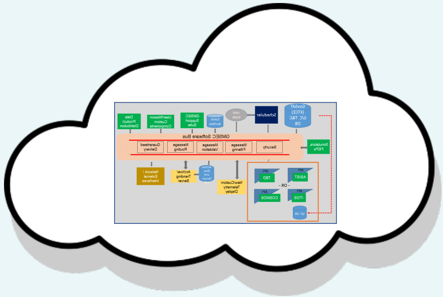


Virtualized for single or multi-mission and enterprise support.



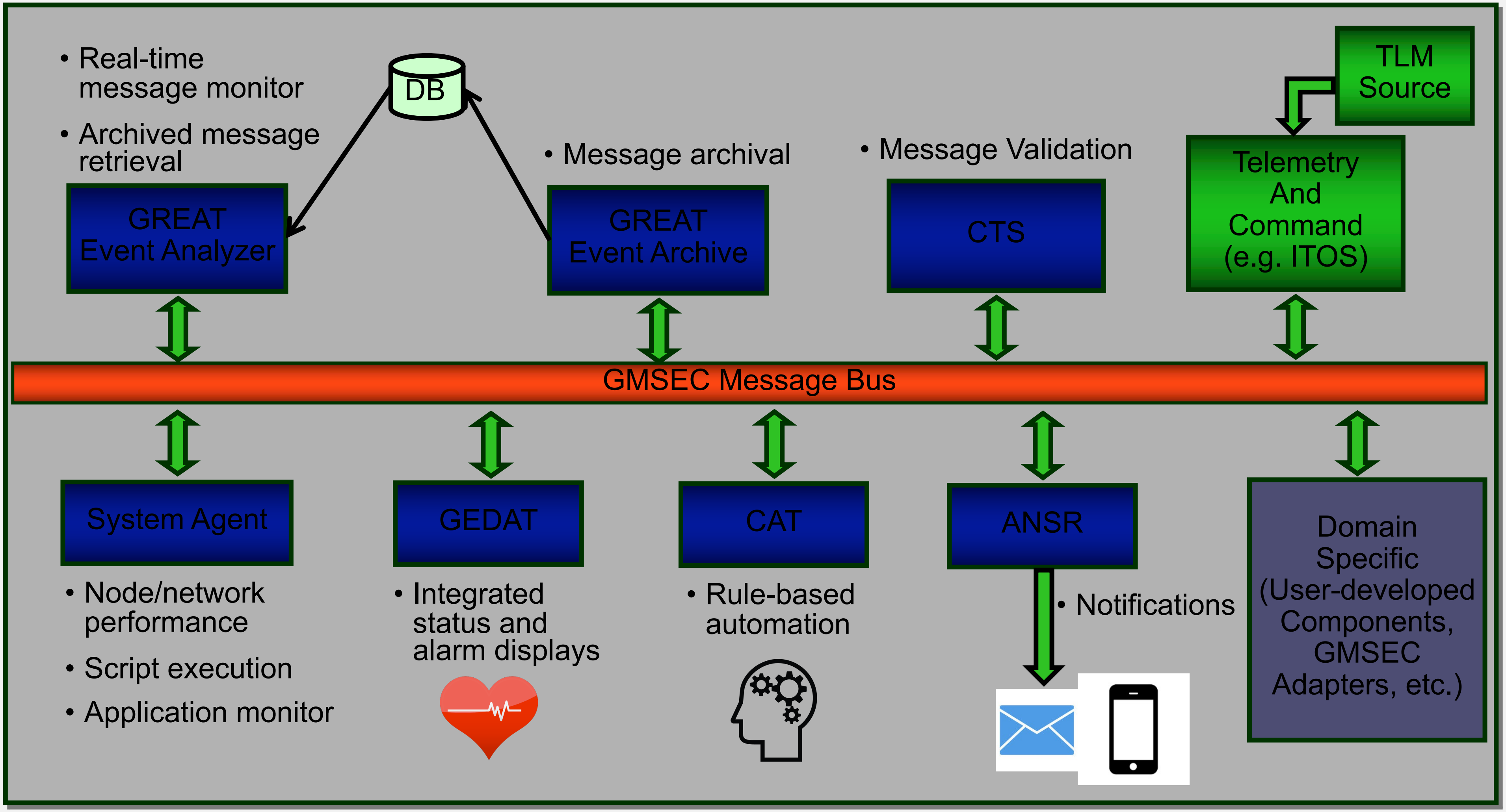


Low-cost mini-MOC



Transition to the cloud–partial or complete.

