DATA SCIENCE &
SCIENTIFIC COMPUTING

UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

Designing and deploying a FAIR-by-design data pipeline and platform for electron microscopy laboratories

Research thesis in: Data Management

Supervisor
Dott. Federica Bazzocchi

Candidate
Nicola Perin

University of Trieste

19 settembre 2025

Outline

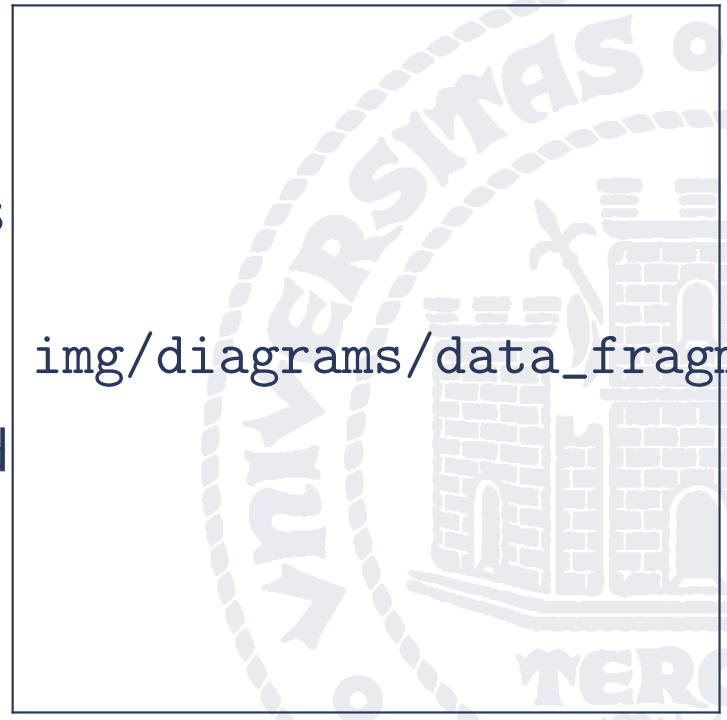
Electron microscopy at a glance

- Different modes: TEM, SEM, STEM $\rightarrow$ images, diffraction patterns, spectra.
- The data: large, complex, and very diverse in shape and size.
- The reality: every vendor has their own format, metadata is often incomplete.

The key question: how do we make these outputs easier to reuse and share?

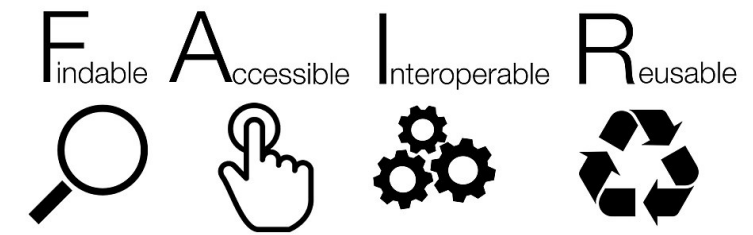
Current Problems and What's Needed

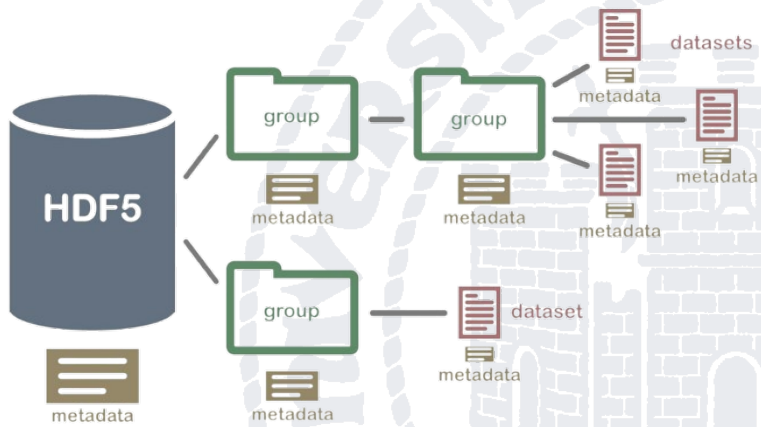
- **Fragmentation**: many formats, weak or missing metadata.
- **Friction**: manual copying, endless zip files, confusing naming.
- **Collaboration**: unclear provenance, scattered access, hard to reuse.
- **Need**: structured metadata, persistent IDs, scalable storage, simple tools.



