

FAIRmat Tutorial 8:

Using NOMAD as an Electronic lab notebook (ELN) for FAIR data

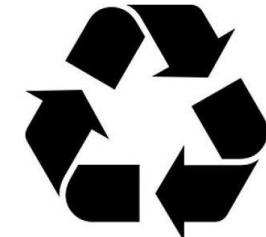
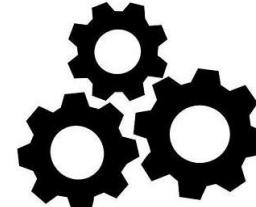
Organized by FAIRmat Area A Synthesis



FAIRmat Area A: Synthesis

Why Structured Data?

- the structure highlights relationships between data points
(semantic **interoperability**)
- machine-readable**
- ideal for classification, regression, and clustering with **AI**
- searchable**
- shareable** (using a community agreed structure)
- leads to data-driven decisions



Research Data Management

Raw Data

ELN

Data Analysis

Research Data Management

Raw Data

ELN

Data Analysis

- Log files from instruments
- Recipe files from process software
- Spreadsheet files



https://de.freepik.com/vektoren-kostenlos/illustration-des-biedienfeldkonzepts_13662974.htm

Research Data Management

Raw Data

- Log files from instruments
- Recipe files from process software
- Spreadsheet files



https://de.freepik.com/vektoren-kostenlos/illustration-des-biedienfeldkonzepts_13662974.htm

ELN

- Manually entered data and metadata



Data Analysis

Research Data Management

Raw Data

- Log files from instruments
- Recipe files from process software
- Spreadsheet files



https://de.freepik.com/vektoren-kostenlos/illustration-des-bedenfeldkonzepts_13662974.htm

ELN

- Manually entered data and metadata



https://de.freepik.com/vektoren-kostenlos/laptop-mit-bildungssymbol-isoliert_11691038.htm

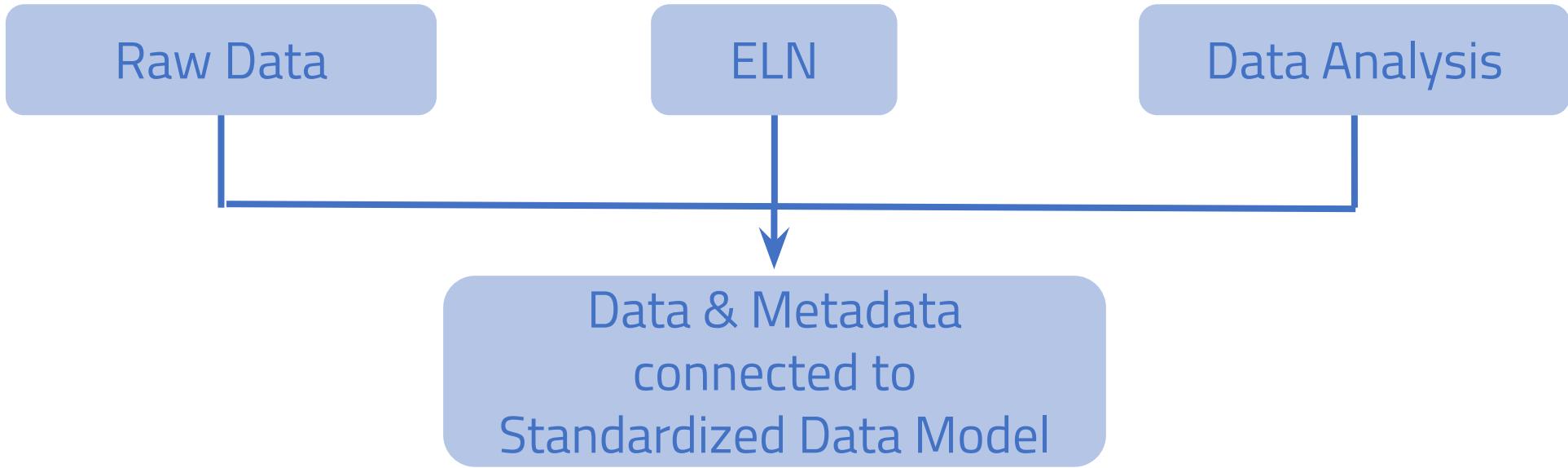
Data Analysis

- Post processing software
- User-tailored scripts

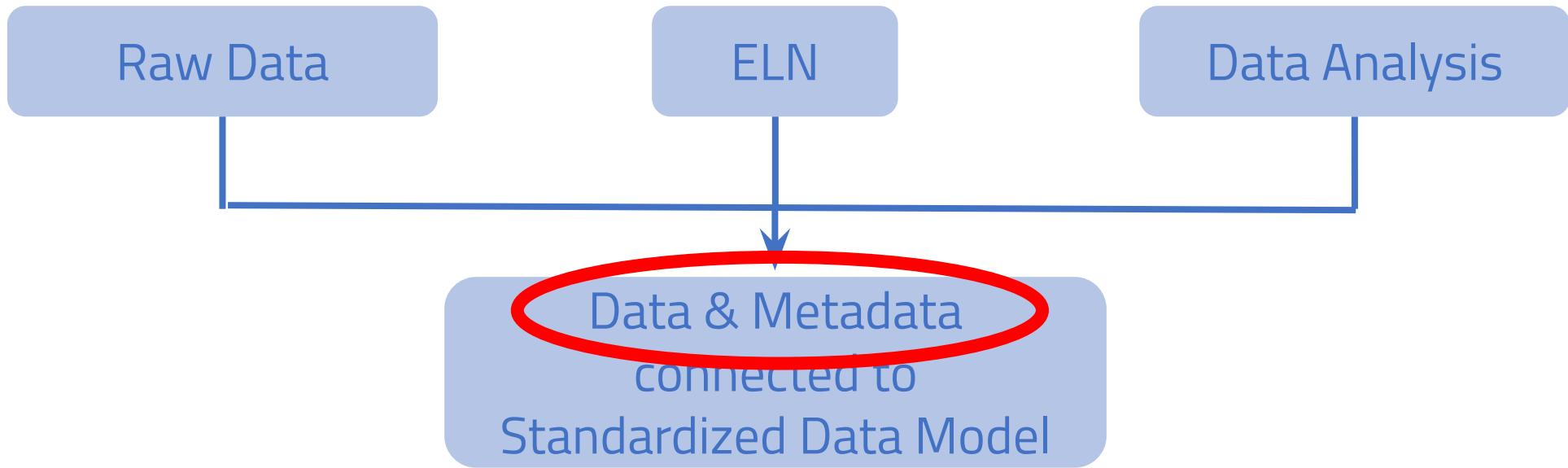


https://de.freepik.com/vektoren-kostenlos/site-statistik-konzeptillustration_7140739.htm

