

An Event-Driven Workflow and Monitoring System for the Atmospheric Radiation Measurement (ARM) Program

ELVIS OFFOR, CARINA LANSING, EROL CROMWELL, SHERMAN BEUS, BRIAN ERMOLD, KRISTA GAUSTAD

PNNL

DOE Data Days 2024

Atmospheric Radiation Measurement (ARM) Program

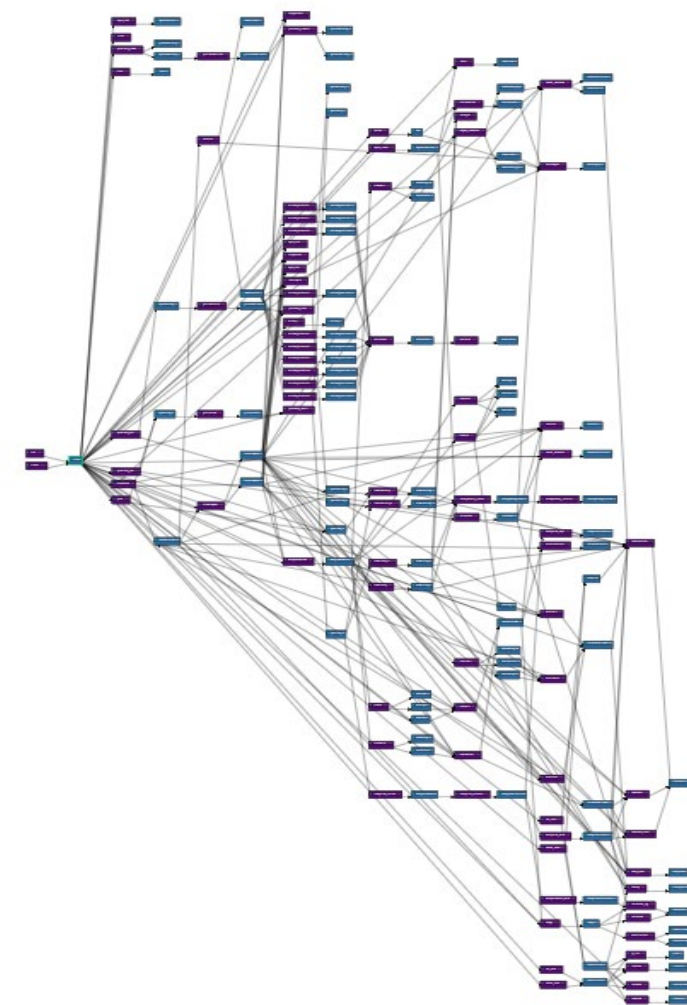
- ▶ The ARM Program is one of the largest and most influential data-collecting efforts in climate research.
- ▶ ARM operates a global network of atmospheric observatories in climatically significant locations.
- ▶ Data are collected from hundreds of instruments at locations around the world.
- ▶ The program is currently generating approximately **50 terabytes of data per month**.
- ▶ ARM manages a **complex hierarchy of processes** that provide the scientific community with quality-assured data products in near real time.



