

The logo for nos<sup>3</sup> features the letters 'nos' in a dark grey, lowercase sans-serif font, with a large orange '3' as a superscript. To the left of the text is a decorative graphic of eight orange circles of varying sizes arranged in a loose arc.

# nos<sup>3</sup>

NASA Operational Simulator  
for Small Satellites

Tools for Software Based Validation  
and Verification of Small Satellites

Matt Grubb

[Matthew.d.grubb@nasa.gov](mailto:Matthew.d.grubb@nasa.gov)

Small Sat Workshop 2016

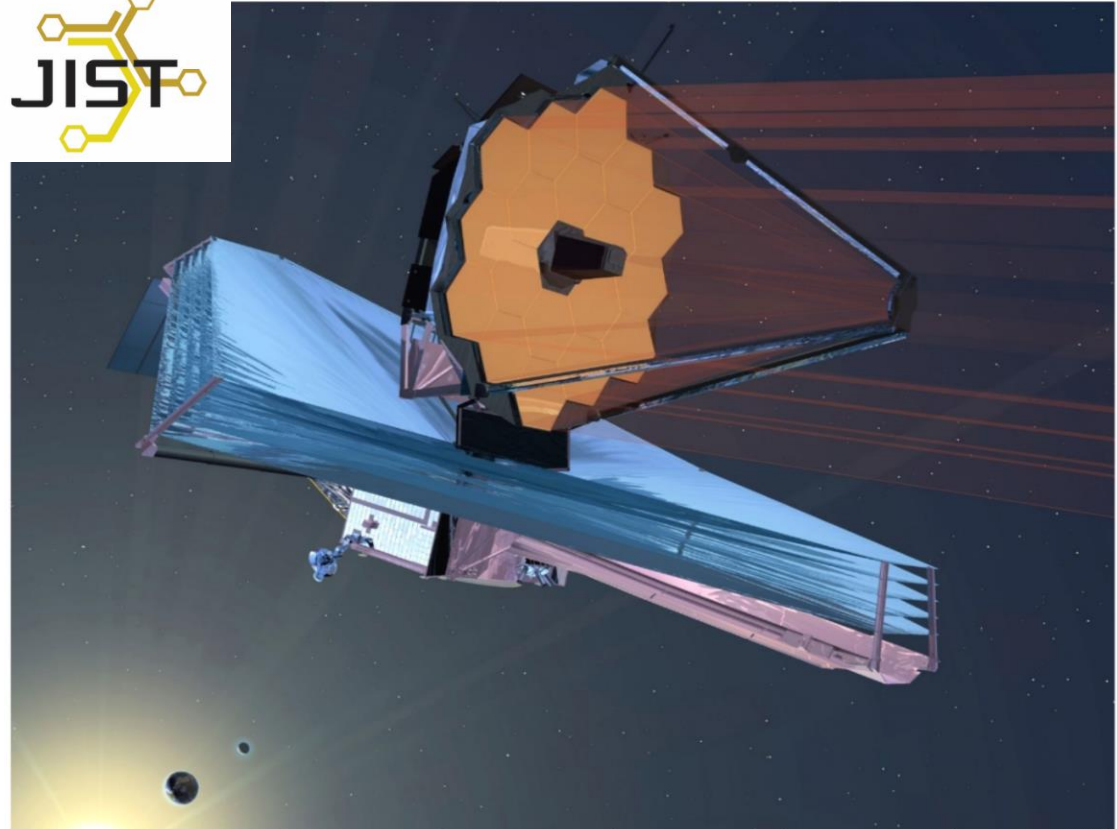


# ITC Developed Simulators

**S3**



**JIST**



# Overview









## What is NOS<sup>3</sup>?

- Based upon Simulation-to-Flight 1 (STF-1) hardware, but sufficiently generic
- Easily-interfaces to Core Flight System (CFS), but CFS not required
- A collection of Linux executable and libraries, deployed ready-to-run
- Allows the testing of flight software as it is flown