Research Data Management

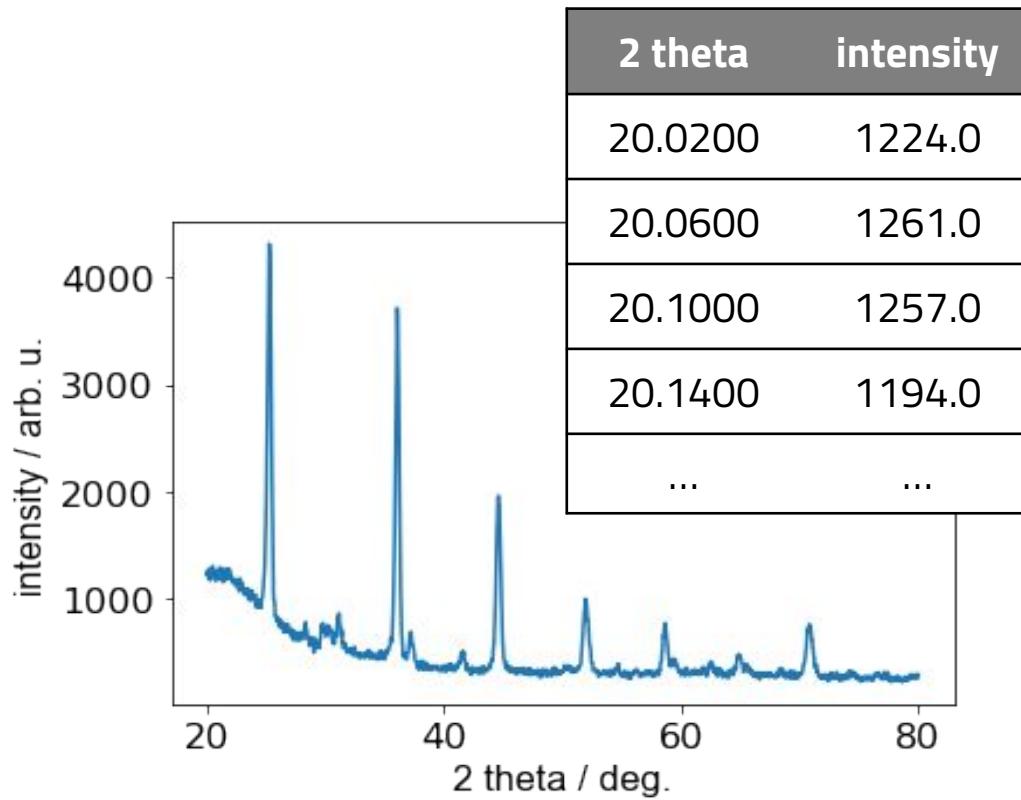


Research Data Management



Data & Metadata

Data:
actual content of information



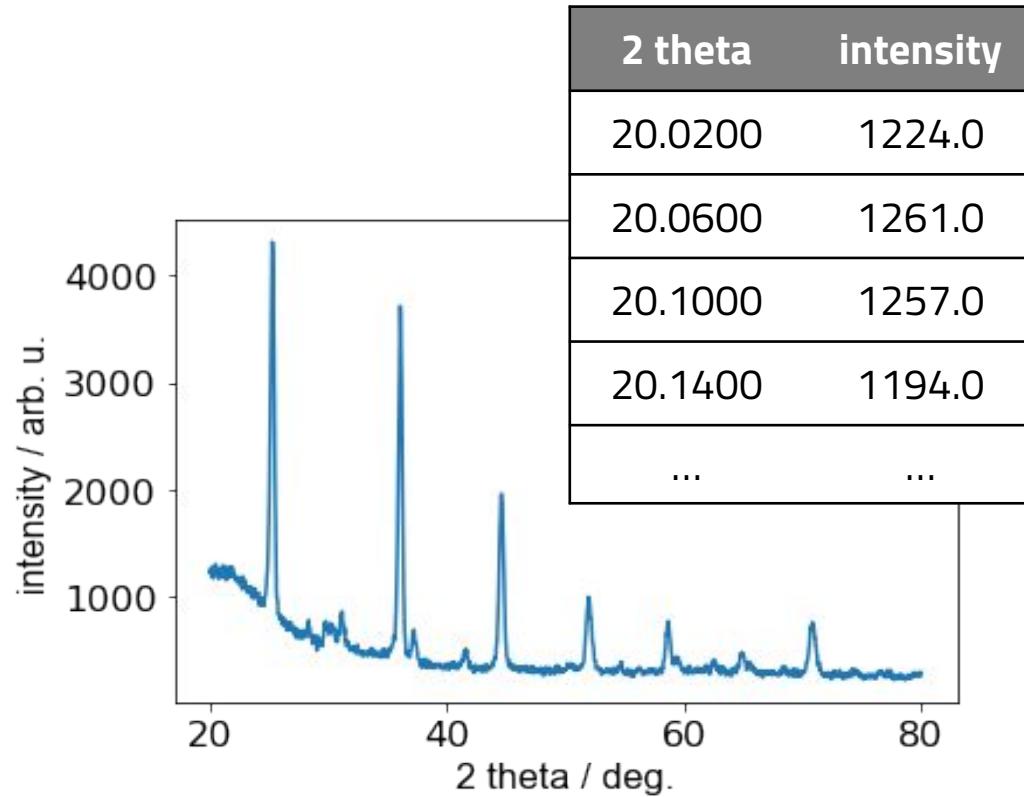
Data & Metadata

Data:

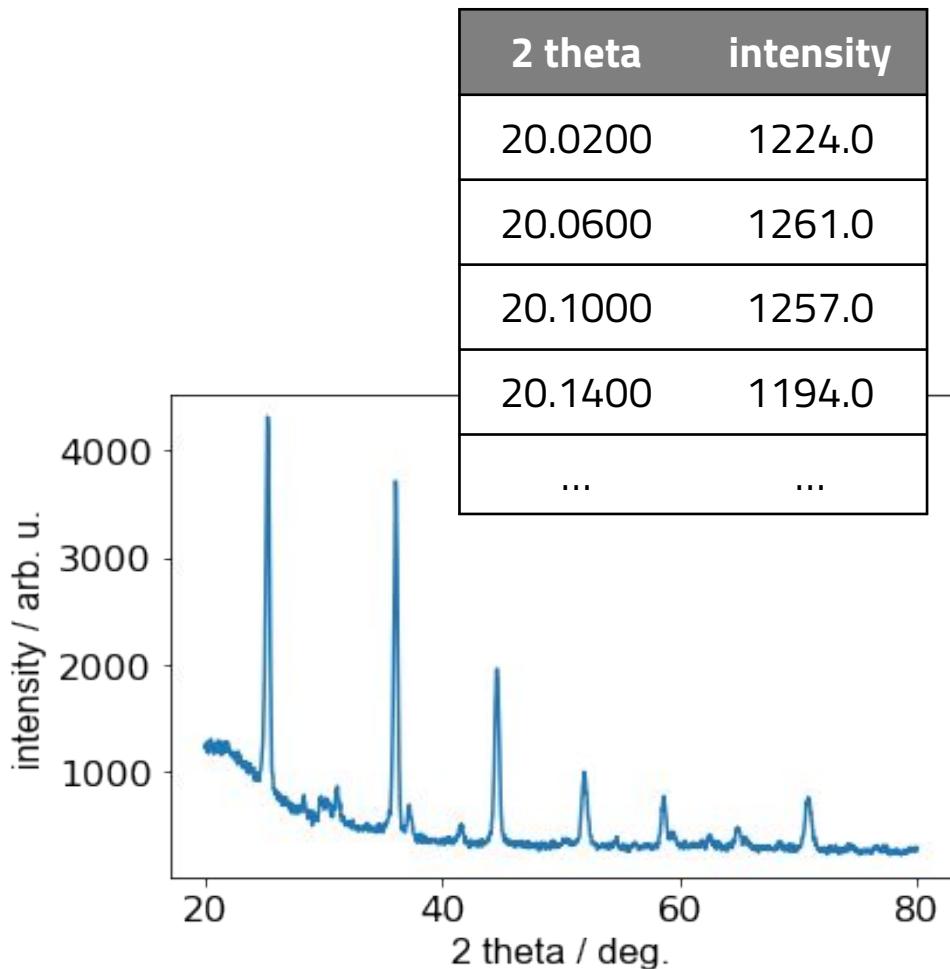
actual content of information

Metadata:

data that provides information about other data



Data & Metadata



Data:

actual content of information

Structural Metadata:

provides information about
containers of data

XRD Measurement:

2 theta:

description: The 2-theta angle of the diffractogram.

unit: °

type: float

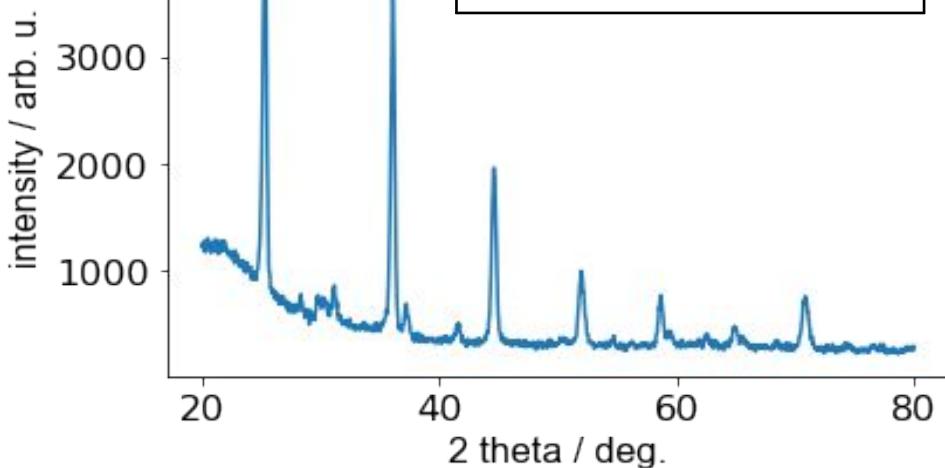
intensity:

description: The count at each 2-theta value.