ARM Data Processing Challenges

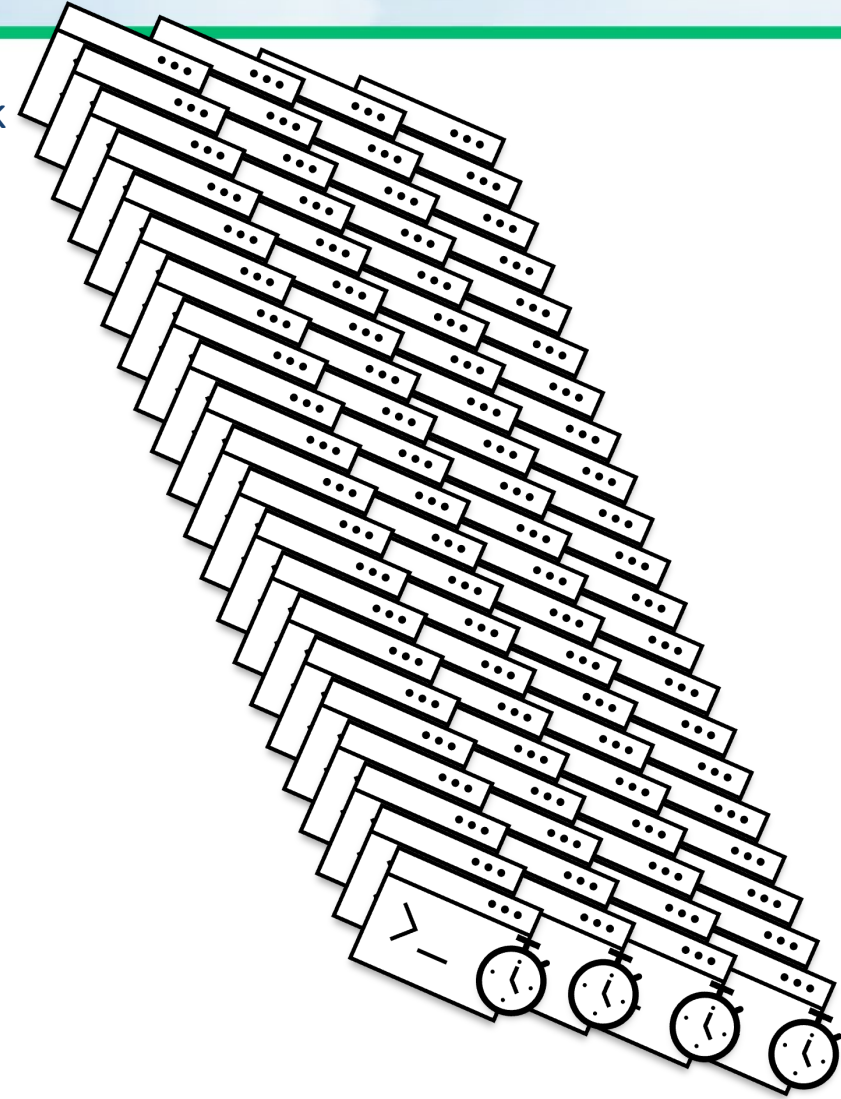
- ▶ **Data Variety and Complexity.** ARM data includes different types such as radiometric measurements, meteorological observations, aerosol concentrations, and spectral information.
- ▶ **Multi-Scale Data.** The data also vary across different spatial and temporal scales, requiring alignment to study complex climate processes.
- ▶ **Complex Data Analysis.** Extracting meaningful insights from ARM data requires sophisticated statistical and computational models that can handle high levels of variability and uncertainty.
- ▶ **Real-Time Data Processing Requirements.** For many ARM applications, data needs to be processed in near real-time.
- ▶ **Data Transmission and Integrity Issues.** A variety of issues can affect the accuracy and continuity from instruments, including network disruptions and equipment failures.

Data product graph for a single ARM instrument.



Limitations of ARM's Legacy Data-Processing Framework

- ▶ ARM has been operating for decades, and its legacy data-processing framework involves a massive number of independently executed cron jobs.
- ▶ Cron-based system creates several inherent challenges to data operations:
 - Downed pipelines are difficult to detect.
 - Impossible to pause processing for downstream products.
 - Processes often run even if no data are available.
 - Difficult to troubleshoot and correct problems.
 - Many manual steps involved in processing and error correction.
 - Many processes have one-off special logic.
 - Time lag between when data are produced and when they are released to the public.



ARMFlow: ARM Event-driven Process Control And Monitoring

- **Event-driven processing** – faster data delivery to users
- **Expanded process monitoring** – supports better troubleshooting and reporting
- **Visual dashboard with all info at your fingertips** – easier to detect and troubleshoot errors
- **Automates previously manual steps** – faster end-to-end and less error prone
- **Process-specific configuration** – full transparency of processing rules
- **Integration with other ARM systems** – reduces manual steps/learning curve and improves task tracking & monitoring
- **Intelligent processing** – reduces unnecessary runs and false positive errors; provides self-healing error correction when possible