Example GMSEC System



- Real-time message monitor
- Archived message retrieval

- Message archival

- Message Validation

DB
GREAT Event Analyzer
GREAT Event Archive
CTS
TLM Source
Telemetry And Command (e.g. ITOS)
GMSEC Message Bus
System Agent
GEDAT
CAT
ANSR
Domain Specific (User-developed Components, GMSEC Adapters, etc.)

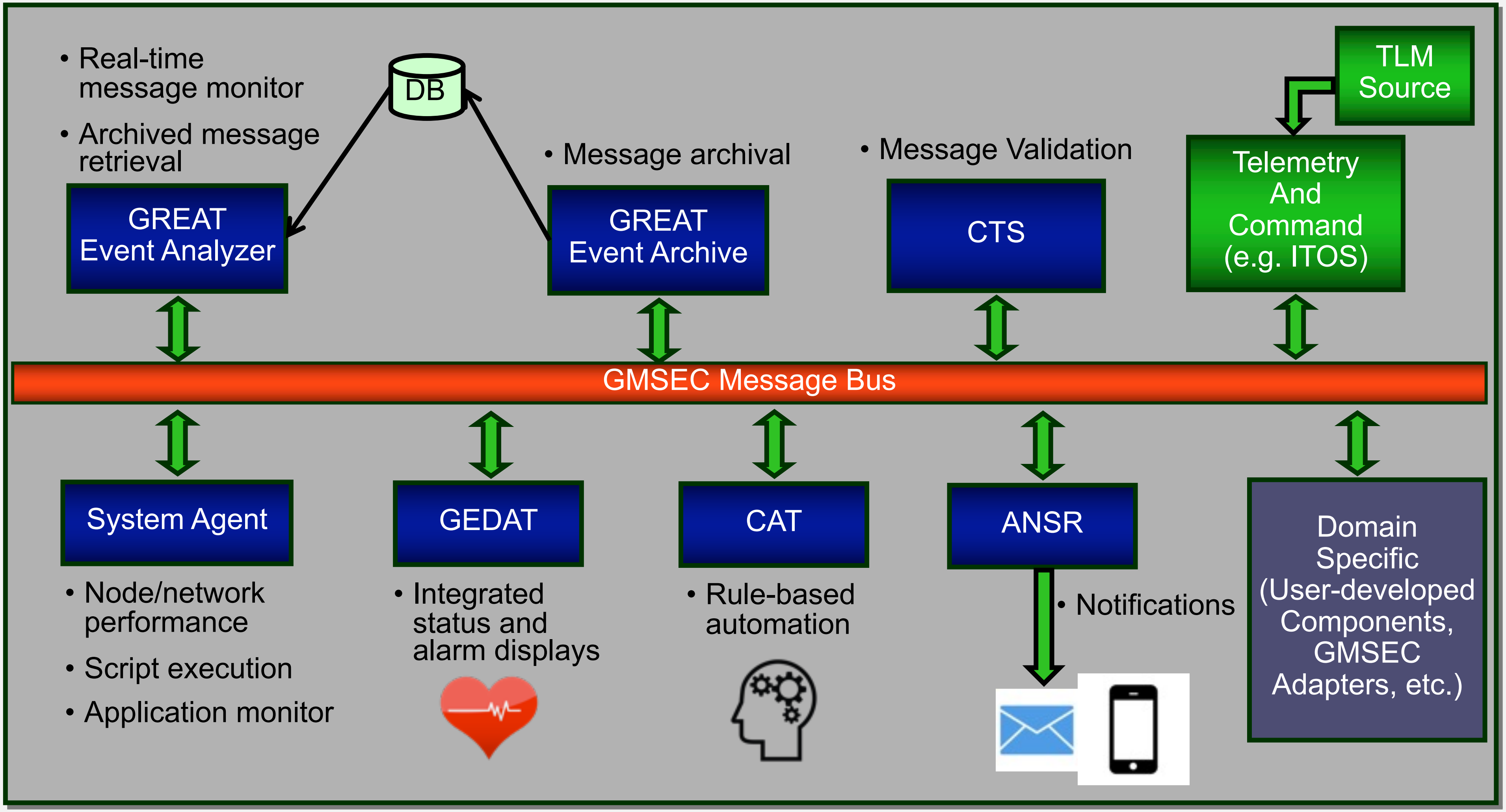
- Node/network performance
- Script execution
- Application monitor