unit: dimensionless

type: float

Data & Metadata



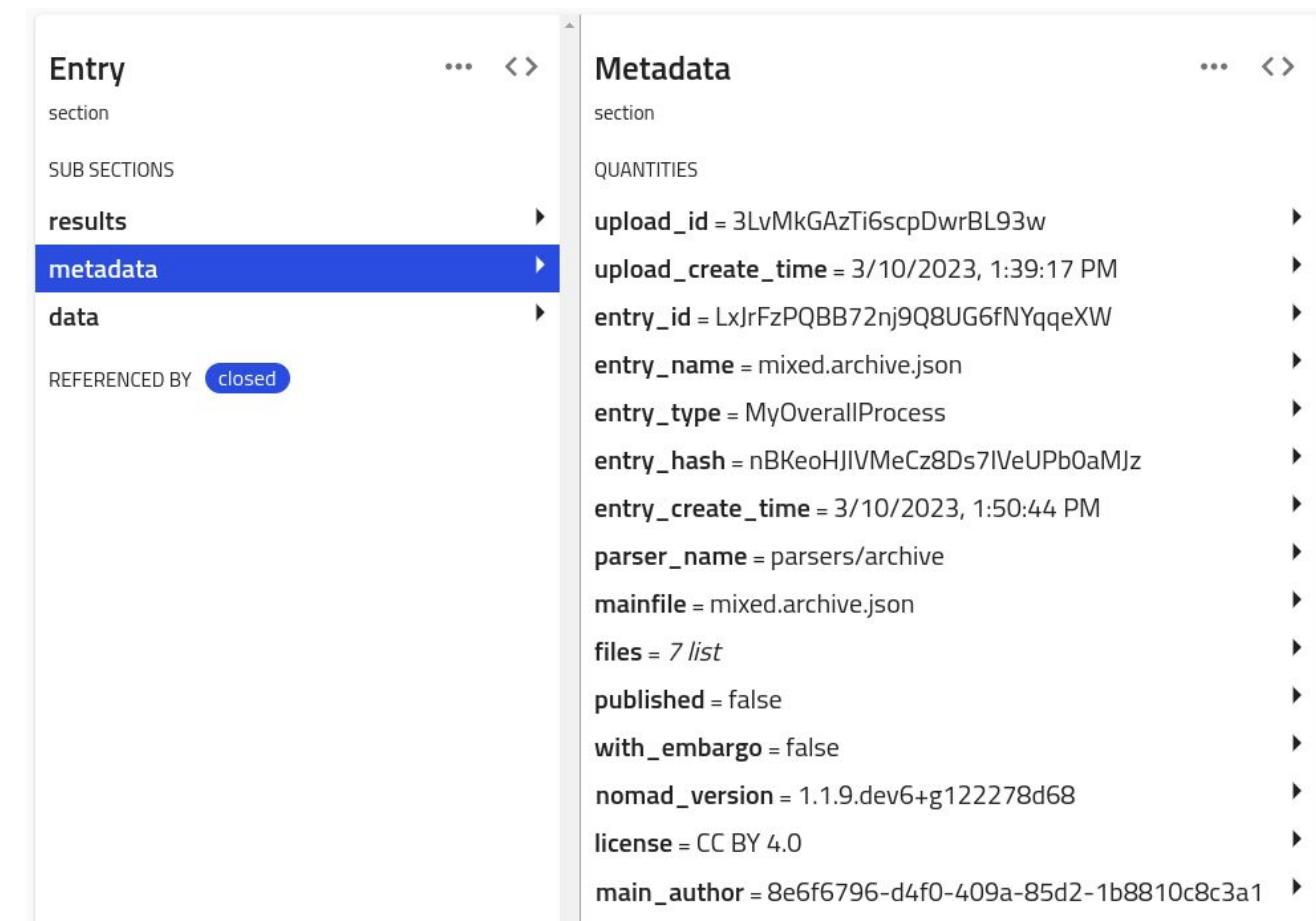
Data:

actual content of information

2 theta	intensity
20.0200	1224.0
20.0600	1261.0
20.1000	1257.0
20.1400	1194.0
...	...

Structural Metadata:

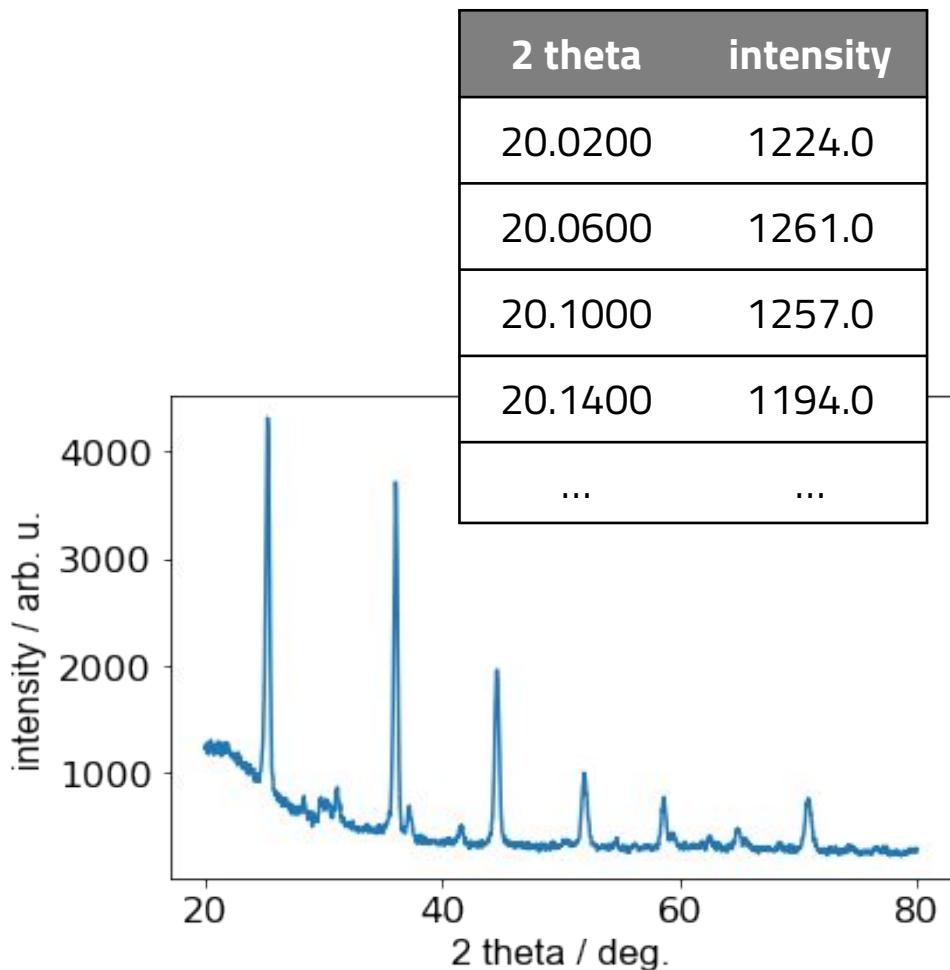
provides information about containers of data



Data & Metadata

Data:

actual content of information



Descriptive Metadata:

provides important context
about data

Date of the experiment?

