Distributed Data Management

Rucio for the CLAS12 Collaboration

Anil Panta

2026

Feature overview + JLab status

WHY THIS MATTERS

Data management is no longer optional

DOE Mandate

- Federally funded research must follow data management requirements covering sharing, preservation, and access.
- Goal: accelerate discovery and make data useful beyond the originating experiment.
- Applies directly to CLAS12 raw and derived data products.

energy.gov/datamanagement

Jefferson Lab's Response

- Lab-wide Data Management Requirements are being revised now.
- Better compliance will be matched with better tools — not just paperwork.
- Rucio is that tool: a production data management system, already running for Hall data at JLab.

THE STANDARD WE'RE BUILDING TOWARD

FAIR: the principles behind all data management Modernization

Findable

Every dataset gets a persistent, unique identifier and rich, searchable metadata.

Accessible

Data is retrievable via open, standardized protocols — with authentication where needed.

Interoperable

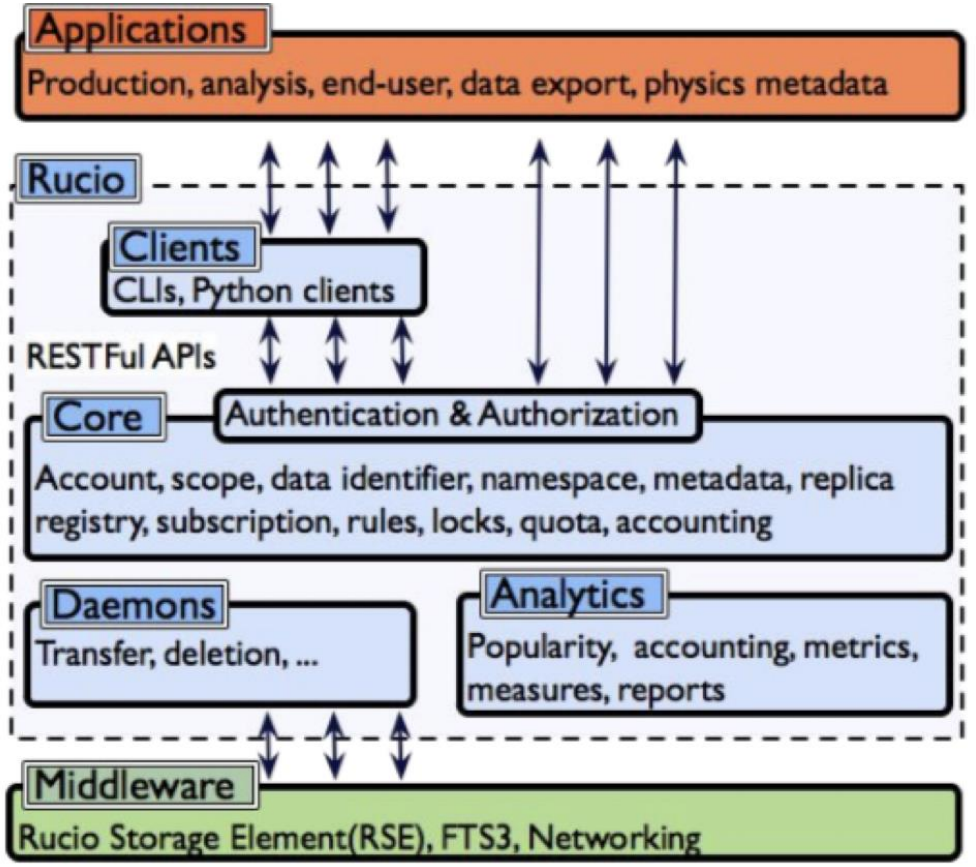
Metadata uses shared, formal vocabularies that other tools and sites can read.

Reusable

Rich provenance and clear usage terms mean data outlives the person who made it.

Rucio: a scientific data management system

- Free, open-source (Apache 2.0) — github.com/rucio
- Built and proven at CERN for LHC-scale data: over a decade of operational experience
- Single global namespace across facilities, regardless of administrative domain
- Handles the full lifecycle: creation, location, transfer, deletion, annotation, access



CORE CONCEPT

One identifier, no matter where the file lives

Every file or dataset gets a Data Identifier (DID), written as **scope:name**.

```
clas12:/RunGroupA/recon/run-018532/clas12_018532_recon.hipo
```

scope : name

Unique, every time

No collisions across run periods, halls, or processing passes.

