SYSTEMS ENGINEERING RESEARCH CENTER

New Observing Strategies Testbed (NOS-T) Design and Development

ART-015 (PI: Dr. Paul Grogan)

Presenter: Dr. Matthew J. LeVine

Sponsor: NASA Earth Science Technology Office

STEVENS INSTITUTE OF TECHNOLOGY

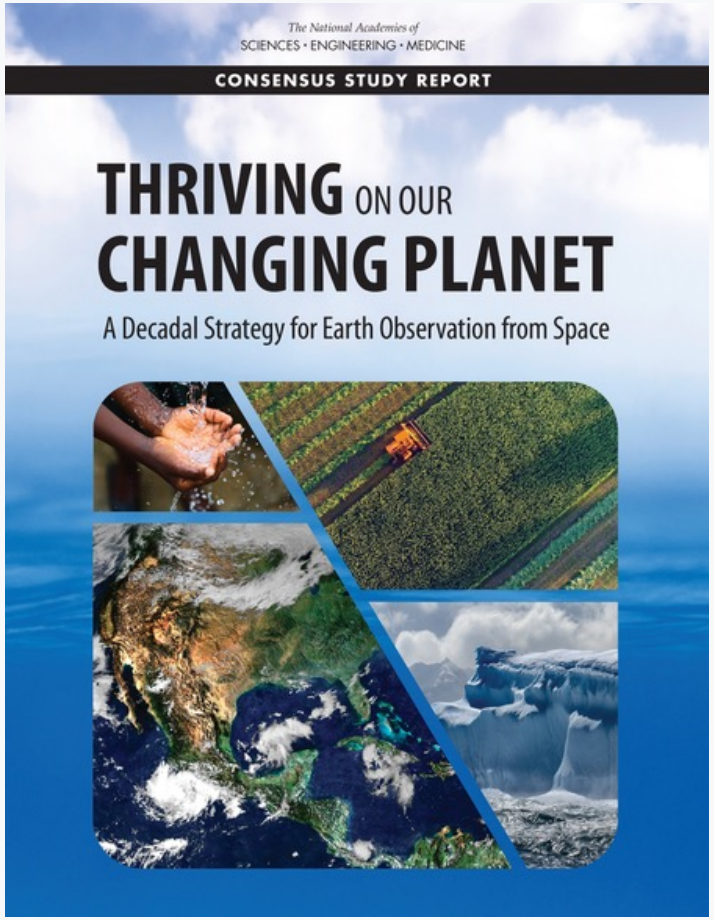
ANNUAL RESEARCH REVIEW 2022

Presentation Overview

- Context and Background
  - NASA's Earth Science Program
  - New Observing Strategies (NOS)
- NOS Testbed Framework
  - Development Principles
  - Concept of Operations
  - Governance
  - System Architecture
- Example Application Cases:
  - Simulated Mission
  - Real-time Mission
  - Tasking/Collecting Demonstration
- Technical Architecture
  - Event-driven Architecture
  - Application Interfaces

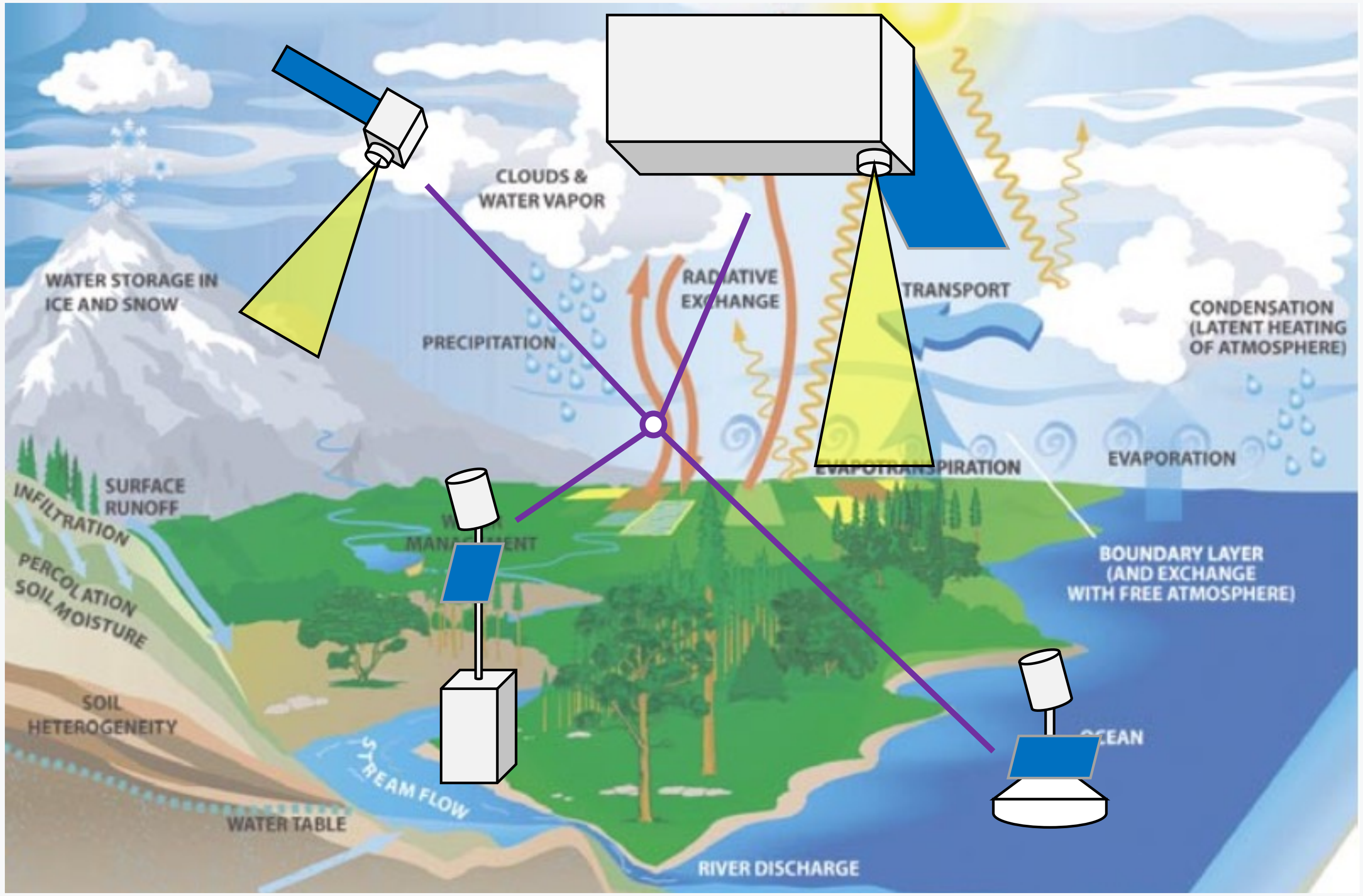
Earth Science Decadal Survey Strategy Elements

1. Sustained Science and Applications
2. **Innovative Methodologies**
3. Cross-Benefit of Science and Applications
4. **External Resources and Partnerships**
5. Programmatic Agility and Balance
6. **External Trends**
7. Competition
8. Ambitious Science and Applications



