Leveraging Adaptive Software Standards to Enable the Rapid Standup of Small Satellite Ground Systems

Mike Sotak, Kratos Defense

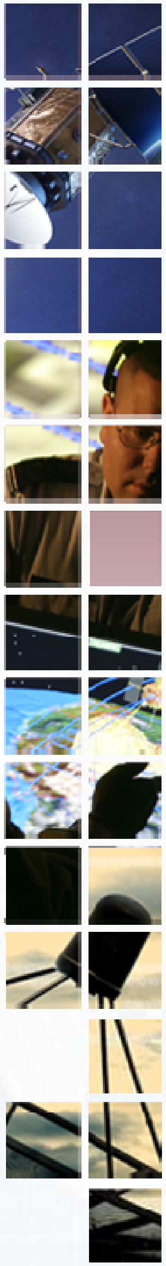
1 March 2016

KRATOS

Outline

- SmallSat Overview
  - SmallSat Market Trends
  - Realities
- Standards in SmallSat Ground Systems
  - XTCE
  - GEMS
- Use of Adaptive Methodologies
  - Representational State Transfer (REST)
  - Scheduling Use Case
- Summary

KRATOS

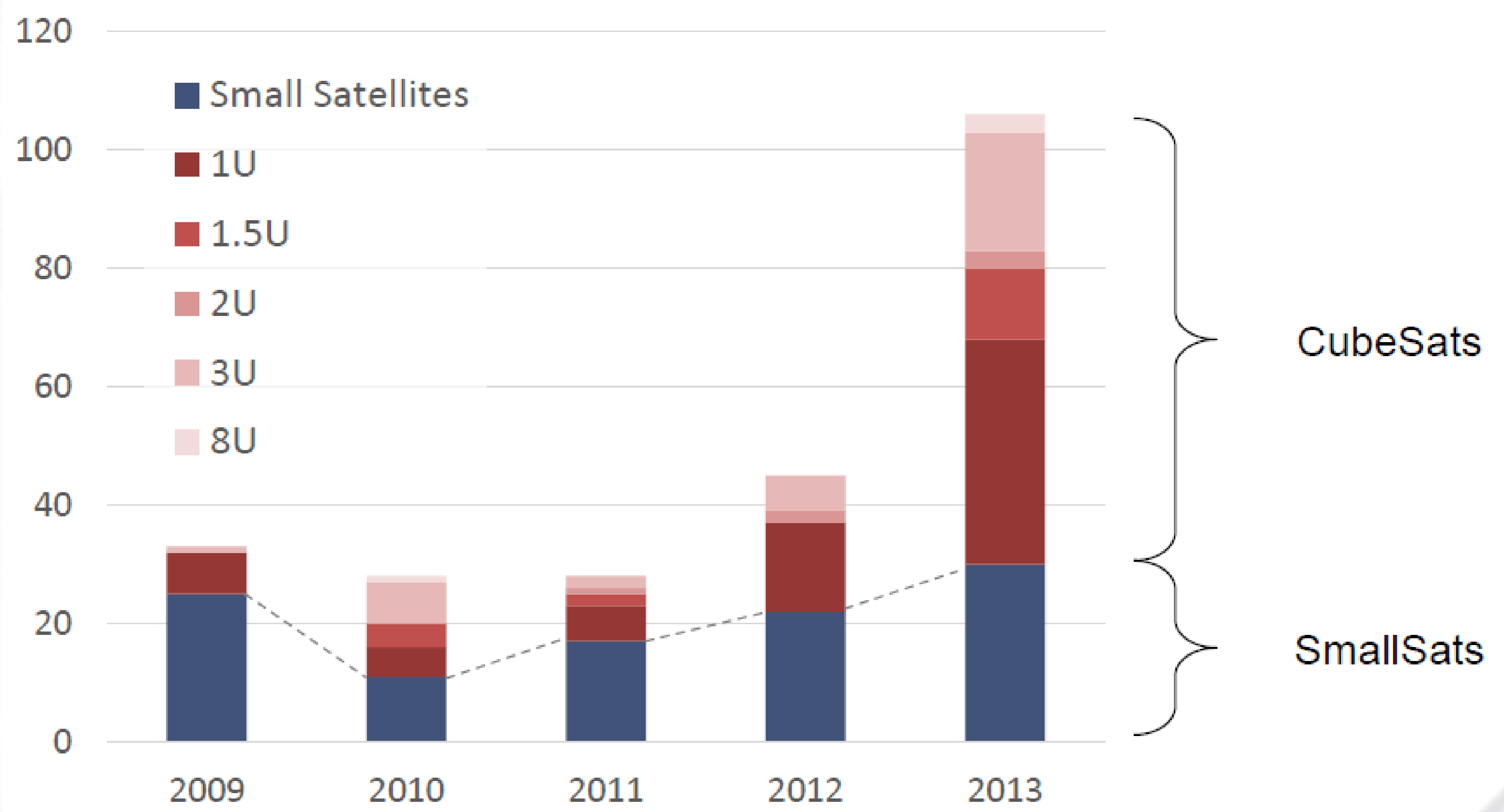


Introduction

- The availability of small satellite technologies has made space accessible to a broader range of individuals and organizations
- This has brought about a new wave of rapidly innovating space systems that challenge and push the norm for ground system architectures
