ITC
Independent Test Capability

# NASA Operational Simulator (NOS)

## A Reusable Software-Only Verification & Validation (V&V) Architecture

http://www.nasa.gov/centers/ivv/JSTAR/ITC.html

Justin R Morris
Justin.R.Morris@nasa.gov

NASA IV&V Facility
100 University Drive
Fairmont, WV 26554

# Agenda

- **Independent Test Capability (ITC)**
  - Jon McBride Software Testing & Research Lab (JSTAR)
- **NASA Operational Simulator (NOS)**
  - Architecture
  - Middleware
- **NOS Utilization**
  - Global Precipitation Measurement (GPM) Operational Simulator (GO-SIM)
  - James Webb Space Telescope (JWST) Integrated Simulation and Test (JIST)
- **Closing Remarks**

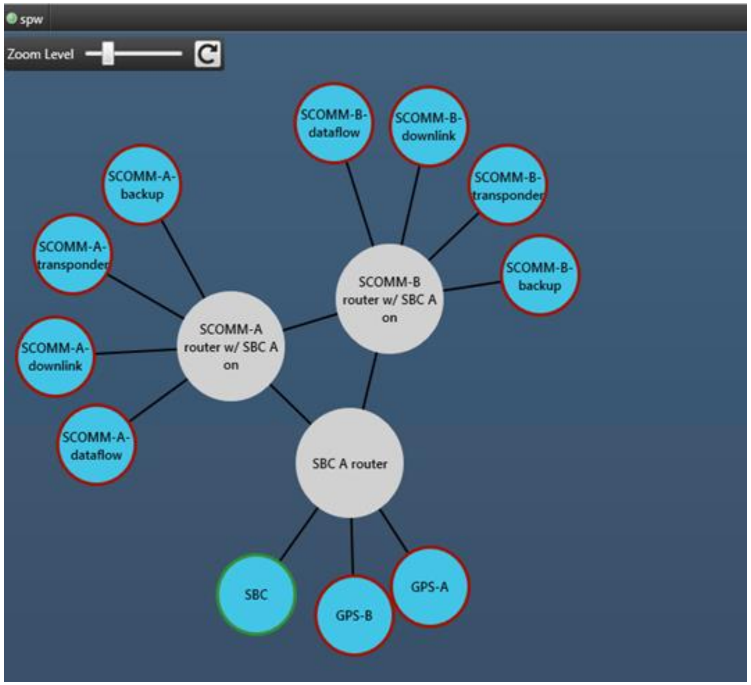
# Independent Test Capability (ITC)

## Charter

Acquire, develop, and manage adaptable test environments that enables the dynamic analysis of software behaviors for multiple NASA missions

# Independent Test Capability (ITC)

- ITC Develops System Simulators
  - Experts in Hardware Modeling and Distributed Simulation
  - Experts in Simulator & Software Integration
- NOS Architecture
  - Reusable Hardware Models
  - Custom Middleware
- System Test Automation
- Typical NOS Users
  - V&V and IV&V Engineers
  - Project Developers
  - Operators and Testers



# Jon McBride Software Testing & Research (JSTAR) Laboratory

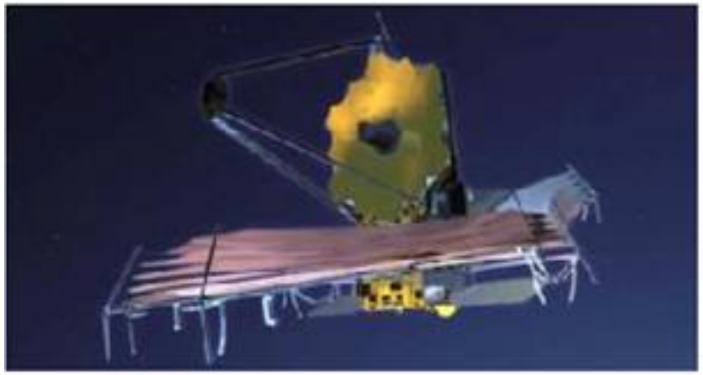
- Cloud-based infrastructure using server and desktop virtualization
- Large scale simulator deployments
- Hardware-in-the-loop and software-only test environments
- Integration of COTS and GOTS software tools to support V&V activities

NASA ITC Independent Test Capability

# Mission Support

## Global Precipitation Measurement (GPM) Operational Simulator (GO-SIM)

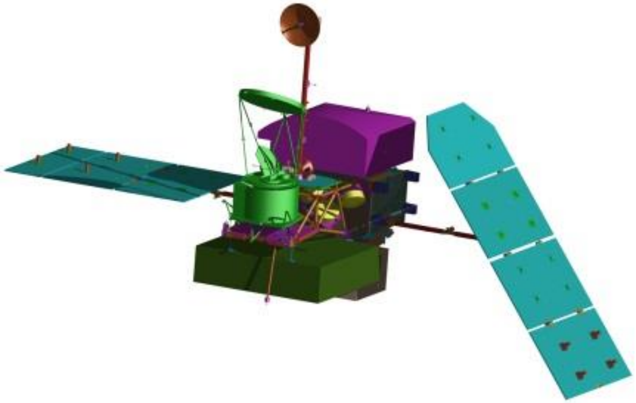
Closed-loop simulator including **unmodified** operational ground system, **unmodified** flight software, environmental simulator, and science instrument simulators

