Science and Technology Facilities Council

Scientific Computing

Apache Airflow and its use in PSDI *

Alexander Belozerov, Vasily Bunakov, Amali Pawula Hewage, Stuart Meacham, Tom Underwood

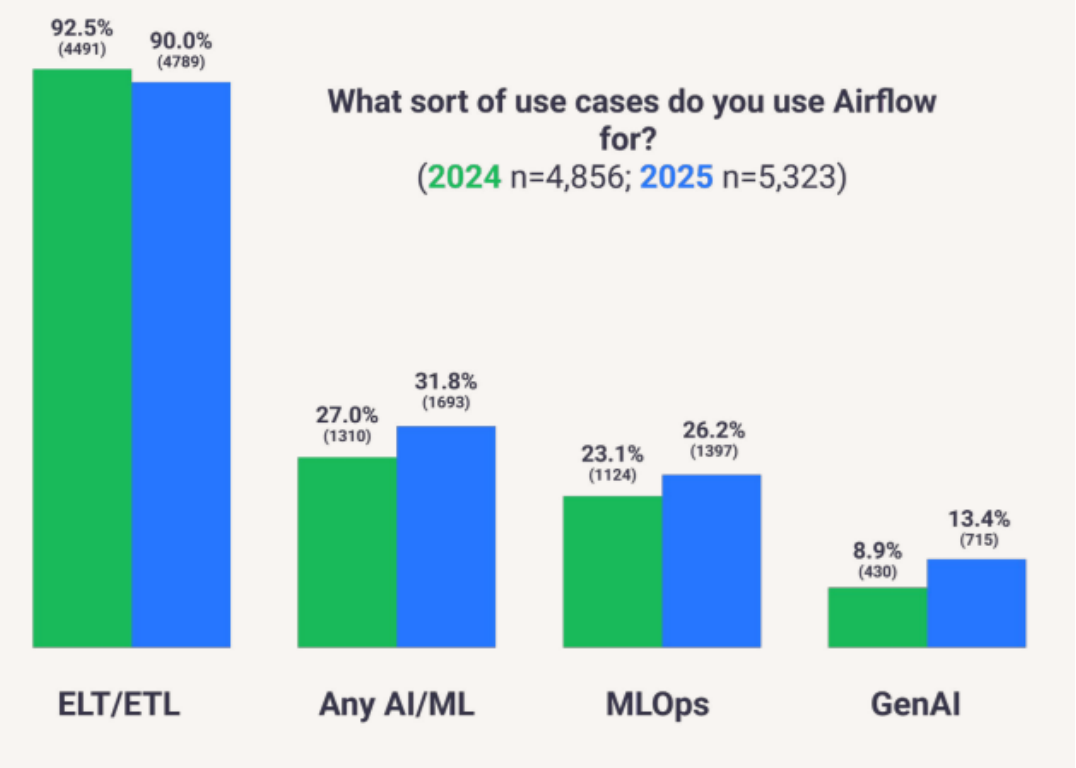
In Data Technology Group extended meeting with RAL Space and ISIS, 1 April 2026