Storage can change underneath it

A file can move from cache to disk to tape — the DID stays the same.

A stable anchor

Other systems (logs, conditions DBs, analysis configs) can reference the DID and stay correct.

CORE CONCEPT

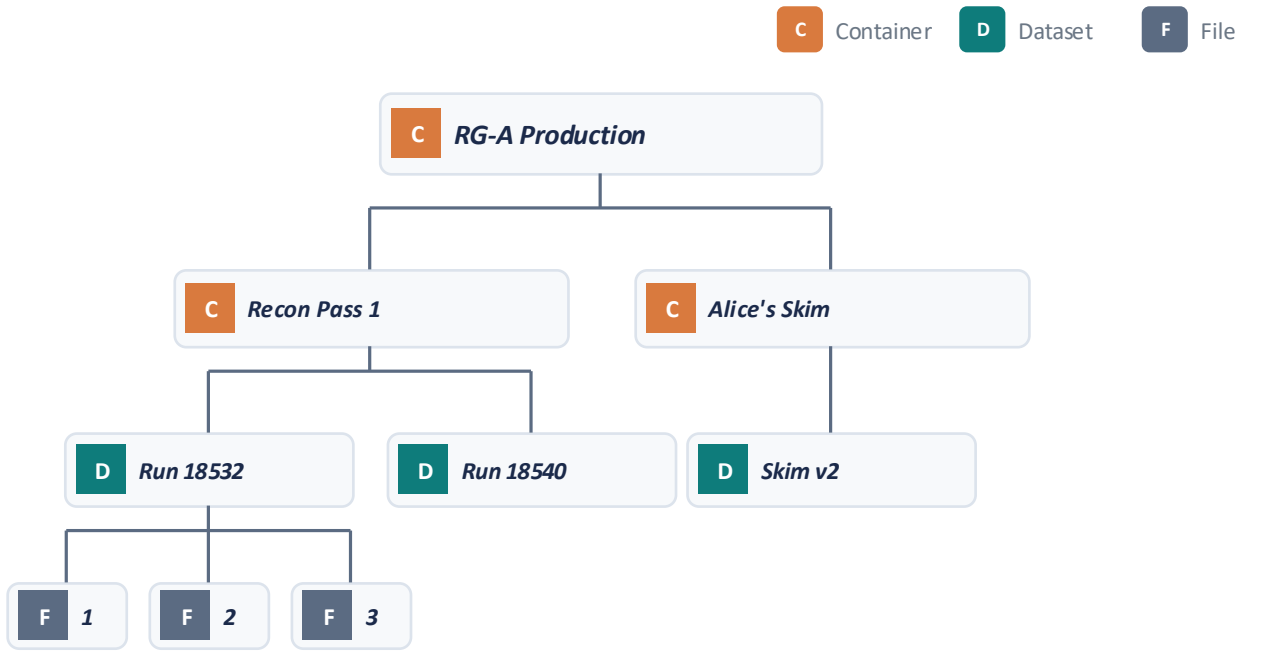
Namespace arrangement: files, datasets, containers

Files are the smallest operational unit — data is physically stored as files, and sub-file operations aren't possible.

Datasets are a named set of files.

Containers are a named set of datasets — or, recursively, other containers.

*All three are just names that refer to data — so we call any of them a **data identifier** (DID): the name of a single file, dataset, or container.*



"RG-A Production is physics ready" — the production group tags the top container once; everyone downstream inherits it.

