Piano Nazionale di Ripresa e Resilienza

NeXus and Electron Microscopy

Federica Bazzocchi,

Istituto di Ricerca per l’Innovazione Tecnologica (RIT) Trieste

Optical, Electron, and Scanning Probe Microscopy Online Workshop

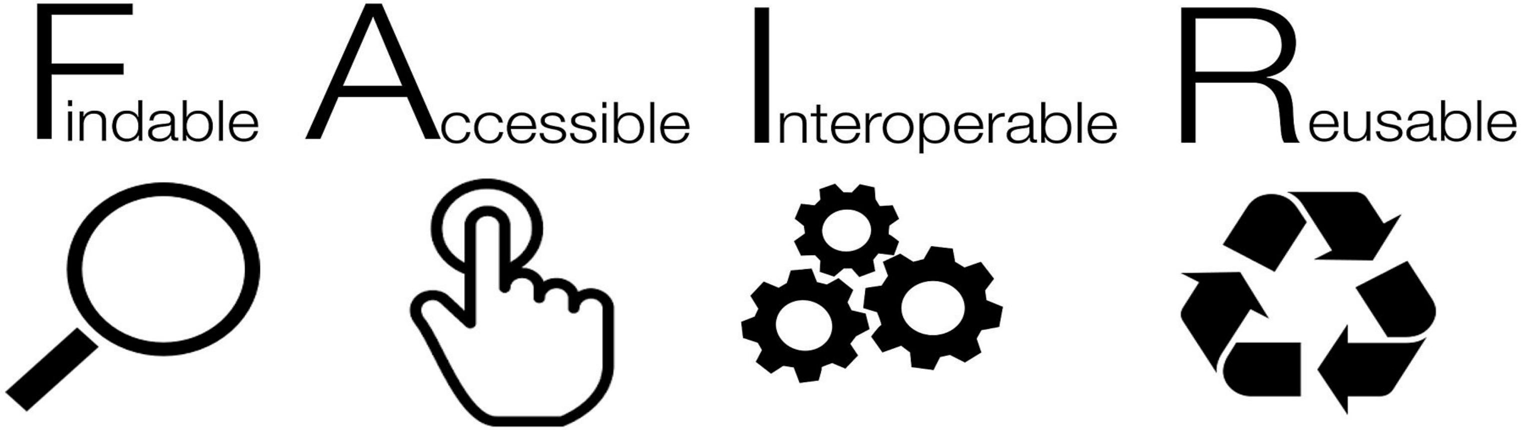
6.11.2024

OUTLINE

- □ Brief recap on FAIR principles
- □ NeXus file format
- □ Nxem
- □ Final remarks

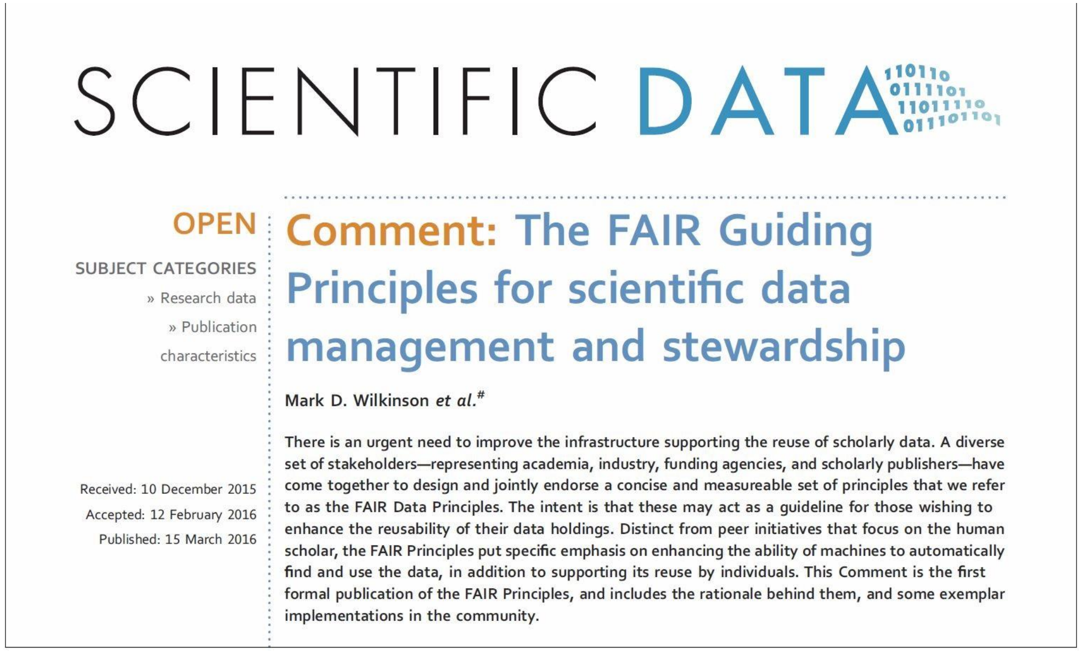
AREA SCIENCE PARK

❑ Brief recap on FAIR principles



Findable Accessible Interoperable Reusable

Door SangyaPundir - Eigen werk, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=53414062



SCIENTIFIC DATA

OPEN

SUBJECT CATEGORIES
» Research data
» Publication characteristics

Received: 10 December 2015
Accepted: 12 February 2016
Published: 15 March 2016

Comment: The FAIR Guiding Principles for scientific data management and stewardship

Mark D. Wilkinson *et al.*#

There is an urgent need to improve the infrastructure supporting the reuse of scholarly data. A diverse set of stakeholders—representing academia, industry, funding agencies, and scholarly publishers—have come together to design and jointly endorse a concise and measureable set of principles that we refer to as the FAIR Data Principles. The intent is that these may act as a guideline for those wishing to enhance the reusability of their data holdings. Distinct from peer initiatives that focus on the human scholar, the FAIR Principles put specific emphasis on enhancing the ability of machines to automatically find and use the data, in addition to supporting its reuse by individuals. This Comment is the first formal publication of the FAIR Principles, and includes the rationale behind them, and some exemplar implementations in the community.

https://doi.org/10.1038/sdata.2016.18

❑ Brief recap on FAIR principles: the metadata role

F

F1: (Meta) data are assigned globally unique and persistent identifiers
F2: Data are described with rich metadata
F3: Metadata clearly and explicitly include the identifier of the data they describe
F4: (Meta)data are registered or indexed in a searchable resource

A

A1: (Meta)data are retrievable by their identifier using a standardized communication protocol
A1.1: The protocol is open, free and universally implementable
A1.2: The protocol allows for an authentication and authorization procedure where necessary
A2: Metadata should be accessible even when the data is no longer available