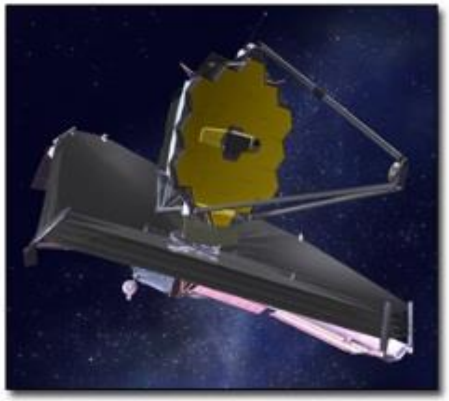


## James Webb Space Telescope (JWST) Integrated Simulation and Test (JIST)

Simulator that demonstrates reusable NOS technologies can be applied to other NASA missions

# NASA Operational Simulator (NOS)

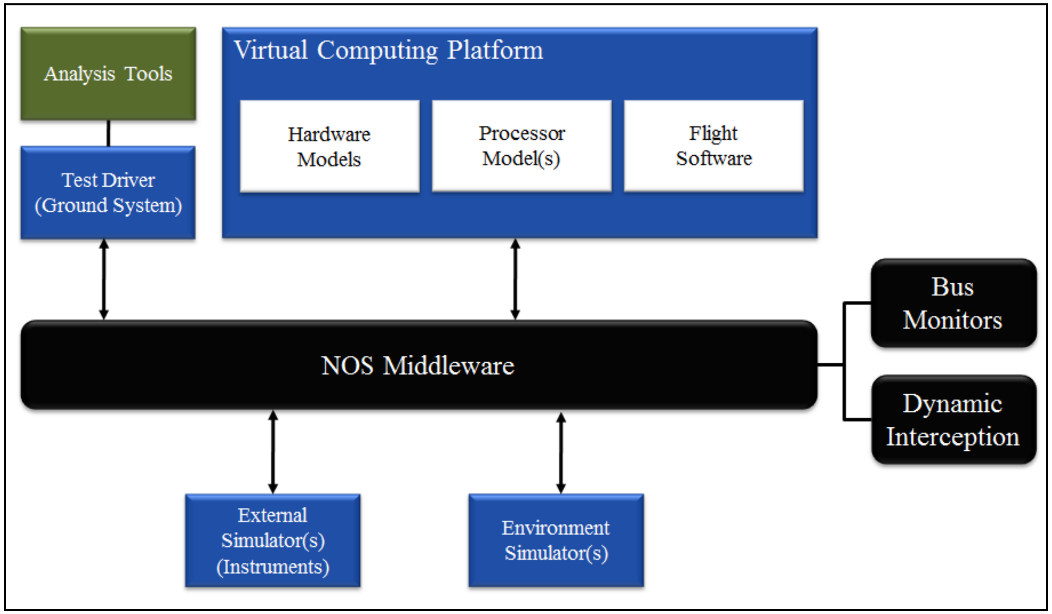




# NASA Operational Simulator (NOS)

- Software-only simulation architecture (common components from in-house software simulation development)
- Capable of executing unmodified flight software executable(s)
- Custom layered-architecture middleware
- Dynamic interception capability
- Reusable software modules and scripts
- Virtual machine deployment

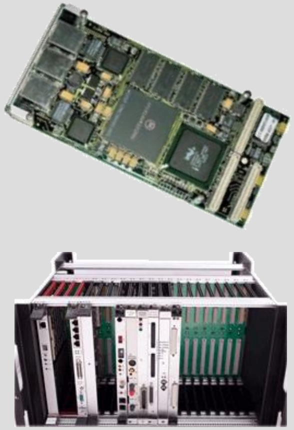
Typical NOS Architecture (Space Domain)



NASA ITC
Independent Test Capability

# NOS Feature Set

## Plug-and-Play Hardware Models



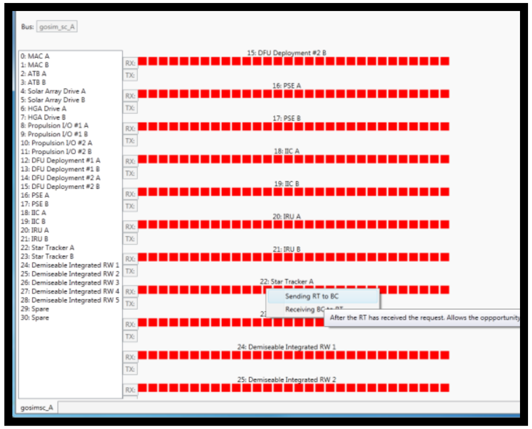
*Processors,*
*Boards,*
*Racks*

## Use of Operational Ground Systems Software

## Instrument Model Framework

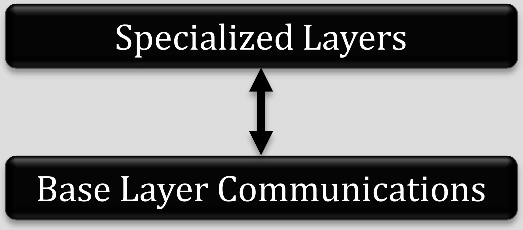
```
Instrument1
Subaddress HandlerA → FunctionA
Subaddress HandlerB → FunctionB
…
Subaddress HandlerN → FunctionN

InstrumentX
Subaddress HandlerA → FunctionA
Subaddress HandlerB → FunctionB
…
Subaddress HandlerN → FunctionN
```

