



NFDI4
BIOIMAGE

NFDI4BioImage TA3 Hackathon @ UoC



NFDI4
BIOIMAGE

Interoperability

*Eik Dahms, Lukas Weil, Christoph Möhl,
Michele Bortolomeazzi, Niraj Kandpal,
Mohamed Abdrabbou, Josh Moore, Jens
Wendt*

OMERO-ARC-Interaction

3 approaches with increasing complexity

30.11.23_Morning discussion

Reference model

OMERO and/or ARC store reference IDs (URLs) of the data objects in the other Database.

This would technically be the easiest solution, but specifications need to be updated. More hard constrain: Do we want, to have data and metadata split, with the possibility of links pointing to nothing in the future?

Snapshot model

On demand snapshotting of information of one model and putting it into the other. The user decides when and what information to merge into what.

There is great control wise and might be very flexible in general. Snapshots could of course be exported from and imported into other tools as well.

The downside is, that desynchronization might occur.

Synchronization model

Full information is stored in OMER0 and DataHub, and being directly synchronized with each other.

This would be the holy grail solution, where users select their tool based on the current goal they have, and changes also appear on the other one.

Technically, this is very difficult to implement. Merge-conflicts might be showstopper.

Snapshot model use case

Omero ARC Exporter

A CLI tool to export Omero projects to arc is currently developed as an extension of [omero-cli-transfer](#):

[omero-cli-transfer with arc export option](#) (arc branch)

Workflow

Two options:

- Create a new ARC from an Omero project
- Add an Omero project as new study and associated Assays (Omero datasets) to an existing ARC.

```
omero transfer pack --arc Project:1343 path/to/my/new_arc  
omero transfer pack --arc Project:1343 path/to/my/existing_arc
```



Project-Dataset

- ## Screen-Plate-Well

- Diagram illustrating the workflow for processing and archiving microscopy images:
- Omero** (Left side):
- Starts with a **PK** (Project Key) node.
 - Leads to a **D** (Dataset) node.
 - The **D** node branches into two **I** (Image) nodes.
 - Each **I** node leads to an **FS** (File System) node.
 - The **FS** nodes are connected to a box labeled **Image**.
- ARCO** (Right side):
- Starts with a **PK** (Project Key) node.
 - Leads to a **S** (Study) node.
 - The **S** node branches into two **P₁** (Process 1) nodes.
 - Each **P₁** node leads to a **P₂** (Process 2) node.
 - The **P₂** nodes lead to a **FS** (File System) node.
 - The **FS** node is connected to a box labeled **Image**.
- Process/Storage** (Bottom):
- Leads to a **File + Pointer** node.
 - The **File + Pointer** node branches into **Directory** and **File** nodes.
 - The **Directory** node is connected to a box labeled **Image**.

Image mapping

General thoughts

- Files and images are a N to M mapping.

In order to represent the measurement of a sample resulting in the creation of a specific image, pointing to the file is not enough.

Maybe this could be solved by data fragment selectors (see [PR on ISA](#)). Probably there would be a selector syntax needed for each file format.

Also in some cases it would be more practical to have the **Image** **File** column point to a directory instead. Does this break ISA/ARC?

- Natively mappable properties should be only natively mapped

As a native property in a schema we define a property that has a fixed key. Natively mappable then means that the same piece of information has a specific place and key in both schemas.