- Increasingly flexible ground systems are required to establish a common ground for providing:
  - Command and Control (C2) to these small satellites
  - Monitor and Control (M&C) of their respective ground equipment.
- Adaptive software standards include the XML Telemetric and Command Exchange (XTCE) and the Ground Equipment Monitoring Service (GEMS)
  - Combined with Representational State Transfer (REST), these adaptive software standards form a flexible architecture
  - Enables rapid turnaround and standup of small satellite ground systems

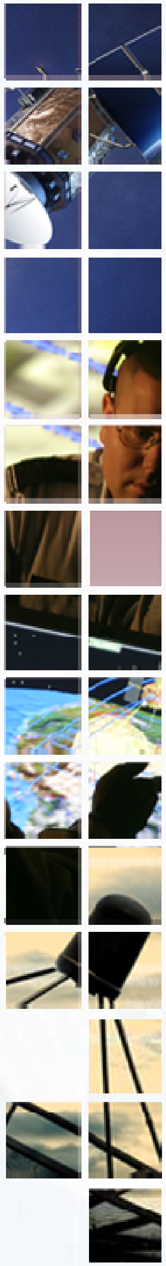
KRATOS

SmallSat Market Trends



Source: "Small Satellite Trends, 2009-2013," Aerospace Corporation, Accessed January 12, 2016. http://digitalcommons.usu.edu/cgi/viewcontent.cgi?filename=0&article=3212&context=smallsat&type=additional

