

EARTH SYSTEM observatory

ATMOSPHERE

Open Source Science for ESO Mission Processing Study

Workshop #1 October 19-20, 2021

AOS Mission Data Processing System Perspective Robert Wolfe, Curt Tilmes and Dave Meyer

NASA Ground Stations

Center Ground Data

External Data

Providers

System

Raw Instrument Data

Ancillary Data

Pointing, etc.)

Auxiliary/Model

Data

(Ephemeris/Orbit,

AOS Project Data Processing System

Receive/Ingest Data

- Instrument & spacecraft data
- Externally provided data
- Standard Data Products

Algorithm Development & Validation

- Develop algorithms
- Integrate algorithms into production container images
- Provide access to software and container images
- Host instrument calibration tools
- Host cal/val & analysis tools

Data Processing

.

- Level 0 Processing
- Level 1 Processing
- Level 2 Processing
- Level 3+ Processing
- Bulk Reprocessing Campaigns
- Low Latency Processing

Store/Catalog Data Products

- Standard Data Products
- Ancillary/Auxiliary Data
- Internal & Temporary Data

Data Access

- Deliver to DAAC, External Partner SDS
- Provide User interface/ Search for data users

Quality Assessment

- Develop analysis tools
- Perform data quality assessment

Report Generation

• Data accountability, quality, system performance, product latency, etc.



AOS Data Processing System Architecture

Pre-Decisional



AOS Data Processing System Architecture

Pre-Decisional

Current Implementation Plan

EARTH SYSTEM OBSERVATORY

- Early development of simulation data and processing in the Analytic Collaborative Environment
- Algorithm Software will decomposed into reusable parts (libraries, executables, etc.)
 - Some parts can be shared across multiple algorithms.
 - Some parts can come from legacy missions.
 - Each algorithm will need to be analyzed and decomposed.
- Software will be developed within a Configuration Management Software Repository with automated Continuous Integration / Continuous Delivery / Continuous Deployment
- Production will be based on Algorithm Software fully integrated with runtime environment into Container Images that can run across multiple platforms
- Legacy Missions:
 - MODIS/VIIRS, OMI/OMPS, GPM, CrIS, CloudSat, CALIPSO, etc.

Project Schedule

Pre-Phase A Start	2021-06-01
Preliminary SDS Architecture and Concept of Operations	2022-03-01
SDS Verification and Validation Approach	2022-04-01
Preliminary SDS Requirements	2022-04-01
Misson Concept Review	2022-05-15
KDP-A	2022-06-30
Preliminary SDS Verification and Validation Plan	2023-02-15
Preliminary SDS Configuration Management Plan	2023-02-15
SDS System Requirements Review	2023-02-15
Mission SRR	2023-04-30
KDP-B	2023-06-30
Define At-Launch Products/Processes and Draft Early ATBDs	2024-08-01
SDS Preliminary Design Review	2024-08-01
Mission PDR	2024-10-01
KDP-C; Mission Confirmation	2024-12-01
SDS Critical Design Review	2025-08-01
Mission CDR	2025-10-01
KDP-D (Inclined)	2027-08-01
I Launch	2028-09-01
P Launch	2030-05-02

EARTH SYSTEM OBSERVATORY

Supporting Open Science

- AOS will support Open Science: A collaborative culture enabled by technology that empowers the open sharing of data, information, and knowledge within the scientific community and the wider public to accelerate scientific research and understanding.¹
- Given appropriate maturity, intellectual property, legal rights and credentials (Full public access where possible), the following will be available:
 - Open Data will be stored in an Open Object Store
 - Open Source Code, including libraries, production algorithms, analysis software, etc. will be stored in an Open Configuration Management Repository
 - Open Container Images which encapsulate the runtime environment with the integrated algorithm allowing complete production reproducibility on multiple platforms
- To the maximum possible extent software will be developed in the open with full transparency

¹Ramachandran, R., Bugbee, K., & Murphy, K. J. Moving from Open Data to Open Science. Earth and Space Science, Wiley Publication https://doi.org/10.1029/2020EA001562

Pre-Decisional

Other

- Allow new processing software frameworks to be adopted and adapted rather than standardizing on one "stagnant" solution that tries to fit everyone
- Standardize *interface* from algorithm to processing framework
 - Encapsulate runtime environment with container images compatible with Open Container Initiative to ensure broadest possible support
 - It should be possible to plug in any algorithm into any processing framework
 - Keep interface standards as simple as possible as a "lowest common denominator" to allow maximum flexibility rather than trying to dictate how everything gets done
 - Allow flexibility in production rule specification