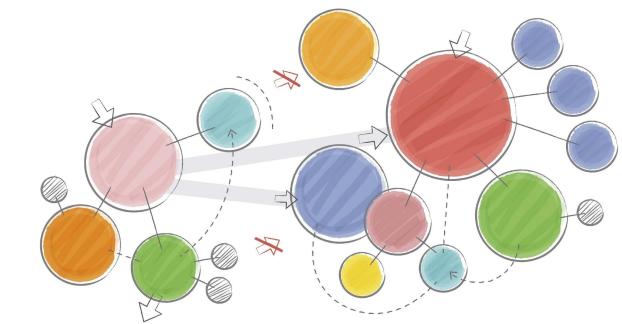
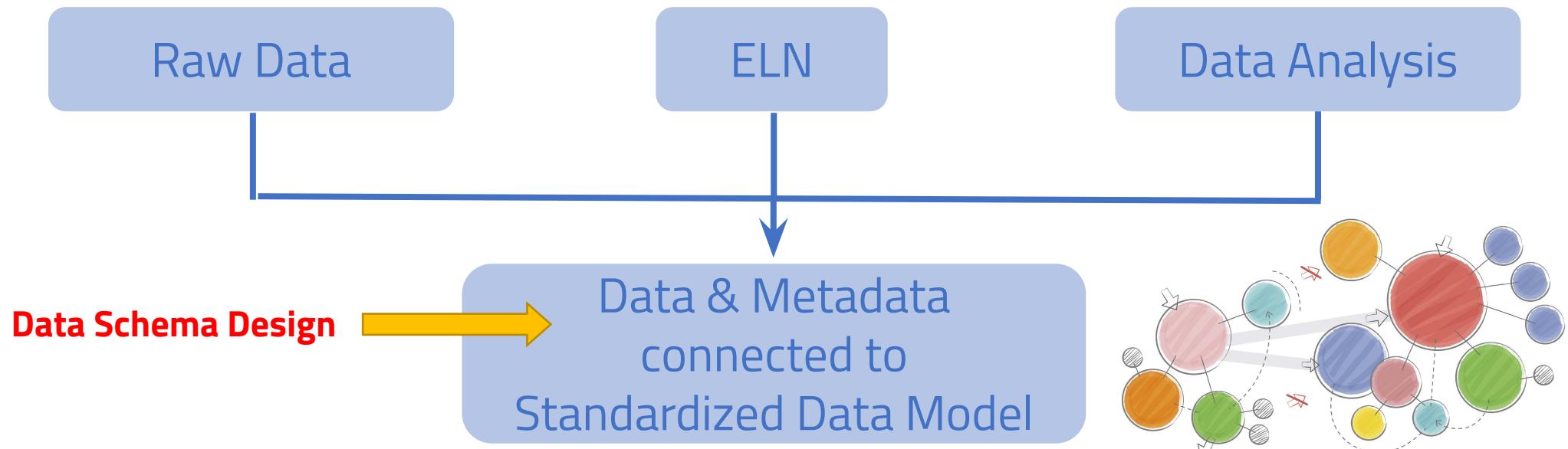
Temperature?

Instrument specifications?

Specimen specifications?

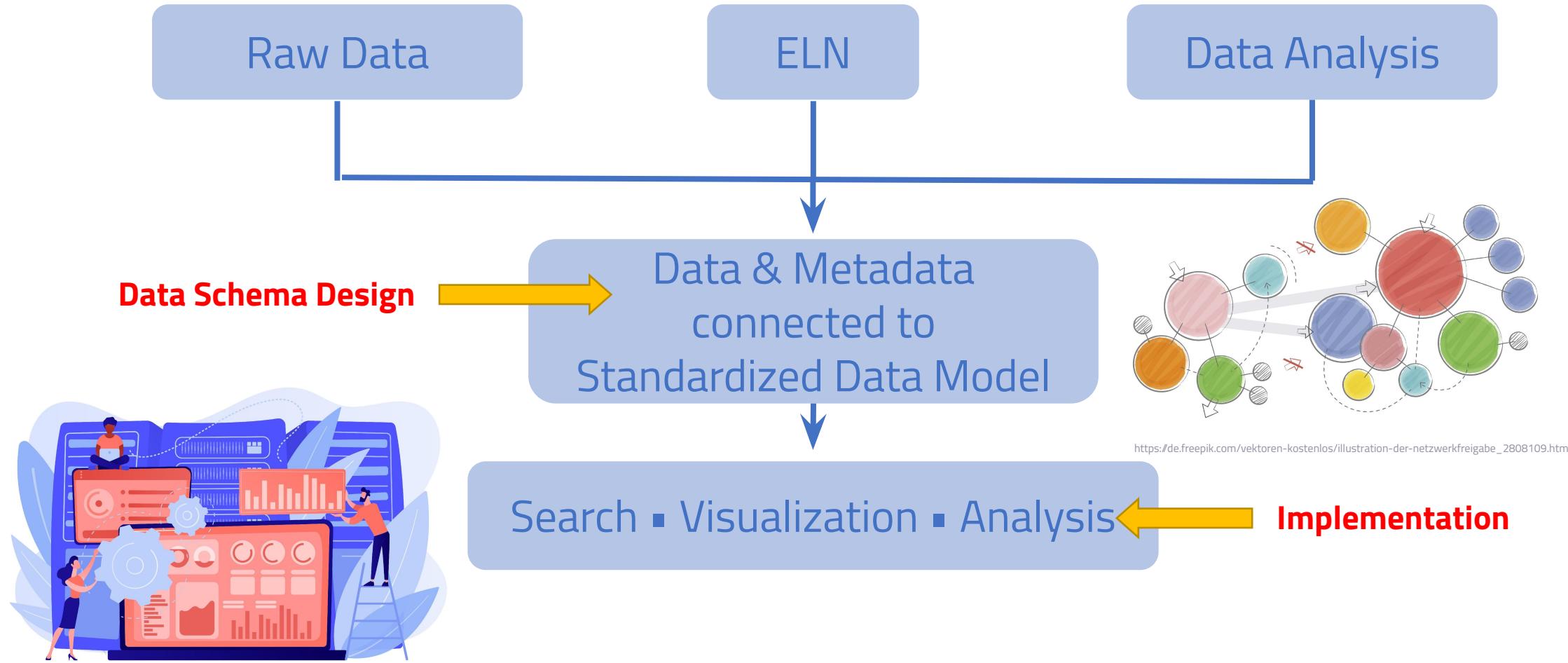
Humidity in the room?

Research Data Management



https://de.freepik.com/vektoren-kostenlos/illustration-der-netzwerkfreigabe_2808109.htm