FINDABILITY

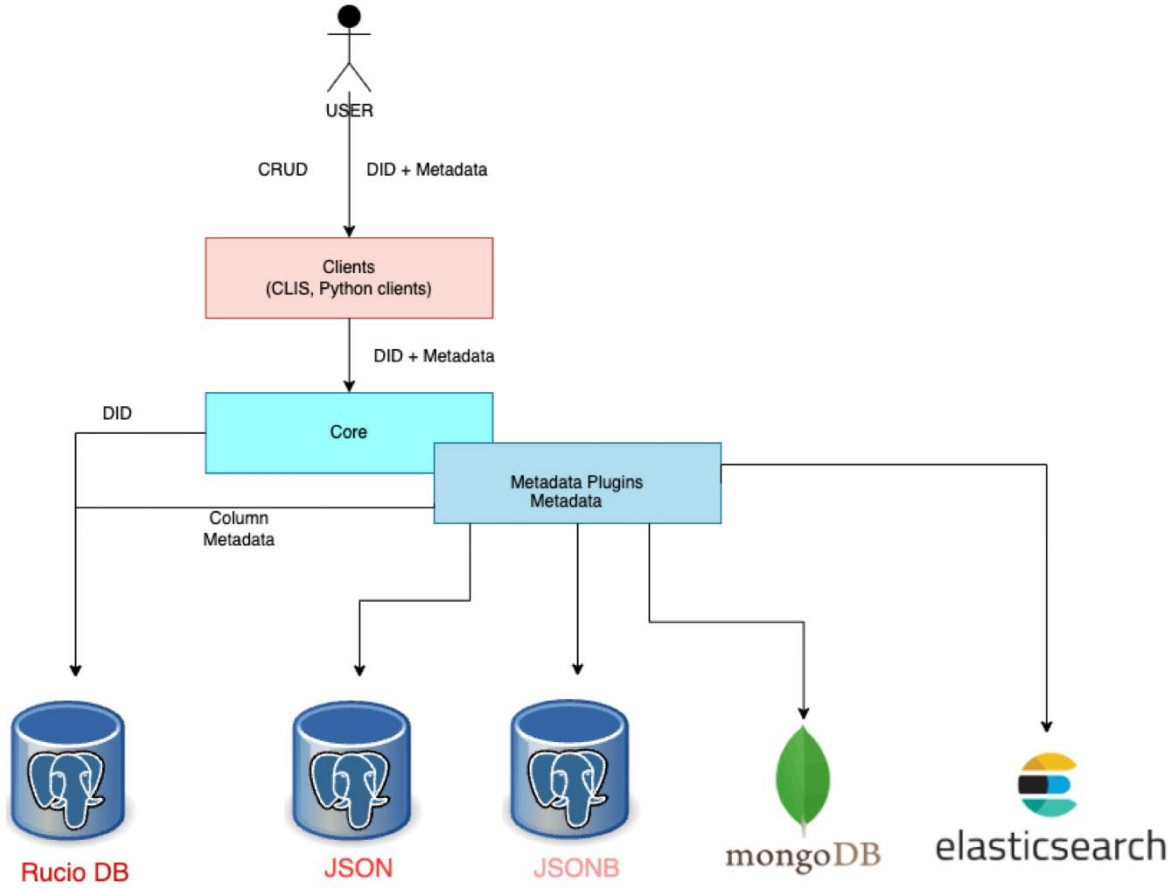
Metadata lives with the data

Any key/value metadata can be attached to a DID — run number, run group, processing pass, calibration version, anything your analysis needs to filter on.

This directly satisfies Findable and Accessible in FAIR.

```
rucio did --filter "run_group==rg-a,type==raw,run=18532"
```

```
rucio did --filter "run_group==rg-a,type==recon,pass=1"
```



Metadata lives with the data

```
$ rucio did list --filter="software_release=26.04.1-stable, generator=pythia8" 'epic:*'
```

```
(.rucio_venv) [panta@ifarm2401 ~/eic]$ rucio did list --short --filter="software_release=26.04.1-stable, generator=pythia8" 'epic:*'
epic:/RECO/26.04.1/epic_craterlake/DIS/NC/10x100/minQ2=10
epic:/RECO/26.04.1/epic_craterlake/DIS/NC/5x100/minQ2=10
epic:/RECO/26.04.1/epic_craterlake/DIS/CC/5x41/minQ2=100
epic:/RECO/26.04.1/epic_craterlake/DIS/NC/5x41/minQ2=1
epic:/RECO/26.04.1/epic_craterlake/DIS/CC/10x275/minQ2=100
epic:/RECO/26.04.1/epic_craterlake/DIS/NC/18x275/minQ2=1
epic:/RECO/26.04.1/epic_craterlake/DIS/CC/18x275/minQ2=100
epic:/RECO/26.04.1/epic_craterlake/DIS/NC/18x275/minQ2=10
epic:/RECO/26.04.1/epic_craterlake/DIS/CC/10x100/minQ2=100
epic:/RECO/26.04.1/epic_craterlake/DIS/NC/10x275/minQ2=1000
epic:/RECO/26.04.1/epic_craterlake/DIS/CC/5x100/minQ2=100
```

Use AI to interact and search via Rucio-MCP

[redacted] 3:16 PM

do we have any 9 GeV samples in Rucio?