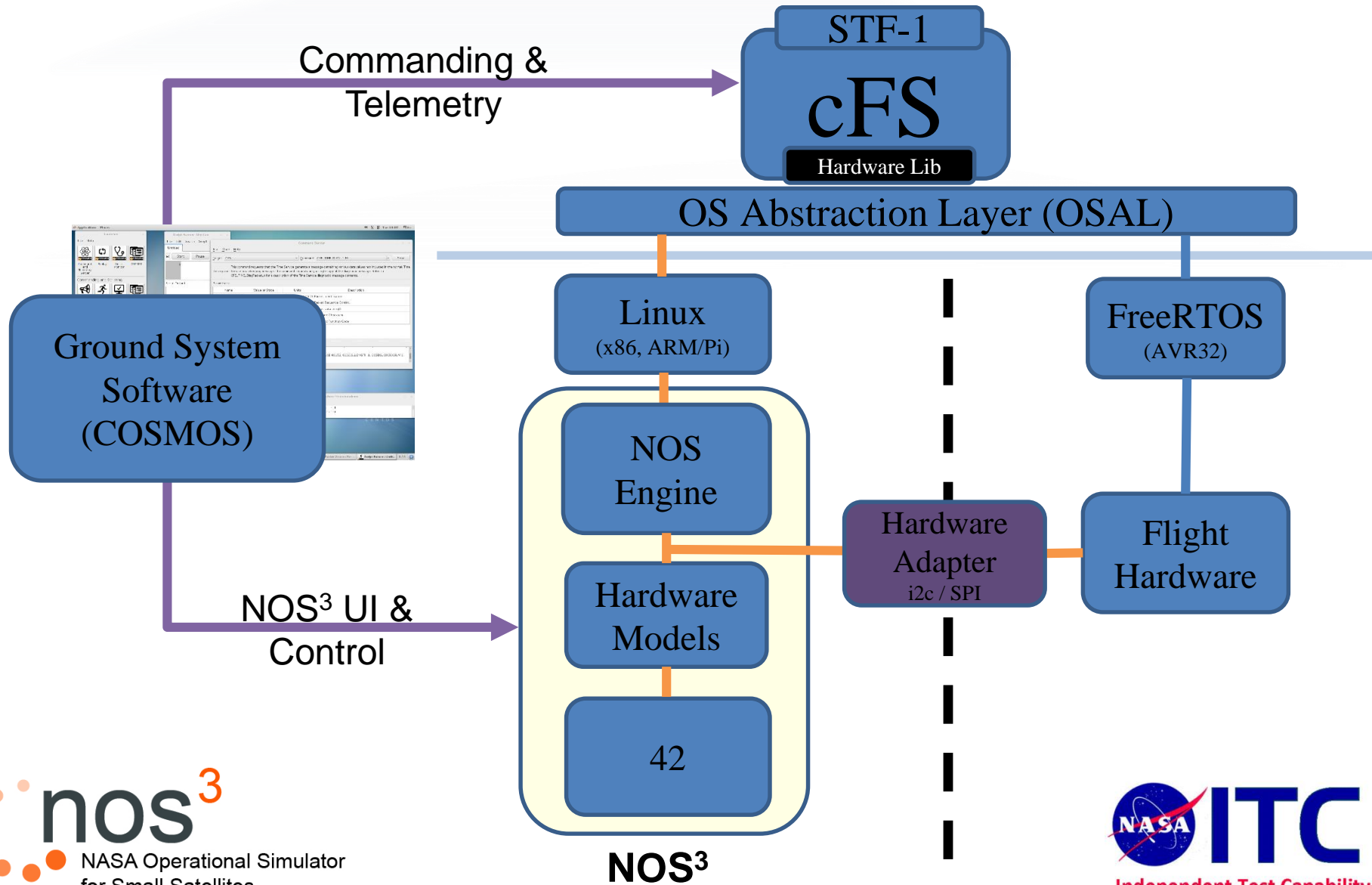
## What is it used for?

- FSW early-development – NOS<sup>3</sup> provides real-world inputs to FSW
- FSW V&V – Testing FSW, invalid inputs, behavior, stress conditions
- FSW Integration – Applications can be tested with simulated hardware
- Mission Planning – Example: power analysis, command and telemetry

# NOS<sup>3</sup> Components

-  • Virtual Machine – NOS<sup>3</sup> environment
-  • NOS Engine Middleware
-  • Hardware Models
-  • FSW Hardware Abstraction Layer
-  • Orbit Inview & Power Prediction (OIPP) Tool
-  • CFS – Flight Software
-  • 42 – Dynamics Simulation and Visualization
-  • COSMOS – Commanding & Telemetry