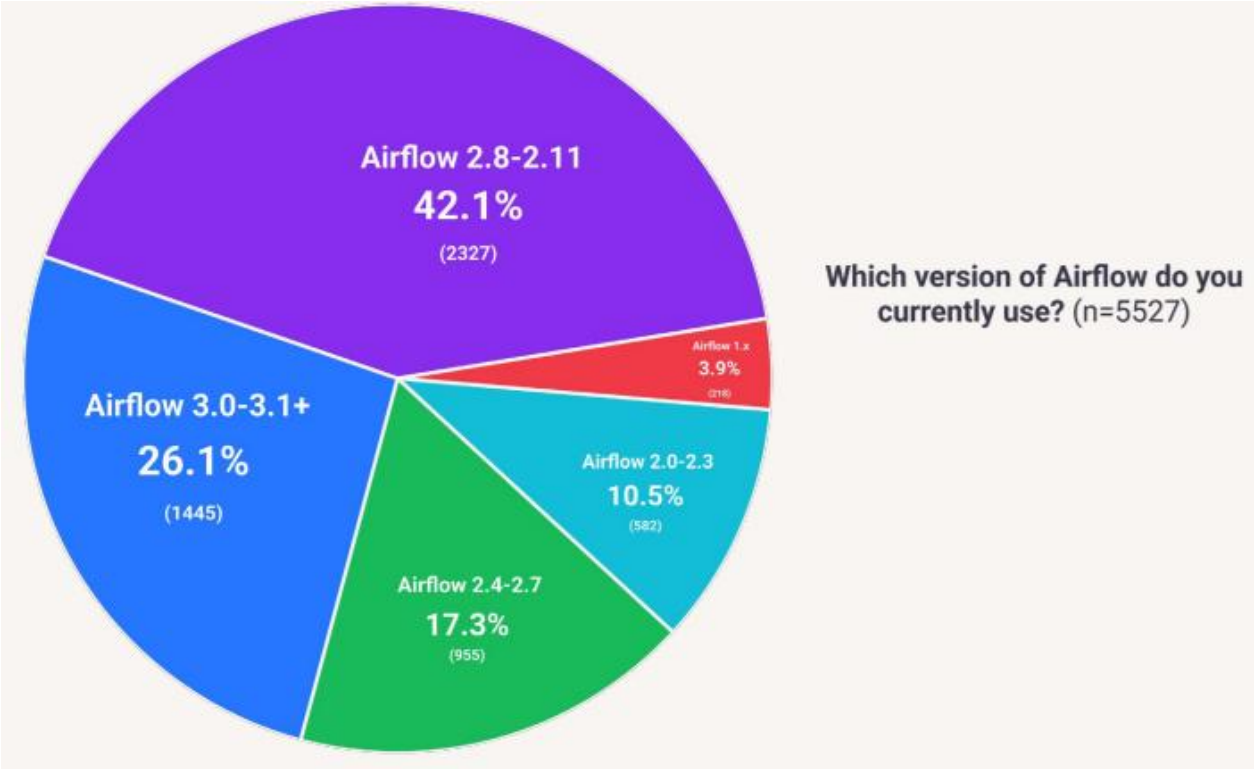
* STFC effort in the current phase of PSDI (Physical Sciences Data Infrastructure) is funded by EPSRC grant EP/X032663/1

Importance of data pipelines

- Current technology trends make volatile the boundaries between “infrastructure”, “system”, “platform” or “application” – which are all just “solutions” to data-related problems
- In modern data engineering, data pipelines are first-class citizens as important as “systems”, “platforms” or “applications”
- How you make data pipelines, how you monitor them and how you evolve them is critical for data-centric services
- Apache Airflow is just one of possible responses to this challenge – or a part of a bigger response that may be required depending on a use case
- Apache Airflow can be (and is) commonly used for ELT/ETL, but it is a general-purpose orchestration solution for all sorts of IT jobs