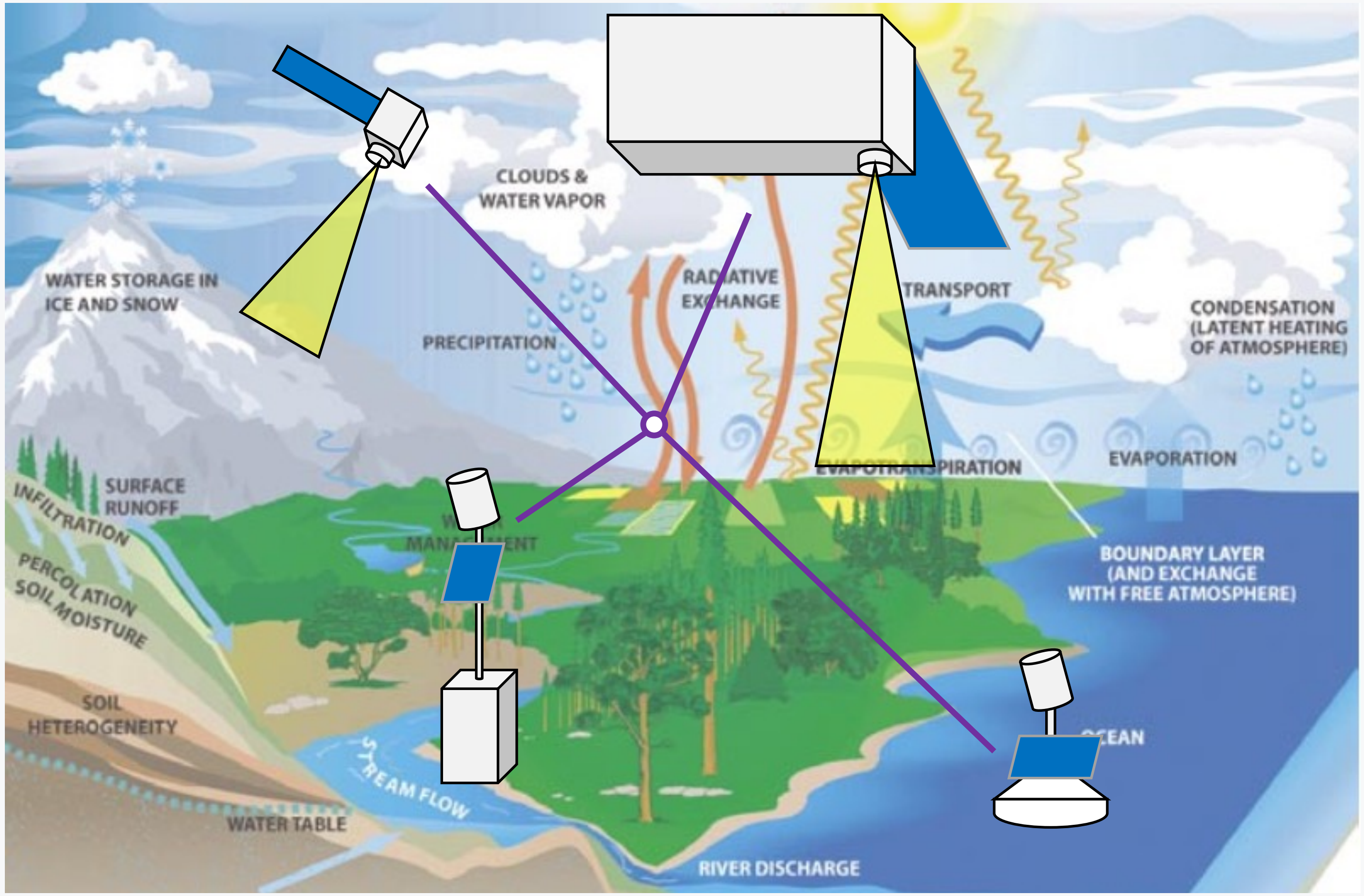
https://doi.org/10.17226/24938

New Observing Strategies (NOS)



- **Optimize** measurement acquisition using diverse observing capabilities
- **Observe** phenomena from different spatial, temporal, and spectral vantage points
- **Coordinate** observations based on events, forecasts, or science models
- **Leverage** NASA and non-NASA assets and data sources

NOS Testbed (NOS-T) Programmatic Objectives

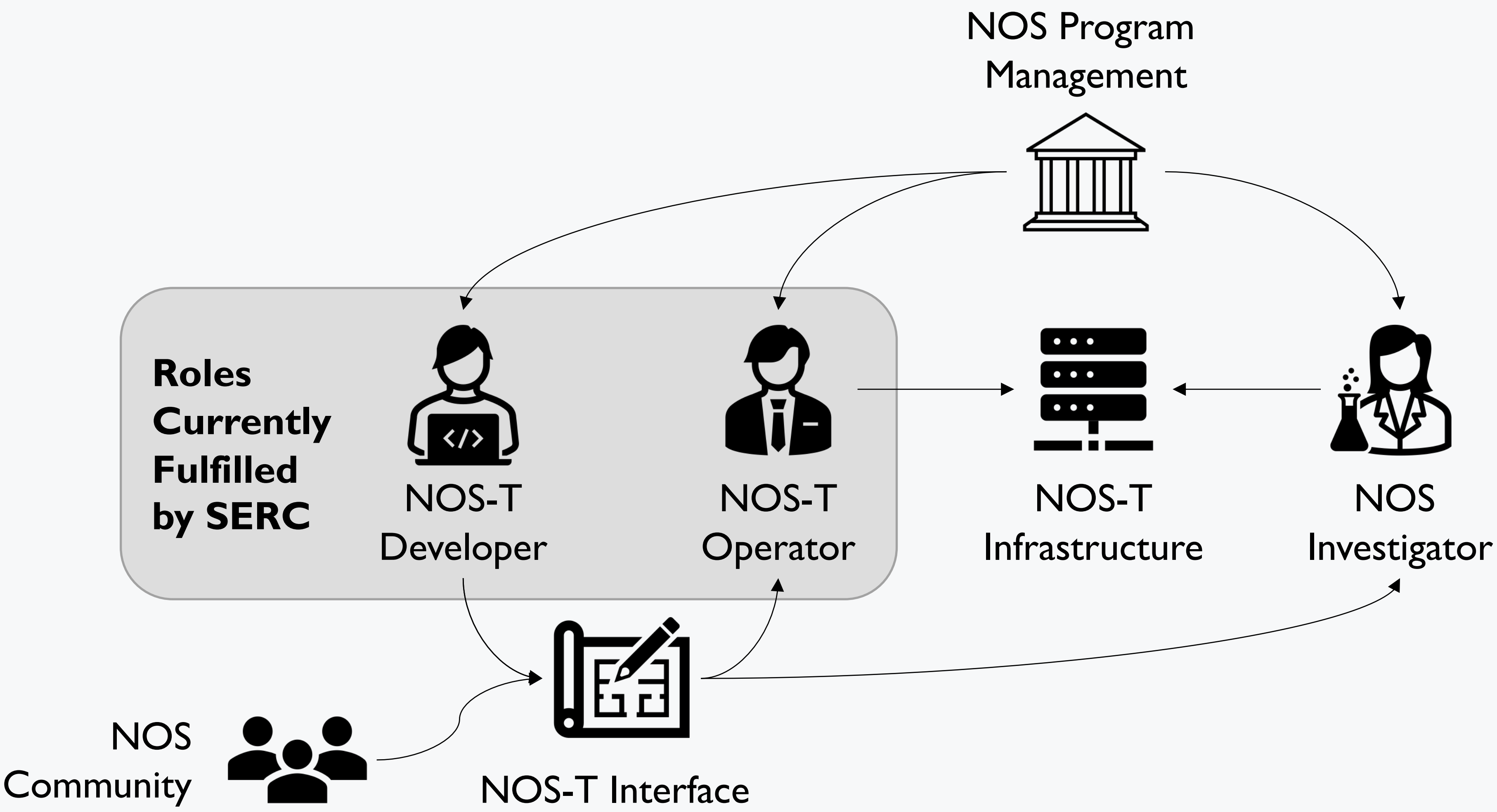


- **Validate** NOS technologies independently and as a system
- **Demonstrate** new distributed operational concepts
- **Enable** comparisons of competing technologies
- **Socialize** new technologies and concepts with the science community and reduce risk