## Internal Bus Monitoring



After the RT has received the request. Allows the oppportunity

## NOS Middleware



Specialized Layers

Base Layer Communications

## Deployment & Maintenance

*Virtualization*

NASA ITC
Independent Test Capability

# NOS Middleware

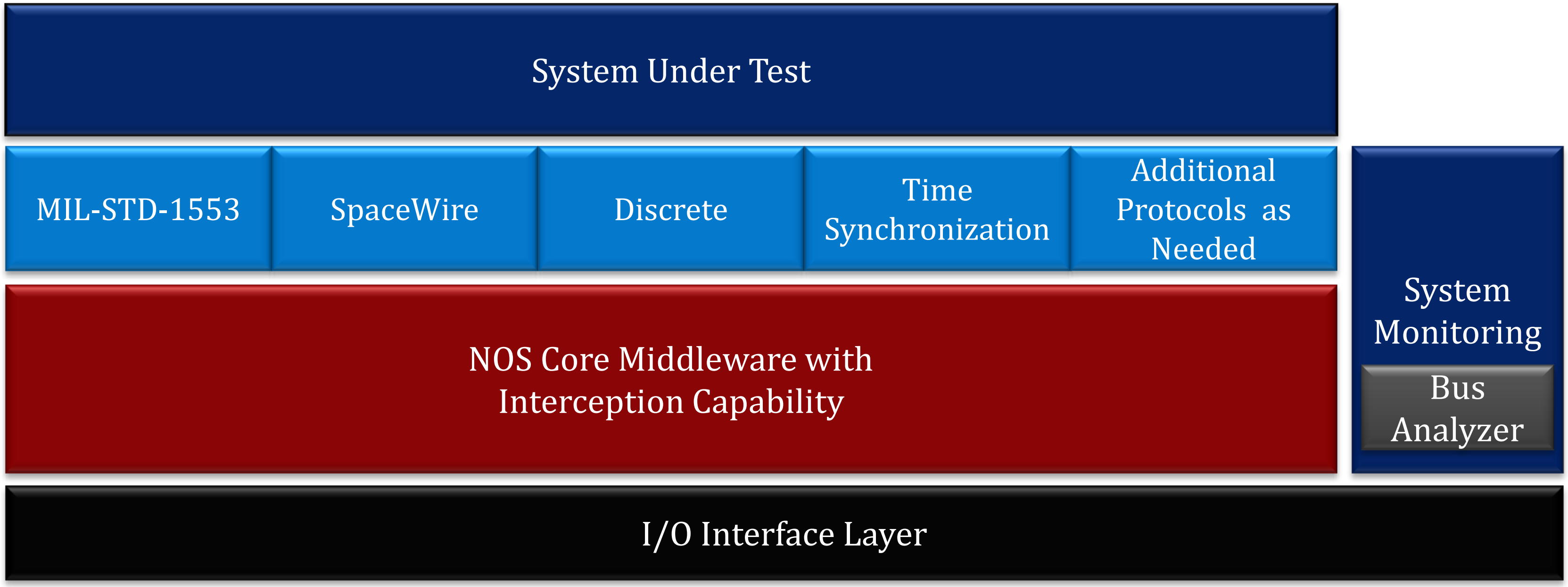
## Overview

- ✓ Offers re-usable communication mechanism
  - Ensures consistent and correct data passing
- ✓ Provides synchronization between distributed applications
- ✓ Flexible and extensible design
  - Can be extended to incorporate any communication protocol

## Features

- ✓ Transport agnostic
- ✓ Cross platform C++ implementation
- ✓ Robust User API
- ✓ Specialized User API Layers
  - MIL-STD-1553B
  - ESA SpaceWire
  - Discrete Signals
  - Time Synchronization
- ✓ Interception allows for V&V analysis
  - No modification to software-under-test