PanDA bot BOT 3:16 PM

Yes — 6 datasets with 9 GeV beam energy in 26.05.0 campaign (epic scope):

Dataset	Beam Config	Q^2 Range	Generator
`/RECO/26.05.0/epic_craterlake/DIS/pythia8.316-1.0/NC/noRad/ep/9x130/q2_10to100`	9×130	10–100 GeV²	pythia8.316
`/RECO/26.05.0/epic_craterlake/DIS/pythia8.316-1.0/NC/noRad/ep/9x275/q2_10to100`	9×275	10–100 GeV²	pythia8.316
`/RECO/26.05.0/epic_craterlake/DIS/pythia8.316-1.0/NC/noRad/ep/9x130/q2_100to1000`	9×130	100–1000 GeV²	pythia8.316
`/RECO/26.05.0/epic_craterlake/DIS/pythia8.316-1.0/NC/noRad/ep/9x275/q2_1to10`	9×275	1–10 GeV²	pythia8.316
`/RECO/26.05.0/epic_craterlake/DIS/pythia8.316-1.0/NC/noRad/ep/9x130/q2_1to10`	9×130	1–10 GeV²	pythia8.316
`/RECO/26.05.0/epic_craterlake/DIS/pythia8.316-1.0/NC/noRad/ep/9x275/q2_100to1000`	9×275	100–1000 GeV²	pythia8.316

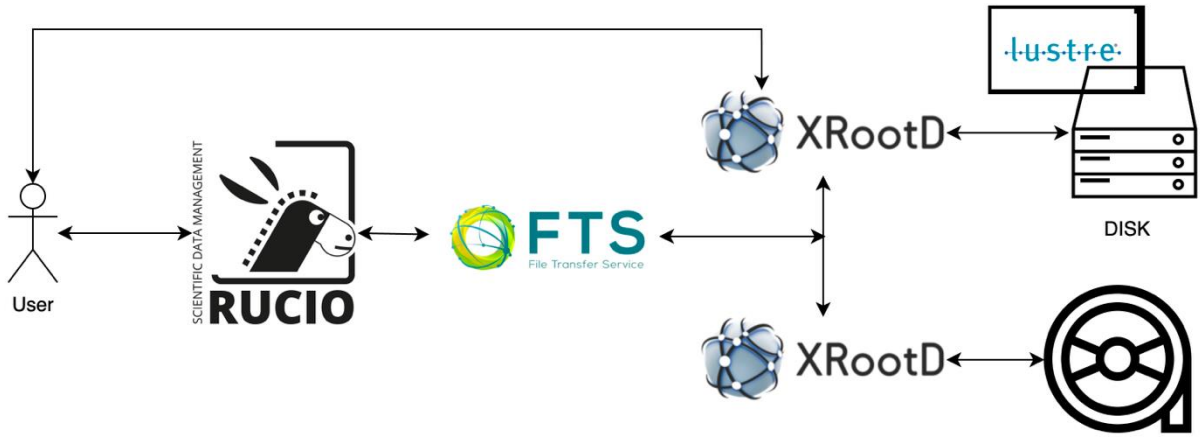
All **DIS NC** (no radiative corrections), pythia8.316, epic_craterlake detector. Covering 9×130 and 9×275 beam configs across Q^2 ranges 1–1000 GeV².

(tools suggested: jlab_rucio_list_files:0.43, jlab_rucio_list_scopes:0.43, jlab_rucio_list_content:0.42, jlab_rucio_get_did_metadata:0.41)
(tools used: jlab_rucio_list_dids, bnl_rucio_list_dids)

ACCESSIBILITY

Beyond the silo:

- CLAS12 data becomes reachable from anywhere — not just JLab filesystems.
- Disk storage is exposed through XRootD; Rucio talks to both DISK and TAPE.
- One download command, regardless of where the bytes physically sit.