KRATOS



“Traditional” vs. Small Satellite Domains

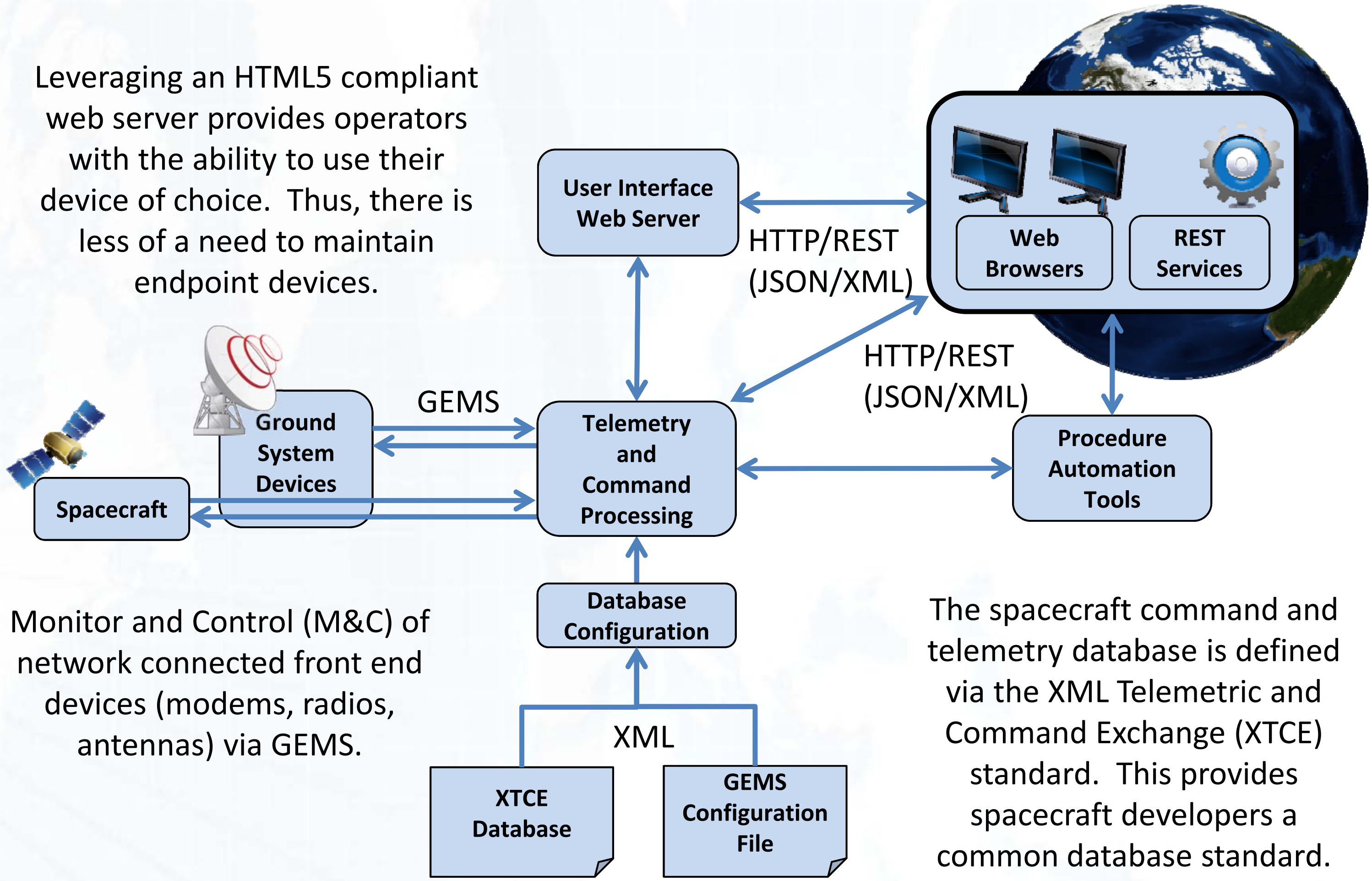
Traditional Satellites	Driver	Small Satellites
Large	**Mission Size**	Small
$$$$	**Cost**	$
Low	**Risk Tolerance**	High
Low (Heritage)	**Innovation**	High
Goal	**Automation**	Foundational

Realities

- A diverse range of players in the SmallSat market, with a diverse familiarity with ground systems
  - Government
  - Small and Large Private Corporations
  - Universities
  - Satellite Developers
  - Ground Equipment Developers
- What can ground system providers do to ensure systems mesh seamlessly, allowing for operators of ranging experience to fly Small Satellites?

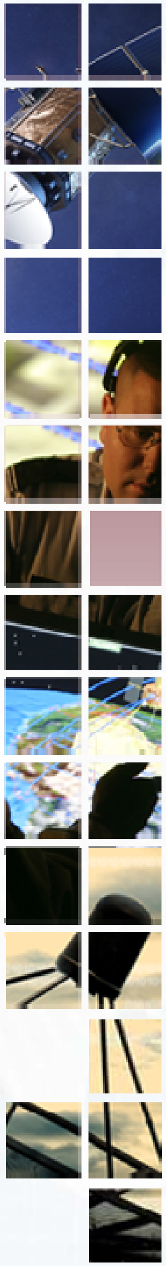
Standards in SmallSat Ground Systems

Leveraging an HTML5 compliant web server provides operators with the ability to use their device of choice. Thus, there is less of a need to maintain endpoint devices.



Monitor and Control (M&C) of network connected front end devices (modems, radios, antennas) via GEMS.