# NOS Middleware Architecture



System Under Test

MIL-STD-1553

SpaceWire

Discrete

Time Synchronization

Additional Protocols as Needed

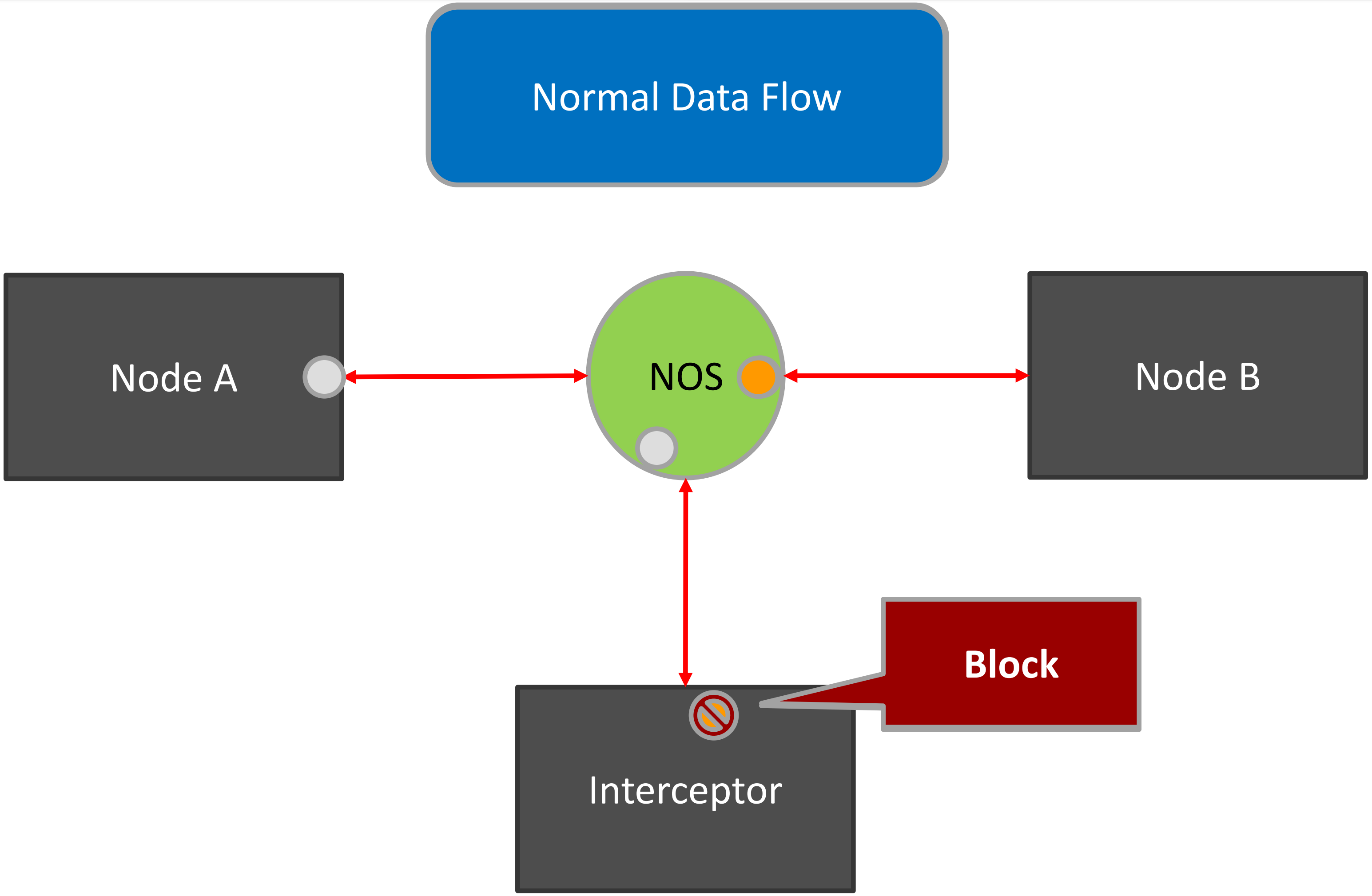
NOS Core Middleware with Interception Capability