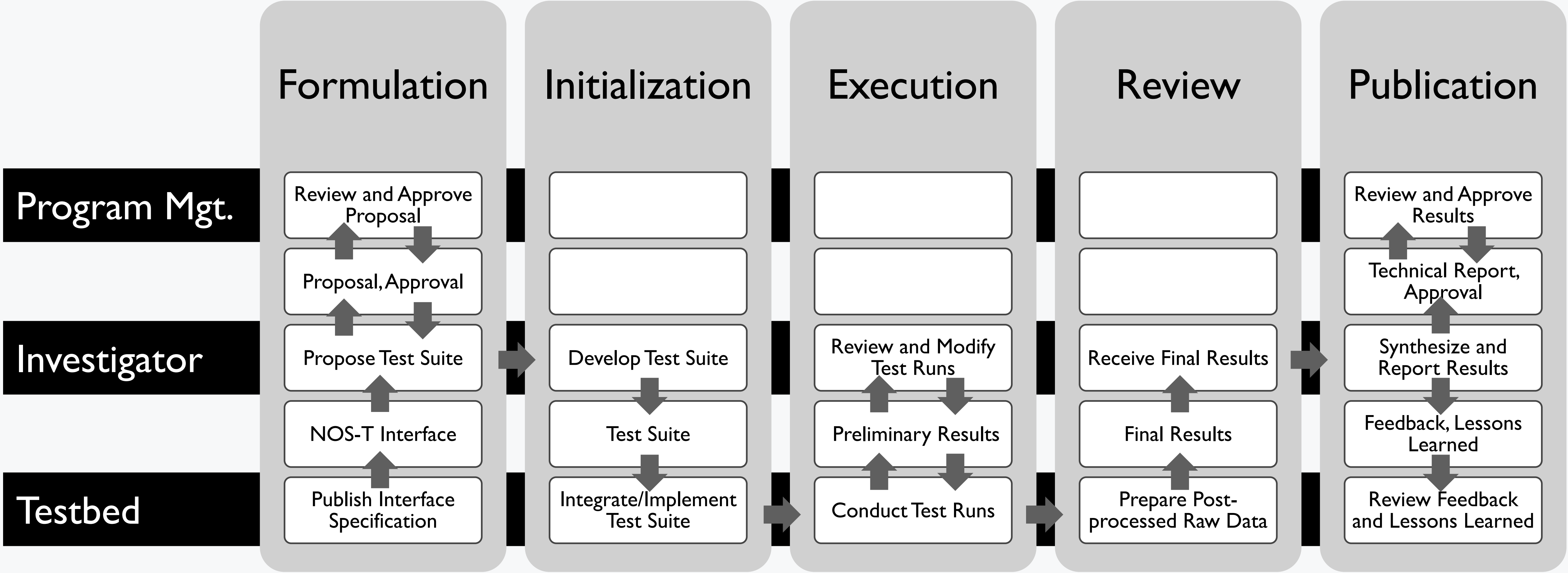
NOS-T Design and Development Objectives

- **Enable disparate organizations to propose and participate in developing NOS software and information tech using the Testbed**
- NOS-T Framework Architecture:
  - Governance
  - Concept of Operations
  - Technical Protocols and Interfaces
- Iteratively develop prototypes and demonstrate NOS-T operation for a representative Earth science mission with at least three nodes
  - Version 1.0 (18 months ending February 2022)
  - Version 2.0 (18 months ending August 2023)

NOS-T Framework: Governance



NOS-T Framework: Concept of Operations



NOS-T Framework: Technical Principles

Geographic distribution: user applications interconnect using standard network interfaces

Multi-party participation: user applications exchange limited information via standard messaging protocols

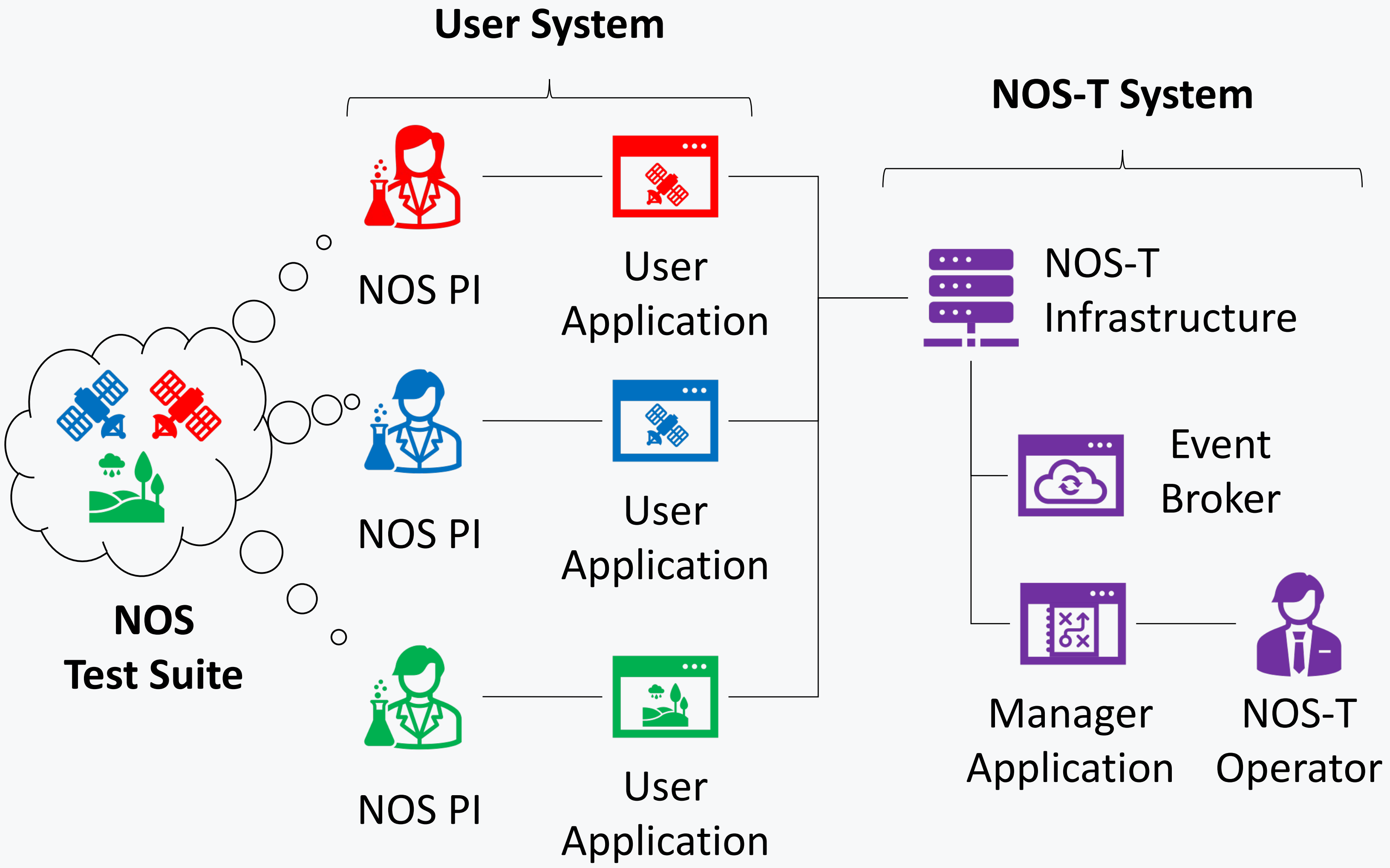
Security: encrypt transport data, provide fine-grain access control rules, monitor hosted infrastructure on authorized information systems

Modularity: loose coupling allows components to be added or updated without modifying the testbed

Extensibility: vary the number or capabilities of user applications to explore a wide range of test cases

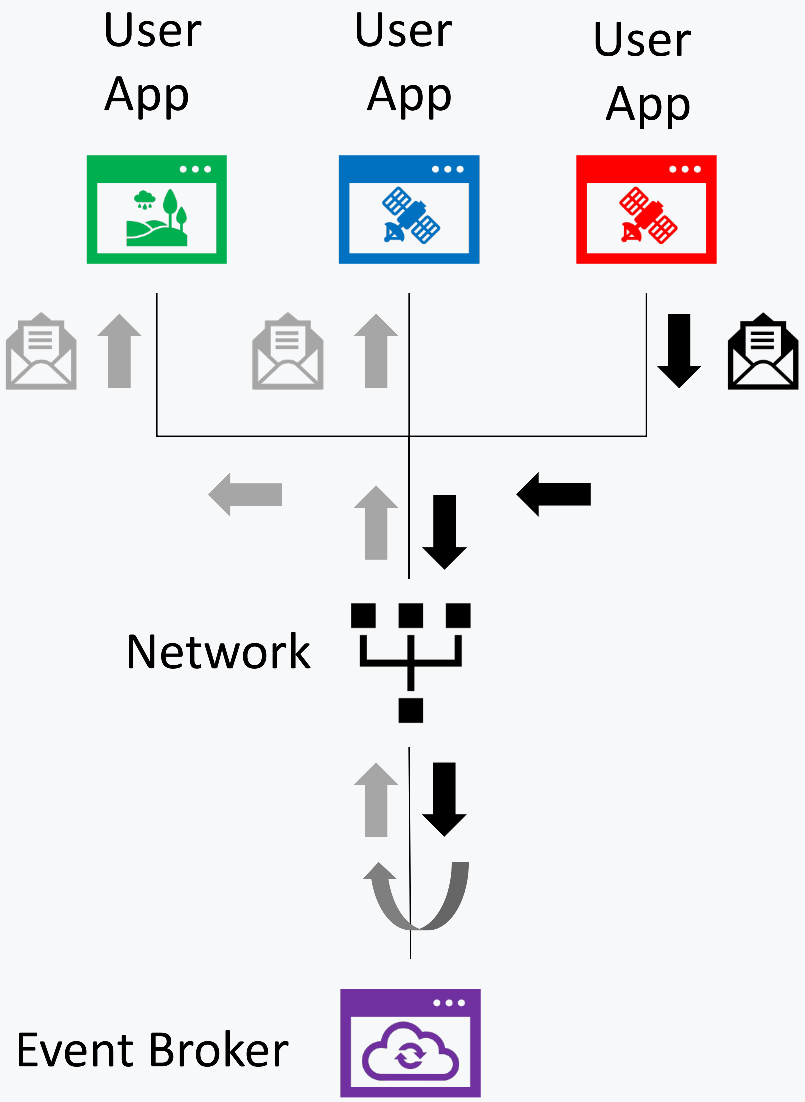
Usability: allow members of the Earth science community to develop test cases and user applications without a substantial learning curve