No need to be bound to JLab filesystem access only

```
$ rucio download clas12:/RunGroupA/raw/run-018532/clas12_018532_000.evio
```

THE KEY IDEA

Declarative data management

You state the outcome you want — Rucio figures out how to get there.

"I need 1 copy of this dataset on DISK for 1 week."

```
$ rucio rule add clas12:run-018532  --copies 1  --rses JLAB-DISK-SE  --lifetime 604800
```

1

Lives on TAPE

Dataset starts as a tape-only replica — not directly job-readable.

2

Rucio stages it

The rule triggers replication; Rucio moves a copy to DISK automatically.

3

Cleans up on its own

After 7 days, the disk replica is deleted — no ticket, no manual step.

FOR ANALYZERS

Run your analysis wherever your compute is

Home institution clusters, national HPC, Open Science Grid — two ways to get CLAS12 data there:

1 Direct File access (download or remote open)

- Rucio resolves the DID to a storage URL.
- Open that URL directly in ROOT C++ or Python (uproot, etc).
- No download step — stream what you need. Speed depends on network path.
- OR you can download where you want and use that

```python
import uproot
from rucio.client import Client

# 1. Resolve Rucio DID to a Physical File Name (PFN)
c = Client()
reps = c.list_replicas([{'scope': 'gluex', 'name': 'myfile'}])
surl = next(reps)['rses']['JLAB-DISK-SE'][0]

# 2. Open and stream specific branches
with uproot.open(surl) as f:
    tree = f["T"]
    arrays = tree.arrays(["px", "py"])
```

```cpp
// Direct open using the XRootD URL provided by Rucio
TFile *f =TFile::Open("root://dtn-gluex.jlab.org//cache/rucio/gluex/path/to/file.root");
```

FOR ANALYZERS

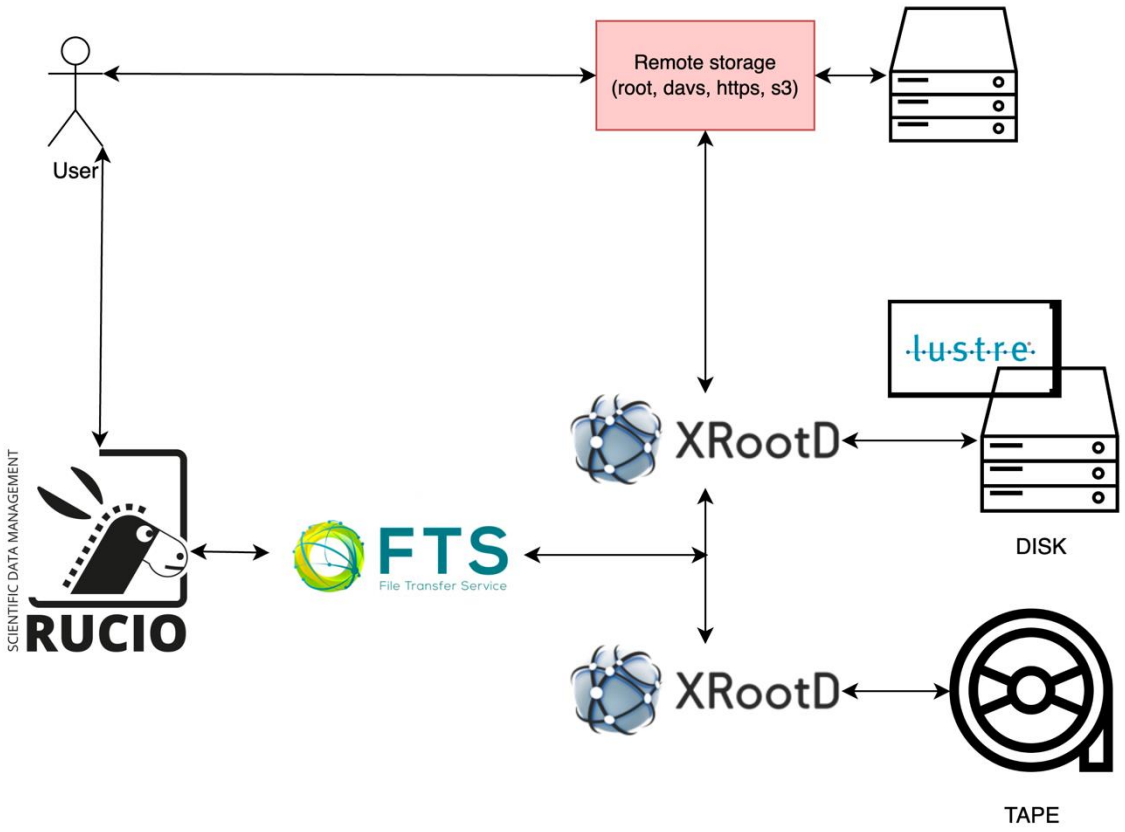
Run your analysis wherever your compute is

