OAK RIDGE National Laboratory

WebMon – overview and recent work

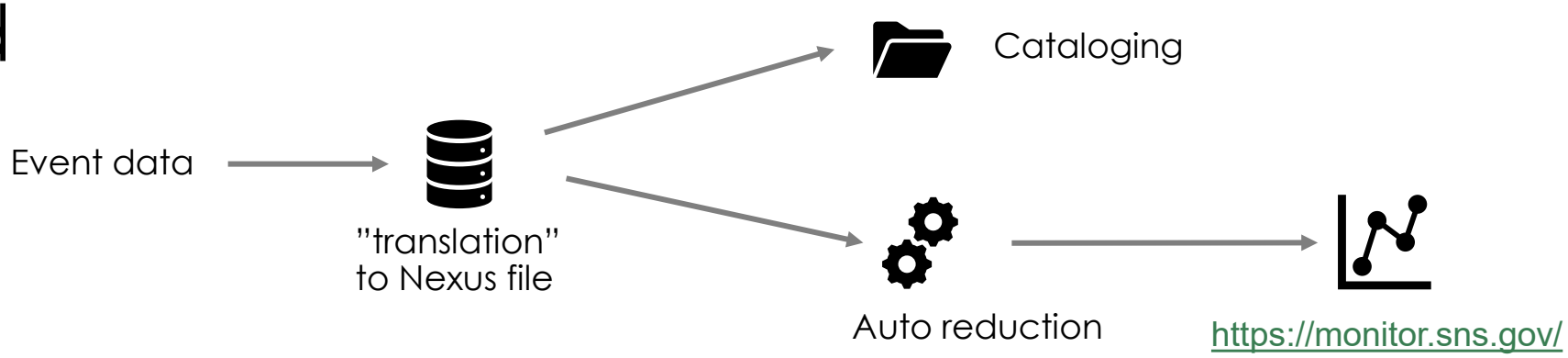
Marie Backman

June 5, 2024

ORNL is managed by UT-Battelle LLC for the US Department of Energy

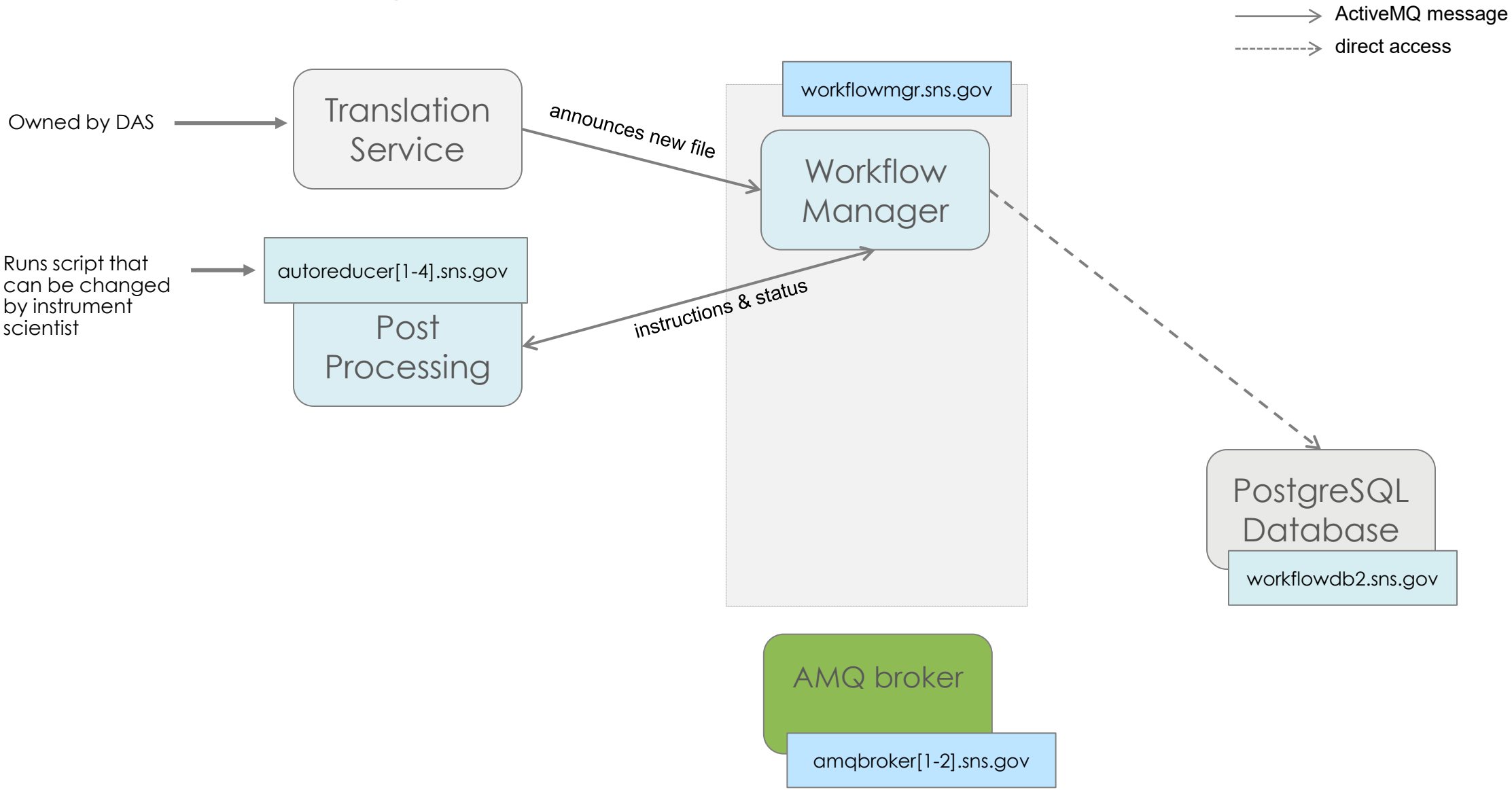