Higher data throughput



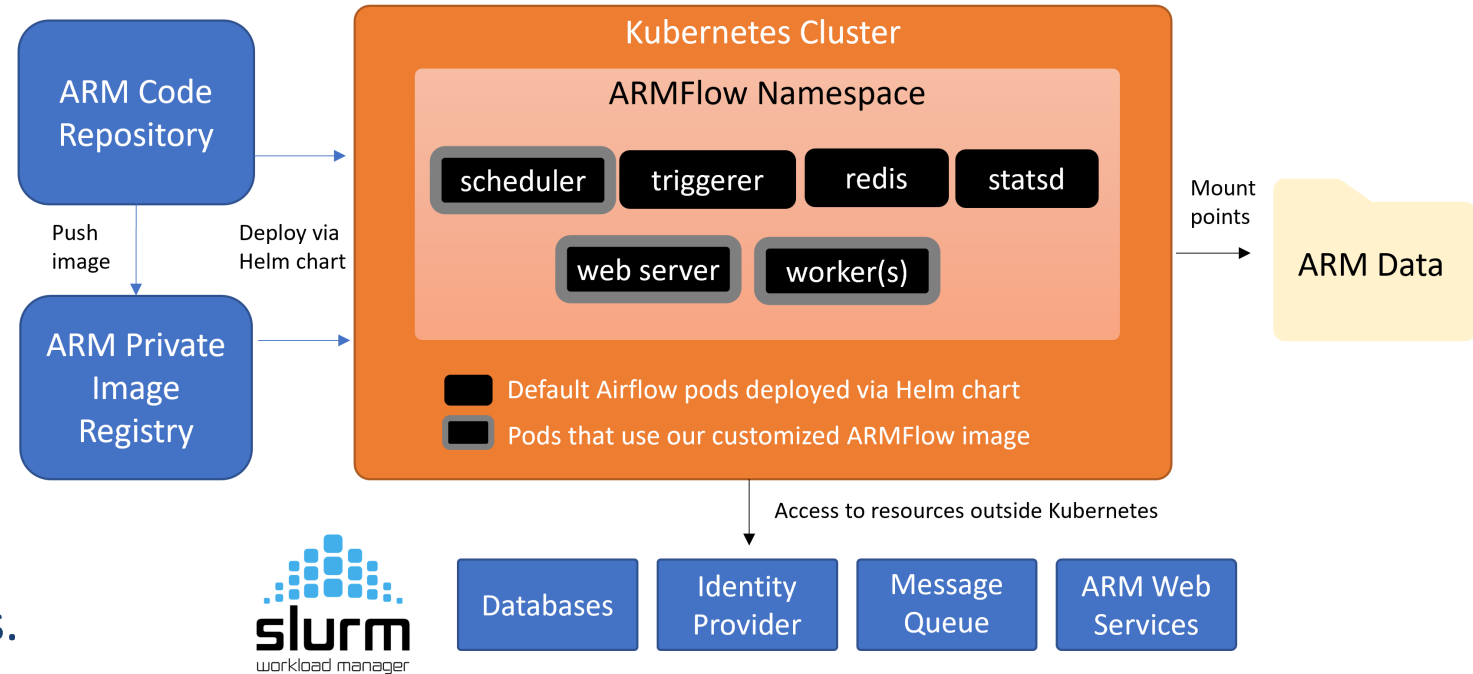
Higher quality data produced



Decreased operating costs

ARMFlow Architecture

- ▶ Based upon the Apache Airflow workflow framework.
- ▶ Running on a Kubernetes cluster at the ARM Data Center.
- ▶ Deployed to Kubernetes via Airflow's Helm chart.
- ▶ ARMFlow includes:
 - Customized Airflow Docker image
 - A suite of custom workflows
 - A custom user dashboard
- ▶ Actual processing runs on Slurm cluster.
- ▶ Integrated with ARM services via events.