System Monitoring

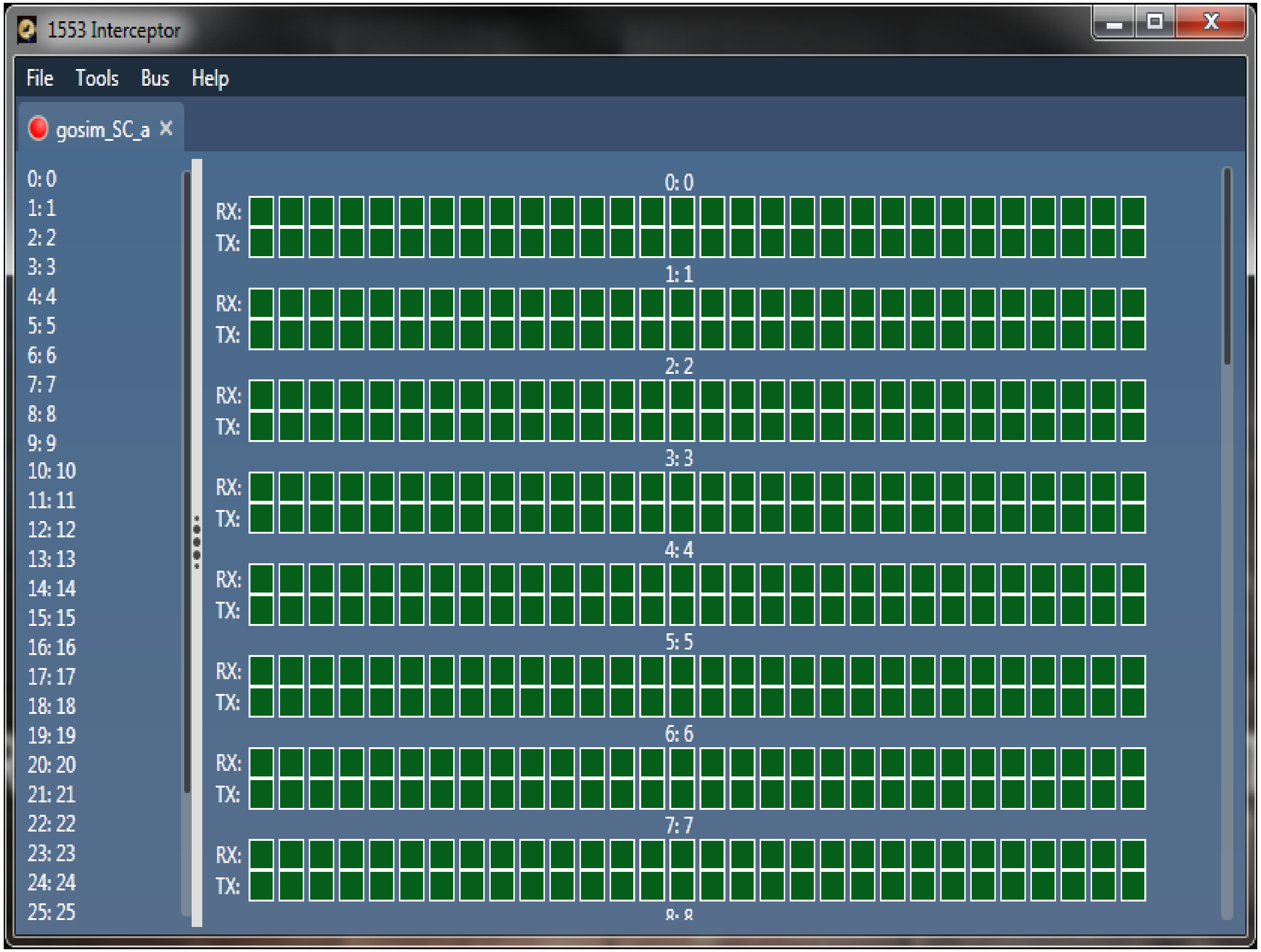
Bus Analyzer

I/O Interface Layer

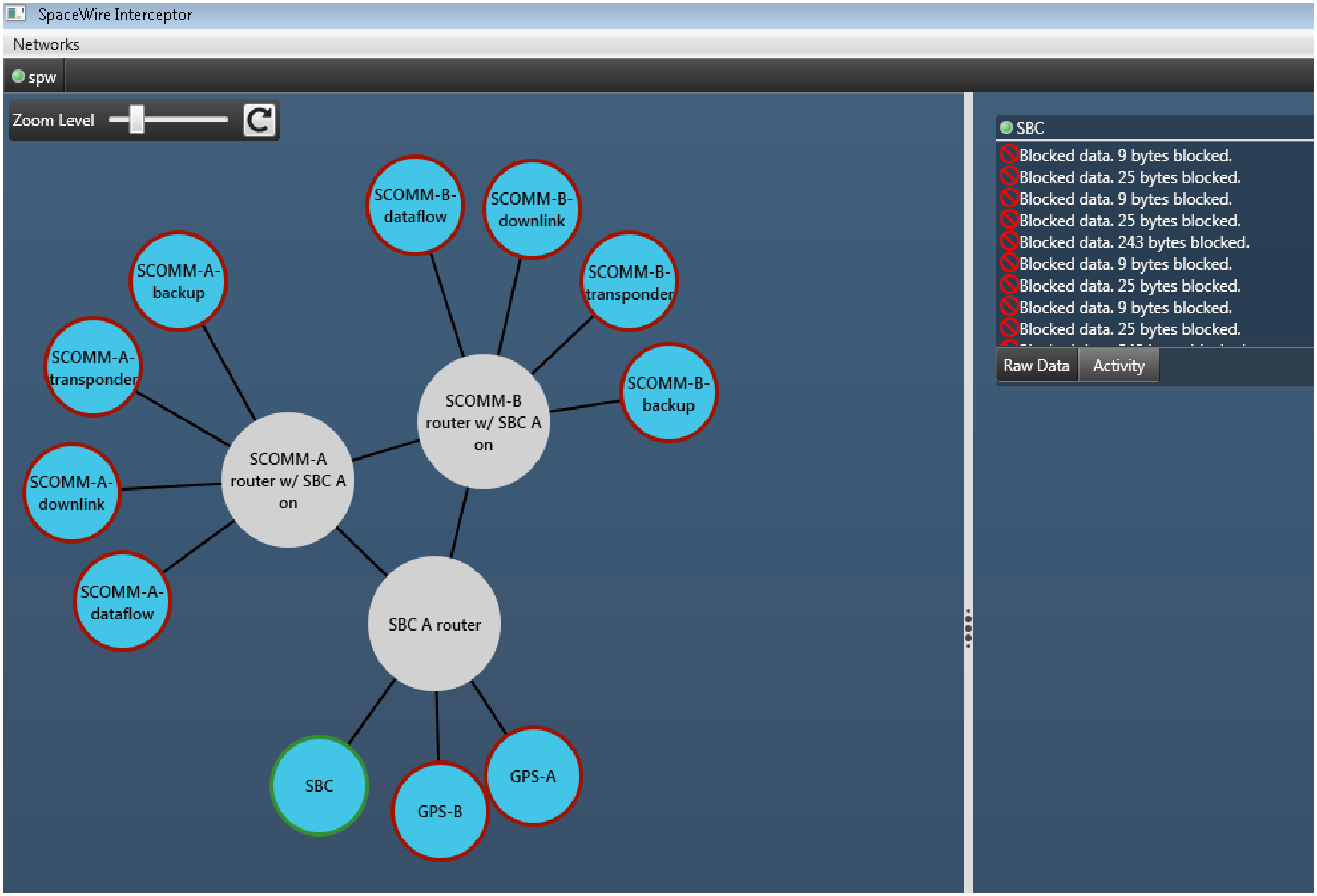
# NOS Dynamic Interception



# NOS User Interfaces
# MIL-STD-1553



# NOS User Interfaces SpaceWire

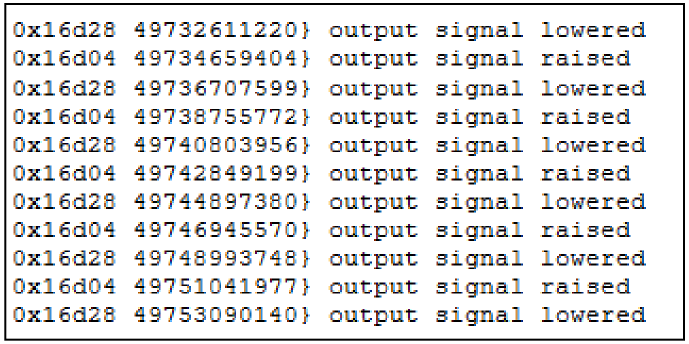


SBC

```
Blocked data. 9 bytes blocked.
Blocked data. 25 bytes blocked.
Blocked data. 9 bytes blocked.
Blocked data. 25 bytes blocked.
Blocked data. 243 bytes blocked.
Blocked data. 9 bytes blocked.
Blocked data. 25 bytes blocked.
Blocked data. 9 bytes blocked.
Blocked data. 25 bytes blocked.
```

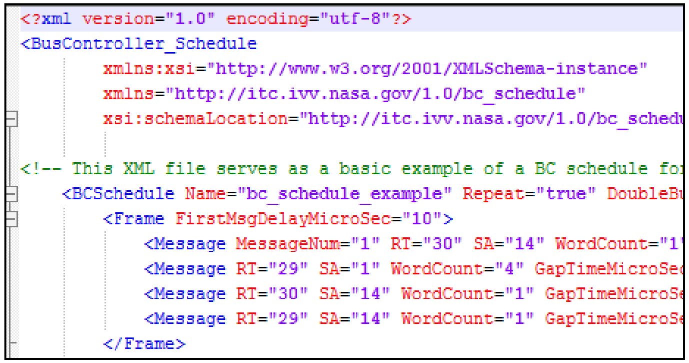
# NOS Software Utilities

- Virtual Oscilloscope
  - Virtual CompactPCI (cPCI) Analysis