U.S. DEPARTMENT OF ENERGY

Background

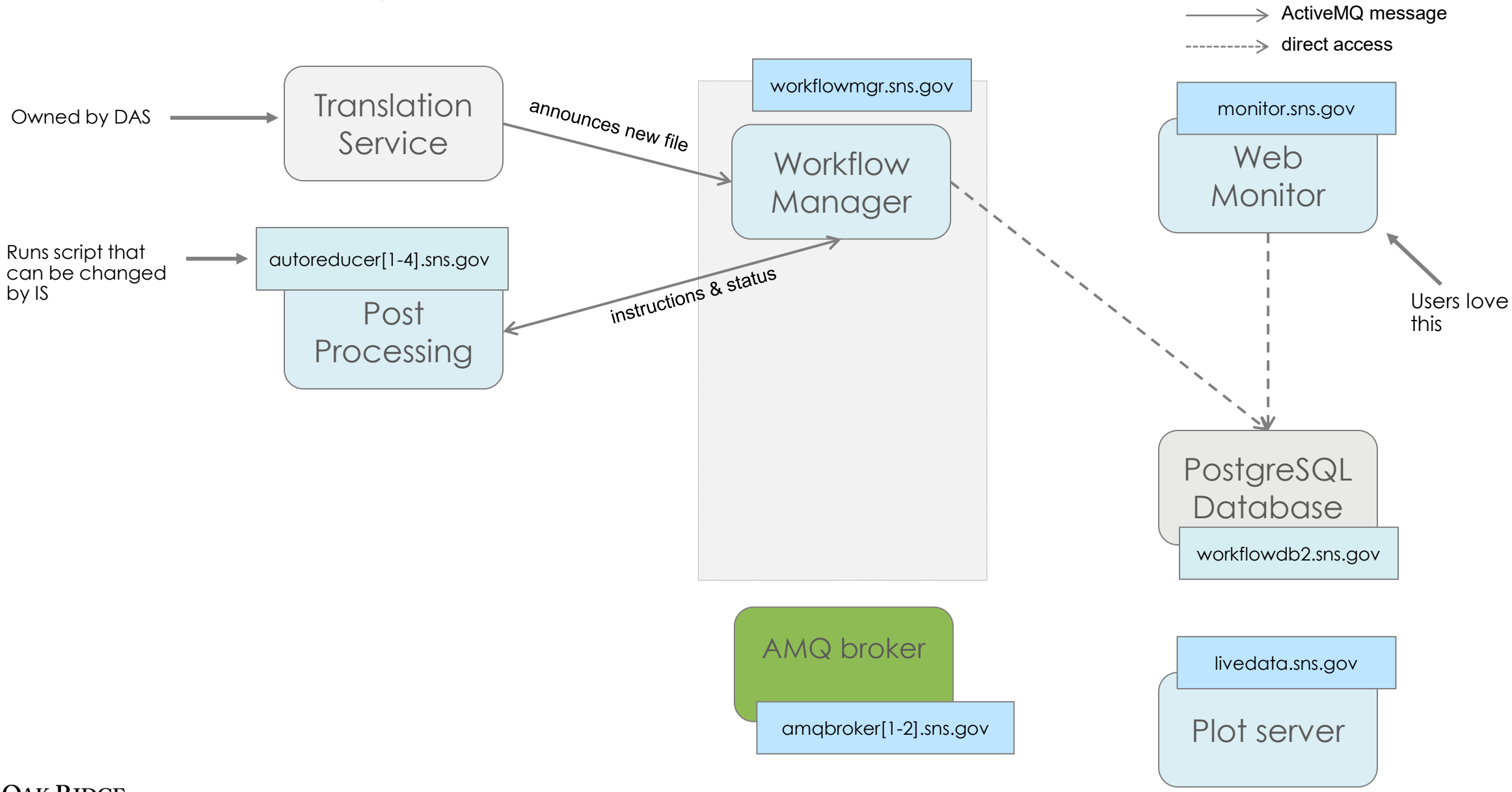


- Originally, "translation" to Nexus files was done after a run completed and took a lot of time
- Around 2012, the ADARA project changed everything to event streaming
  - "Translation" now starts at the beginning of the run
- As part of that effort, the automated reduction workflow was created
  - The web monitor was initially a diagnostics tool for developers but quickly became popular