NOS-T Graphical Concept

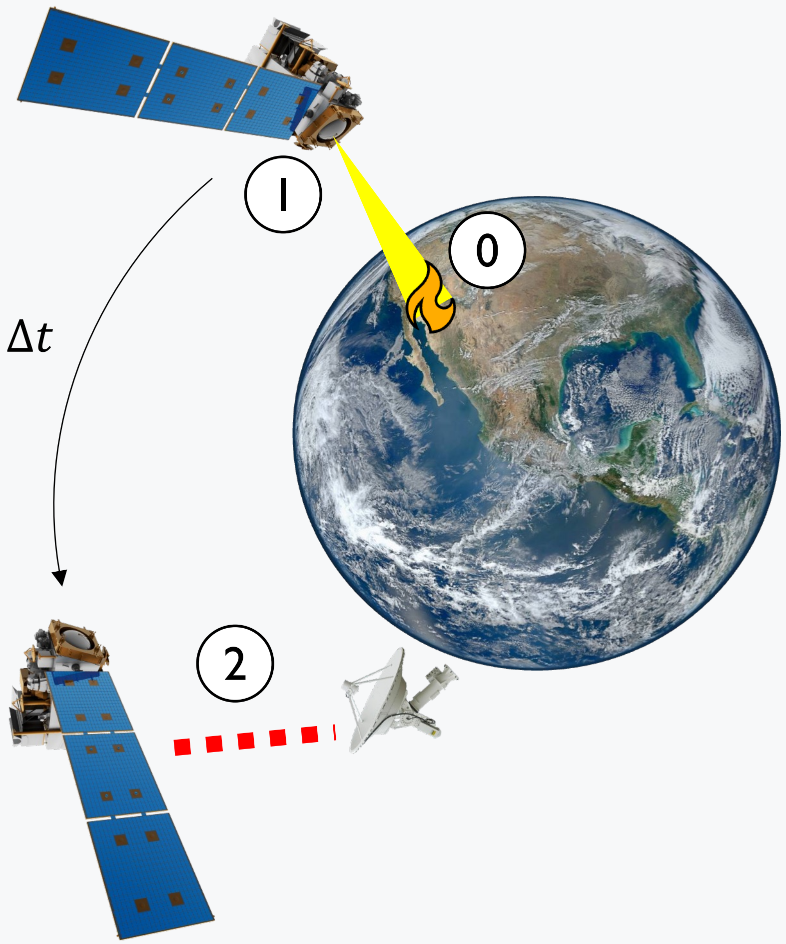


Event-driven Architecture

- Applications communicate state changes via *events* (messages)
  - **Published** to topics
  - Event broker notifies all **subscribers**
- Broker: *Solace PubSub+ Standard*
  - MQTT messaging protocol
  - Up to 1000 concurrent connections and 10,000 messages/second
  - Hosted on the Science Managed Cloud Environment (SMCE), a FISMA Low cloud information system



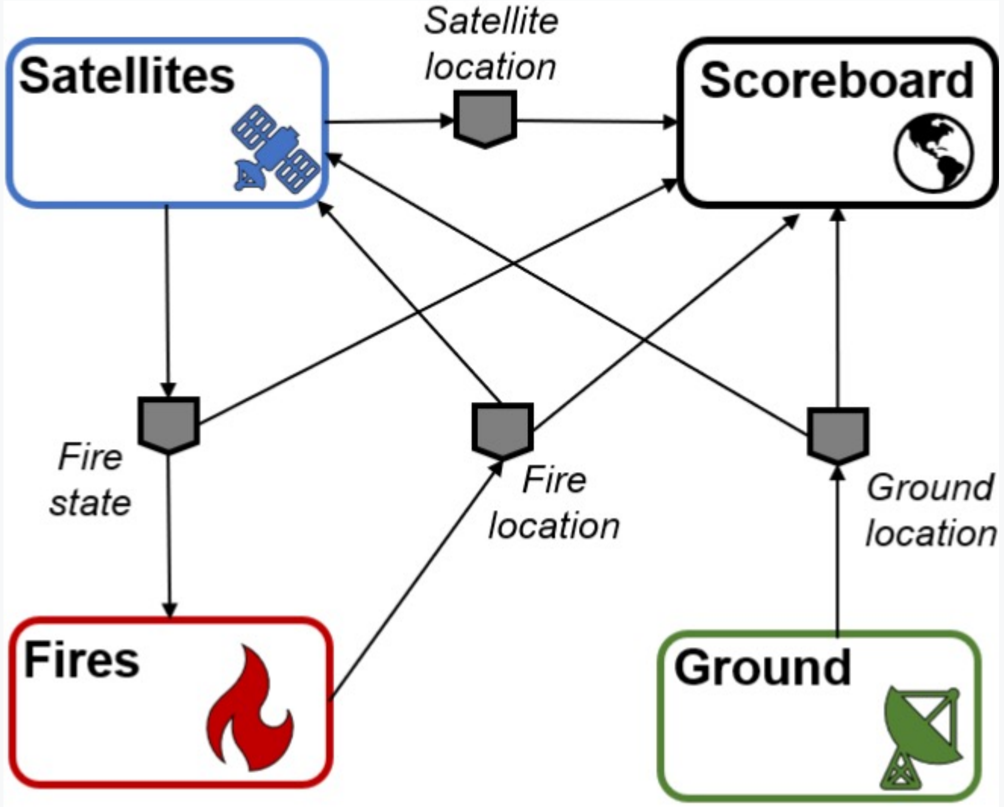
Application Case 1: Simulated Time Execution



- Investigate fire hazard detection in the continental U.S.
  - Initiate fires using 2020 VIIRS data
  - Remote observation by three-satellite constellation
  - Data downlink to ground station
  - Evaluation of key performance measures (observation latency)
- Extensible to design-of-experiment studies to assess observation system variables

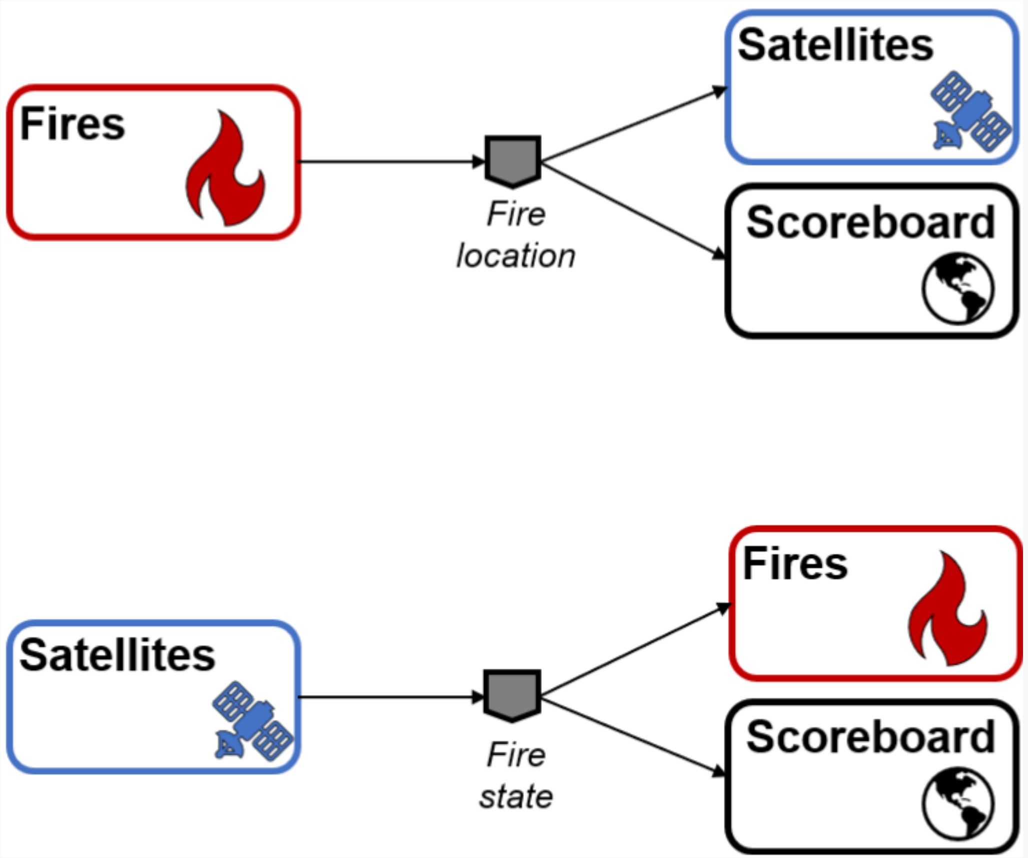
5-day (at 60x Scale) Scenario; Playback Speed ~90x