I

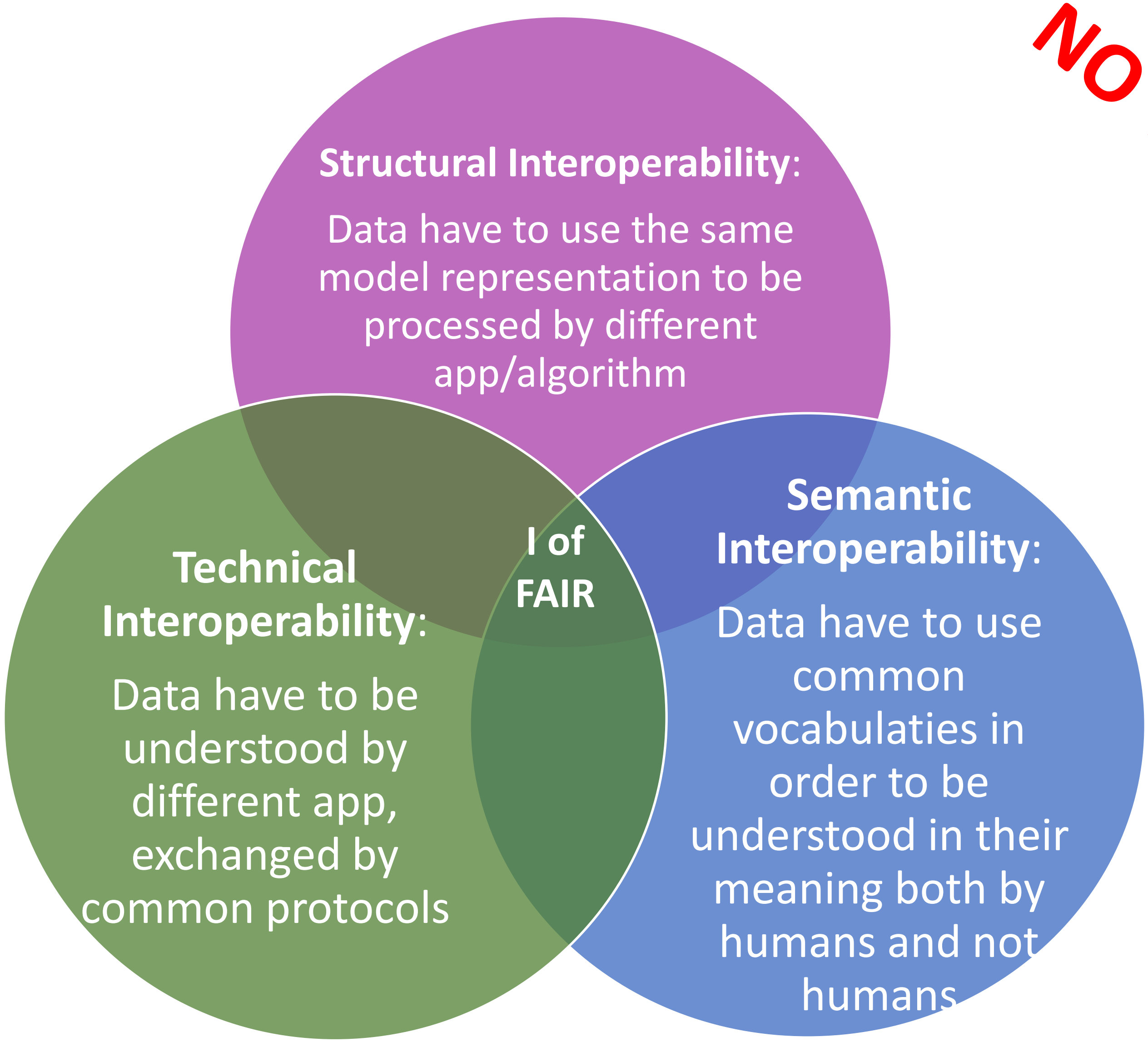
I1: (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation
I2: (Meta)data use vocabularies that follow the FAIR principles
I3: (Meta)data include qualified references to other (meta)data

R

R1: (Meta)data are richly described with a plurality of accurate and relevant attributes
R1.1: (Meta)data are released with a clear and accessible data usage license
R1.2: (Meta)data are associated with detailed provenance
R1.3: (Meta)data meet domain-relevant community standards

❑ Brief recap on FAIR principles: the interoperability challenge

NO TRIVIAL !!!



Structural Interoperability:

Data have to use the same model representation to be processed by different app/algorithm

Technical Interoperability:

Data have to be understood by different app, exchanged by common protocols

Semantic Interoperability:

Data have to use common vocabulaties in order to be understood in their meaning both by humans and not humans

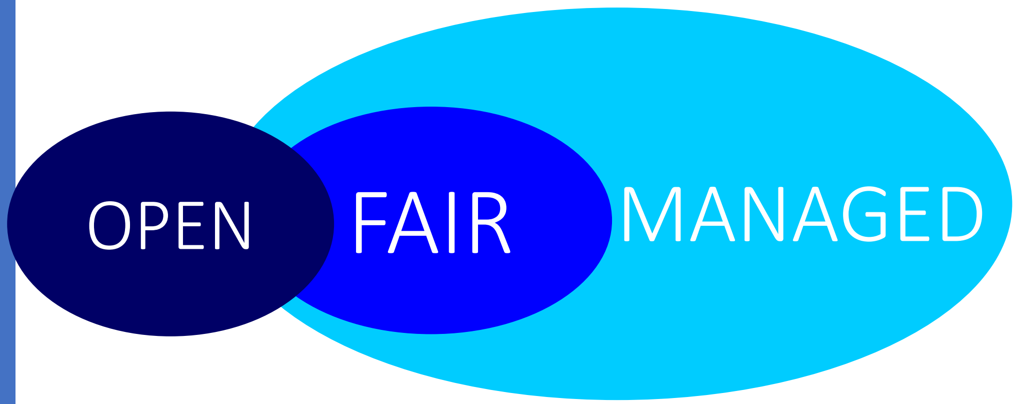
I of FAIR

AREA SCIENCE PARK

Managing data properly is in the primary interest of any researcher, as the whole research process results streamlined and more effective