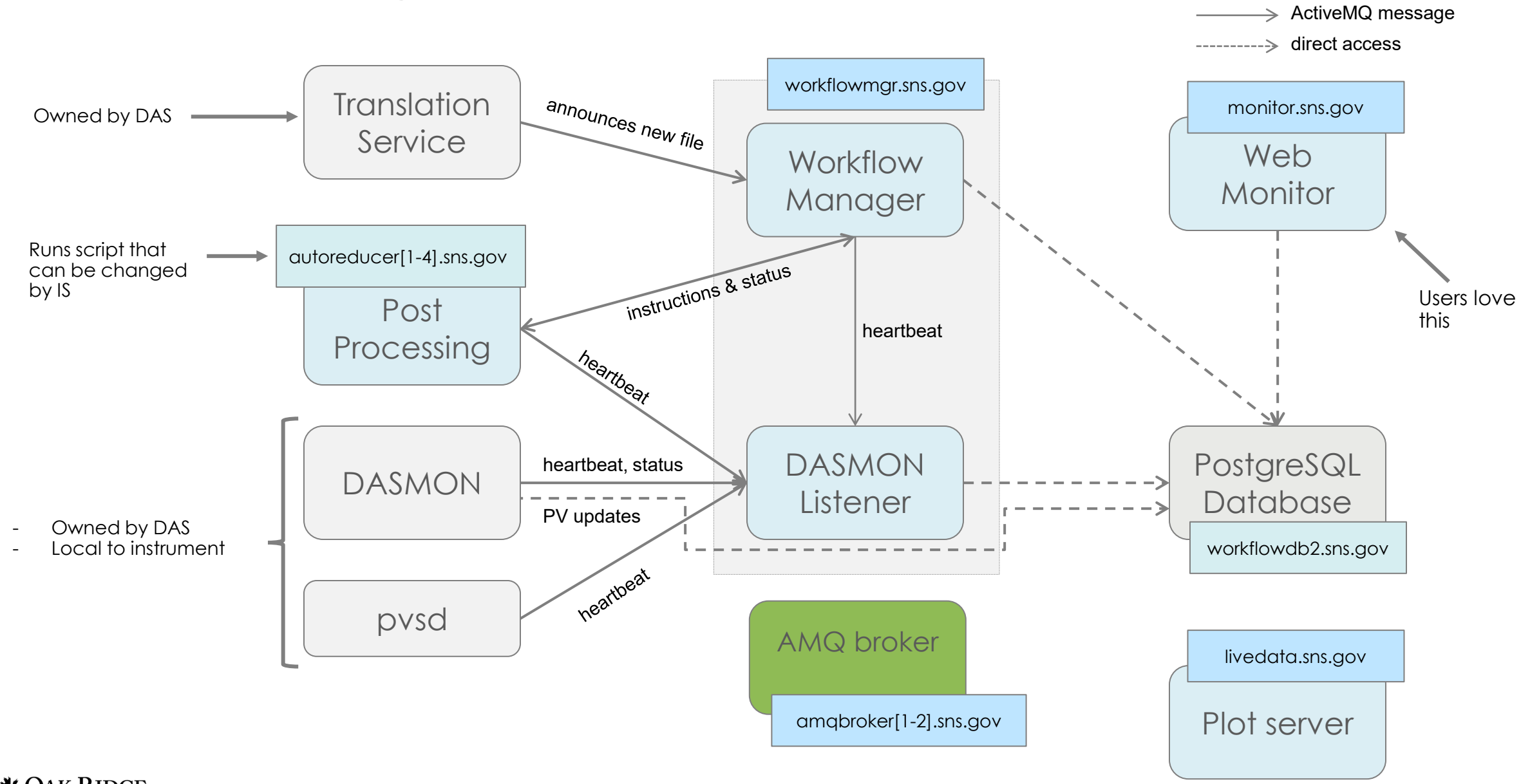
Post-Processing Architecture



Post-Processing Architecture

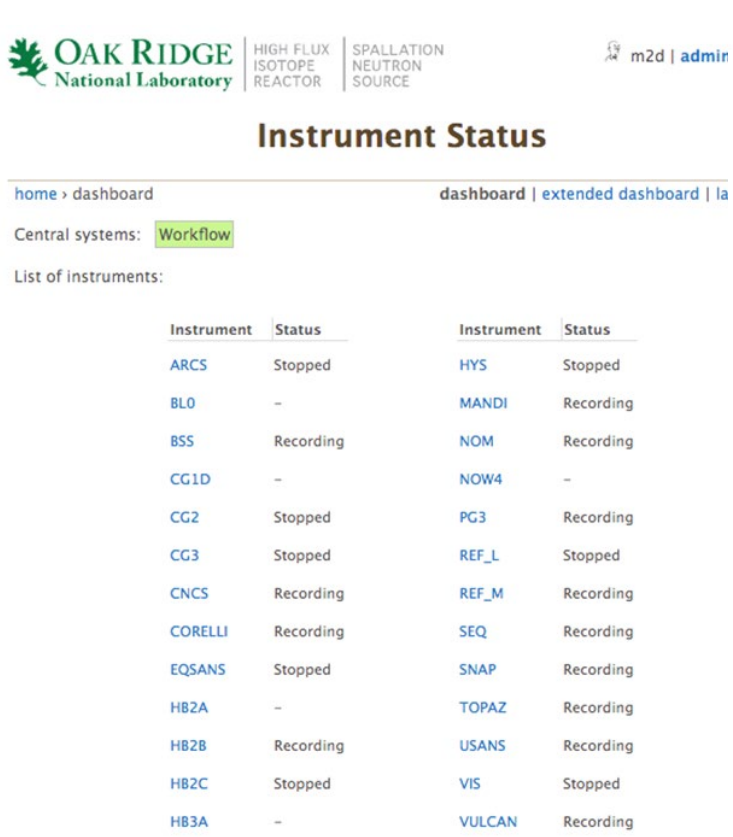


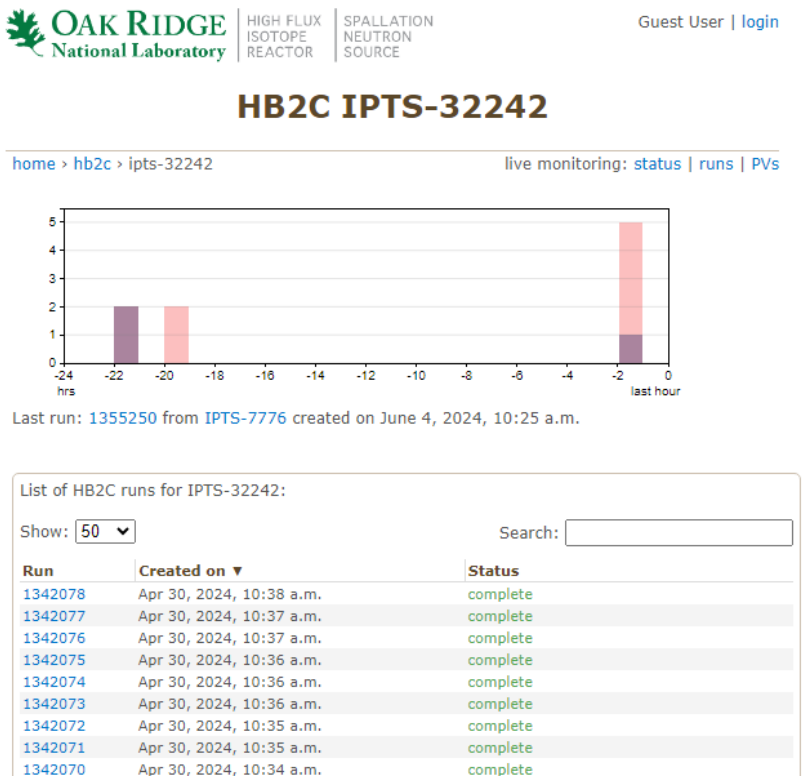
Post-Processing Architecture

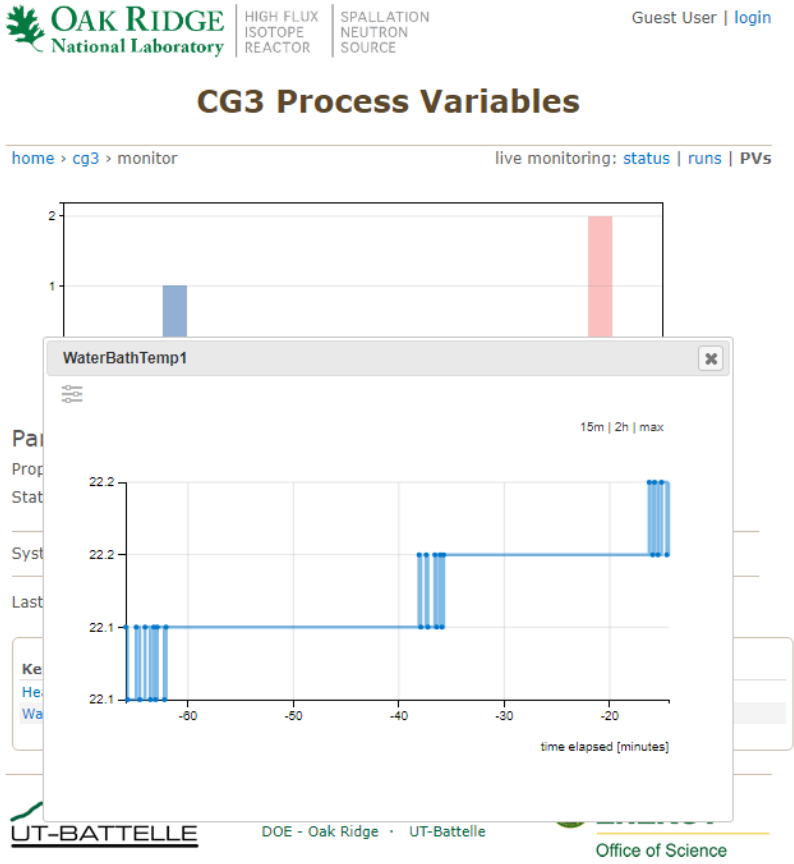


Web monitor – monitor.sns.gov

- User-facing part of WebMon



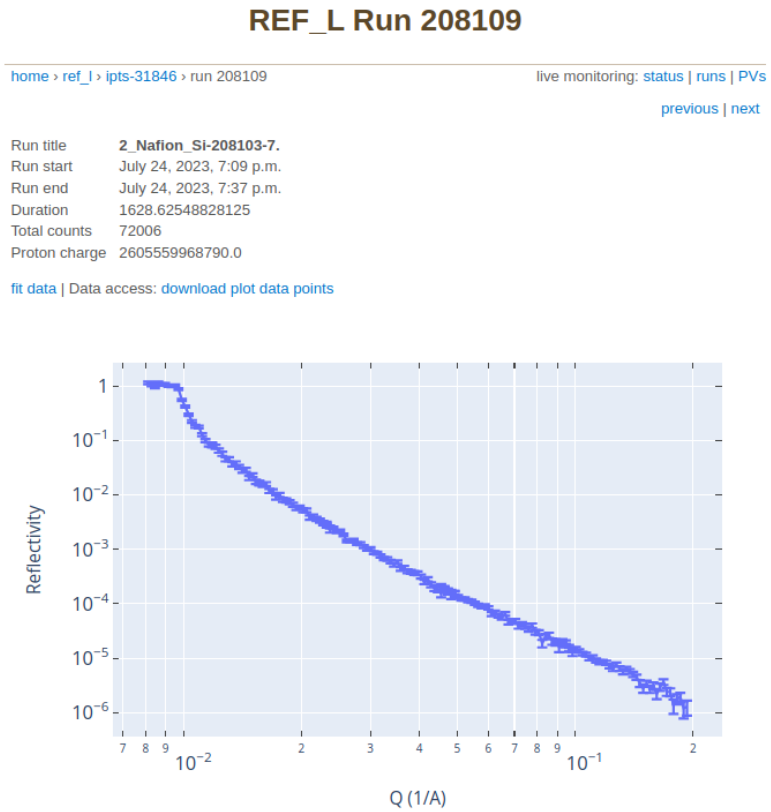




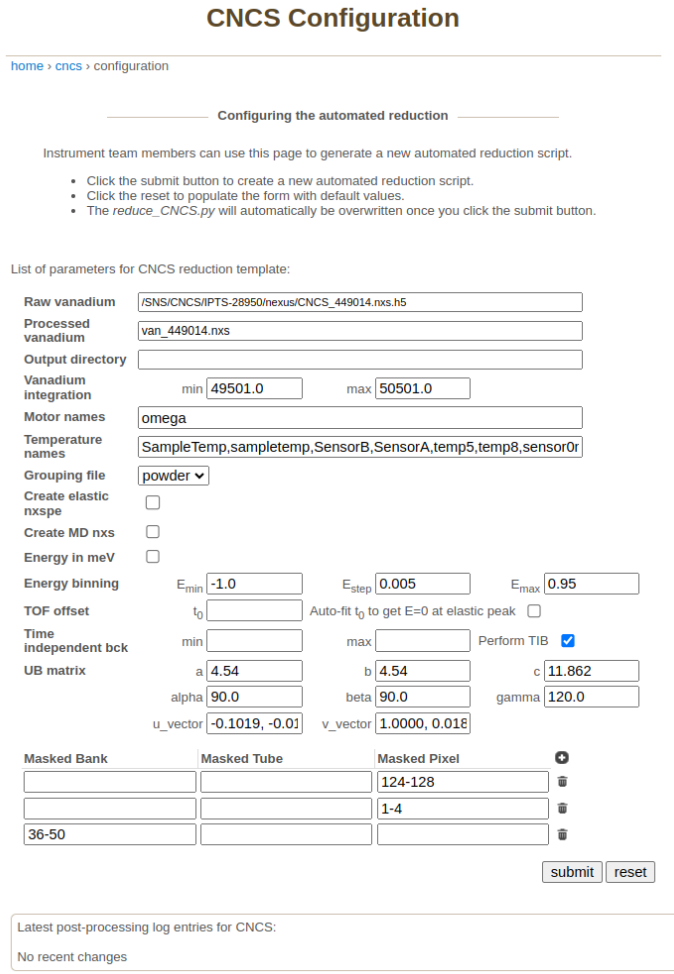
OAK RIDGE National Laboratory

Web monitor – monitor.sns.gov

Rerun cataloging and reduction



Configure autoreduction (IS only)



OAK RIDGE National Laboratory

Technologies

- ActiveMQ Classic message broker
- Python clients using the STOMP messaging protocol
  - Workflow manager
  - Dasmon listener
  - Autoreducer
- 2 Django apps
  - Web monitor (monitor.sns.gov)
  - Live Data Server (livedata.sns.gov)
- 2 PostgreSQL databases (workflowdb2.sns.gov and livedata.sns.gov)