# NOS<sup>3</sup> Architecture



# Ease of Deployment

- Ready-to-run after unpacking a .tar
- Install *Vagrant* and *VirtualBox*
- Run *nos3\_installer*
- Developer build tools installed
- Convenience scripts for building/running

The screenshot displays the COSMOS software suite running on an Ubuntu desktop. Key components visible include:

- COSMOS Command and Telemetry Server - STF1 Configuration:** A window with tabs for Interfaces, Targets, Cmd Packets, Tlm Packets, Routers, Logging, and Status. It contains a table with columns: Interface, Connect/Disconnect, Connected?, Clients, Tx Q Size, Rx Q Size, Bytes Tx, Bytes Rx, and Cmd Pkt. The table shows two interfaces: CFS\_INT and COSMOSINT, both with a status of 'true' and 0 clients.
- NOS Time Driver:** A terminal window showing log entries for time synchronization, such as '2016-05-19 14:33:49.Xf [ INFO ] - TimeDriver::send\_tick\_to\_nos\_engine: tick = 443, absolute time 552110843.000000'.
- NOS Engine Standalone Server:** A terminal window showing the process of creating transports for fsw and nos3, and a query menu with options like Buses, Data Nodes, Time Clients, Time Sender, and Exit.
- Command Sender:** A window for sending commands to a target (STF1). It includes a 'Command' dropdown set to 'ENABLE\_TELEMETRY' and a 'Send' button. Below is a table of parameters for the command.
- STF1 Flight Software:** A terminal window showing the initialization status of various components, including MGR, SEN, NAV, SCH, EPS, CADET, and TO telemetry output.

STF1 Flight Software

NOS Engine Standalone Server

Simulators

# NOS Engine Middleware

- ITC developed middleware
- Common server to communicate to all data nodes (CFS, Hardware simulators, Time ticker, Command terminals)
- C/C++ API
- I2C, UART and SPI protocols
- Asynchronous and Synchronous