  - Board-Level Signal Analysis



```
0x16d28 49732611220} output signal lowered
0x16d04 49734659404} output signal raised
0x16d28 49736707599} output signal lowered
0x16d04 49738755772} output signal raised
0x16d28 49740803956} output signal lowered
0x16d04 49742849199} output signal raised
0x16d28 49744897380} output signal lowered
0x16d04 49746945570} output signal raised
0x16d28 49748993748} output signal lowered
0x16d04 49751041977} output signal raised
0x16d28 49753090140} output signal lowered
```

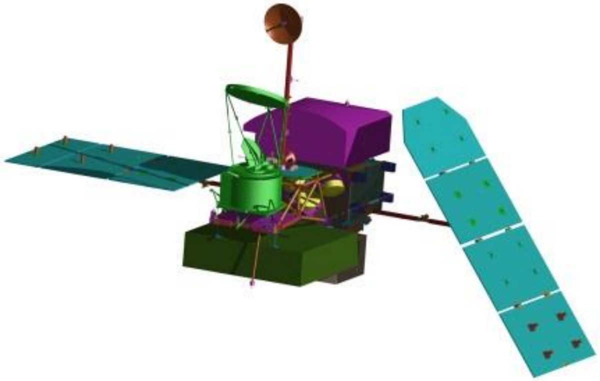
- Virtual MIL-STD-1553 Bus
  - Bus Controller with XML Defined Schedules
  - Remote Terminal
  - Bus Monitor/Logger
  - PASS3200 Software Emulator

