GSAW 2015

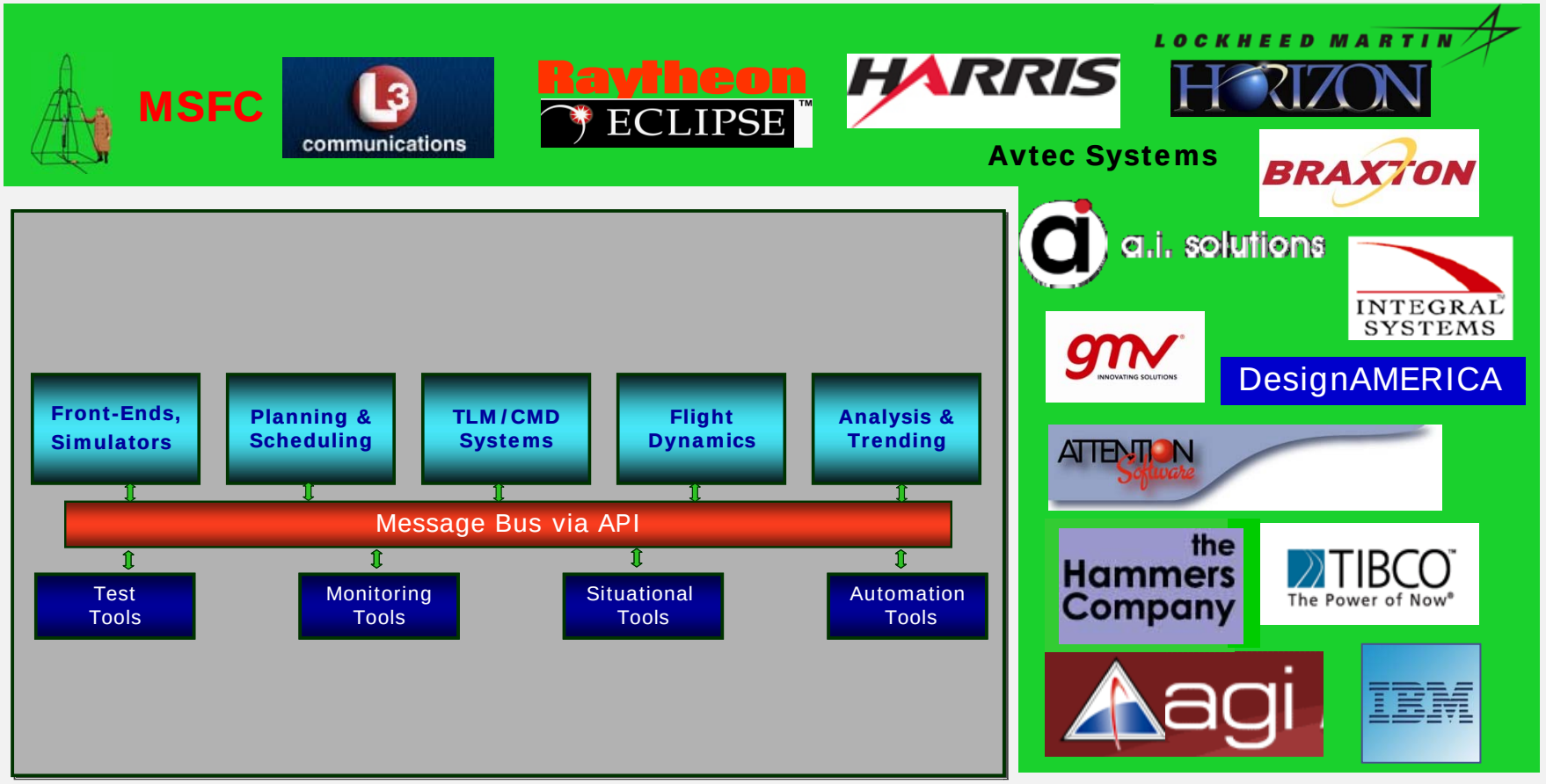
Session 11B: Frameworks in Action – A Foundation for Service Based Architectures

NASA Goddard Space Flight Center's Open Architecture Framework

March 4, 2015
Los Angeles, California

Dan Smith
GMSEC Project Manager
NASA Goddard Space Flight Center
Software Engineering Division
Dan.Smith@nasa.gov

GMSEC



Users can choose the best products for their needs.
Nearly all COTS command and control systems are now GMSEC compatible.

GMSEC Framework

- The GMSEC Framework consists of the GMSEC application programming interface (API), standardized GMSEC messages, an underlying middleware to interface with other components, and a suite of support tools to aid both development and operations.
  - Standard API available as NASA Open Source.
  - “CompactC2 API” available for government use; includes security additions.
  - 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++, Java, Perl, Python
  - **Middleware Products:** TIBCO SmartSockets, Apache ActiveMQ, IBM WebSphere MQ, GMSEC Bolt, Oracle Weblogic
  - **Operating Systems:** Microsoft Windows 7 (32 & 64 bit), Microsoft Windows 2003 Server; Red Hat Linux 3, 4, 5 & 6 (32 bit & 64 bit); Solaris 10