FAIR ≠ OPEN

FAIRness VS OPENness



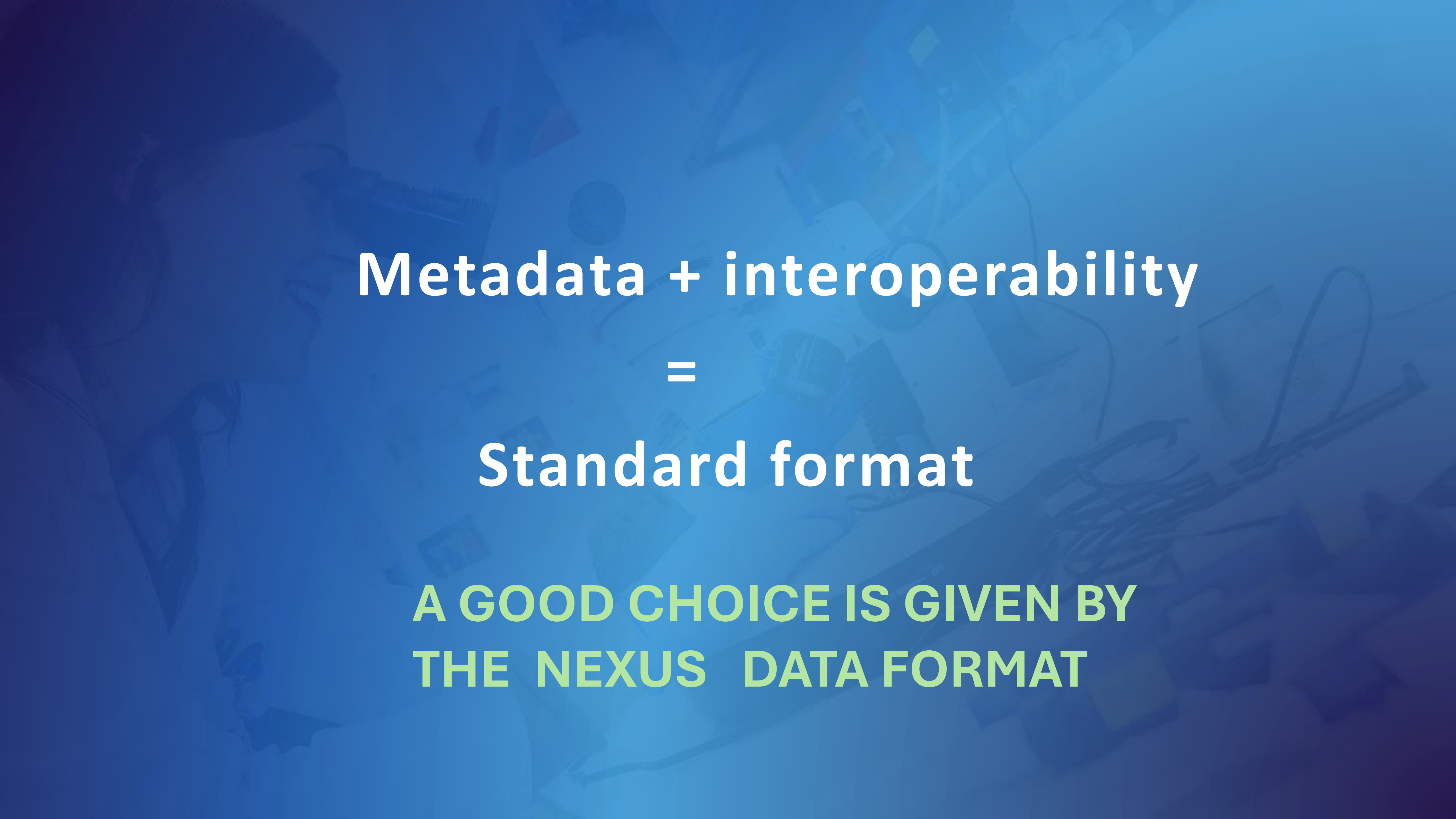
1. DATA SHOULD BE AS OPEN AS POSSIBLE

2. BUT IF DATA ARE NOT «FAIR», OPENING IS RISKY (MISUSE, MISINTERPRETATION, …)

3. IF DATA ARE NOT PROPERLY MANAGED FROM THE BEGINNING, IT'S ALMOST IMPOSSIBLE TO MAKE THEM «FAIR» [WITH EOSC MANAGED/FAIR INCREASINGLY OVERLAPPING, **«FAIR-BY-DESIGN»]**

In Horizon Europe data should be «as open as possible and as closed as necessary»

Slides by Elena Giglia DOI 10.5281/zenodo.6761944



Metadata + interoperability

=

Standard format

A GOOD CHOICE IS GIVEN BY THE NEXUS DATA FORMAT

NeXus data format

NeXus

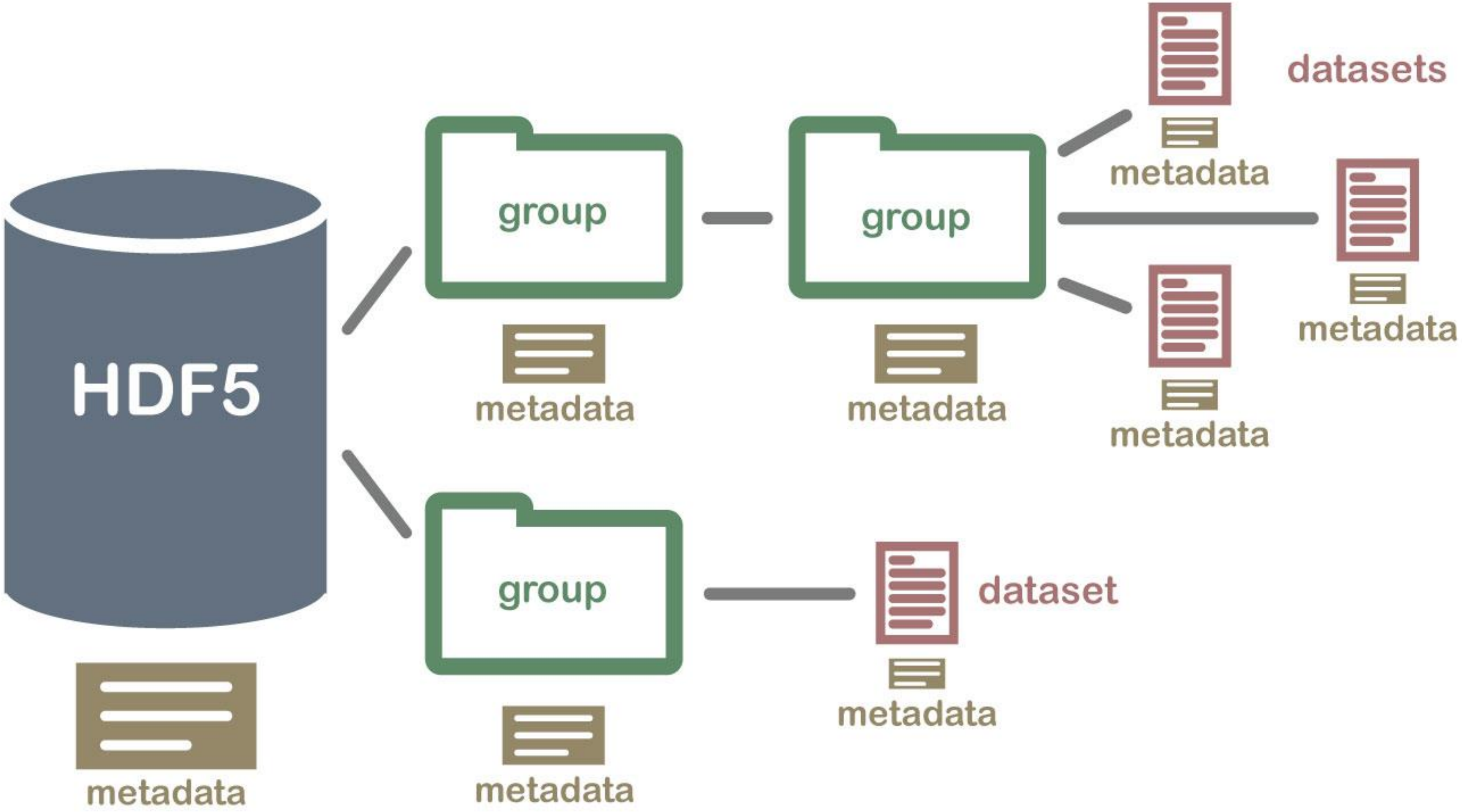
NeXus is a common data exchange format for neutron, X-ray, and muon experiments. NeXus is built on top of the scientific data format HDF5 and adds domain-specific rules for organizing data within HDF5 files in addition to a dictionary of well-defined domain-specific field names. Application definition are supervised/regulated by NIAC

The NeXus data format has three purposes:

- **Raw data**: NeXus defines a format that can serve as a container for all relevant data associated with a scientific instrument or beamline.
- **Processed data**: NeXus also defines standards for processed data. This is data which has underwent some form of data reduction or data analysis. NeXus allows storing the results of such processing together with documentation about how the processed data was generated.
- **Standards**: NeXus defines standards in the form of application definitions for the exchange of data between applications. NeXus provides standards for both raw and processed data.