How Apache Airflow is used globally *





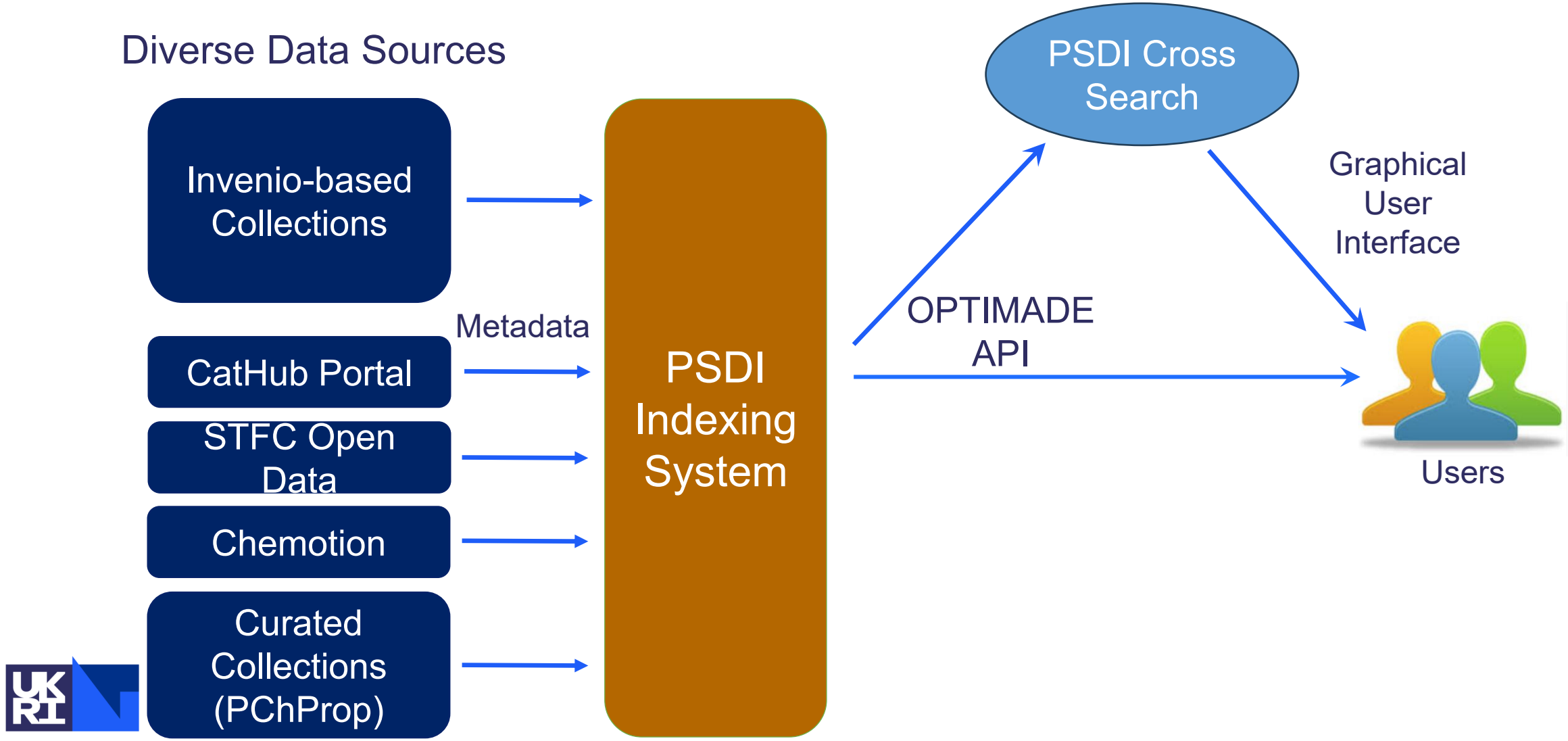
Science and Technology Facilities Council
Scientific Computing

* As per 2026 State of Apache Airflow report by Astronomer.io – the company contributing more than a half of Apache Airflow code

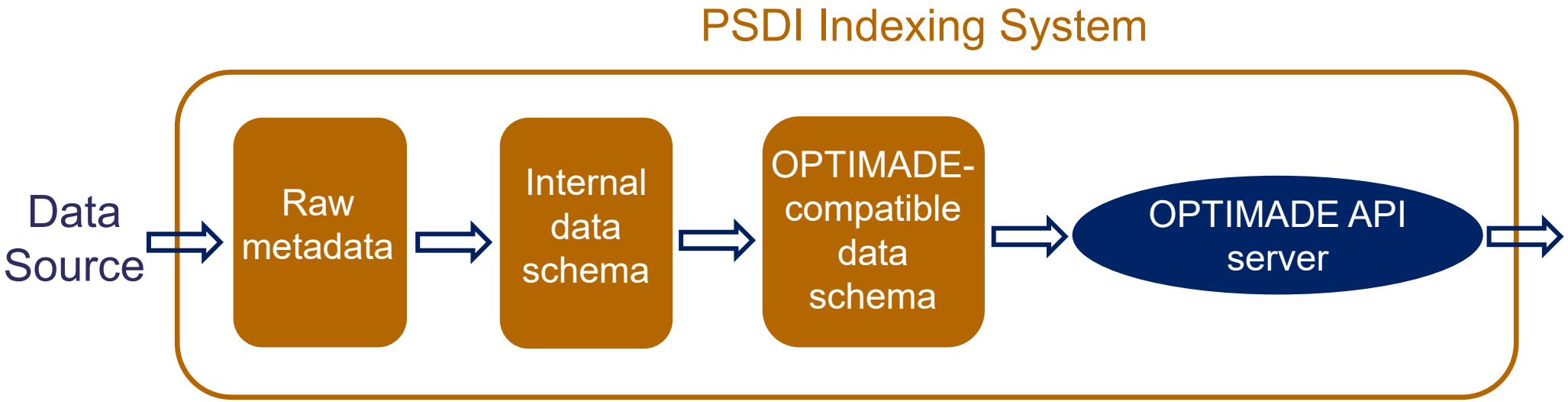
Indexing and application of Apache Airflow in it

Science and Technology Facilities Council
Scientific Computing

Indexing System: A Brief Overview



Indexing System: A Deep Dive



Technology stack:

Apache Airflow

MongoDB

HashiCorp Vault

kubernetes

+ automated CI/CD pipeline

OPTIMADE API

www.optimade.org

That is a standardized REST API designed for accessing and exchanging materials science data.