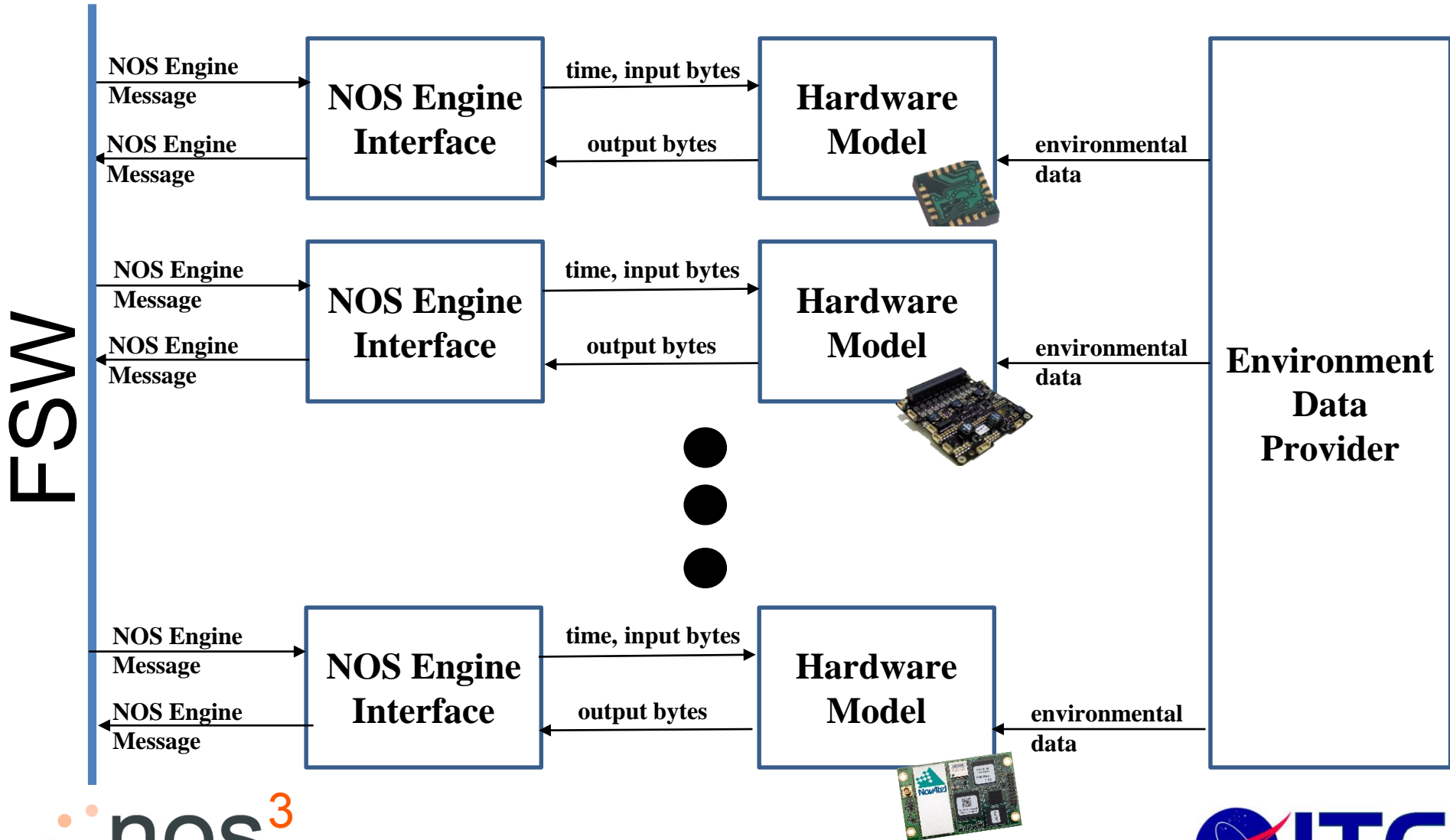




# Hardware Models

- Modeled based on characteristic data, or manufacturers data specifications
- Currently have modeled
  - Novatel GPS
  - Clyde Space EPS
  - Honeywell Magnetometer
  - ISISpace Antenna System
  - Gomspace A3200 support chips (FRAM, Gyro)

# Hardware Models



# Flight Software (CFS)

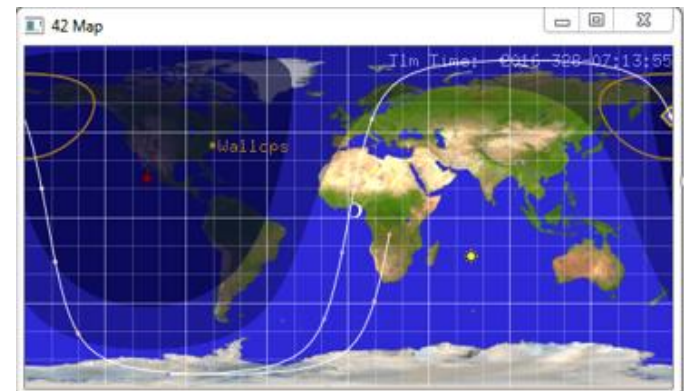
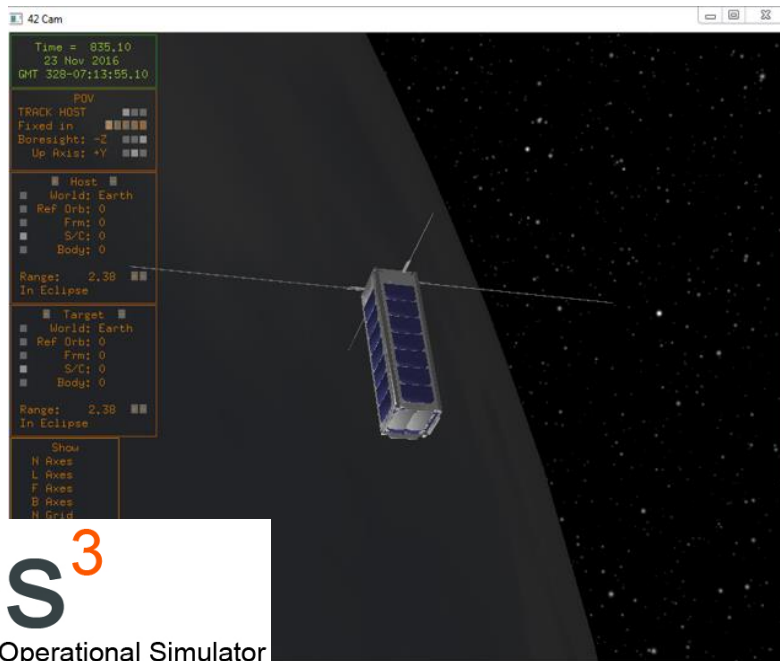
- Open source flight software developed by GSFC
- Includes an OS Abstraction Layer
  - Allows building for flight and NOS<sup>3</sup> targets on same machine without source code changes
- Additional Platform-Support-Package (PSP) added to sync CFS time with NOS<sup>3</sup>