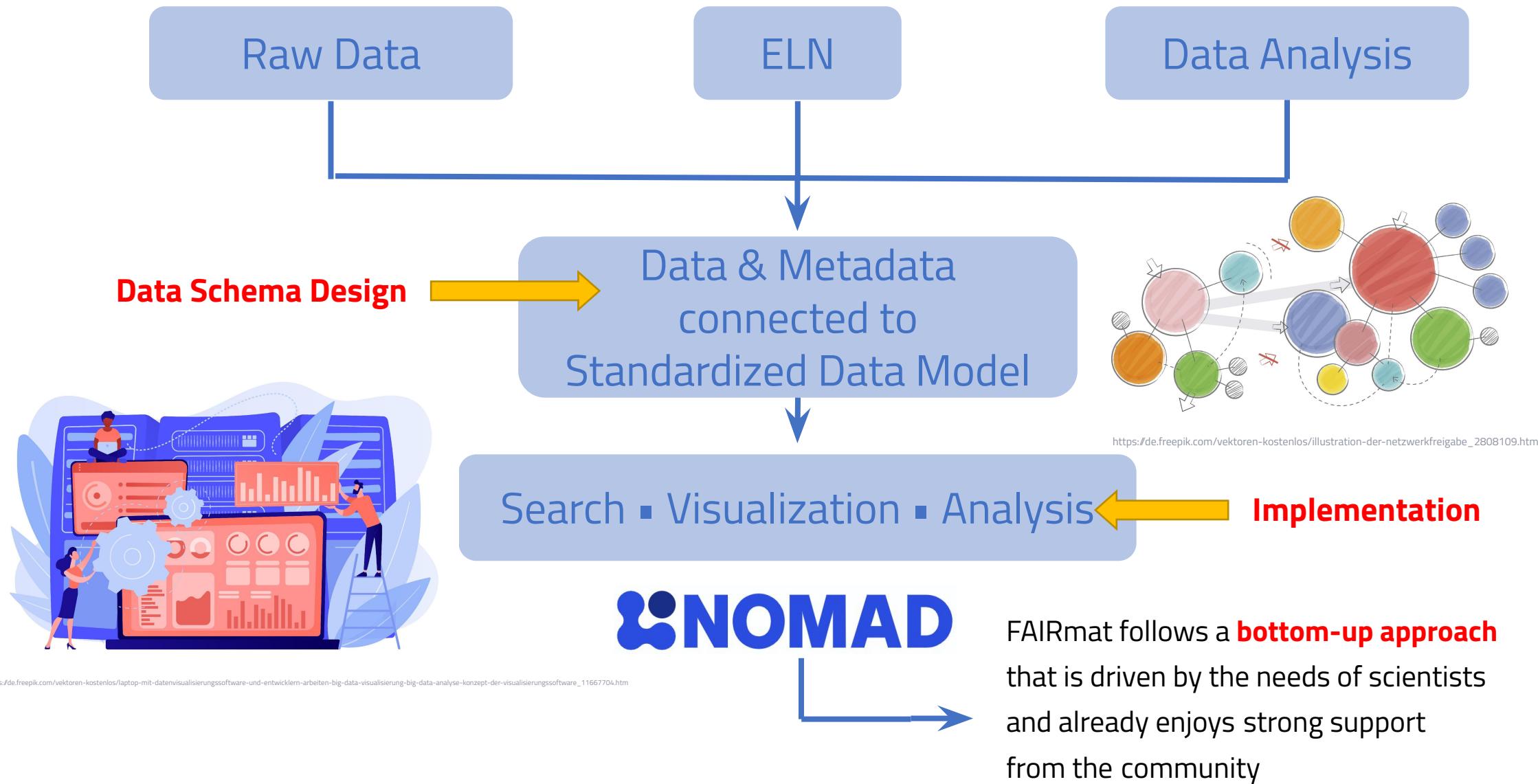
Research Data Management



https://de.freepik.com/vektoren-kostenlos/illustration-der-netzwerkfreigabe_2808109.htm

https://de.freepik.com/vektoren-kostenlos/laptop-mit-datenvisualisierungssoftware-und-entwickeln-arbeiten-big-data-visualisierung-big-data-analyse-konzept-der-visualisierungssoftware_11667704.htm

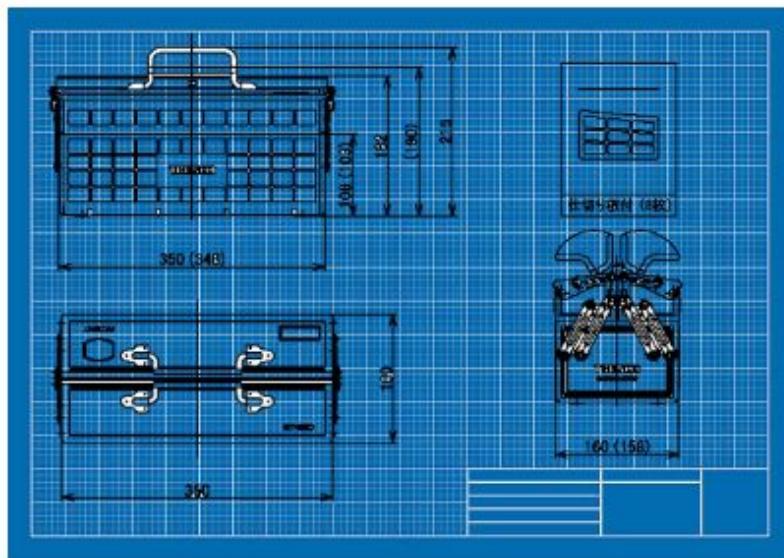
Research Data Management



Schema and Template Concepts

Schema:

A formal description of data, data types, and data file structures, such as XML files.



The blueprint of a toolbox

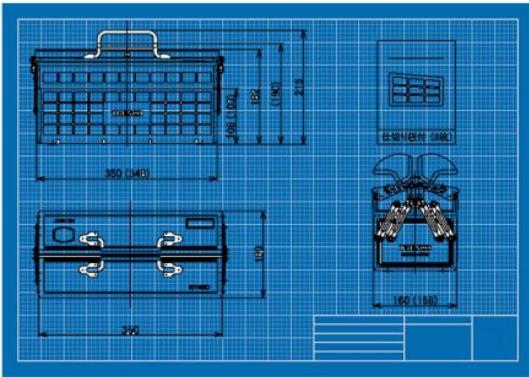
Template:

A physical object from which other objects are based or derived.