FireSat+ Test Case Architecture



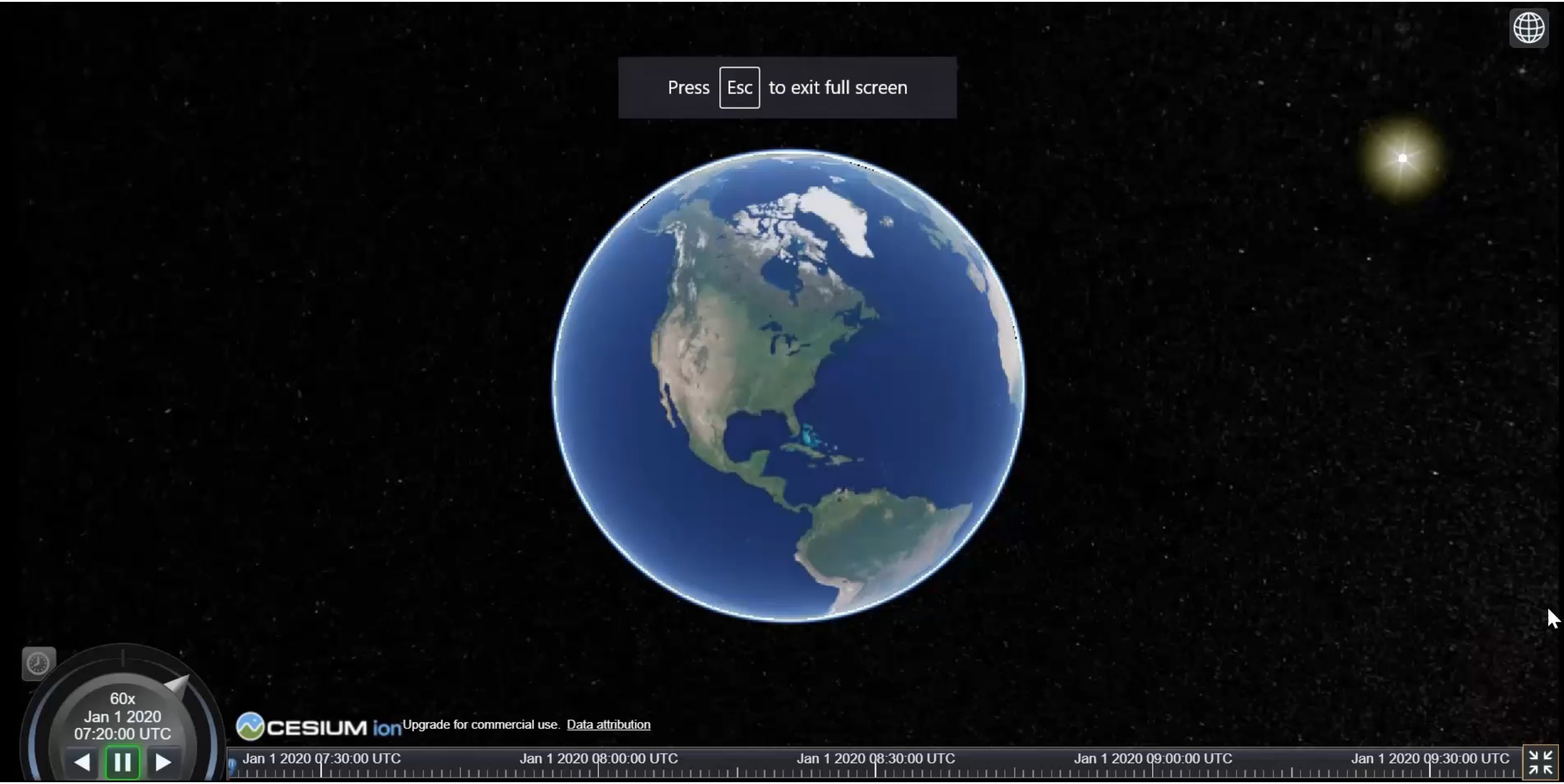
- **Fires:** publishes fire location, records times started, detected, reported
- **Ground:** publishes ground station location
- **Satellites:** models orbit propagation, detects fires, reports fires when link to Ground is possible
- **Scoreboard:** displays graphical representation of mission

FireSat+ Interface Sample



- Topic:
  - ➢ nost-001/fires/location
- Payload:
  - ➢ Fire ID, ignition lat/lon, timestamp

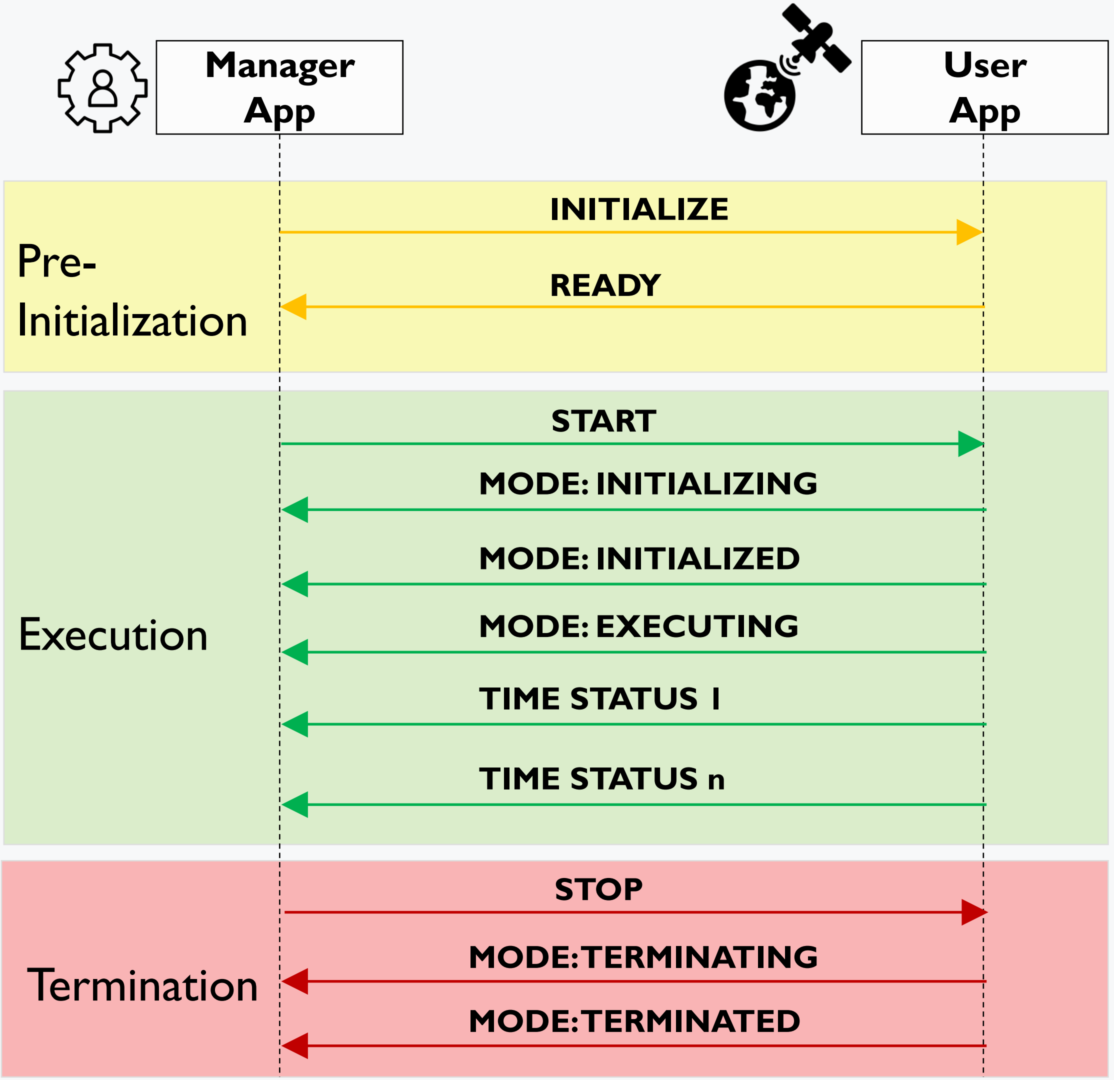
- Topic:
  - ➢ nost-001/satellite/detected
  - ➢ nost-001/satellite/reported
- Payload:
  - ➢ Fire ID, satellite ID, timestamp, state