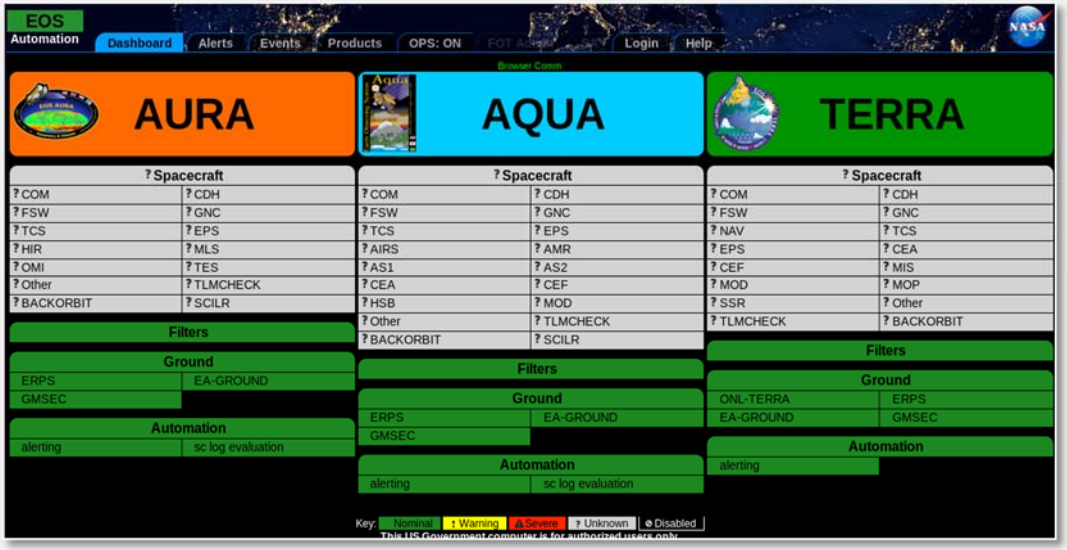
NASA and GMSEC Update

- At NASA/GSFC, missions are gaining sophistication in their use of the open, message-oriented GMSEC architecture
  - Automation capabilities are getting more and more complex
  - Situational awareness tools are now evolving
  - Most new missions find it much easier to integrate their new systems

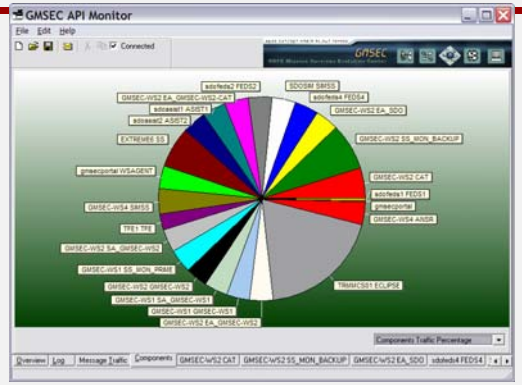
- Some increase in use of multi-mission operations centers

- Agency-wide study now underway to determine future strategy

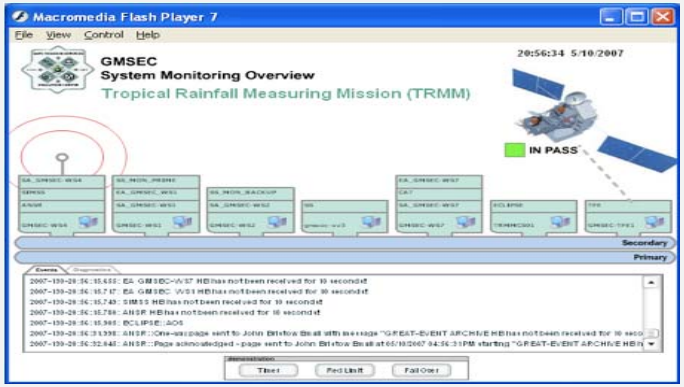
- Discussions with other government agencies have increased. We are all facing the same challenges.



Framework-enabled capabilities are still maturing



First steps. Subscribe to all messages and show their distribution.



Learning more. Subscribe to log messages and show actual configuration and activities as they occur.

Use GMSEC messages and automation rules to show progression of flight dynamics processing steps and automate the sequence of activities and notification.

Today. Process 1,000's of messages to provide drill-down status of multiple satellites. Fully configurable, available on mobile apps.

Technical Directions

2015 marks the 10th year of GMSEC operational use!

API Work

- Some restructuring is needed to better match more recent standard approaches for things like JAVA error handling.
- Scripting languages such as Perl, Python, and Ruby now conform to a new JAVA scripting standard. Will write a single adapter/binding plus multiple small config files so that a wide range of scripting products can be supported.
- **Major Factor:** Modernizing the API will break backward-compatibility for the first time! A detailed multi-year plan is being developed for how to address the change.
  - Should the software support two approaches at once? Should we give 2-year's notice?
  - For how long should we maintain support for the existing approach? 2 years past the release of the new version? Until all missions run out? Two years and then as funded by missions for extended support?

COMPONENTS

- Planning a common restructuring so that any of the key support products can run either as a traditional local console utility or as a webservice, with the choice made in a system config file.

NEW COMPONENTS/FOCUS AREAS

- DAT – Data Access Toolkit. Now in mission test/evaluation phase.
- Low-Cost Data Generator – Currently studying options; make vs buy benefits
- Increase use of cloud, multi-mission awareness, web access.

Challenges as we move towards broader use

1. Who is in charge? What does “in charge” even mean?
2. Governance of the message specification and the API software
3. Organizational approach for enterprise management
4. Funding models for sustainment and advancement when dealing with multiple internal and external partnership organizations
5. Acquisition strategy changes to deal with a infrastructure, common applications, and mission specific needs
6. How do all the organizations stay on the same page? Do they even need to?

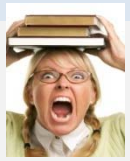
Backup

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 Release Process	Subject Matter Experts	Lessons Learned Documents	List of experienced contractors
			User Group 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.

DoD Open Systems Architecture

“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.