Home institution clusters, national HPC, Open Science Grid — two ways to get CLAS12 data there:

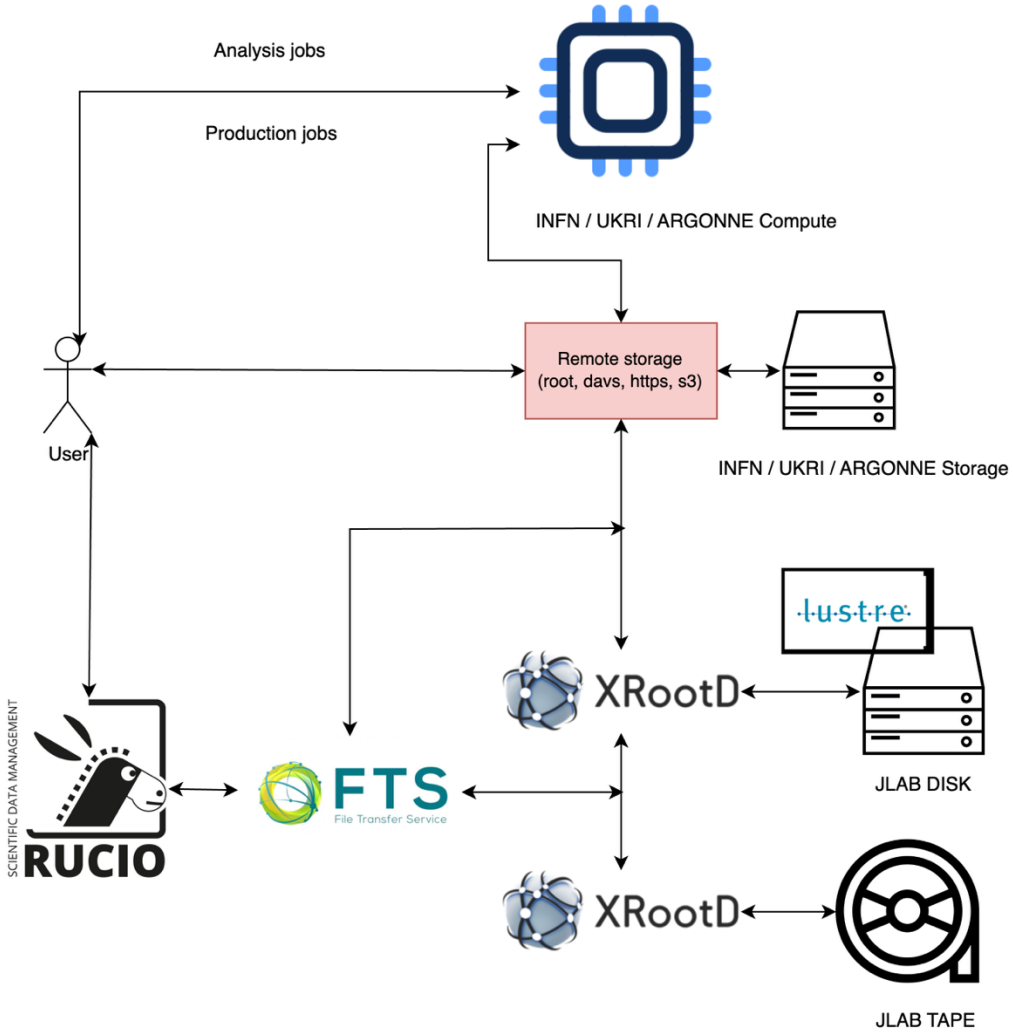
2 Attach External Storage

- Bring your own storage — root, https/davs, S3, or Globus endpoints all work.
- Rucio replicates data there on request, on a lifetime you set.
- Good fit when network egress from JLab is the bottleneck, not the analysis itself.