Press Esc to exit full screen
60x
Jan 1 2020
07:20:00 UTC
CESIUM ion
Upgrade for commercial use. Data attribution
Jan 1 2020 07:30:00 UTC
Jan 1 2020 08:00:00 UTC
Jan 1 2020 08:30:00 UTC
Jan 1 2020 09:00:00 UTC
Jan 1 2020 09:30:00 UTC

NOS-T Technical Interface

Manager App Messages

- **Initialize** = Start application
- **Start** = Begin Test Run
- **Stop** = Stop Test Run

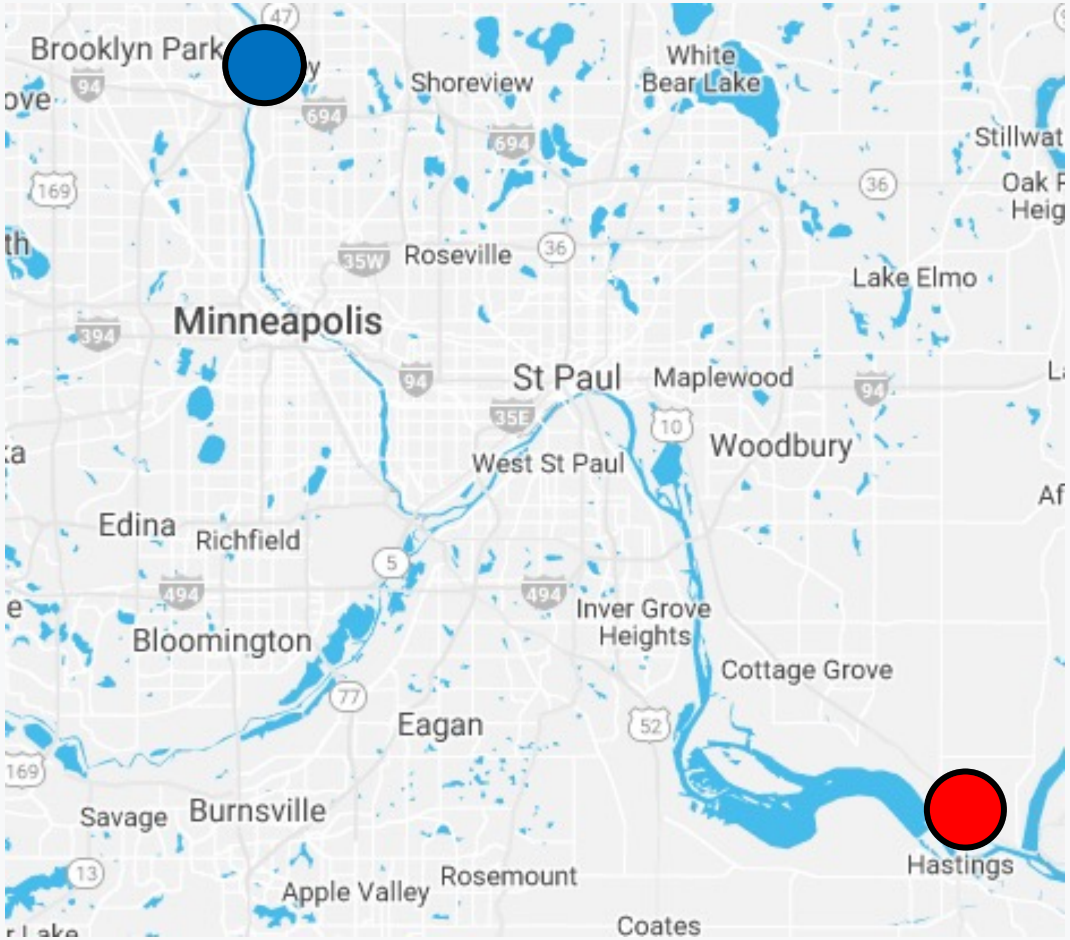


User App Messages

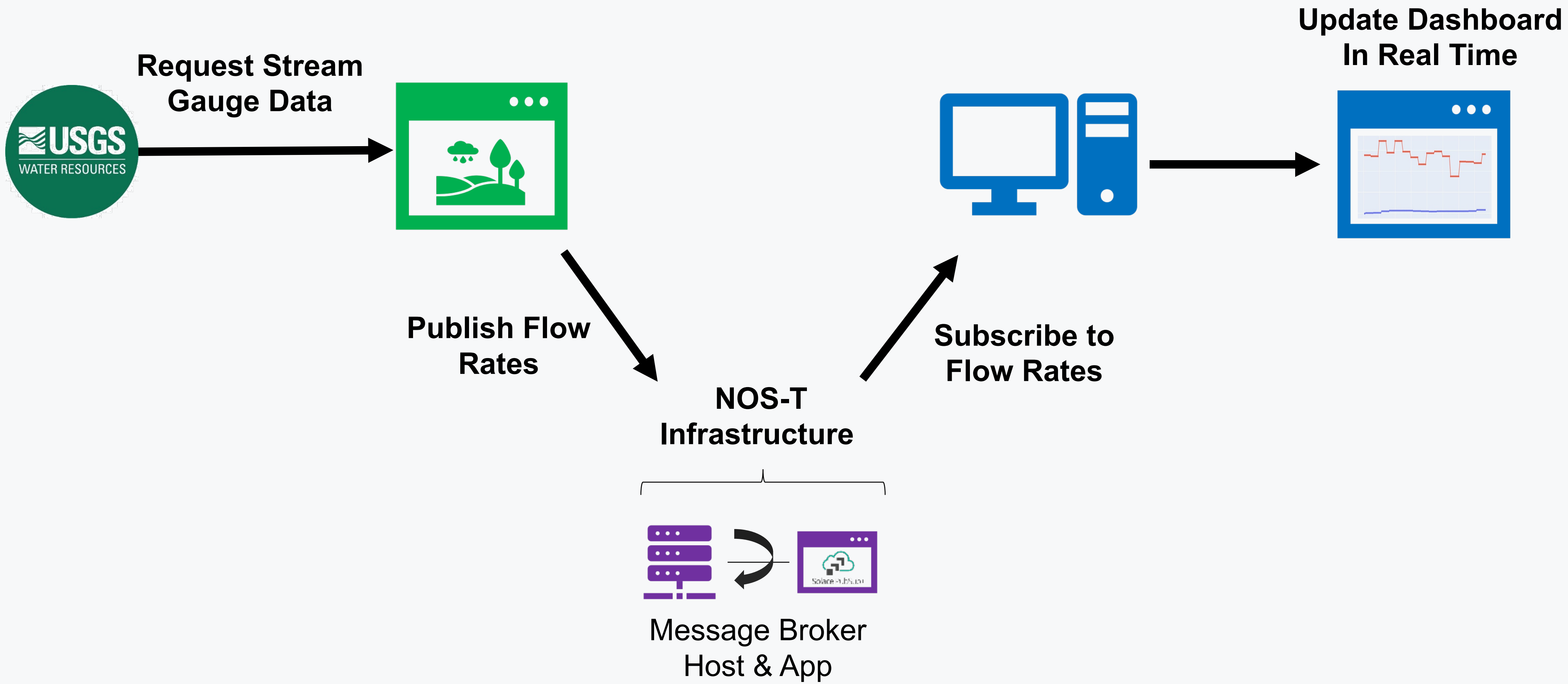
- **Ready** = Ready to Start Test Run
- **Mode: Initializing** = App begins initialization
- **Mode: Initialized** = App is initialized
- **Mode: Executing** = App is Executing
- **Time Status** = Heartbeat
- **Mode:Terminating** = App begins termination
- **Mode:Terminated** = App has terminated run