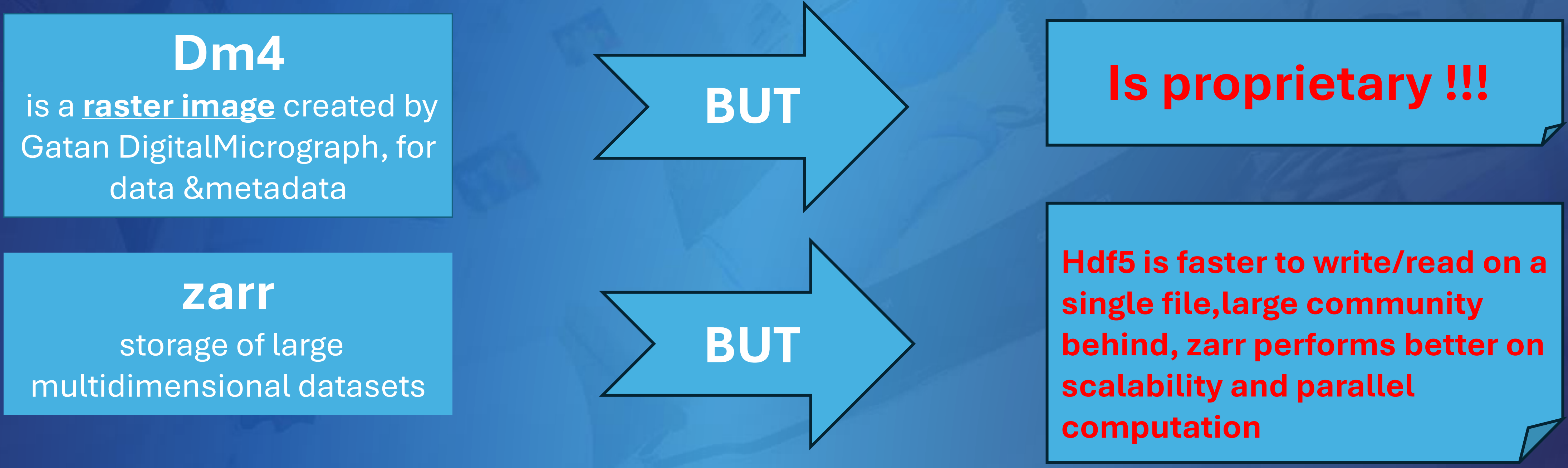
What is HDF5?

Hierarchical Data Format



NeXus is the best choice for TEM?

Huge amount of data, do we have other file formats that could perform better?



Dm4

is a **raster image** created by Gatan DigitalMicrograph, for data &metadata

BUT

Is proprietary !!!

zarr

storage of large multidimensional datasets

BUT

Hdf5 is faster to write/read on a single file,large community behind, zarr performs better on scalability and parallel computation

NXDL and NeXus class definition

NeXus

The set of rules for storing information in NeXus data files is declared using the **NeXus Definition Language**. NXDL itself is governed by a set of rules (a schema) and is written as an XML Schema, hence it is machine-readable using industry-standard and widely-available software tools for XML files.

Class definitions are specified in each domain specific NXDL scheme:

- **Base class definitions** define the complete set of terms that **might** be used in an instance of that class.
- **Application definitions** define the **minimum** set of terms that **must** be used in an instance of that class.
- **Contributed definitions** include propositions from the community for NeXus base classes or application definitions

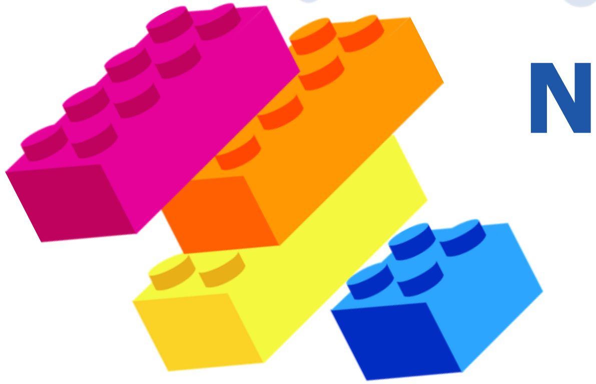
AREA SCIENCE PARK

an analogy....

Application definition

Application definition

Base Classes



Nexus : Game Elements

NeXus

Base class definitions and **application definitions** are made of

- □ Groups

  Levels in the NeXus hierarchy. May contain fields and other groups.
- □ Fields

  Multidimensional arrays and scalars representing the actual data to be stored.
- □ Attributes

  Attributes containing additional metadata can be assigned to groups, fields, or files.
- □ Links

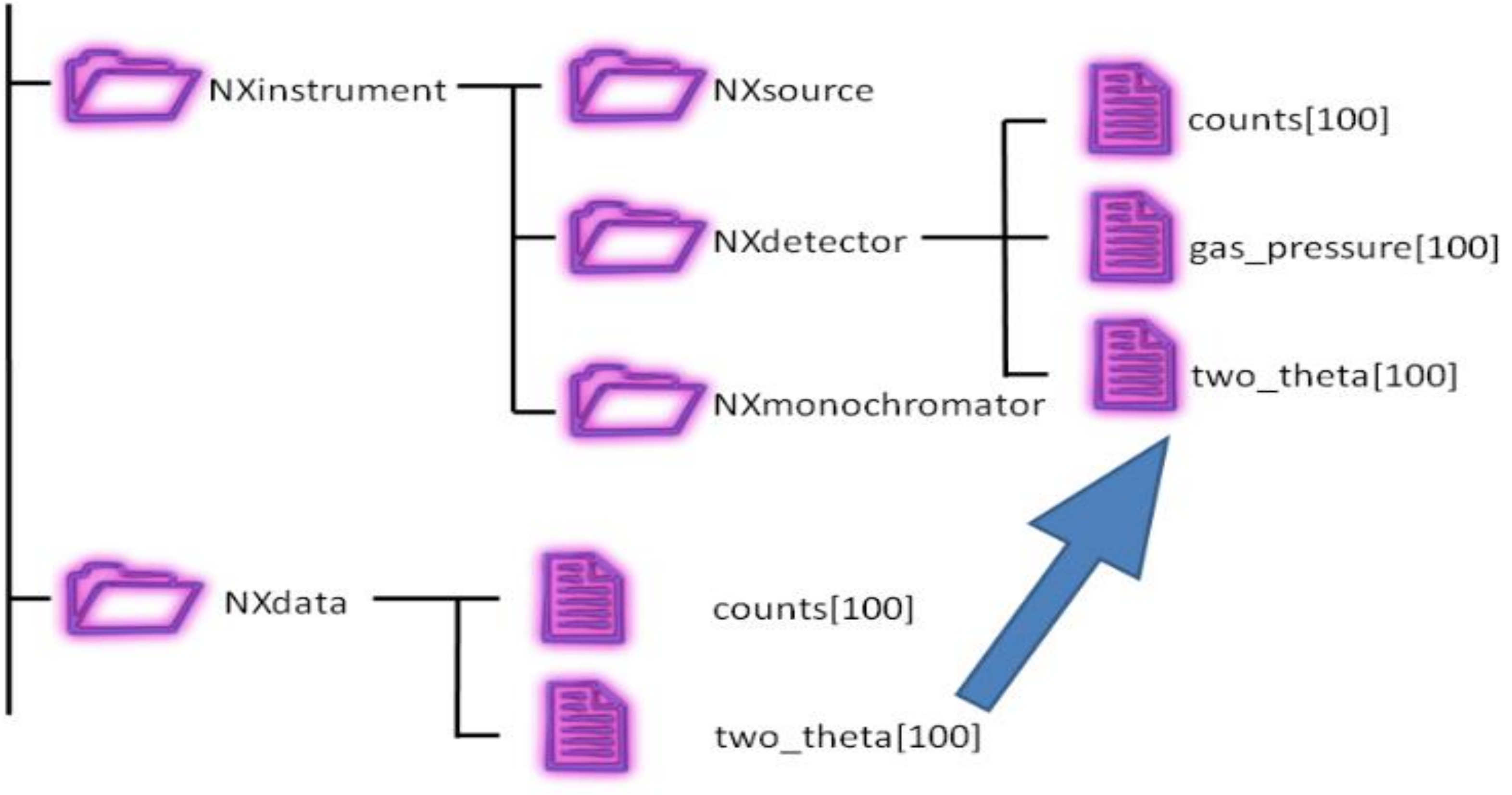
  Elements which point to data stored in another place in the file hierarchy.