"I need 1 copy of X data in my external storage, for Y lifetime."



Run your analysis wherever your compute is



OPERATIONS

Governance built in

Accounts & Quota

User- and group-based operations. Each account gets a storage quota — no surprise space clashes between groups.
Federated logins, token management e.t.c

Data Popularity

Access traces show what's actually being used. "This dataset hasn't been touched in months" becomes a policy decision, not a guess.

Declarative Lifecycle

New campaign: place on disk for N days, then archive to tape. Old campaign: drop the disk replica once the new one lands — automatically.

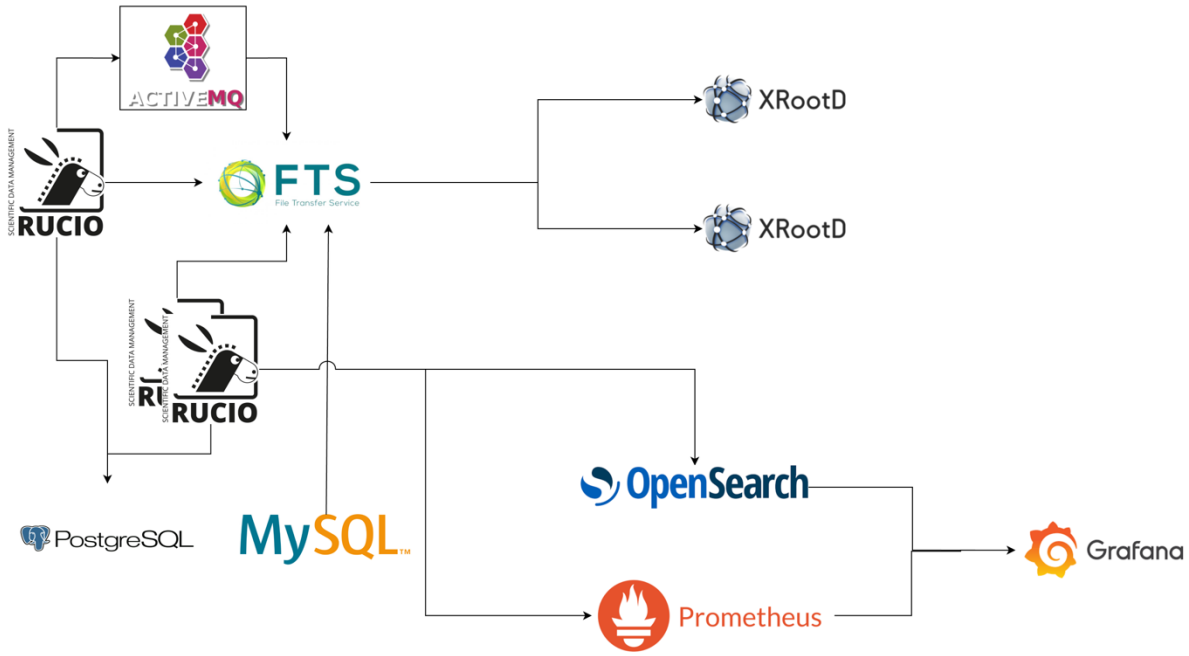
In practice: an unused analysis dataset or Monte Carlo dataset doesn't get regenerated for the next production campaign — the popularity trace already told us nobody reads it.