Application Case 2: Real-time Test Case

- Real-time stream gauge data retrieved via web requests from the USGS National Water Information System (NWIS)
  - ➢ Displays flow rates from two sensors on a dashboard – Mississippi River above and below Minneapolis/St. Paul
  - ➢ Demonstrates ability to use real-time data for a test case
- Extensible to trigger spacecraft observations when certain flow rates are met



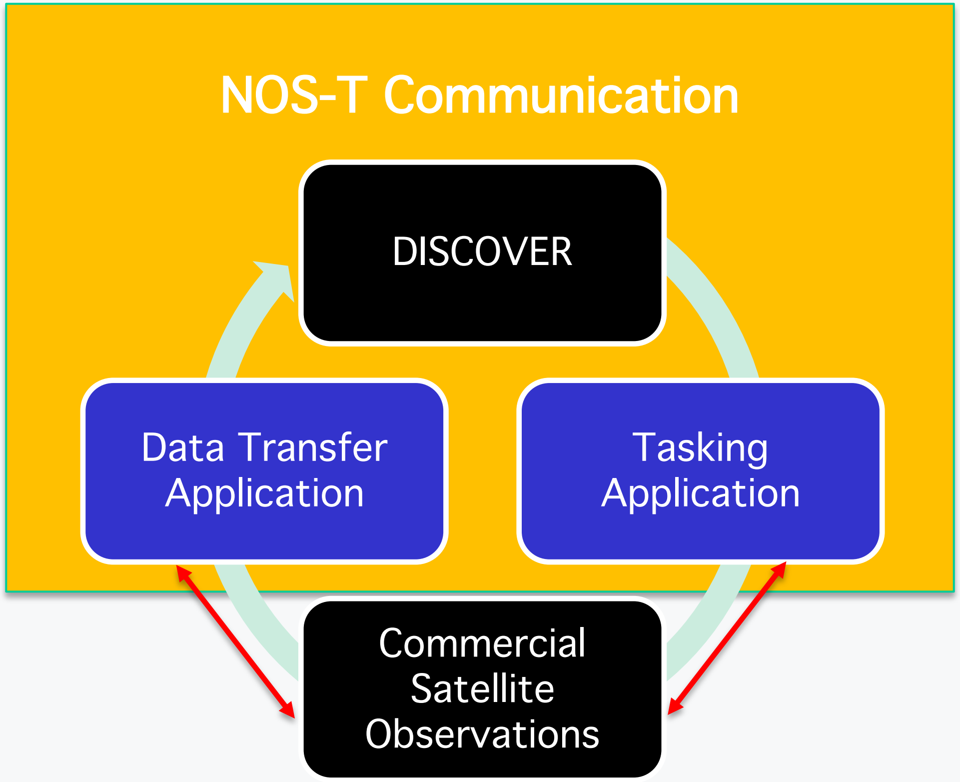
Real-time Case: Application Architecture



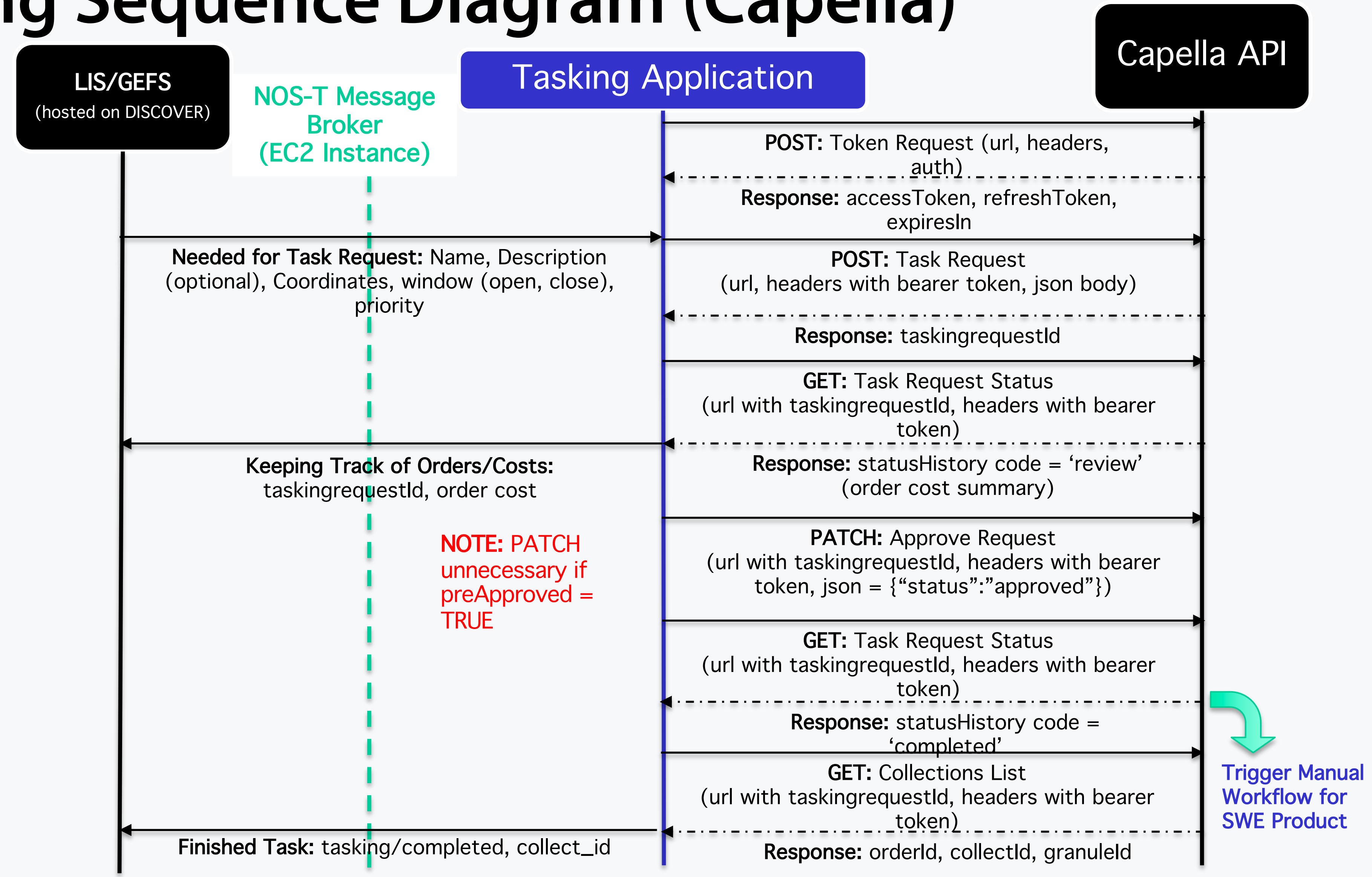
15-hour (real-time) scenario; 2000x playback