AREA SCIENCE PARK

FAIRmat Github

Nexus : Game Elements

NeXus



Linking in a NeXus file

AREA SCIENCE PARK

FAIRmat Github

Nexus : Game Rules

NeXus

The tree syntax is a very condensed version (with high information density) meant to convey the structure of the HDF file.

- **Groups** have an appended to their name (with NeXus class name shown).
- **Indentation** shows membership in the lesser indented parent above.
- **Fields** have a data type and value appended (for arrays, this may be an abbreviated view).
- **Attributes** (of groups or fields) are prefixed with `@`.
- NeXus-style **links** are described with some sort of arrow notation

AREA SCIENCE PARK

FAIRmat Github

NeXus

Let’s put everything together…

AREA
SCIENCE PARK

An example: Nxem

(contributed definition Partner consortia in the German National Research Data Infrastructure are here e.g. NFDI-MatWerk, NFDI4Ing, NFDI-BioImage, NFDI-Microbiota, NFDI4Health, and e.g. NFDI-Neuro)

NXem is a NeXus *application* definition for the normalized representation of electron microscopy research. It is an extension of the Nxem_base base class.

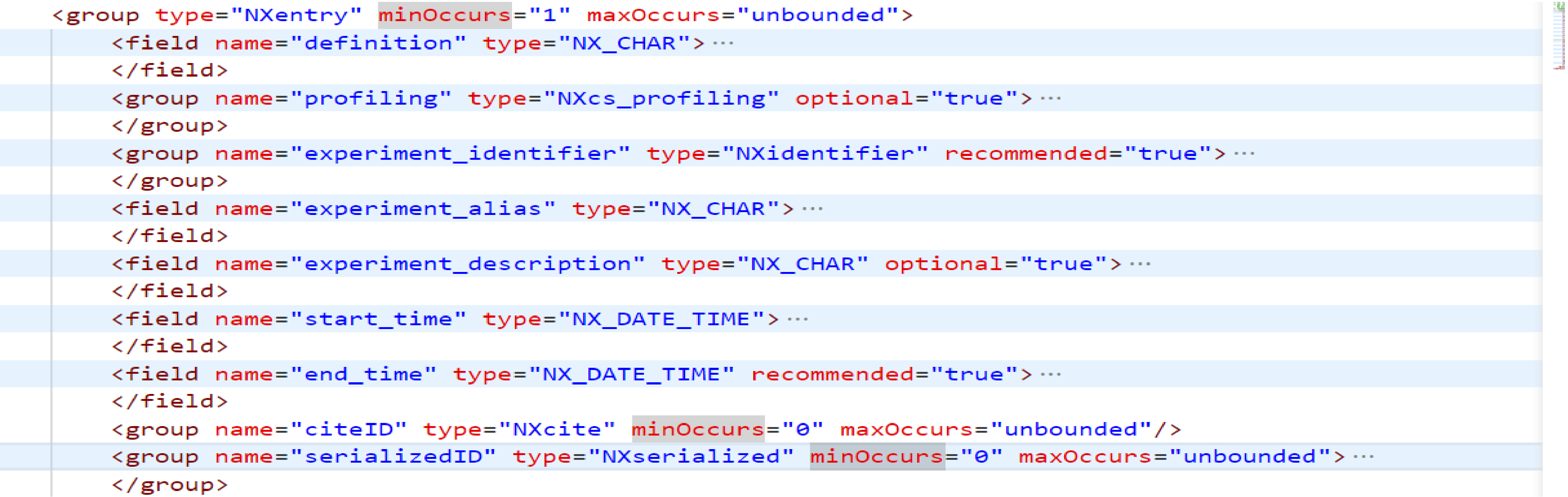
This application definition is thus an example of a general description with which to normalize specific pieces of information and data collected within electron microscopy research.

This application definition is also a blueprint which shows how users can build specific application definitions by reusing em-specific base classes - and thus represent electron-microscopy-specific content.

NXem

```
<?xml version='1.0' encoding='UTF-8'?>
<?xml-stylesheet type="text/xsl" href="nxdlformat.xsl"?>

<definition xmlns="http://definition.nexusformat.org/nxdl/3.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" category="application" type="group" name="NXem"
extends="NXobject" xsi:schemaLocation="http://definition.nexusformat.org/nxdl/3.1 ../nxdl.xsd">
```