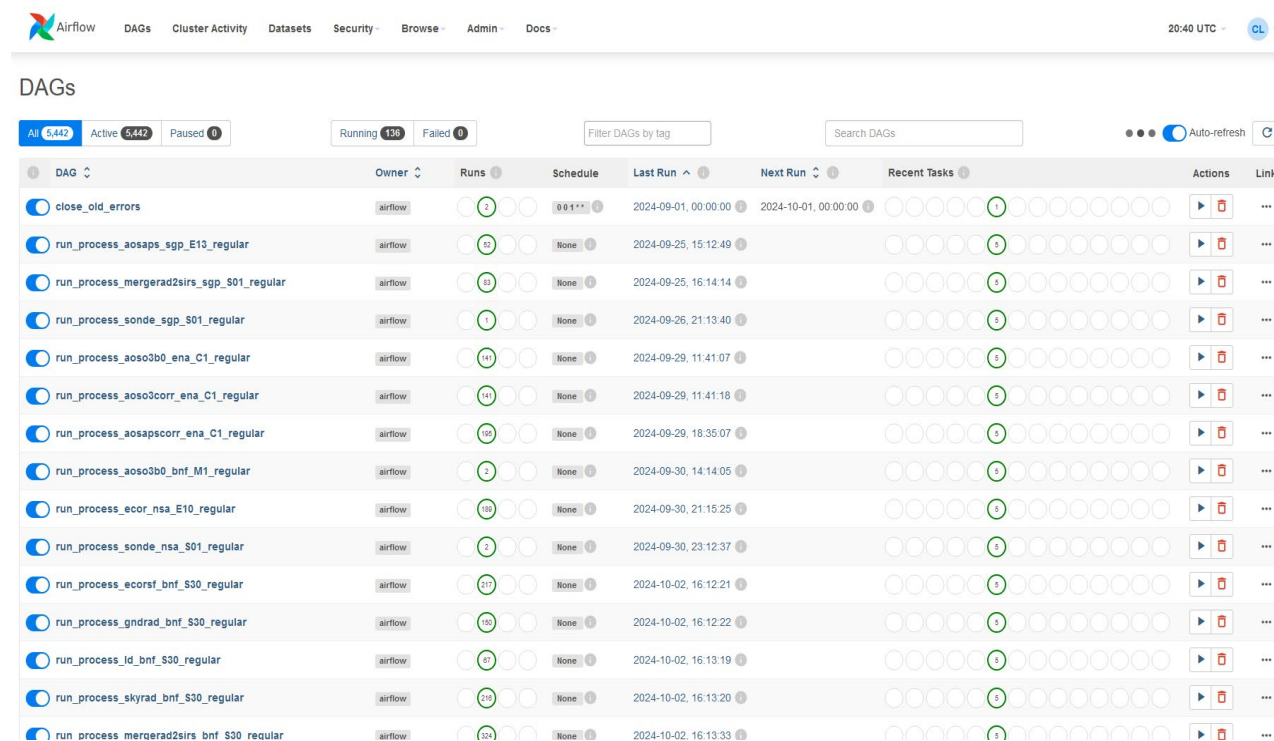


ARMFlow Needed a Custom Dashboard

▶ Close to 8000 workflows running in Airflow!

▶ The Airflow dashboard is not helpful:

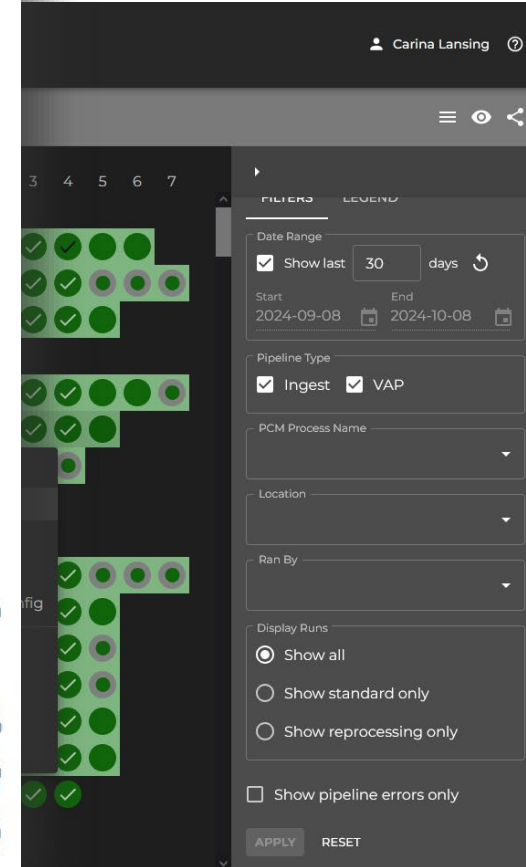
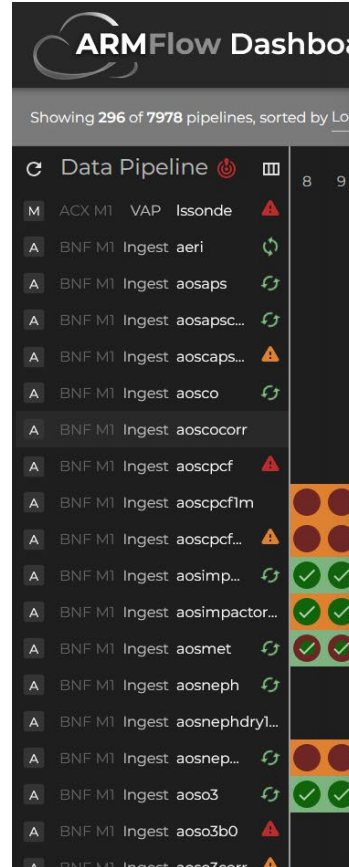
- Shows all workflows individually with no roll up.
- Can't group workflows by instrument or location.
- Must page through workflows one at a time – no filters.
- Can't filter out workflows that are not running.
- Headers are not fixed.
- Can't tell if an error shown is for a current or past error.
- Airflow only reports workflow crashes, not ARM data processing errors, so many error conditions would be missed.
- No access to ARM-specific information or logs.



DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
close_old_errors	airflow	3	00 1 * * *	2024-09-01, 00:00:00	2024-10-01, 00:00:00	3	▶ 🗑️	...
run_process_aosaps_sgp_E10_regular	airflow	51	None	2024-09-25, 15:12:49		5	▶ 🗑️	...
run_process_mergerad2sirs_sgp_S01_regular	airflow	53	None	2024-09-25, 16:14:14		5	▶ 🗑️	...
run_process_sonde_sgp_S01_regular	airflow	1	None	2024-09-26, 21:13:40		5	▶ 🗑️	...
run_process_aoso3b0_ena_C1_regular	airflow	14	None	2024-09-29, 11:41:07		5	▶ 🗑️	...
run_process_aoso3corr_ena_C1_regular	airflow	14	None	2024-09-29, 11:41:18		5	▶ 🗑️	...
run_process_aosapscorr_ena_C1_regular	airflow	10	None	2024-09-29, 18:35:07		5	▶ 🗑️	...
run_process_aoso3b0_bnf_M1_regular	airflow	3	None	2024-09-30, 14:14:05		5	▶ 🗑️	...
run_process_ecor_nsa_E10_regular	airflow	10	None	2024-09-30, 21:15:25		5	▶ 🗑️	...
run_process_sonde_nsa_S01_regular	airflow	3	None	2024-09-30, 23:12:37		5	▶ 🗑️	...
run_process_ecorsf_bnf_S30_regular	airflow	21	None	2024-10-02, 16:12:21		5	▶ 🗑️	...
run_process_gndrad_bnf_S30_regular	airflow	10	None	2024-10-02, 16:12:22		5	▶ 🗑️	...
run_process_id_bnf_S30_regular	airflow	5	None	2024-10-02, 16:13:19		5	▶ 🗑️	...
run_process_skyrad_bnf_S30_regular	airflow	21	None	2024-10-02, 16:13:20		5	▶ 🗑️	...
run_process_mergerad2sirs_bnf_S30_regular	airflow	14	None	2024-10-02, 16:13:33		5	▶ 🗑️	...

ARMFlow's Custom Dashboard

- ▶ Workflows grouped by instrument name/location (“pipeline”)
- ▶ UI rolls up daily processing and data states for a pipeline.
- ▶ Pipelines that are currently stopped visible via red alert triangle.
- ▶ Errors report both processing and ARM data violations.
- ▶ Advanced filtering capabilities:
 - Hide pipelines that are not running
 - Filter by instrument name, location, or user
 - Select date range
- ▶ Details and all log files accessible from UI



A key goal for ARMFlow's UI is to allow users to spot and fix errors quickly. If errors are fixed quickly, there are few to no long-term consequences!

UI Highlight: Running Processes

- ▶ ARMFlow validates parameters and workflow configuration before running and won't allow if conditions are not valid.

Run Ingest/VAP

PCM Process Name
aopsapavg

Location
ENA C1

Reprocessing

Start
2024-08-01

End
2024-08-02

Optional args
Custom command line arguments

Form errors detected. Please check and revise the respective field(s):

- Overlapping dates detected - this run requires reprocessing.

CANCEL RUN

Run VAP aopsapavg at ENA C1

Based upon available data and processing rules, ARMFlow will try to run the following dates:

Start Date: Not available *

End Date: Not available *

* Not enough input data are available to run.

CANCEL RUN

- ▶ User can specify dates. If not specified, ARMFlow will determine if it can run based upon available input data.