Autoreducers / post-processing agent

- Repo: https://github.com/neutrons/post_processing_agent
- Postprocessing tasks it handles:
  - Cataloging of raw data in ONCat (https://oncat.ornl.gov/)
    - Users can download their data from ONCat
  - **Autoreduction**
    - **Runs instrument-specific data reduction script**
    - **Publishes plots to Live Data Server**
  - Cataloging of reduced data in ONCat
  - Adjusting parameters in the autoreduction scripts (available to instrument scientists)
- 4 instances running on autoreducer[1-4].sns.gov

Testing strategy

- Unit tests
- System tests part of CI and for local development
  - System of 11 docker containers for message broker, clients, DB, web server etc.
  - https://github.com/neutrons/data_workflow/blob/next/docker-compose.yml
- Developer test environment in CADES – webmon-test.ornl.gov
  - https://code.ornl.gov/sns-hfir-scse/infrastructure/neutrons-test-environment
- https://data-workflow.readthedocs.io/en/latest/developer/index.html

RHEL 7 – RHEL 9 upgrade

- The SNS/HFIR servers were running Red Hat Enterprise Linux 7 (RHEL 7) (EOL: 2024-06-30)
- Neutrons Data Project responsibility: port our (mainly data reduction) software from RHEL 7 to RHEL 9
- Python 2 is not distributed with RHEL 9
- Why not containerize everything?
  - Growing tech debt
  - Cyber security!

Porting Post-Processing Agent

- Removed obsolete code
- Added integration tests with containerized ActiveMQ broker and agent
- Migrated from Python 2.7 to Python 3.9 (RHEL 9 default Python version)
- Still distributed as RPM (RPM Package Manager)

Porting Live Data Server

- Serves plots to monitor.sns.gov
- Production instance used Apache HTTP Server, but initial work had been done in the test environment to move to Nginx and containerize Live Data Server
- Migrated from Python 2.7 + Django 1.9 to Python 3.6 + Django 2.0
  - Story in the backlog to move to latest Python, Django and PostgreSQL
- Tested new RHEL 9 production with WebMon test environment
- Purged old data to shrink PostgreSQL DB from 1.4 TB to 550 GB

Migration from ActiveMQ Classic to ActiveMQ Artemis

- Main difference was message addressing and routing
- Classic has queues (point-to-point) and topics (publish-subscribe)
- Artemis only has queues and uses addresses to achieve different routing mechanisms.
  - Multicast routing: all subscribers to the address get their own queue
  - Anycast routing: only one queue for the address that all consumers subscribe to
- https://activemq.apache.org/components/artemis/migration-documentation/

Thank you!