- Integrated status and alarm displays

- Rule-based automation

- Notifications

Example GMSEC System (Mission version)



- Real-time message monitor
- Archived message retrieval

- Message archival

- Message Validation

- Node/network performance
- Script execution
- Application monitor

- Integrated status and alarm displays

- Rule-based automation

- Notifications

DB
GREAT Event Analyzer
GREAT Event Archive
CTS
TLM Source
Telemetry And Command (e.g. ITOS)
GMSEC Message Bus
System Agent
GEDAT
CAT
ANSR
Domain Specific (User-developed Components, GMSEC Adapters, etc.)

Event-Based Automation Lessons Learned / Advice

- Provide a very flexible tool. Avoid having to hard-code the automation steps
- Centralize the management of the actions to be taken
  - Simplifies configuration management
  - Fewer “why did that just happen?” moments
- For complex scenarios, separate the problem detection from the action
  - Have independent routines that monitor conditions and report problems
  - Have the automation system watch for the problem report messages
- Gain the confidence in the system before you trust it to manage your system
  - First rule to implement: “Tell me at 5 pm that it is time to go home”
  - Until you trust that it will tell you at exactly 5 pm and every single day, don’t move on to more complicated rules
- Only automate what you really really understand
  - Don’t try and anticipate all the things you may want to automate
  - Wait until a problem is recurring, has a unique signature, and a proven correction strategy

Programmatic View

- GMSEC is a NASA project developed at GSFC
  - Funded primarily by the Space Science Mission Operations (SSMO) and Earth Science Mission Operations (ESMO) organizations at GSFC
  - Some supplemental funding from missions and external users, including other government agencies
- Available free of charge for use on any U.S. government program
  - The core API software is available as Open Source for even broader availability
  - Missions (NASA and non-NASA) are encouraged to do their own integration using GMSEC; we can provide support if requested but it is normally at a very low level.
- Very engaged with the commercial satellite control product community
- Very engaged with other U.S. government space organizations that also fly LEO and GEO satellites
- We prefer not to write and maintain a lot of software – we'd rather develop the system that allows missions to use GMSEC-compliant software developed by others.

The Current “Store Front” ConOps

- GMSEC is run as a store front operation, offering products and services to NASA ground system teams and other organizations

- Service Offerings
  - General consulting, training
  - Arranging of vendor demos
  - Support during early integration phases, on-call throughout lifecycle

- Product Offerings
  - GMSEC API, GMSEC Components, Message Specification document
  - Software enhancements as needed

- GMSEC team does not build the actual control centers
  - Many former “GMSECers” are now on various development teams
  - Each mission, or set of missions, provide local network support, general ground system support, etc.

The “GMSEC” Problem - GMSEC is many things

GMSEC Project Team

The people

Jim Sec

The person

GMSEC Bus/Framework

A concept that includes the middleware, brokers, network, GMSEC API, and standard messages

GMSEC Components

The GOTS applications

GMSEC Lab/Demo Facility

The Goddard location

GMSEC API*

The s/w layer on top of the message transport (aka CompatC2 API)