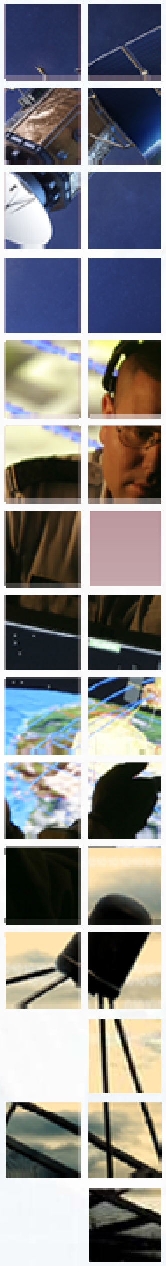
The spacecraft command and telemetry database is defined via the XML Telemetric and Command Exchange (XTCE) standard. This provides spacecraft developers a common database standard.



XTCE Overview

- The XML Telemetric and Command Exchange (XTCE) standard, co-adopted by the Object Management Group (OMG) and CCSDS, is used for satellite database ingest
  - XML is an easy language to use, debug, and understand
  - The XTCE standard capitalizes on these attributes to define satellite databases
    - Multiple parameters can be defined to include telemetry frame locations, frame sizes, calibrations/conversions, and limits and command locations, sizes, and arguments
    - Hierarchy within XTCE allows telemetry to be rolled up by subsystem for notification and alerts
- Using XTCE eliminates the labor needed to define/negotiate a custom space to ground ICD and ingest script
  - As a documented standard, XTCE provides all of the capability needed in a satellite telemetry and command database

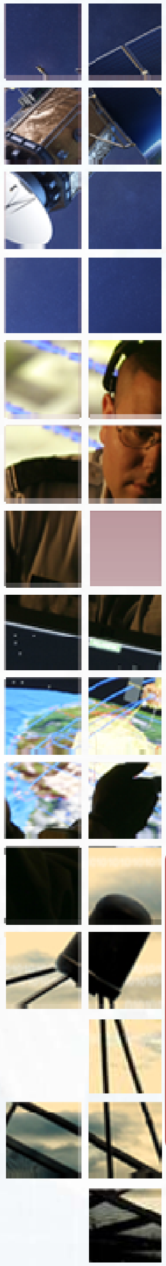
KRATOS



GEMS Overview

- The Ground Equipment Monitoring Service (GEMS) standard, also adopted by OMG, provides a capable interface for monitoring and controlling ground equipment
  - – The interface works as an exchange for data, directives, and status values
- Allows for numerous devices to be integrated
  - – Define directives and status points in an XML file and can be ingested along with the XTCE database
  - – Ground equipment can be commanded/monitored the same way
- Ground directives could also be included in automation procedures
  - – Automate pre-pass ground hardware configuration and set up and post-pass ground hardware deconfiguration and clean up

KRATOS



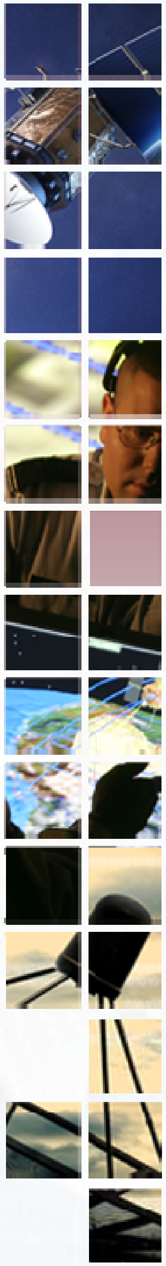
Typical Ground Segment Functions

RF Reception / Transmission Antenna	Frame or Packet Decommutation	Command Generation	Procedure Scripting	Memory Management
Antenna Pointing and Control	Point Context Check	Command Formatting	Display Building	Mission Data Processing
RF Up / Down Conversion	Engineering Unit Conversion	Command Authority Check	Real-Time User Interface	Automated Anomaly Processing
Modulate / Demodulate Signal (MODEM)	Measurand Limit Check and Alarming	Transmission and Tracking	Ops Automation	Flight Dynamics
Encryption / Decryption (optional per mission)	Point and Track File Generation	Verification	Plotting and Trending	External Interfaces
	Ground Device Monitor and Control	Logging and Messaging	Raw Telemetry and Processed File Retrieval	Payload and Bus Mission Planning

Functionality Legend

- Ground Hardware
- Telemetry
- Commanding
- Additional C2 Req'ts
- Mission Unique

Use of standards helps to unify these components and pave the way to an adaptive mission.