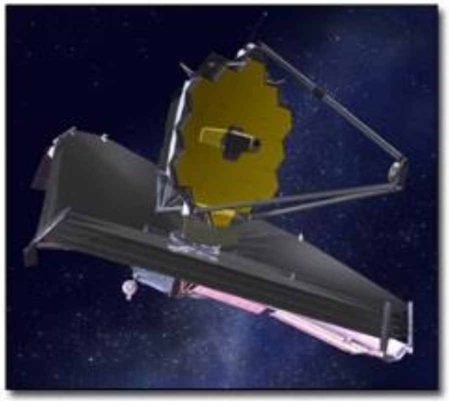


```
<?xml version="1.0" encoding="utf-8"?>
<BusController_Schedule
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns="http://itc.ivv.nasa.gov/1.0/bc_schedule"
        xsi:schemaLocation="http://itc.ivv.nasa.gov/1.0/bc_schedu

<!-- This XML file serves as a basic example of a BC schedule fo
    <BCSchedule Name="bc_schedule_example" Repeat="true" DoubleBu
        <Frame FirstMsgDelayMicroSec="10">
            <Message MessageNum="1" RT="30" SA="14" WordCount="1
            <Message RT="29" SA="1" WordCount="4" GapTimeMicroSe
            <Message RT="30" SA="14" WordCount="1" GapTimeMicroSe
            <Message RT="29" SA="14" WordCount="1" GapTimeMicroSe
        </Frame>
```

- Virtual SpaceWire Router

# NOS Utilization

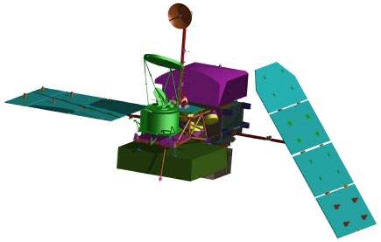




# NOS Utilization

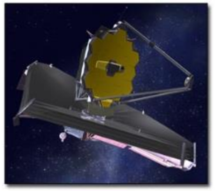
## GO-SIM

1. **GPM GSFC Flight Software Testers**
   - ✓ Dry run test procedures; reduce required use of lab resources
2. **GPM Software Safety**
   - ✓ Tool kit to support safety studies
3. **GPM IV&V Engineers**
   - ✓ Provides flexible testing platform for IV&V personnel
   - ✓ Independent Testing & Risk Reduction



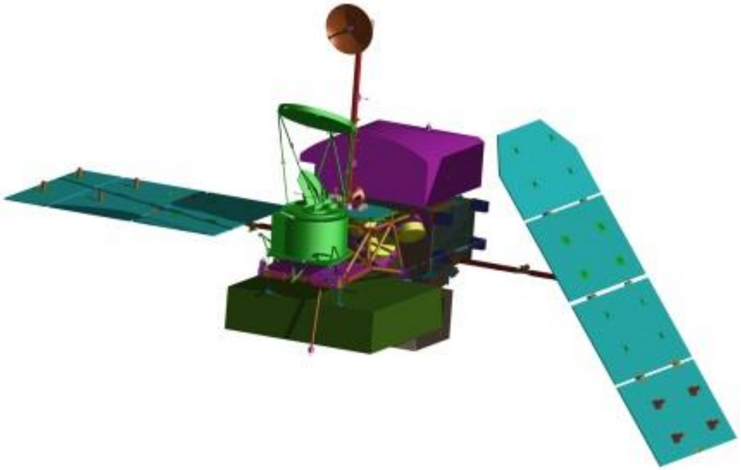
## JIST

4. **JIST Development Team**
   - ✓ Spacecraft simulation environment setup in ½ time due to GO-SIM architecture
5. **JWST IV&V Engineers**
   - ✓ Risk reduction simulator under development
   - ✓ Supported processor offline mode test efforts



# NOS Utilization

## GPM Operational Simulator (GO-SIM)



Independent Test Capability

# GPM Operational Simulator GO-SIM

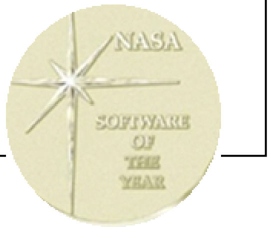
## Components

- COTS Emulator
- Primary Instrument Simulations (GMI/DPR)
- GPM Ground System
- GSFC Goddard Dynamic Simulator (GDS)
- NOS Middleware
- GPM Hardware Models

## Capabilities

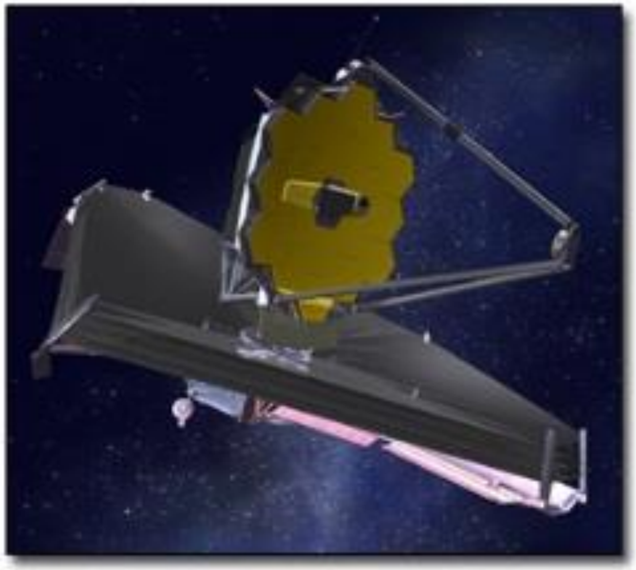
- Load and run unmodified flight software binaries
- Execute test flight scripts
- Single-step debugging
- Inject errors via ground system and NOS middleware
- Stress system under test