A toolbox made for a specific set of tools

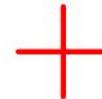
Schema and Template Concepts



Schema



Template



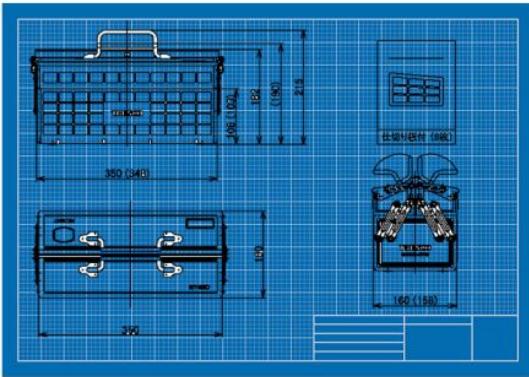
Structured Archive File



Data



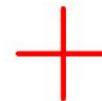
Schema and Template Concepts



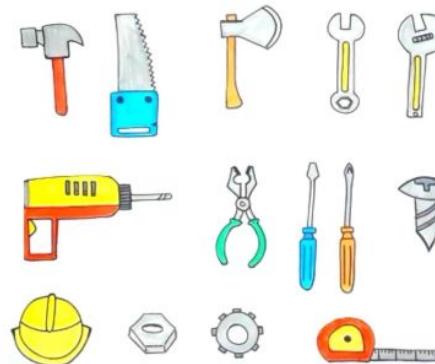
Schema



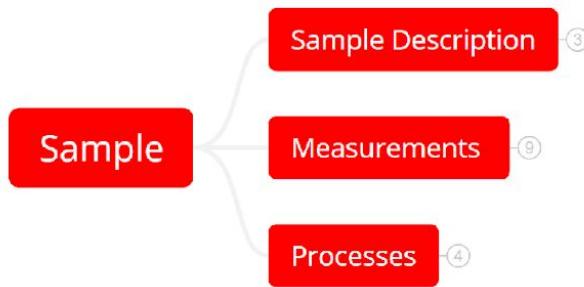
Template



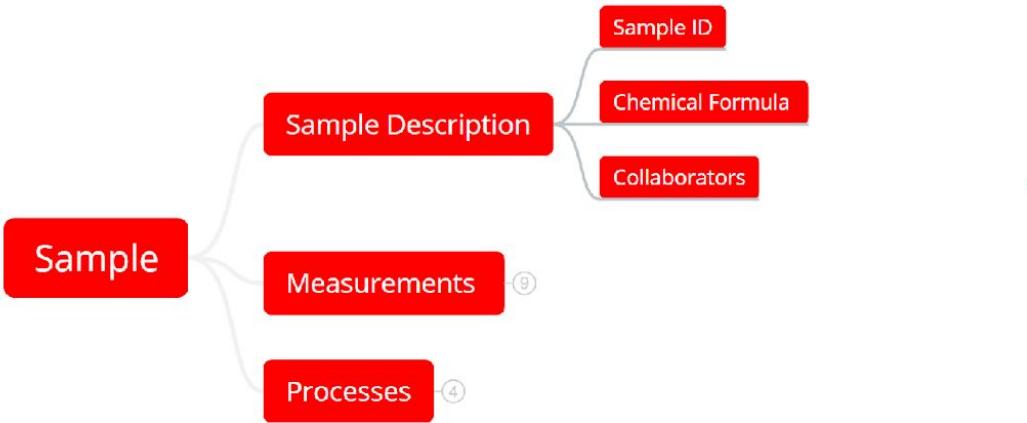
Structured Archive File



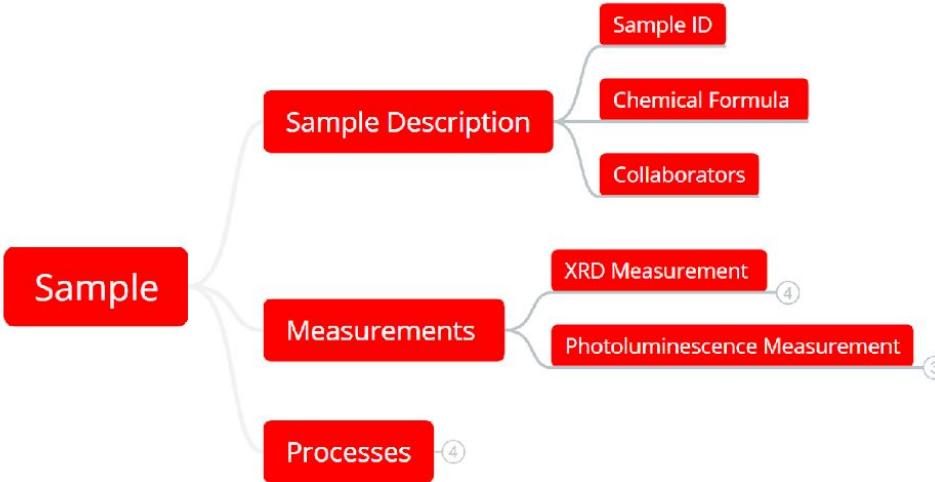
Data



Example of file formats: **XML, JSON, YAML, or HDF5**

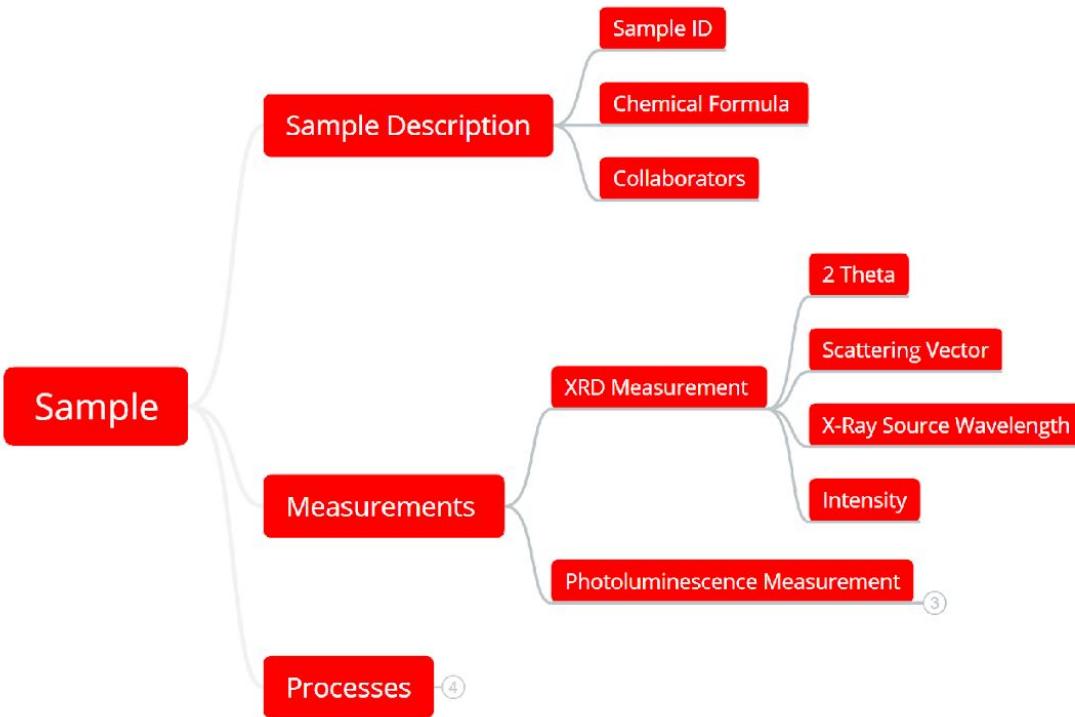


Example of file formats: **XML, JSON, YAML, or HDF5**



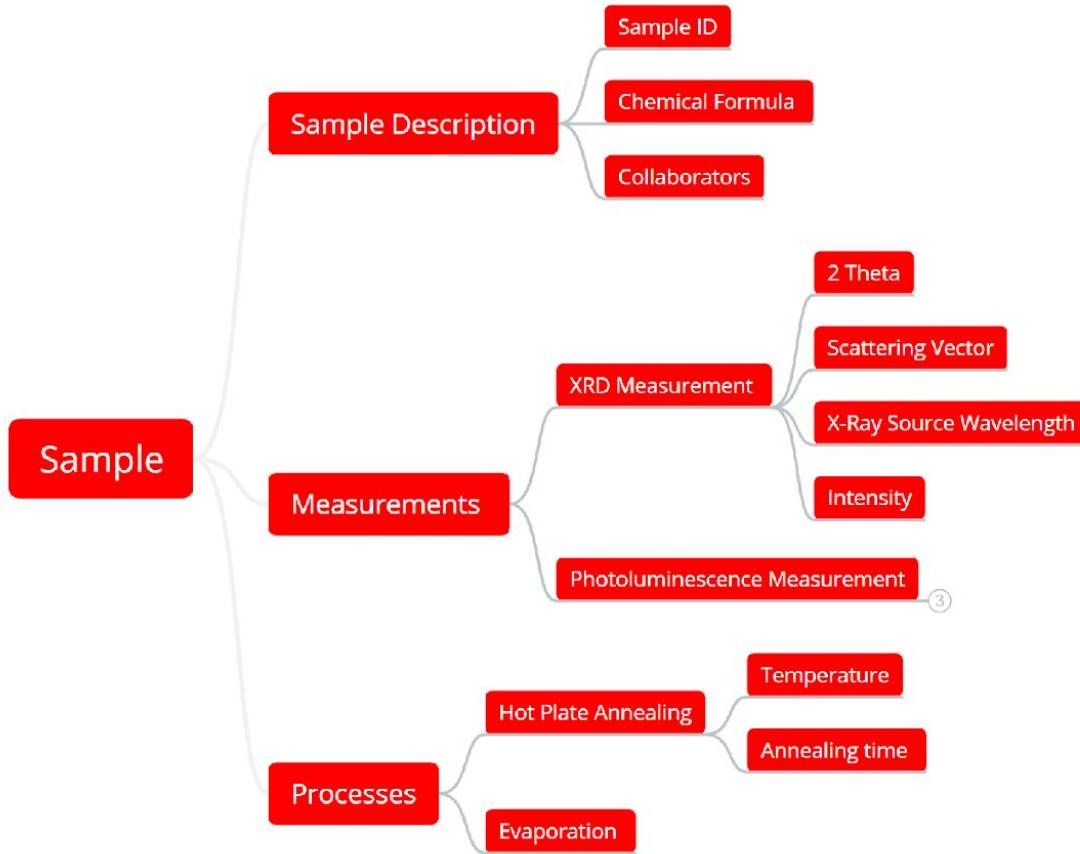
Example of file formats: **XML, JSON, YAML, or HDF5**

Hierarchical organization



Example of file formats: **XML, JSON, YAML, or HDF5**

Hierarchical organization



Example of file formats: **XML, JSON, YAML, or HDF5**

YAML files to describe data structures

It is a Markup Language defining hierarchy with indentation

```
definitions:  
  name: My Custom Schemas  
  sections:  
    MyProcess:  
      base_sections:  
        - nomad.datamodel.metainfo.eln.BasicEln  
      quantities:  
        data_file:  
          type: str  
        m_annotations:  
          browser:  
            adaptor: RawFileAdaptor  
          eln:  
            component: FileEditQuantity  
      sub_sections:  
        MyProcessesCollection:  
          section:  
            quantities:  
              sample_id:  
                type: str  
              m_annotations:  
                eln:  
                  component: StringEditQuantity
```

- extensible
- structured
- plain text
- human readable

```
definitions:  
  name: My Custom Schemas  
  sections:  
    MyProcess:  
      base_sections:  
        - nomad.datamodel.metainfo.eln.B  
      quantities:  
        data_file:  
          type: str  
        m_annotations:  
          browser:  
            adaptor: RawFileAdaptor  
          eln:  
            component: FileEditQuantity  
      sub_sections:  
        MyProcessesCollection:  
          section:  
            quantities:  
              sample_id:  
                type: str  
              m_annotations:  
                eln:  
                  component: StringEditQuantity  
              roughness:  
                type: np.float64  
                unit: nm  
              m_annotations:  
                eln:  
                  component: NumberEditQuantity  
                  defaultDisplayUnit: nm
```