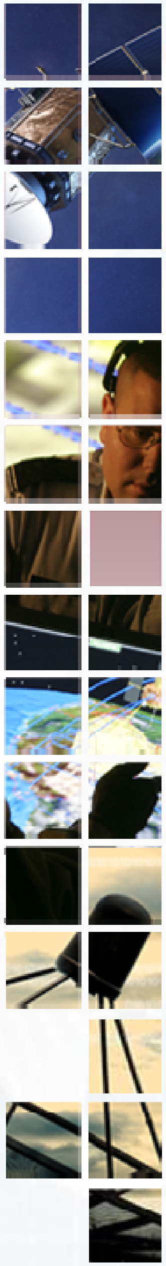


Use of Adaptive Methodologies

- As demonstrated on the last slide, Ground Systems are incredibly complex architectures with tons of functionality
- Adaptive methodologies are needed in order to bring cohesion among these disparate components
  - Definition: A rapid development strategy for deploying software efficiently and effectively
- Using a RESTful methodology allows for mixing and matching software products and functions rather than provisioning a stovepipe system
  - This in turn minimizes system disruptions as individual components can be readily maintained or replaced

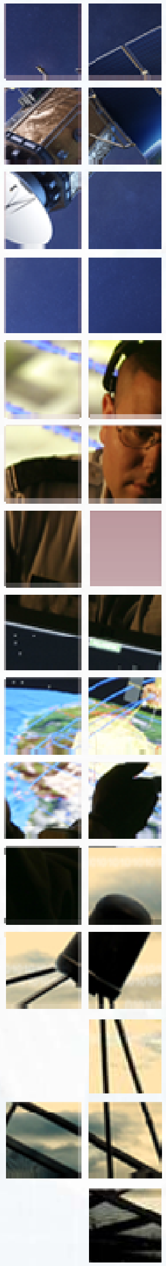
KRATOS

REST



- Representational State Transfer (REST) is an adaptive methodology used in the development of scalable, high performance communications architectures.
  - In terms of satellite ground systems, this allows an open, web-based approach to communicate with ground equipment and satellite busses while leveraging common software interfaces.
  - Use of these simple, standards-based interfaces allows for widespread availability for users and operators to quickly develop and turn around fully functional ground systems.
  - Even further, this model allows for additional spacecraft or equipment to be efficiently added to the enterprise.
  - Messages within REST are recommended to be using standard formats, such as JavaScript Object Notation (JSON) or XML
- To support a constellation of small imaging satellites, new spacecraft databases might be required or additional remote ground locations, with new equipment, might need to be put up in a short time frame.

KRATOS

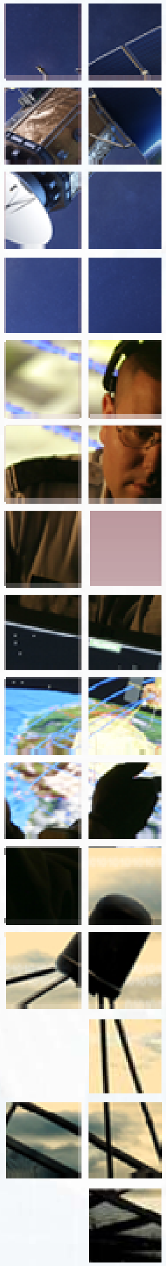


Adapting REST to the Control Center

- REST systems are based on similar verbs utilized by the HyperText Transfer Protocol (HTTP)

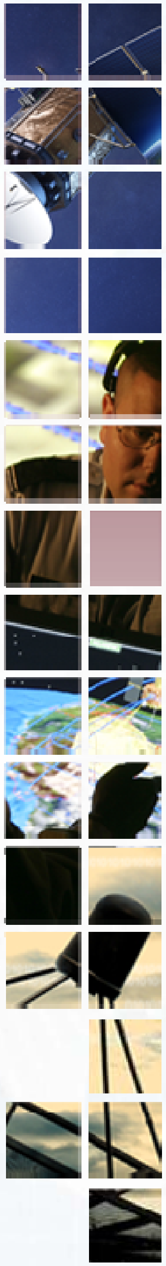
Verb	Potential applications
GET *Current status of a resource/item*	• Receive a schedule from mission planning system • Get current value of a telemetry parameter • Download a generated trending report • Access Two-Line Element Set (TLE) files for nearby neighbors
POST *Save a new resource/item*	• Add a new contact to a mission planning system • Send a spacecraft command • Create a new trending report profile • Perform closest approach prediction
PUT *Save status of an existing resource/item*	• Update the resources associated with a schedule item • Save an updated spacecraft automation procedure • Update archive with re-processed data reflecting updated calibrators • Change location parameters for site antennas
DELETE *Destroy a resource/item*	• Remove an antenna from the schedule system • Delete a spacecraft database • Erase files from history for a de-orbited spacecraft • Clean up out-of-date ephemeris reports