**NASA Software of the Year Honorable Mention 2012**



# NOS Utilization

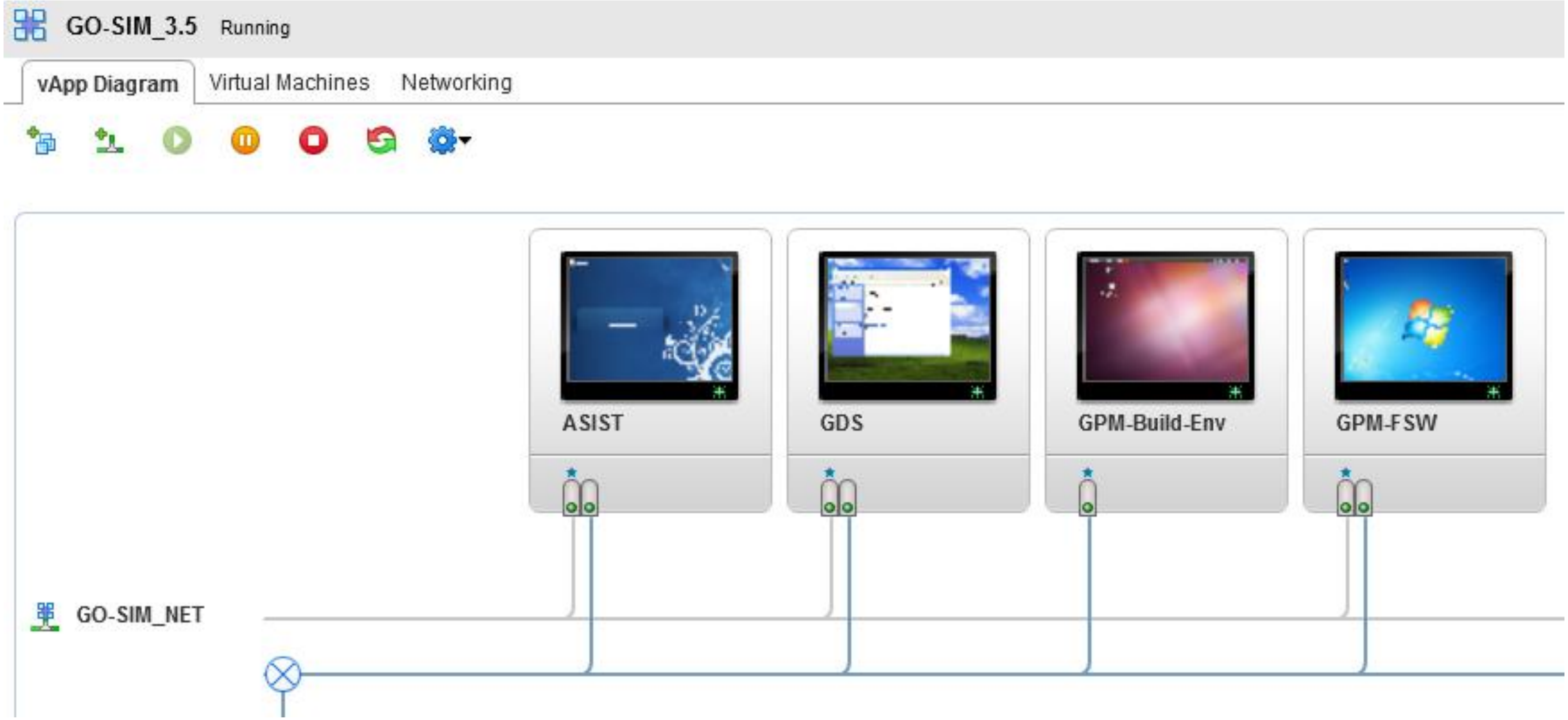
## James Webb Space Telescope (JWST) Integrated Simulation and Test (JIST)



# JWST Integrated Simulation and Test (JIST)

- Software-only spacecraft simulator
- Flexible environment to support V&V activities
- Unmodified ground system and scripts
- Unmodified software-under-test binaries
- Integration of COTS, GOTS and in-house developed components
- Custom hardware models
- Automated Testing Framework

# Virtualized Deployment



# Closing Remarks

- NOS provides a generic software-only simulation architecture that has been  utilized on NASA missions
- NOS architecture is transparent to user
- New instantiations of NOS require customization for missions/projects → NOS has demonstrated significant cost and time savings
- NOS provides reusable hardware models
- NOS provides custom-developed middleware with user APIs and interception
- NOS extends to other domains
  - Large complex systems
  - Distributed components

# Contact Information

- Web Page
  - http://www.nasa.gov/centers/ivv/jstar/JSTAR.html
- E-Mail
  - Justin.R.Morris@nasa.gov
  - Team Mailing List: ivv-itc@lists.nasa.gov