- Developed by the OPTIMADE Consortium in 2020.
- Adopted by PSDI to expose data and power the Cross Data Search.

25	29	59,521,800
PROVIDERS	DATABASES	STRUCTURES

Science and Technology Facilities Council

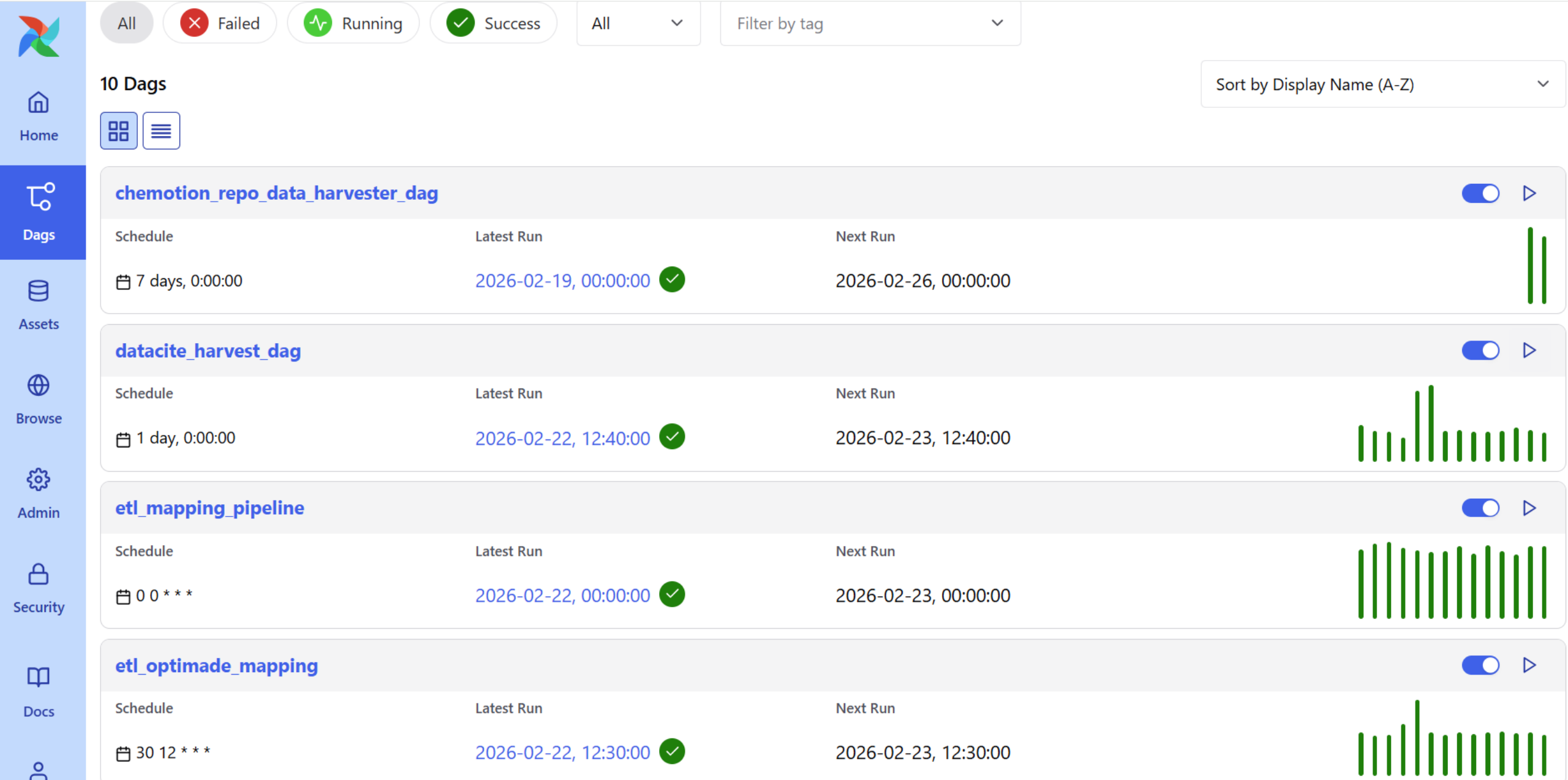
Scientific Computing

Where we use Airflow

1. Harvesting raw metadata and ingesting them into MongoDB
    - A dedicated harvester for each data source
    - Each data source has its own schedule for harvesting