REST has applications across all components within a satellite control center



Scheduling Use Case

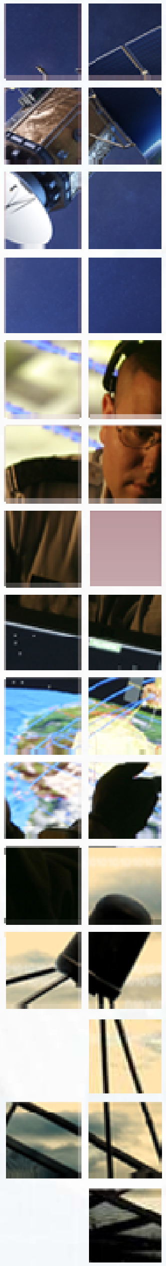
- The following provides an example of using REST to prepare a satellite contact schedule
- Using a message encoded in JSON, this example adds events to the schedule by adhering to the following rules:
  - The events array can contain one to many event definitions.
  - The event must have a unique id.
- An event is made up of a parent event with one to many children, with the children events doing the “work” of the event as specified in the “directive” line



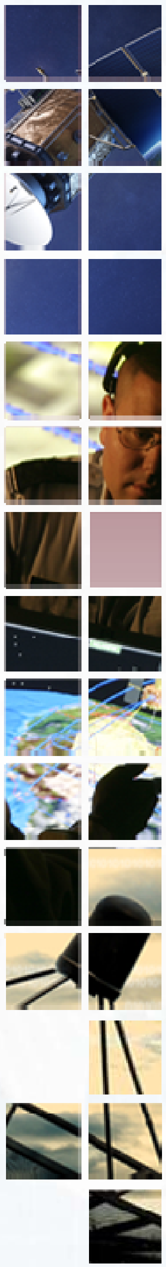
Scheduling Use Case

Attribute	Description
Id	Unique id of event
parentTreeNodeId	
Start	Start time in ISO8601 format (UTC)
End	Event time in ISO8601 format (UTC)
Title	Title of event
startMode	Run mode starting event. 1 = manual run, 0 = automatic run
endMode	Run mode of stopping event. 1 = manual run, 0 = automatic run
autoExtend	Extend the event automatically (true, false).
runStatus	The run status of the event. "initial" for an event that is first created and not run yet.
Attributes	The list of keyword value attributes that are the resources for the event. The keywords should match the resources database entries and the values should match the resource_values database entries.
parentEventId	Optional id if this event is a child event of a container event. Note: container events hold the "attributes" tag, child events have no attributes.
startSource	Bridge name to send the event startProps text message to. The bridge the message is sent to is responsible for actually doing the "execution" work. Note: only child events have this tag.
startProps	The text message that will be sent to the target bridge for execution of the event. Note: only child events have this tag.