# 42

## GSFC Open Source Dynamics Simulator

- NOS<sup>3</sup> TCP/IP Socket Integration
- Simulation time synchronized with NOS<sup>3</sup>
- Moving toward closed loop



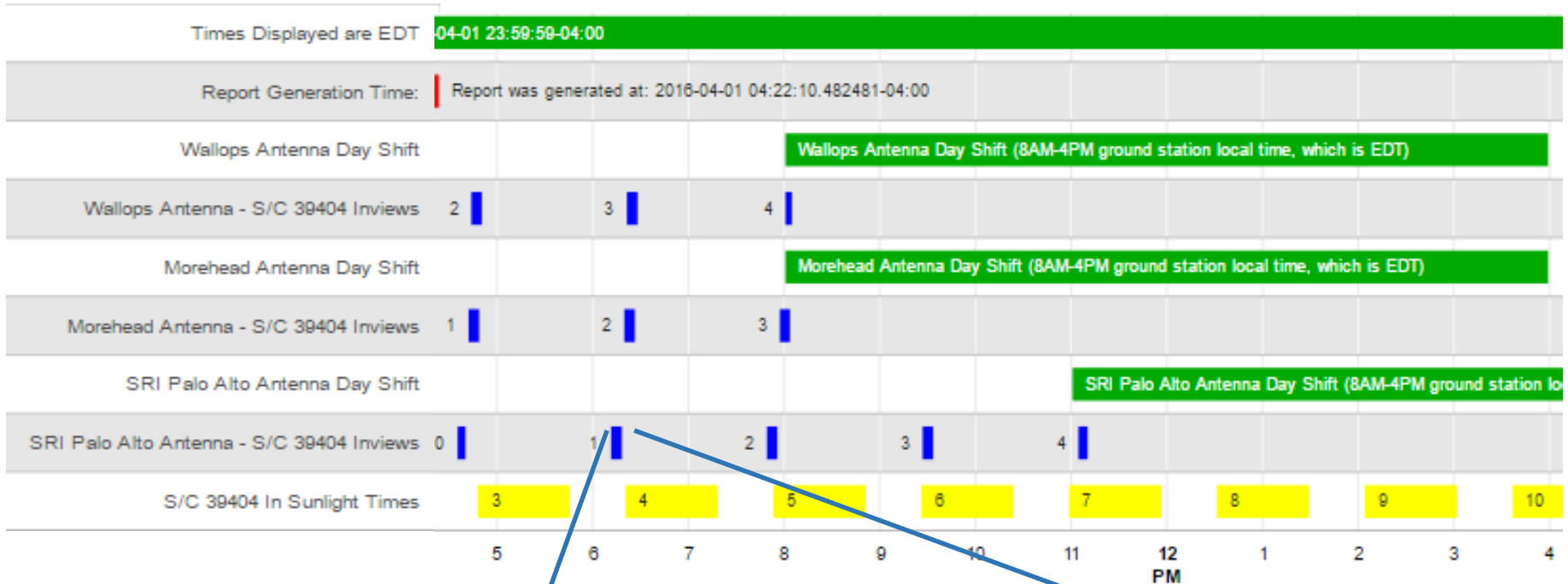
# COSMOS

- Open Source software for embedded system commanding and telemetry
- Currently connects to CFS TO\_lab
  - Future plan is to have radio simulator to replace TO\_lab
- Can be used for operator training, testing table loads to SC, verifying command and telem databases, etc.

# Orbit, Inview, and Power Prediction

- Web page: Generated daily by cron job
- TLE Data pulled from <http://celestrak.com> as obtained from NORAD
- Time Periods (configurable)
  - Yesterday, Today, Future
- Displays
  - Ground station in-views
  - Sunlight and Eclipse times

# Orbit, Inview, and Power Prediction (OIPP)



1

Wallops Antenna - S/C 77777 Inviews: 12:43 pm - 12:50 pm

Duration: 0.12 hours



# Acquiring NOS<sup>3</sup>

- Stop by the GSFC booth and see us
- Email us:  
**[ivv-dl-nos3-support@mail.nasa.gov](mailto:ivv-dl-nos3-support@mail.nasa.gov)**