2. Extract Transform Load (ETL) operations:
    - Cleaning data
    - Mapping onto the internal data schema
    - Mapping onto the OPTIMADE schema

Pipeline orchestration using Apache Airflow UI



A typical Airflow DAG for harvesting

Authenticate with HashiCorp Vault

Retrieve database credentials from Vault

Connect to the database and obtain the last harvest timestamp

Harvest metadata from the data source

Load harvested metadata into the database

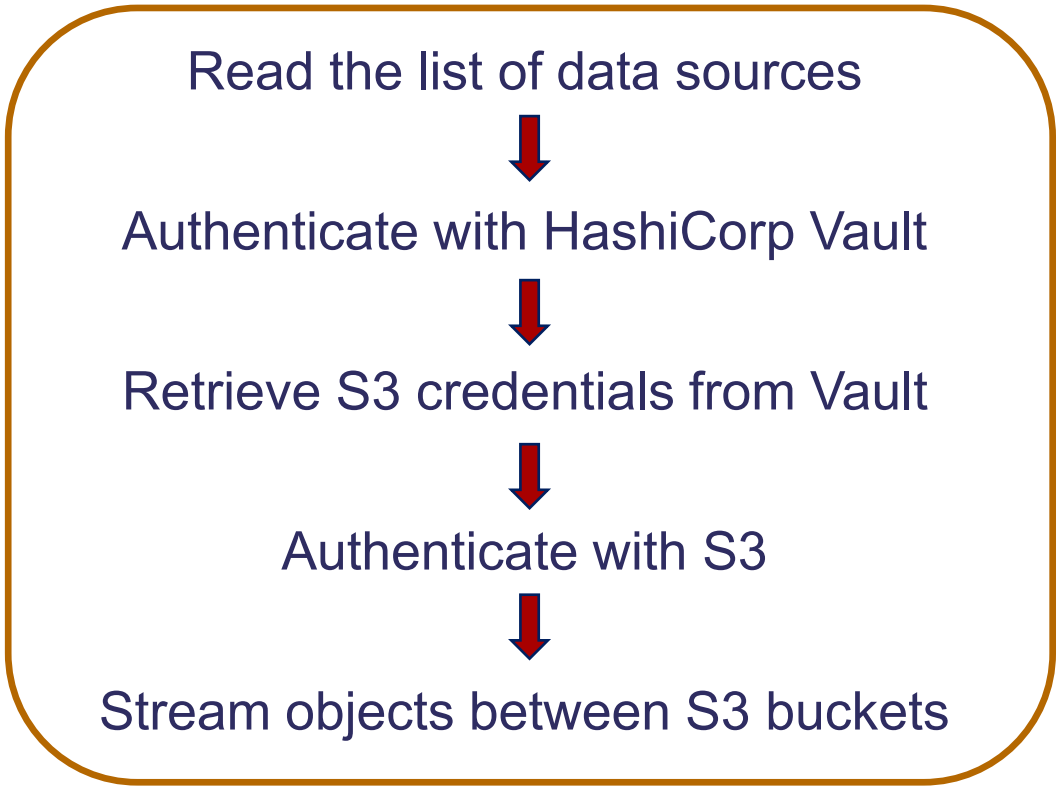
Store the timestamp in the database

First examples of Airflow pipelines for Lakehouse

Science and Technology Facilities Council
Scientific Computing

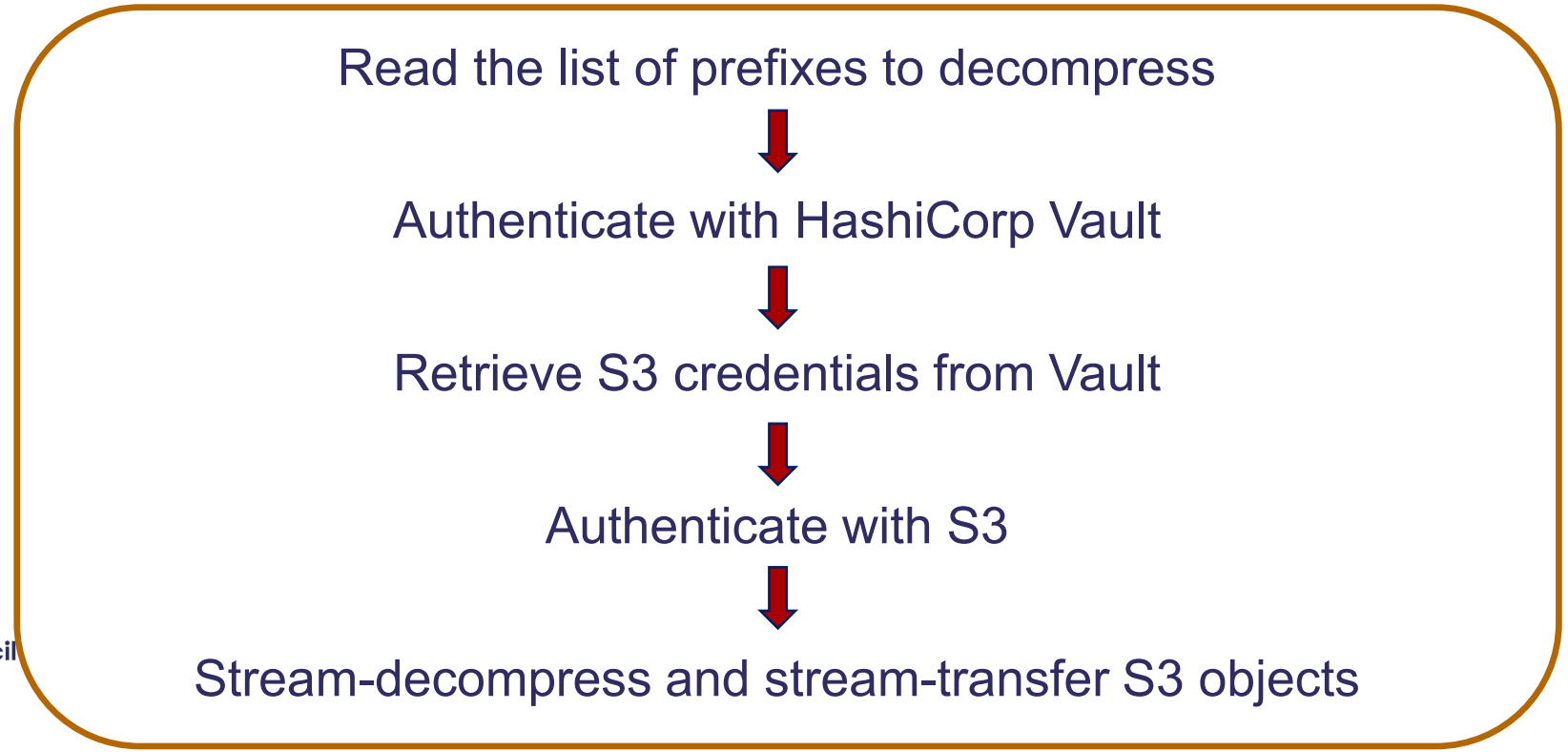
DAG 1: Ingesting Data from Amazon S3 into the Bronze Layer

- Designed for Materials Project data, but applicable to any data stored in S3.
- Extensible: accepts a list of data sources as input.
- Streams data in chunks to minimise memory overhead.
- Transfers only new objects (not present in the destination bucket).



DAG 2: Automated Gzip Decompression from STFC S3 Storage

- Currently used for the Materials Project data.
- Configurable via S3 prefix lists for flexible scope control.
- Streams data in chunks to minimise memory overhead.
- Outputs decompressed objects directly to a configurable destination.



Apache Airflow deployment for PSDI

Science and Technology Facilities Council
Scientific Computing

Airflow Architecture

- Enable independent developer workflows
- Automated builds and deployments
- Scalable
- In cluster testing
- Leverage on external resources/teams where possible

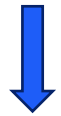
Airflow Technologies

- Airflow (v2 -> v3)
- GitHub Actions for scheduled pipeline and reusable workflows
- ARC runners for on-premises deployment
- CAPI/K8s as deployment environment
- Hashicorp Vault for secrets (pipeline and DAG)
- External PostgreSQL for Airflow metadata

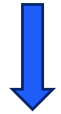
Airflow Deployment

Manual pipeline trigger for prod

Actions based Scheduled Pipeline



DAG aggregation from git sub-modules



Container build -> push to GHCR

Kustomized deployment in K8s