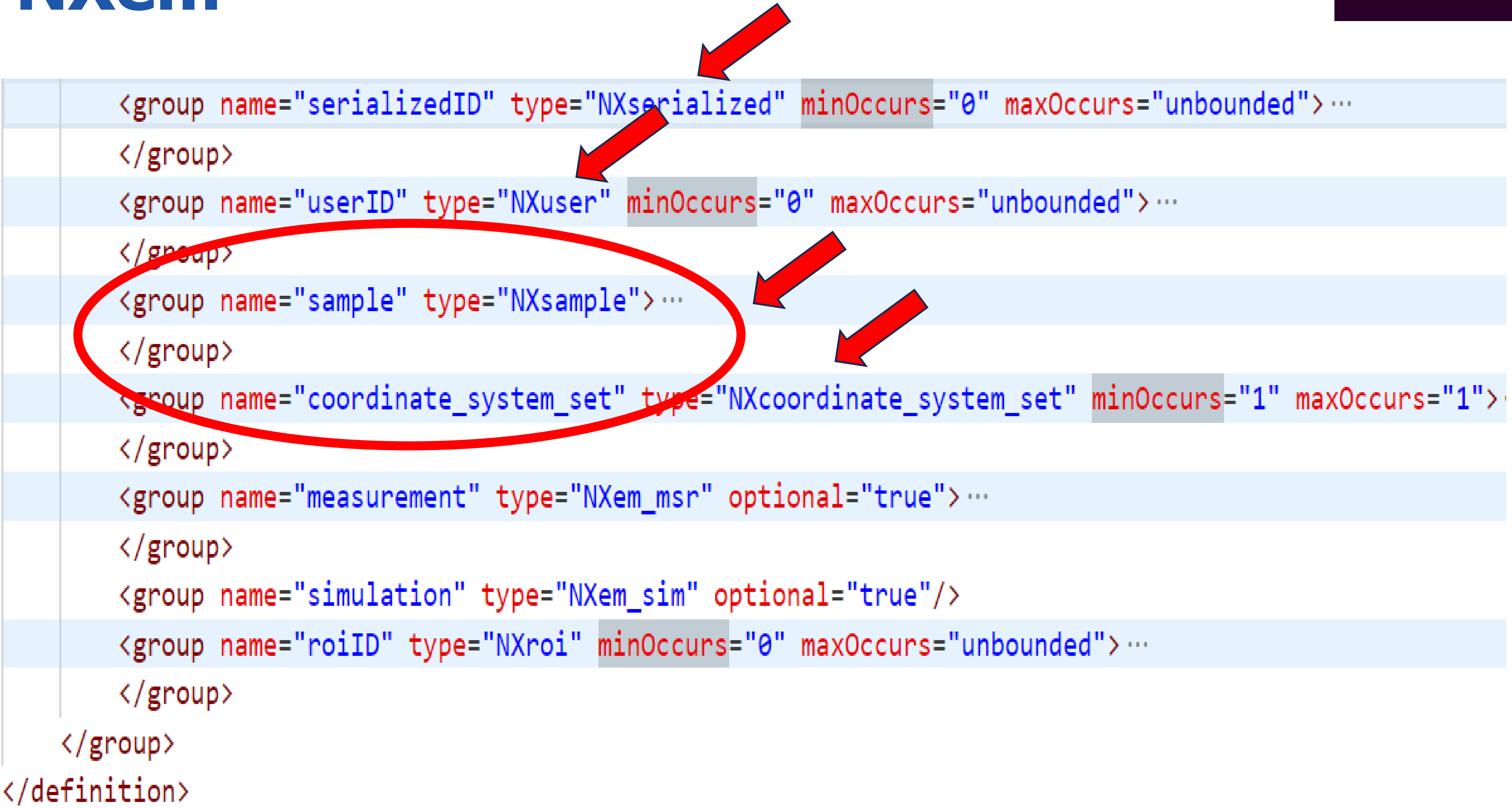


```
 <group type="NXentry" minOccurs="1" maxOccurs="unbounded">
     <field name="definition" type="NX_CHAR">···
     </field>
     <group name="profiling" type="NXcs_profiling" optional="true">···
     </group>
     <group name="experiment_identifier" type="NXidentifier" recommended="true">···
     </group>
     <field name="experiment_alias" type="NX_CHAR">···
     </field>
     <field name="experiment_description" type="NX_CHAR" optional="true">···
     </field>
     <field name="start_time" type="NX_DATE_TIME">···
     </field>
     <field name="end_time" type="NX_DATE_TIME" recommended="true">···
     </field>
     <group name="citeID" type="NXcite" minOccurs="0" maxOccurs="unbounded"/>
     <group name="serializedID" type="NXserialized" minOccurs="0" maxOccurs="unbounded">···
     </group>
```

NXem

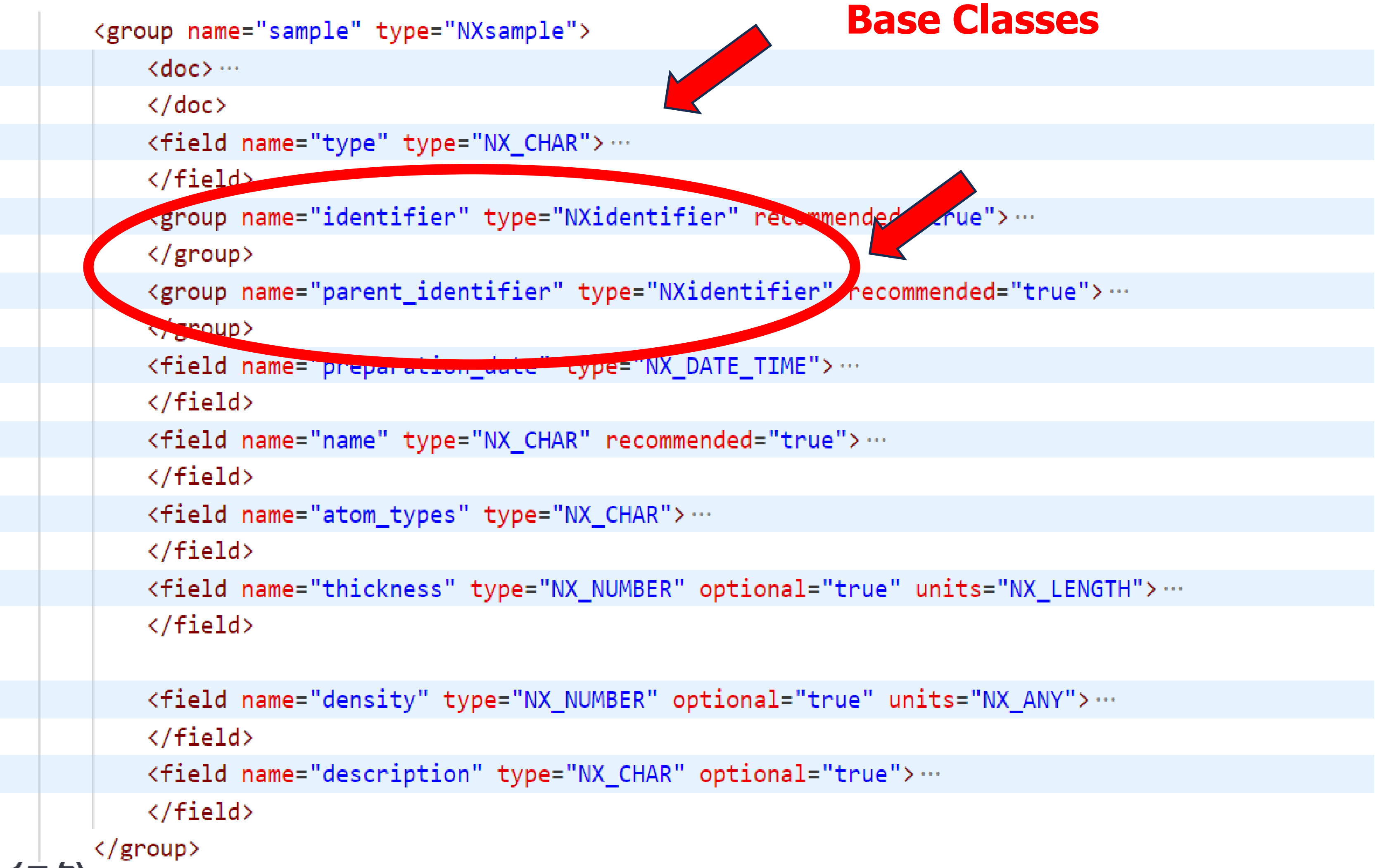
Base Classes

NeXus



```
        <group name="serializedID" type="NXserialized" minOccurs="0" maxOccurs="unbounded">⋯
        </group>
        <group name="userID" type="NXuser" minOccurs="0" maxOccurs="unbounded">⋯
        </group>
        <group name="sample" type="NXsample">⋯
        </group>
        <group name="coordinate_system_set" type="NXcoordinate_system_set" minOccurs="1" maxOccurs="1">
        </group>
        <group name="measurement" type="NXem_msr" optional="true">⋯
        </group>
        <group name="simulation" type="NXem_sim" optional="true"/>
        <group name="roiID" type="NXroi" minOccurs="0" maxOccurs="unbounded">⋯
        </group>
    </group>
</definition>
```