UI Highlight: Error Reporting

Clicking on error icon brings up details

The screenshot shows the ARMFlow Dashboard interface. At the top, it says "ARMFlow Dashboard" and "Showing 9 of 7978 pipelines, sorted by location". Below this is a calendar view for October 2024, with a 370% zoom level. A pipeline named "Data Pipeline" is selected, and a red error icon is visible. A modal dialog box titled "Processing Errors: mfrsr7nch @ BNF S20" is open, displaying a table of error details and a list of actions.

Process Type	Processing Id	Level	Error Codes	Message	Latest Occurrence	State	Assignee	Comments	Actions
Ingest	regular_mfrsr7nch_bnf_S20	Error	adi.processing.other	Could not find re... View All	2024-10-07 21:20:45	New			View INC View Process Log View Slurm Log View Slurm Batch Script View ARMFlow Process Config View Error History Resolve Error

```
Could not find required configuration files:  
ERROR: Could not find required configuration files:  
-> search path: /data/armflow/dev/conf/bnf/bnfmfrsr7nchS20  
-> search pattern: [0-9]{6}\.dat$
```

The error dialog shows root cause

And also provides links to all relevant log files

UI Highlight: Process Details

A	BNF S20 Ingest irt	
A	BNF S20 Ingest mergerad2sirs	
A	BNF S20 Ingest met	



- Daily cells show state.
- Background is roll up of process state. Circle shows the roll up of data state.
- Right clicking on a day brings up detailed information.

Info

- View Ingest log
- View Collections log
- View Bundle log
- Pipeline Process States
- Pipeline Error History
- View ARMFlow Process Config

- Run Bundler
- Run Ingest
- Prune data
- Release data



- Provides detailed information for all processes that ran.
- Provides access to all relevant logs.
- Shows which files **failed** validation checks and why.

Pipeline Information: irt @ BNF S20 - 2024-09-25

^ Ingest

Process

Status	Processing ID	Run Directory	Ran On	Ran By	Logs	
✓	SUCCESS	regular_irt_bnf_S20	N/A	2024-09-25T23:17:11+00:00	data_delivery	View Process Log View ARMFlow Process Config

Outputs

Files or Datastream	Validation Errors	Archive Status	Release By	Release On	Comment
bnfirt200msS20.a1.20240925.000000.cdf	MISSING_RECORDS	Archived	airflow	20240926.011836	
1334 records written < 1411 expected threshold.					
bnfirtS20.b1.20240925.000000.cdf	MISSING_RECORDS	Archived	airflow	20240926.011836	

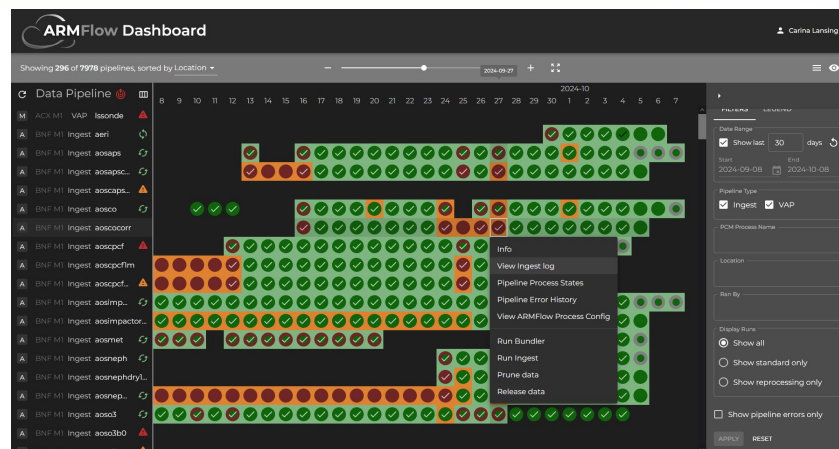
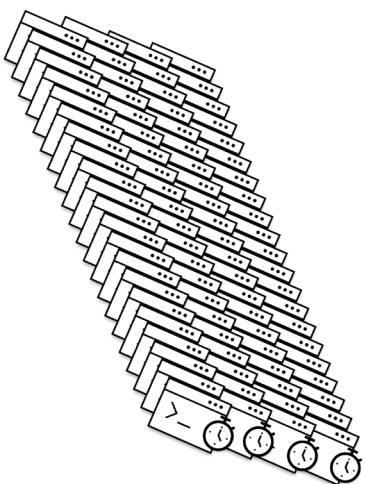
^ Bundler