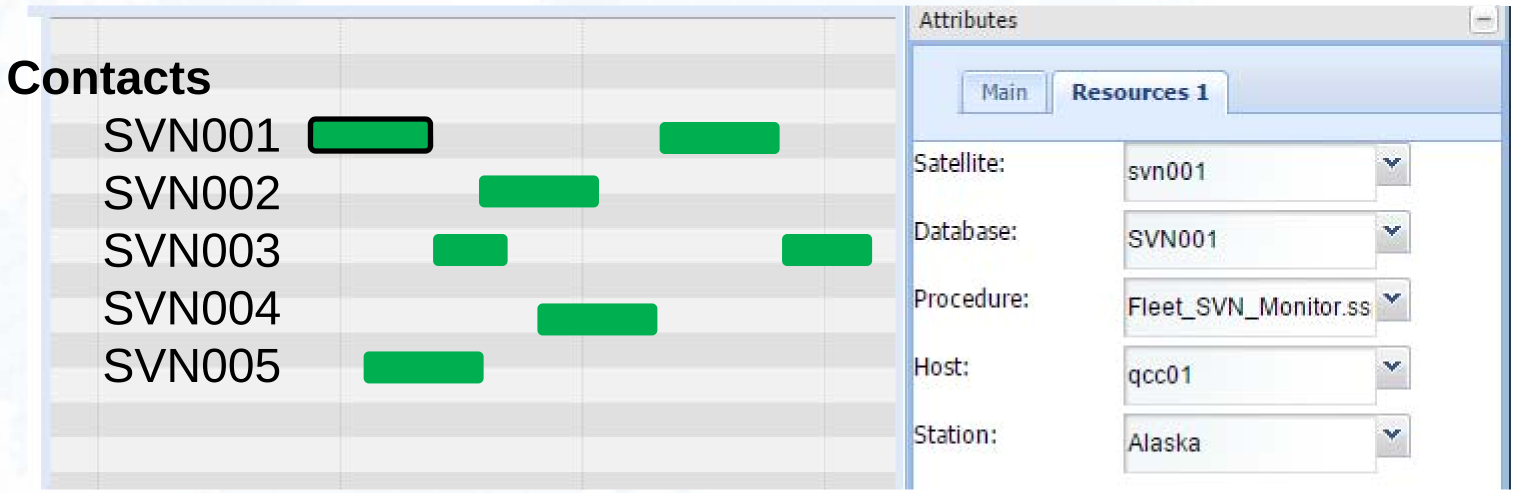
Scheduling Use Case: JSON Message




```
{
    "message": "addEvents",
    "events": [
        {
            "id": "157391399311047654",
            "parentTreeNodeId": "8",
            "start": "2016-01-20T18:16:04.000Z",
            "end": "2016-01-20T18:31:04.000Z",
            "title": "",
            "startMode": 1,
            "endMode": 1,
            "autoExtend": false,
            "runStatus": "initial",
            "attributes": {
                "host": "qcc01",
                "satellite": “SVN001",
                "resource-def2": "0",
                "procedureName": “Fleet_SVN_Monitor",
                "tcProcessor": "qcc01"
            }
        },
        {
            "id": "745615139931104765 4",
            "parentEventId": "157391399311047654",
            "parentTreeNodeId": "8",
            "start": "2016-01-20T18:16:04.000Z",
            "end": "2016-01-20T18:17:04.000Z",
            "title": "Start Stream",
            "startSource": "StreamBridge",
            "startProps": "{\"message\":\"outbound\", \"directive\":\"start %stream% %database% \"}",
            "startMode": 1,
            "endMode": 1,
            "autoExtend": false,
            "runStatus": "initial",
            "attributes": { }
        },
        {
```



Scheduling Use Case: CSV and Front End Example



Satellite	Date	Start	End	Station	Host	Procedure	Auto Start	Auto End	Database
svn001	1/15/2016	T00:17:00.000Z	T00:17:06.000Z	Alaska	qcc01	Fleet Monitor.ssp	YES	YES	SVN001
svn002	1/15/2016	T00:17:10.000Z	T00:17:15.000Z	Sweden	qcc03	Fleet Monitor.ssp	YES	YES	SVN002
svn003	1/15/2016	T00:17:06.000Z	T00:17:11.000Z	Canada	qcc02	Fleet Monitor.ssp	YES	YES	SVN003
svn004	1/15/2016	T00:17:13.000Z	T00:17:18.000Z	USA-W	qcc01	Fleet Monitor.ssp	YES	YES	SVN004
svn005	1/15/2016	T00:17:03.000Z	T00:17:10.000Z	USA-E	qcc04	Fleet Monitor.ssp	YES	YES	SVN005

Summary

- The SmallSat industry is a rapidly growing, diverse community with a large number of stakeholders
- SmallSats typically require a quick turnaround, necessitating streamlined development processes
- Use of standards ensures ground system developers and satellite operators can find a common ground to help accommodate tight production schedules
  - XTCE and GEMS are well-defined standards that provide a common schema for satellite operators to develop satellite and ground equipment databases
  - Use of a RESTful methodology allows for simple, lightweight, but powerful ground system communications

KRATOS