Application Case 3: NOS-Live (NOS-L) Demonstration

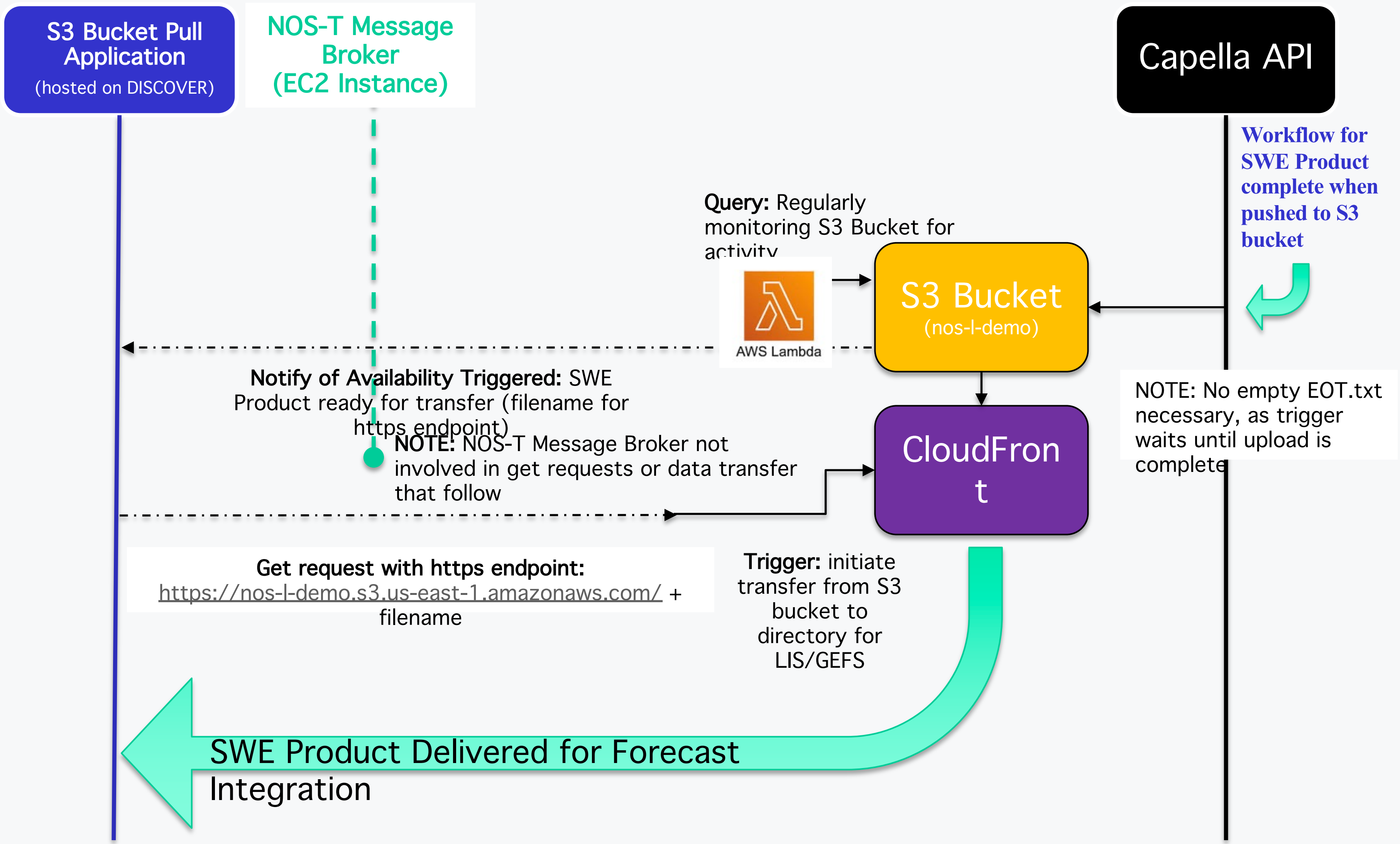
- Design and demonstrate an event-driven architecture to automatically task, collect, and integrate data products into forecasts
- Essentially a four-node problem:
  - ➢ Science application identifying area of interest (Lon, Lat, time window) – NASA GSFC
  - ➢ Tasking application to interface with commercial satellite provider API – NASA JPL
  - ➢ Commercial satellite providers with automated API interface
  - ➢ Data transfer application to move collected data products to correct location – Stevens



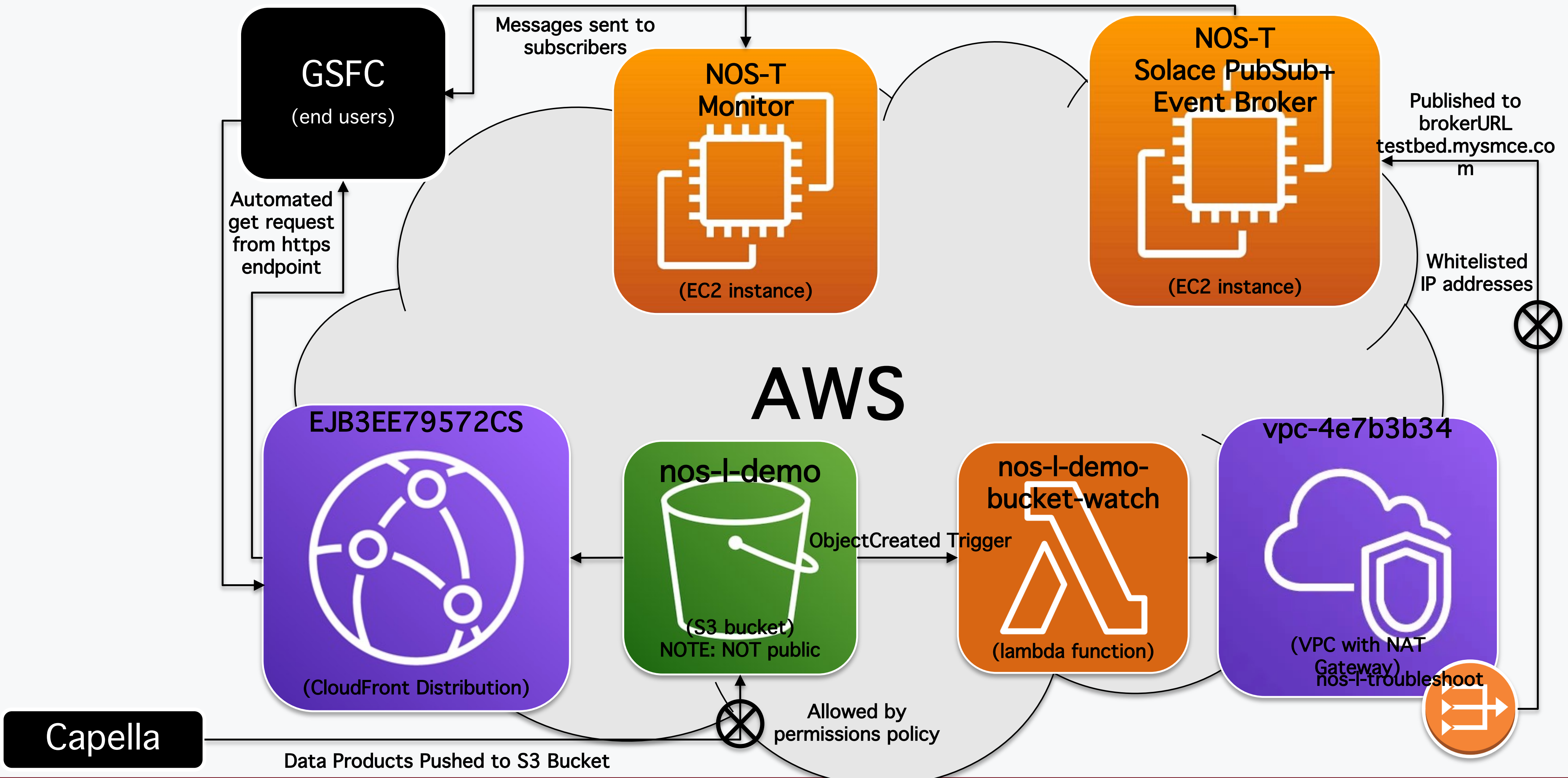
Tasking Sequence Diagram (Capella)



Data Access Sequence Diagram (Capella)



Amazon Web Services: Supporting Architecture



Summary

- NOS-T provides computational platform to prototype NOS missions
- NOS-T framework provides an initial governance model, concept of operations, and technical interface specification
  - Event-driven architecture
  - Simulated and real-time execution
- NOS-T Tools Python library available under an open-source license:
  - Repository: https://github.com/code-lab-org/nost-tools
  - Documentation: https://nost-tools.readthedocs.io/en/latest/

Acknowledgements

This material is based on work supported, in whole or in part, by the U.S. Department of Defense through the Systems Engineering Research Center (SERC) under Contract No. W15QKN-18-D-0040.

Thanks to team members Brian Chell, Leigha Capra, Theodore Sherman and alumni Hayden Daly and Matthew Brand for their contributions.

PI: Paul Grogan, pgrogan@stevens.edu, 201-216-5378

SYSTEMS ENGINEERING RESEARCH CENTER

THANK YOU

Stay connected with us online.