WHERE WE STAND TODAY

Deployed at JLab for EIC and GlueX

Infrastructure deployed

- Rucio server(EIC and GlueX), with XRootD disk and tape interaction
- FTS3 for managed file transfers
- Monitoring Infrastructure



CLAS12 status

- Make GlueX instance Multi-VO
- Add Clas12 as a VO.
- Start by defining the naming scheme of Clas12
- Start registering the old dataset acc. to agreed scheme

ROADMAP

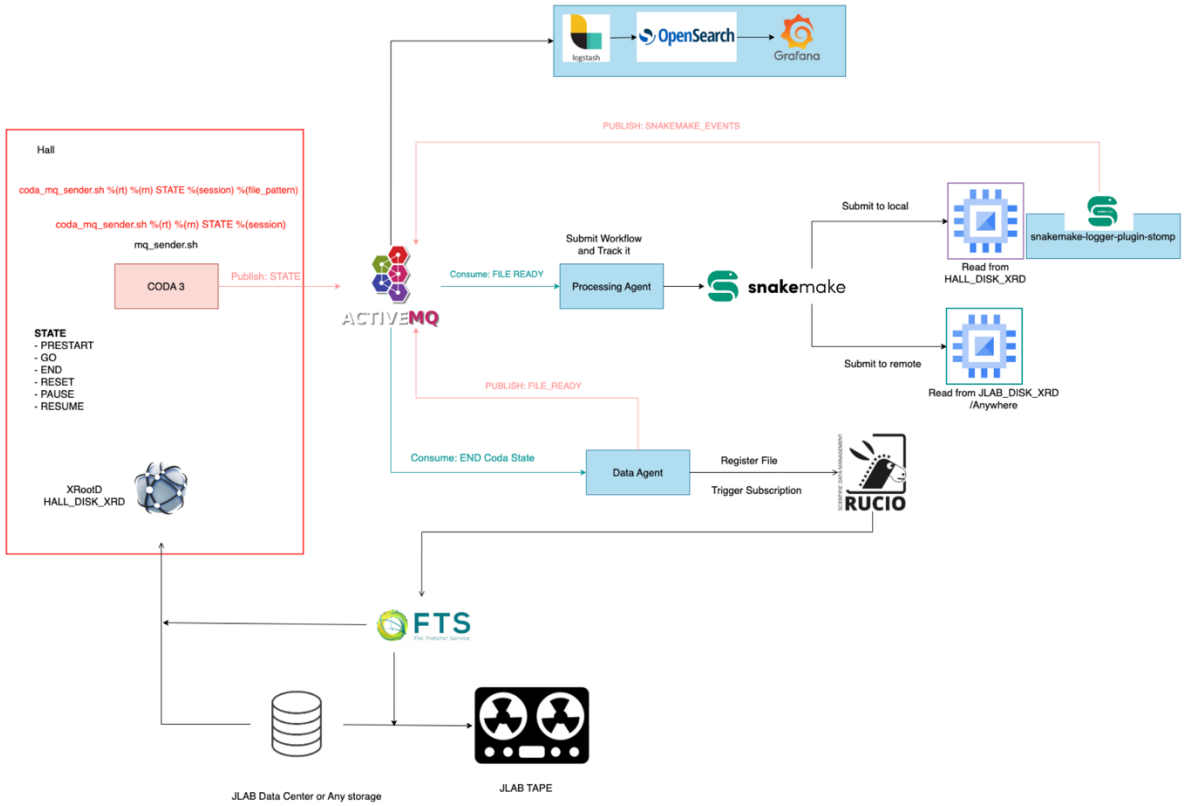
What's next

Automatic replays, prompt reconstruction

System to integrate DAQ to auto register to Rucio and then process automatically with workflow capture.

SWIF integration

SWIF workflows take a Rucio DID directly as an input dataset.



THE BIGGER PICTURE

JLab is rebuilding data management, piece by piece

Rucio is one piece of a larger shift — each system covers a different stage of the data lifecycle.

LIVE

Data Catalog

Rucio

Where files, datasets, and containers live — covered today.

LIVE

Document Repository

InvenioRDM

Publications, notes, and community pages — covered in the earlier talk.

LIVE

Metadata Catalog

Rucio

Searchable run, calibration, and processing metadata attached to every DID.

PLANNED

Workflow Capture

Planned — REANA-style

Capture the steps, not just the output, so an analysis can be re-run and reproduced.

Why this matters together:

- a catalog tells you which data exists and what is the data;
- a repository tells you what was published and related notes;
- a workflow system tells you exactly;

“JLab is working toward all three meeting.”