A way forward: FAIR and NeXus (NXem)

- FAIR principles: **findable, accessible, interoperable, reusable**.
- **HDF5**: efficient format for large, structured datasets.
- **NeXus**: conventions for scientific data (`NXinstrument`, `NXsample`).
- **NXem**: application definition tailored to electron microscopy.





NeXus

At the national level, these FAIR practices are promoted and supported by the NFFA-DI infrastructure.

Introducing NFFA-DI

- **NFFA-DI** = Nano Foundries and Fine Analysis – Digital Infrastructure.
- Italian research initiative connecting major nanoscience centers.
- Goal: open access to advanced instrumentation, FAIR data, and computational resources.
- Acts as the national driver for FAIR data practices in nanoscience.

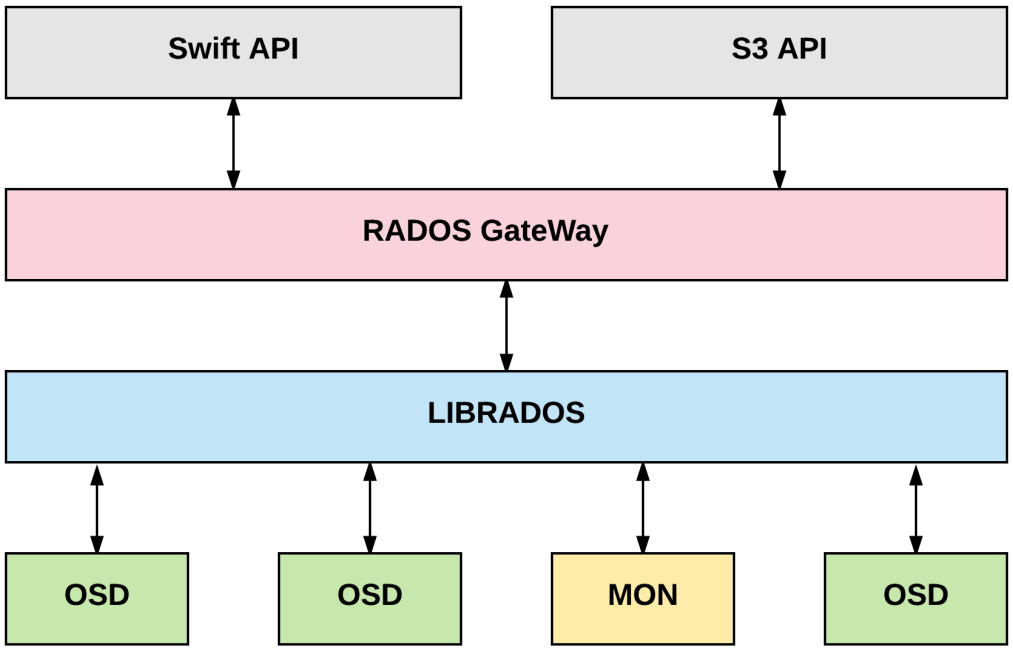
Source: `https://nffa-di.it/en/`

Institutional context

- **LAME** produces multi-terabyte datasets in electron microscopy.
- As part of NFFA-DI, its work depends on sharing data with other partners.
- To support this, **ORFEO** provides the backbone: HPC resources, identity services, and S3-compatible object storage.
- The challenge: connecting LAME's lab workflows with ORFEO's infrastructure in a smooth way.

