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 → 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.

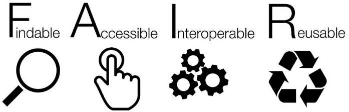
The key question: how do we make these outputs easier to reuse and share?

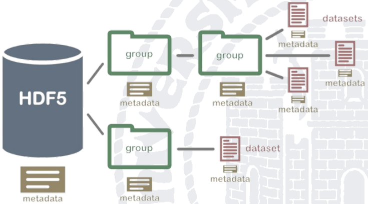
Current Problems and What's Needed

- **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.



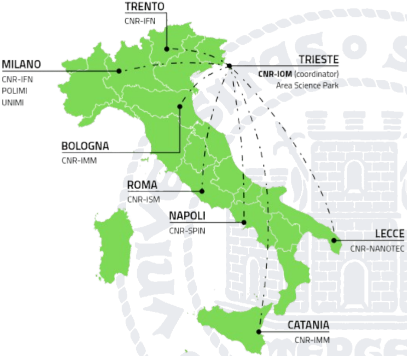


NeXus

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.



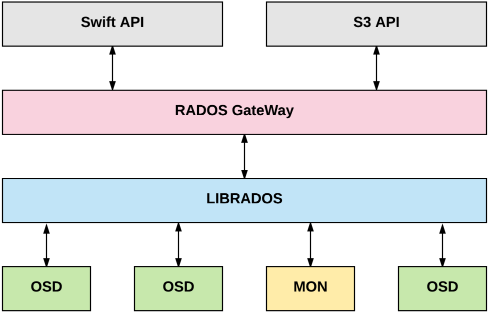
Source: `https://nffa-di.it/en/`

From the lab to the datacenter

- **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.

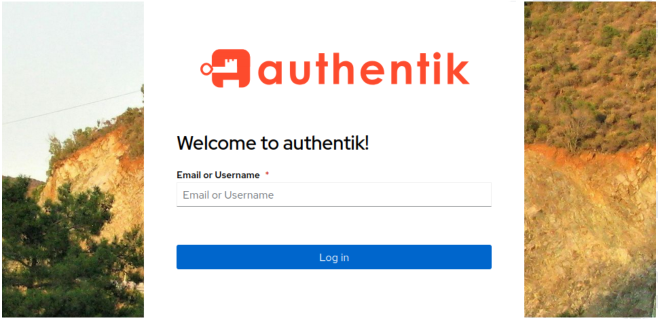
ORFEO's storage model

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.



Welcome to authentik!

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 → hard to reuse or integrate with ORFEO/NFFA-DI.

Our proposal

- **Transfer**: move data directly into ORFEO using the S3 protocol.
- **Transform**: convert outputs (e.g. TIFF) into NeXus/NXem with standardized metadata.
- **Integrate**: build on ORFEO's existing services, with a simple web interface and API.

Choosing a framework

To put our proposal into practice, we need a tool that researchers can actually use. That means building a **web application** that can:

- guide researchers through projects, samples, and experiments,
- handle uploads and metadata in a consistent way,
- connect directly to ORFEO's services (S3 storage, authentication).

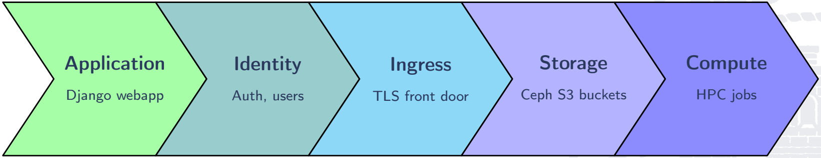
Once this was clear, the next step was to choose the right framework.

Why Django?

We needed a framework that was stable, flexible, and easy to maintain. Django fits well because it provides:

- A structured way to model projects, samples, and experiments.
- Built-in tools to create both a user interface and an API.
- Support for background tasks such as data checks and conversions.
- Strong support for authentication and long-term maintenance.

Thinking about the whole pipeline



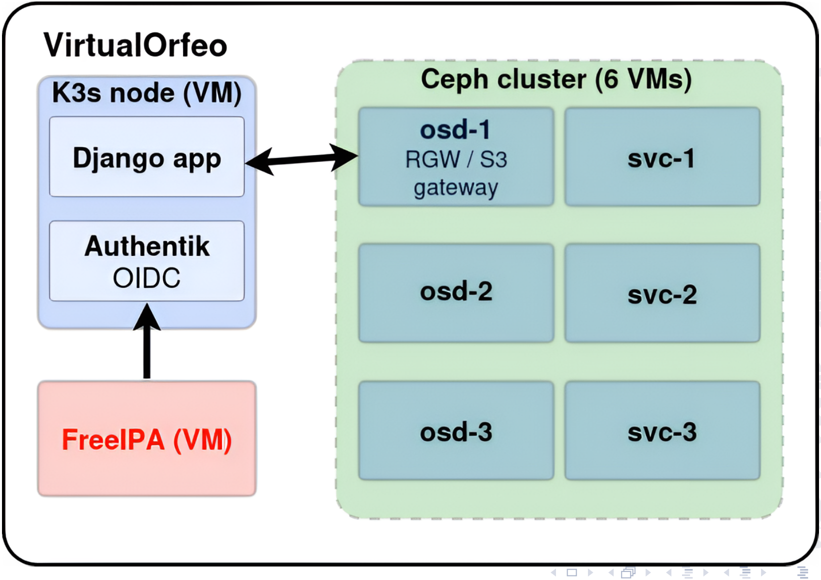
The app is one part of this chain — testing only makes sense when the whole path is reproduced. Our solution: a **digital twin** of ORFEO.

VirtualOrfeo

ORFEO is a complex infrastructure: identity, ingress, storage, and compute. Testing our Django app directly on production would be risky and slow.

- **VirtualOrfeo** is a lightweight clone of ORFEO, built on K3s.
- It uses the same Helm charts and configs as production.
- This lets us deploy the **Django app** in a realistic environment: it can authenticate through Authentik, upload to S3 buckets, and be accessed through the same ingress as in ORFEO.

VirtualOrfeo topology



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?