Process

Status	Processing ID	Run Directory	Ran On	Ran By	Logs	
✓	SUCCESS	regular_irt_bnf_S20	N/A	2024-09-27T00:17:22.304565+00:00	data_delivery	View Bundler Log View Comlog View ARMFlow Process Config

[View Comlog](#) CLOSE

ARMFlow: Where Are We Now



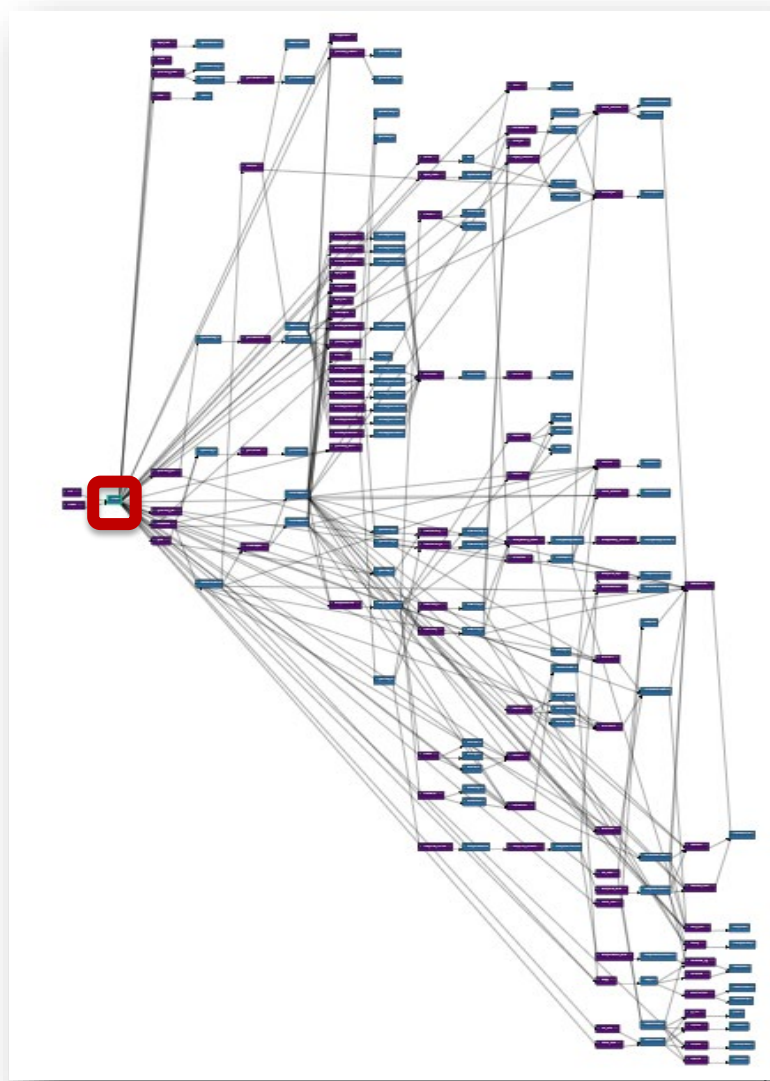
ARMFlow has undergone one full year of beta testing

- ▶ Production deployment is live; gradually rolling out processing pipelines.
- ▶ Beta server mimics production processing so we can thoroughly test data pipelines before releasing them to production.
 - This is critical to catch special edge cases!
- ▶ Rollout transitions seamlessly from existing cron processing to ARMFlow.



ARMFlow Next Steps

- ▶ **Interactive reprocessing:**
 - Visualize data changes
 - Control downstream processing as needed
- ▶ **Interactive corrective actions:**
 - Automate fixes to common processing issues (such as data coming in out of order)
- ▶ **Longer term - incorporate AI to assist operators:**
 - Chat bots to quickly answer questions
 - Instrument failure prediction/detection



Acknowledgements

- ▶ ARM is sponsored by the U.S. Department of Energy's Office of Science under the Biological and Environmental Research (BER) program.
- ▶ For more information on ARM, see our web page at arm.gov.
- ▶ For questions on ARMFlow, feel free to contact me at elvis.offor@pnnl.gov

Thank You!



▶ Questions?