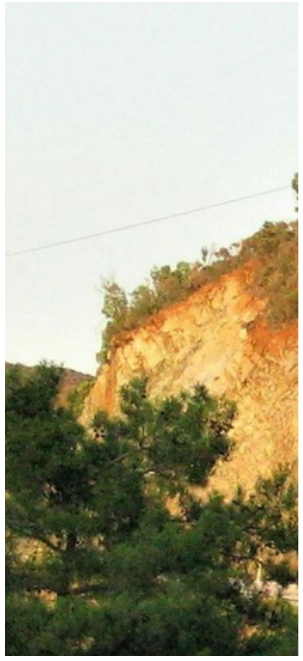
ORFEO's storage model

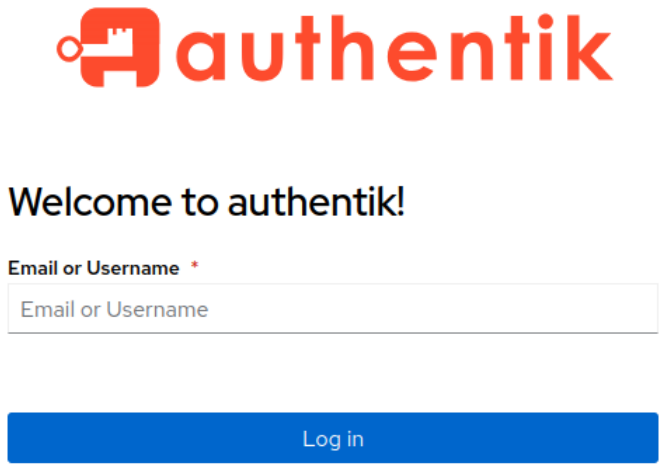
ORFEO does not use a classic file-and-folder hierarchy. Instead, data is stored as **objects** inside **buckets**, managed by Ceph RGW with the S3 protocol. Each object combines the raw data with flexible metadata, which makes it easier to describe and reuse datasets.



Accessing ORFEO

All access to ORFEO's resources goes through a single **Ingress** endpoint. Connections are secured with **TLS certificates**, and identity is handled by **FreeIPA** and **Authentik**, providing single sign-on across services.





From problems to a proposal

What's missing for LAME

- Data often stuck on lab machines or portable drives.
- Transfers are manual, with inconsistent folder structures.
- No ingestion standard $\rightarrow$ hard to reuse or integrate with ORFEO/NFFA-DI.

Our proposal

- Speak **S3** to move data directly into ORFEO.
- Convert outputs to **NeXus/NXem** during ingestion.
- Offer a simple web interface and an API for daily use.

We'll go into details later — this closes the introduction.

The Core Idea

- **Transfer**: data goes from LAME straight into ORFEO buckets.
- **Transform**: TIFF files get converted into NeXus with normalized metadata.
- We integrate existing services, not reinvent them.

Why Django?

- A clear data model for projects, samples, and experiments.
- Django REST Framework $\rightarrow$ easy APIs.
- HTMX $\rightarrow$ dynamic UI without a heavy frontend.
- RQ workers $\rightarrow$ handle background jobs (checksums, NeXus builds).
- Solid ecosystem for authentication, migrations, and testing.

Thinking About the Whole Pipeline

- The app is one piece in a larger chain: identity $\rightarrow$ ingress $\rightarrow$ storage $\rightarrow$ compute.
- It only makes sense if the whole path is tested together.
- Solution: a "digital twin" of ORFEO to try everything before production.

What VirtualOrfeo Is

- A small-scale copy of ORFEO built on K3s.
- Includes ingress, TLS, Authentik/FreeIPA, Ceph RGW (or MinIO).
- Same Helm charts and configs as production.

What It Reproduces

Identity

- FreeIPA + Authentik for SSO.

Storage

- Ceph RGW, presigned uploads/downloads.

Deployment

- App packaged with Helm, running web + workers + DB.

How the App is Structured

- Domain: Project → Proposal → Sample → Experiment → Measurements.
- Data plane: browser S3 using presigned URLs; metadata in Postgres.
- Workers: checksum, metadata extraction, NeXus builds.

Logging In

- Auth handled by Authentik (OIDC).
- Django sees tokens, not passwords.
- Group claims set roles; disabling an account works instantly.

Managing Research Data

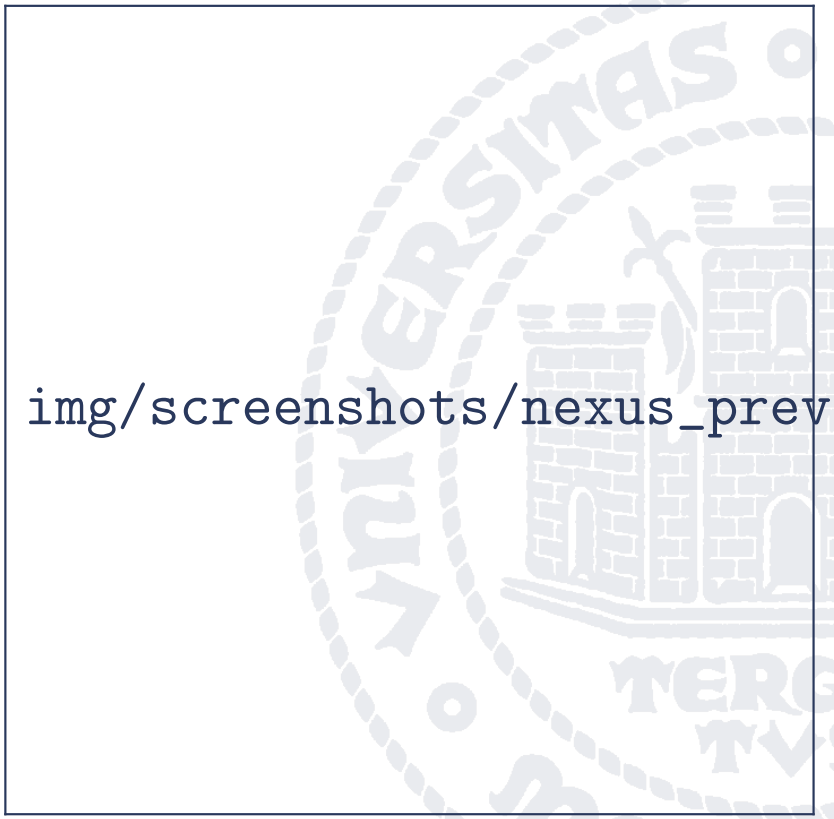
- Three-pane board to create and link entities.
- Context stored next to raw data (`README.txt`).
- Same functionality via REST API for automation.