NXem

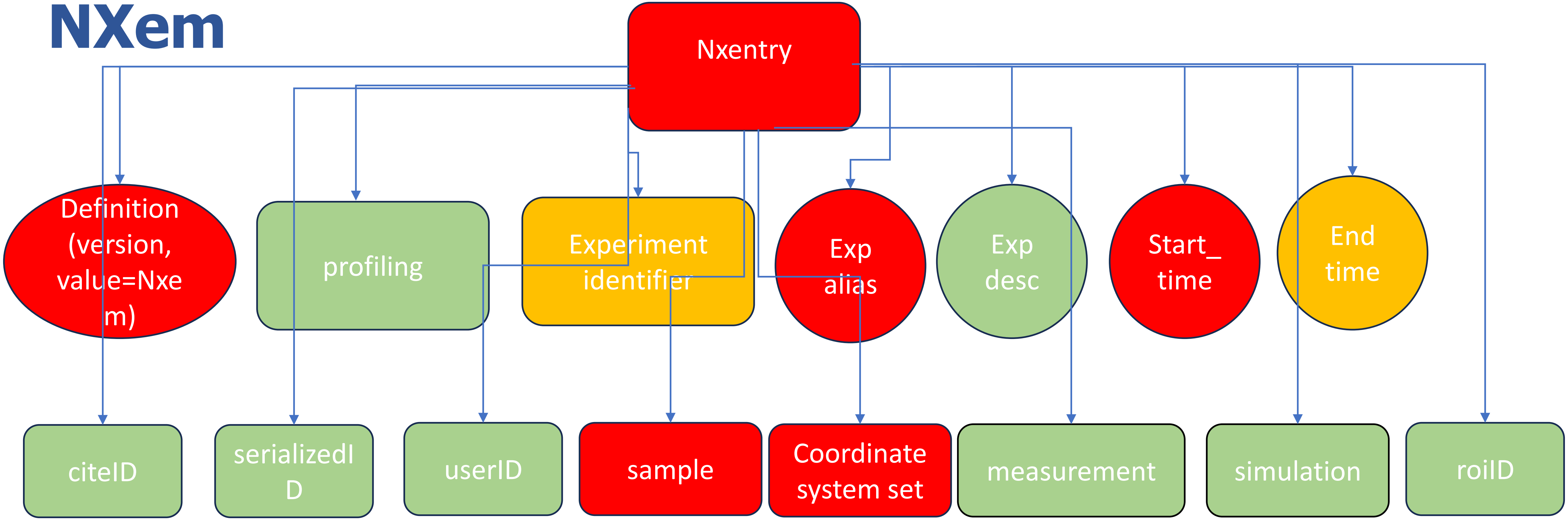


Base Classes

```
<group name="sample" type="NXsample">
    <doc>⋯
    </doc>
    <field name="type" type="NX_CHAR">⋯
    </field>
    <group name="identifier" type="NXidentifier" recommended="true">⋯
    </group>
    <group name="parent_identifier" type="NXidentifier" recommended="true">⋯
    </group>
    <field name="preparation_date" type="NX_DATE_TIME">⋯
    </field>
    <field name="name" type="NX_CHAR" recommended="true">⋯
    </field>
    <field name="atom_types" type="NX_CHAR">⋯
    </field>
    <field name="thickness" type="NX_NUMBER" optional="true" units="NX_LENGTH">⋯
    </field>

    <field name="density" type="NX_NUMBER" optional="true" units="NX_ANY">⋯
    </field>
    <field name="description" type="NX_CHAR" optional="true">⋯
    </field>
</group>
```

NXem



Nxentry

Definition (version, value=Nxem)

profiling

Experiment identifier

Exp alias

Exp desc

Start_ time

End time

citeID

serializedID

userID

sample

Coordinate system set

measurement

simulation

roiID

🟩 optional

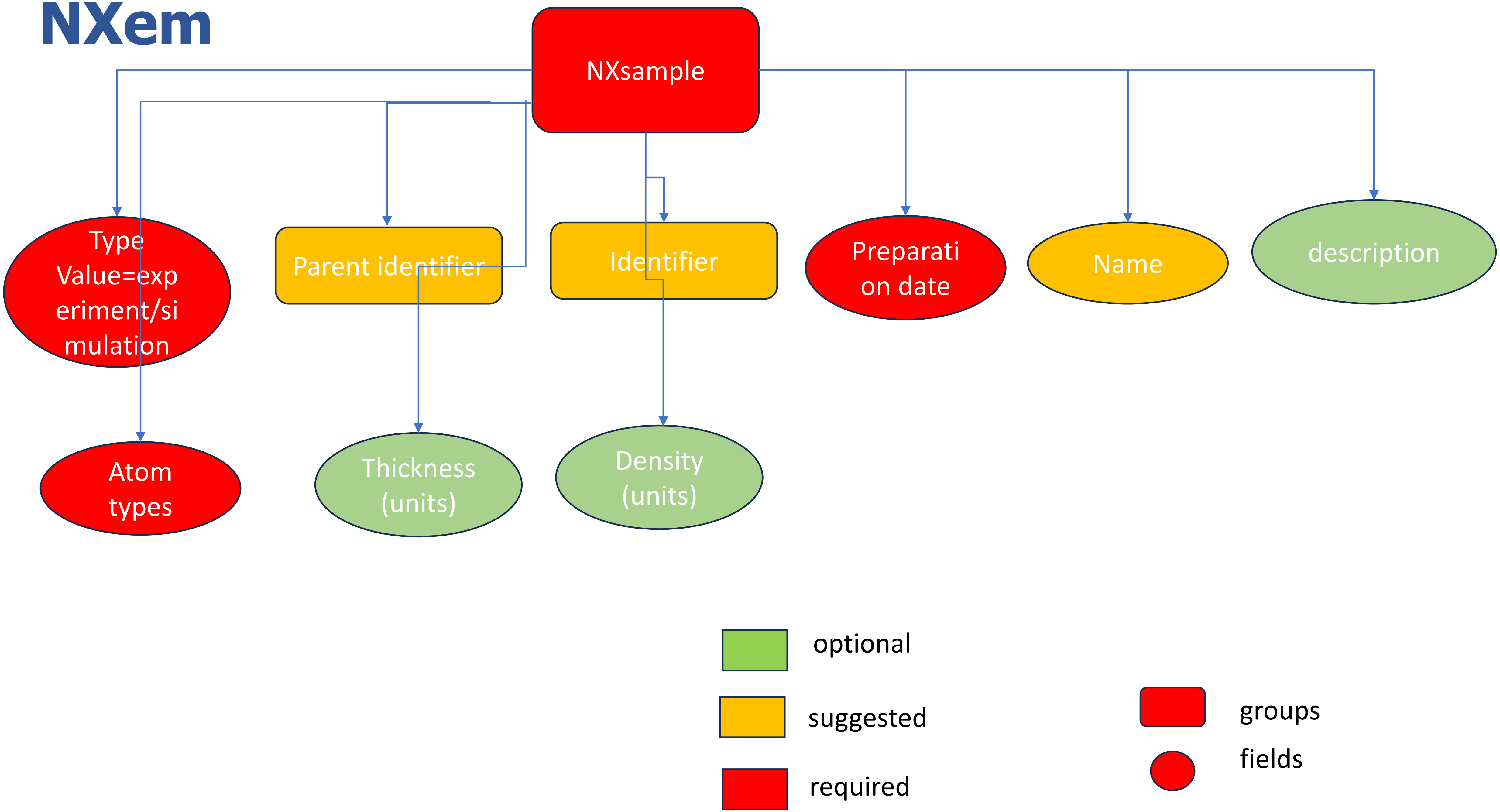
🟨 suggested

🟥 required

🟥 groups

● fields

NXem



NXsample

Type Value=experiment/simulation

Parent identifier

Identifier

Preparation date

Name

description

Atom types

Thickness (units)