The screenshot shows the NOMAD interface with the following components:

- Top Bar:** PUBLISH, EXPLORE, ANALYZE, ABOUT.
- Left Sidebar:** Your uploads / Upload / Entry / Data
- Central Area:** OVERVIEW and FILES tabs. The FILES tab is active, showing a tree structure of data entries.
 - Entry:** A root entry with sections: section, SUB SECTIONS, results, metadata, and data (which is currently selected).
 - MyProcess:** A section under Entry, containing data_file (process_data_row.csv) and SUB SECTIONS (MyProcessesCollection).
 - MyProcessesCollection:** A section under MyProcess, containing entries 0, 1, 2, and aa (which is currently selected).
- Right Area:** A detailed view of the aa entry under MyProcessesCollection. It shows:
 - Myprocessescollection:** section
 - QUANTITIES:**
 - Sample id: aa
 - Roughness: 24 (Unit: nm)
 - Thickness: 45 (Unit: nm)

```

definitions:
  name: My Custom Schemas
  sections:
    MyProcess:
      base_sections:
        - nomad.datamodel.metainfo.eln.B
      quantities:
        data_file: <-->
          type: str
        m_annotations:
          browser:
            adaptor: RawFileAdaptor
          eln:
            component: FileEditQuantity
      sub_sections:
        MyProcessesCollection:
          section:
            quantities:
              sample_id:
                type: str
              m_annotations:
                eln:
                  component: StringEditQuantity
            roughness:
              type: np.float64
              unit: nm
              m_annotations:
                eln:
                  component: NumberEditQuantity
                  defaultDisplayUnit: nm

```

The screenshot shows the NOMAD interface with the following structure:

- Entry**: Contains fields: section, SUB SECT, results, metadata, and data.
- MyProcess**: Contains fields: section, QUANTITIES (data_file: process_data_row.csv), and SUB SECTIONS (MyProcessesCollection).
- MyProcessesCollection**: Contains entries: 0, 1, 2, and aa.
- aa**: Contains fields: Sample id (aa), Roughness (24), and Thickness (45).

Main elements:

- Quantities (data fields int, float, str, datetime)
- Attributes (type, shape, unit, annotations)



```

definitions:
  name: My Custom Schemas
  sections:
    MyProcess:
      base_sections:
        - nomad.datamodel.metainfo.sections.MyProcess
      quantities:
        data_file:
          type: str
        m_annotations:
          browser:
            adaptor: RawFileAdaptor
          eln:
            component: FileEditQuantity
      sub_sections:
        MyProcessesCollection:
          section:
            quantities:
              sample_id:
                type: str
              m_annotations:
                eln:
                  component: StringEditQuantity
            roughness:
              type: np.float64
              unit: nm
              m_annotations:
                eln:
                  component: NumberEditQuantity
                  defaultDisplayUnit: nm

```

Main elements:

- Quantities (data fields int, float, str, datetime)
- Attributes (type, shape, unit, annotations)
- Sections (or Classes, collections of Quantities)

```

definitions:
  name: My Custom Schemas
  sections:
    MyProcess:
      base_sections:
        - nomad.datamodel.metainfo.sections.MyProcess
      quantities:
        data_file:
          type: str
        m_annotations:
          browser:
            adaptor: RawFileAdaptor
          eln:
            component: FileEditQuantity
      sub_sections:
        MyProcessesCollection:
          section:
            quantities:
              sample_id:
                type: str
              m_annotations:
                eln:
                  component: StringEditQuantity
            roughness:
              type: np.float64
              unit: nm
              m_annotations:
                eln:
                  component: NumberEditQuantity
                  defaultDisplayUnit: nm

```

The screenshot shows the NOMAD interface with the following details:

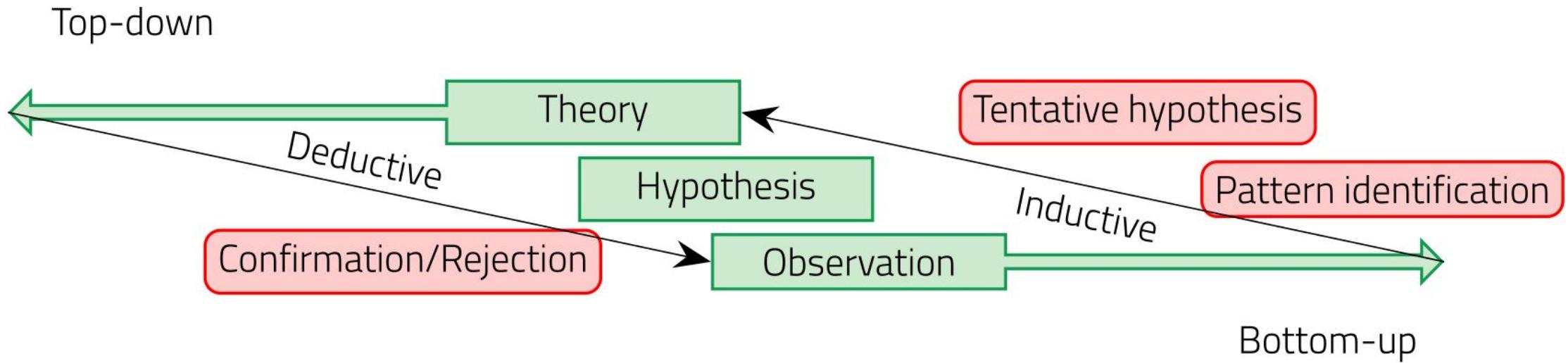
- OVERVIEW** tab is selected.
- Entry** section is expanded, showing:
 - SUB SECTIONS: results, metadata, data
 - data quantity: process_data_row.csv
 - REFERENCED BY: closed
- MyProcess** section is expanded, showing:
 - QUANTITIES: data_file (process_data_row.csv)
 - SUB SECTIONS: MyProcessesCollection
- MyProcessesCollection** section is expanded, showing:
 - 0, 1, 2, aa
 - REFERENCED BY: closed
- FILES** tab is visible on the right.
- DATA** tab is visible at the bottom right.

Several examples are already public:

[github.com/FAIRmat-NFDI/AreaA-data modeling and schemas](https://github.com/FAIRmat-NFDI/AreaA-data-modeling_and_schemas)

Towards a Standard / an Ontology

Towards a Standard



An iterative exchange among scientific communities
is necessary to acquire generality, consistency, usefulness

“Base Classes” approach

Boil down the data structure to elemental building blocks:

Instrument

Sample

Steps

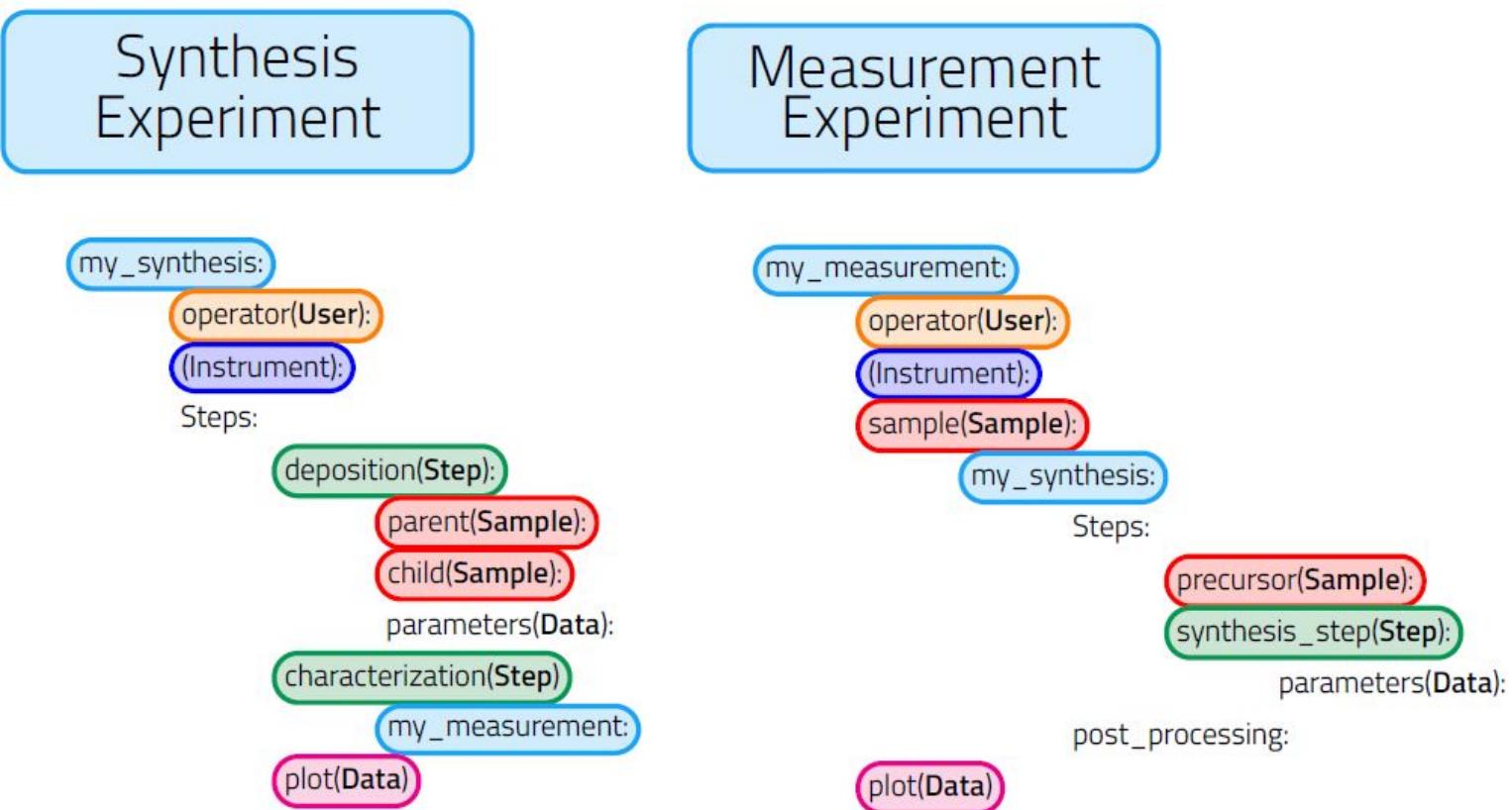
User

Each Base Class contains a set of properties (Quantities)

Allows for additional searchability and processing capabilities in Nomad!

Modularity and Flexibility

Combine base classes into complex structures, depending on single user needs

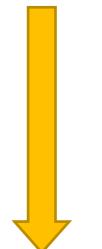


Representing Hierarchy: Coding a Schema

Representing Hierarchy: Coding a Schema

Inheritance

My Gaussmeter



“is a”

Instrument

“My Gaussmeter”
inherits

the properties of
“Instrument”

Representing Hierarchy: Coding a Schema

Inheritance

My Gaussmeter



Instrument

"My Gaussmeter"
inherits

the properties of
"Instrument"

&

Composition

Experiment



My Gaussmeter

"Experiment"
is composed by
an "Instrument"
(a "User", a "Sample", etc.)

How it looks like in Nomad YAML files:

Inheritance

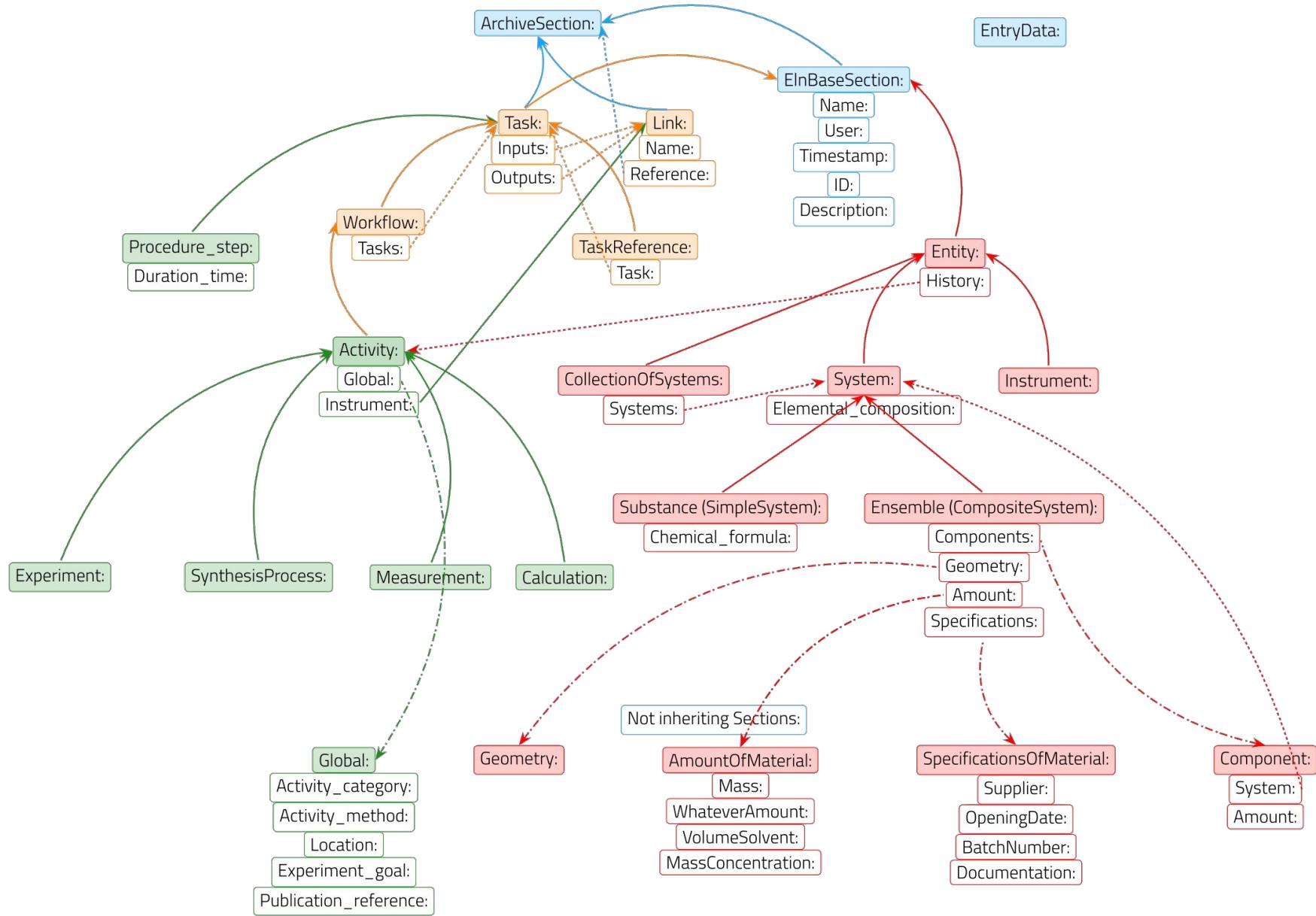
Composition

```
definitions:  
  name: My Custom Schemas  
  sections:  
    MyProcess:  
      quantities:  
        data_file:  
          type: str  
        duration:  
          type: np.float64  
          unit: s  
    MySpecifiedProcess:  
      base_sections:  
        - '#/MyProcess'  
      quantities:  
        carrier_gas:  
          type: str  
      sub_sections:  
        MyProcessesCollection:  
          section:  
            quantities:  
              sample_id:  
                type: str  
              m_annotations:  
                eln:  
                  component: StringEditQuantity
```

"MySpecifiedProcess"
is a
"MyProcess"

"MySpecifiedProcess"
has
quantities
and
sub_sections

Experimental data model is under development



Get Involved

Developed by FAIRmat



sebastian.brueckner@physik.hu-berlin.de

[github.com/FAIRmat-NFDI/AreaA-data modeling and schemas](https://github.com/FAIRmat-NFDI/AreaA-data_modeling_and_schemas)

github.com/nomad-coe/nomad

Acknowledgments

Area D Team: development of the infrastructure

Markus Scheidgen



Theodore Chang



Adam Fekete



Mohammad Nakhaee



Alvin Ladines



Lauri Himanen



Amir Golparvar





Thank you!