Uploading Data

- App gives a one-time URL, browser streams directly to storage.
- Handles large files without overloading the web server.
- Uploads automatically trigger checksum and registration jobs.

From TIFF to NeXus

- Extract metadata from TIFF headers or JSON blocks.
- Normalize values and map into NXem fields.
- Build `.nxs` files with a standard structure.



Background Jobs

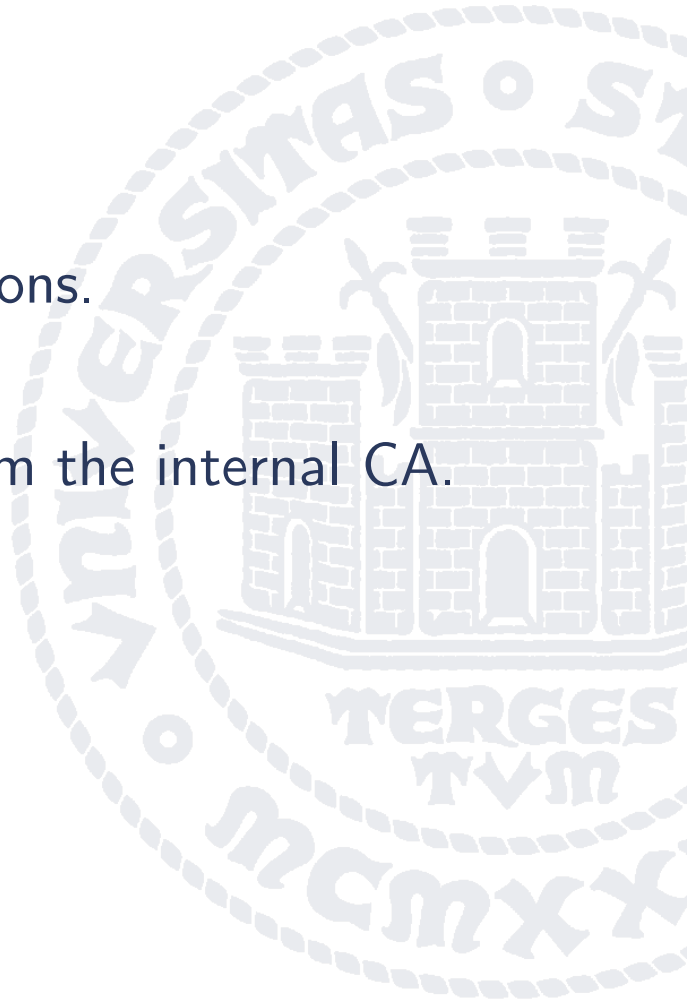
- RQ queues for checksums and NeXus builds.
- Jobs are idempotent, retry automatically if needed.
- Monitoring via web UI and structured logs.

Browsing and Sharing

- Browse buckets, download with presigned links.
- On-the-fly ZIPs for folders; `aria2` manifests for bulk.
- Derived data stored in a mirrored namespace.

Keeping It Secure

- Minimal scopes and group-based permissions.
- Presigned links limited in time and scope.
- Secrets managed by Kubernetes; TLS from the internal CA.



The Payoff

- Data moves smoothly from the lab to ORFEO.
- Files are stored in a standard (NeXus/NXem) from the start.
- Researchers get a simple workflow, and data remains FAIR for the future.

Questions?