GMSEC Architecture

The approach to building and deploying mission applications and systems

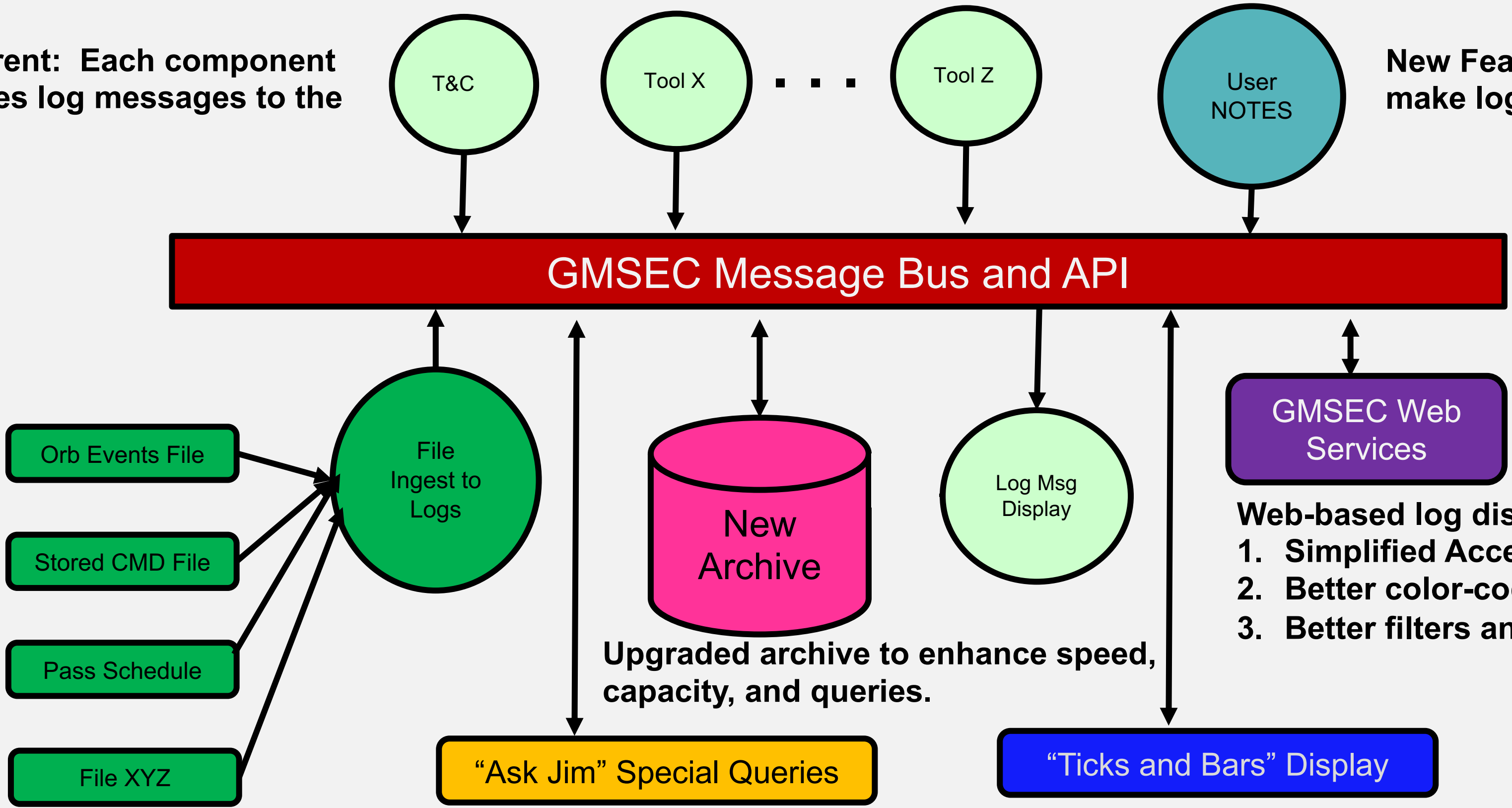
GMSEC ISD (<=2016)

The message spec - C2MS is the evolution of this plus COMPATC2 addendum

GMSEC Log Msg Approach

Current: Each component writes log messages to the bus.

New Feature: Users can make log entries



Applications subscribe to the messages

Web-based log displays provide:

1. **Simplified Access**
2. **Better color-coding**
3. **Better filters and sorts**

Upgraded archive to enhance speed, capacity, and queries.

Converting file contents to event messages creates a single integrated log to help increase situational awareness.

Structured English query of the events log file.

User-defined graphical time-based display of selected events.