Density (units)

🟩 optional

🟨 suggested

🟥 required

▬ groups

● fields

NeXus: python libraries

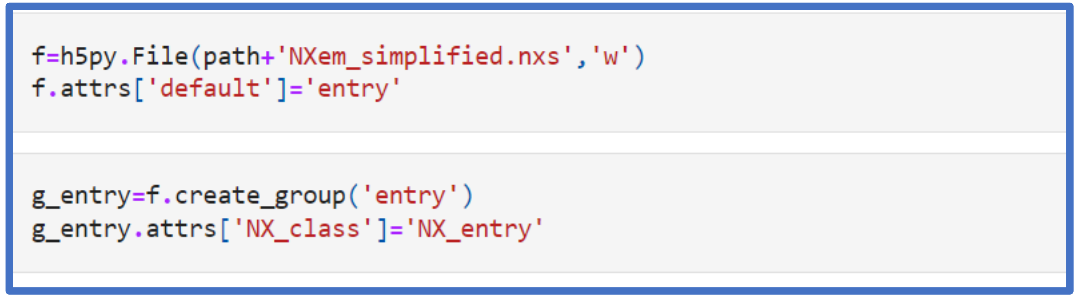
- Nxarray: this library take arrays as inputs and returns nexus

- Pynxtool: library that extends NeXus for experiments and characterization in Materials Science and Materials Engineering and serve as a NOMAD parser implementation for NeXus.

- Nexus-format: This package provides a Python API to open, create, and manipulate NeXus data written in the HDF5 format. The 'nexusformat' package provides the underlying API for NeXpy, which provides a GUI interface for visualizing and analyzing NeXus data.

- Python-nexus: python-nexus provides simple nexus file-format reading/writing tools, and a small collection of nexus manipulation scripts.

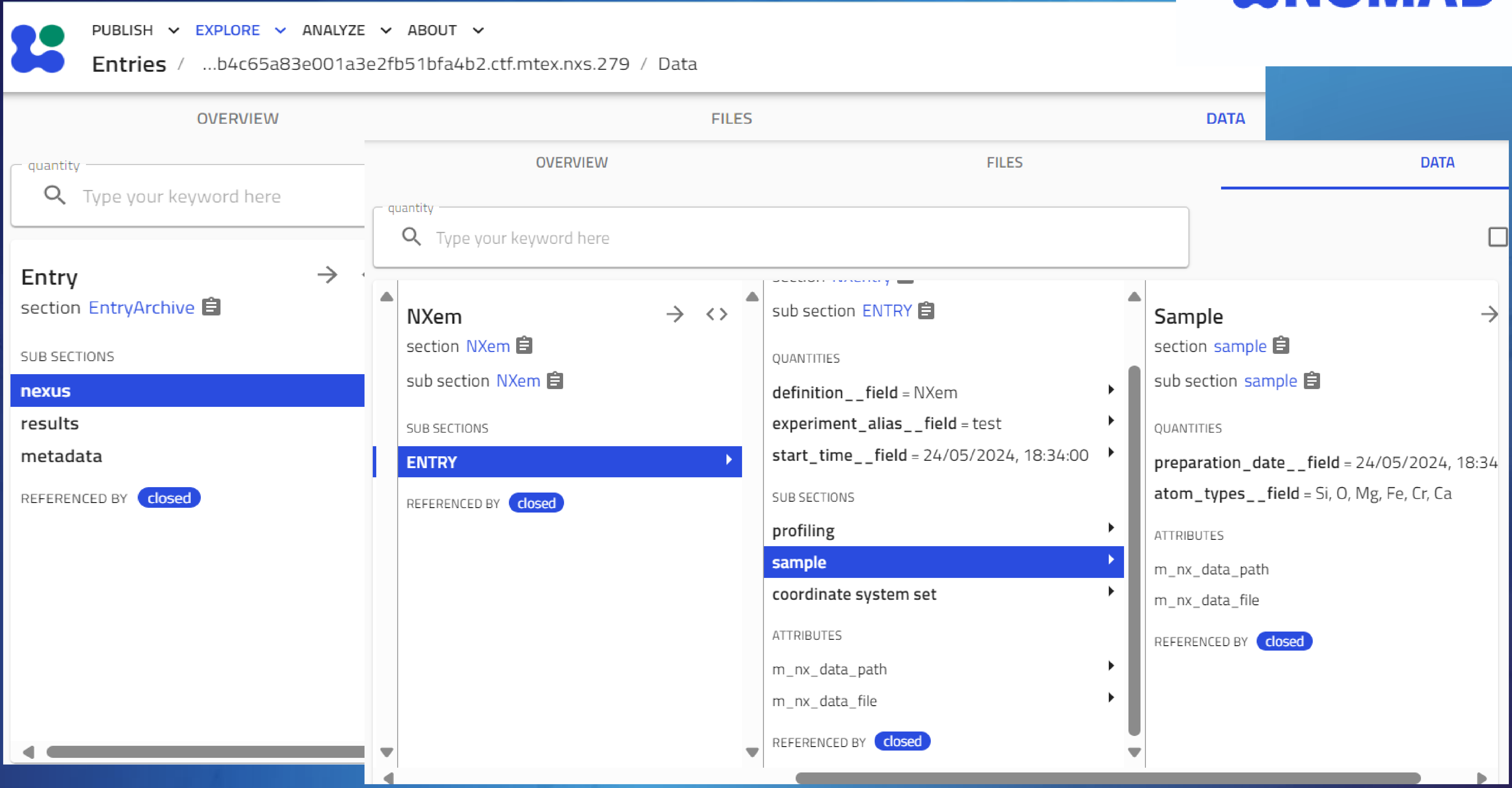
Own scripts



```
f=h5py.File(path+'NXem_simplified.nxs','w')
f.attrs['default']='entry'
```

```
g_entry=f.create_group('entry')
g_entry.attrs['NX_class']='NX_entry'
```

NOMAD



PUBLISH EXPLORE ANALYZE ABOUT

Entries / ...b4c65a83e001a3e2fb51bfa4b2.ctf.mtex.nxs.279 / Data

OVERVIEW FILES DATA

quantity: Type your keyword here

Entry

section EntryArchive

SUB SECTIONS

- nexus
- results
- metadata

REFERENCED BY closed

OVERVIEW FILES DATA

quantity: Type your keyword here

NXem

section NXem

sub section NXem

SUB SECTIONS

- ENTRY

REFERENCED BY closed

sub section ENTRY

QUANTITIES

- definition__field = NXem
- experiment_alias__field = test
- start_time__field = 24/05/2024, 18:34:00

SUB SECTIONS

- profiling
- sample
- coordinate system set

ATTRIBUTES

- m_nx_data_path
- m_nx_data_file

REFERENCED BY closed

Sample

section sample

sub section sample

QUANTITIES

- preparation_date__field = 24/05/2024, 18:34
- atom_types__field = Si, O, Mg, Fe, Cr, Ca

ATTRIBUTES

- m_nx_data_path
- m_nx_data_file

REFERENCED BY closed

FINAL REMARKS

- ❑ To be FAIR compliant we need choosing a standard
- ❑ NeXus file format is a reasonable choice because it has behind a huge active community
- ❑ Nxem may describe the majority of EM experiments

Thanks for your attention!

AREA SCIENCE PARK