

## FAIRmat Tutorial 8:

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

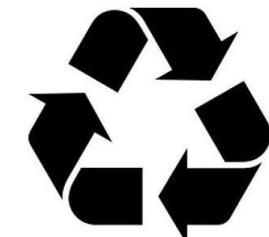
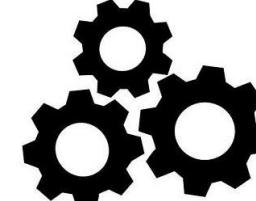
Organized by FAIRmat Area A Synthesis



# FAIRmat Area A: Synthesis

# Why Structured Data?

- ☐ shareable (using a community agreed structure)
- ☐ the structure highlights relationships between data points (semantic interoperability)
- ☐ machine-readable
- ☐ ideal for classification, regression, and clustering with AI
- ☐ searchable
- ☐ 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](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](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-biedienfeldkonzepts\\_13662974.htm](https://de.freepik.com/vektoren-kostenlos/illustration-des-biedienfeldkonzepts_13662974.htm)

## ELN

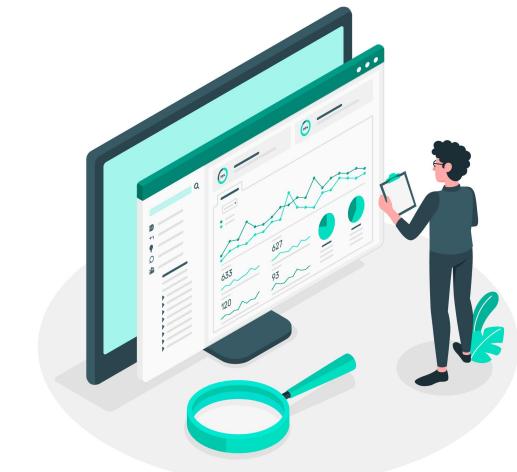
- Manually entered data and metadata



[https://de.freepik.com/vektoren-kostenlos/laptop-mit-bildungssymbol-isoliert\\_11691038.htm](https://de.freepik.com/vektoren-kostenlos/laptop-mit-bildungssymbol-isoliert_11691038.htm)

## Data Analysis

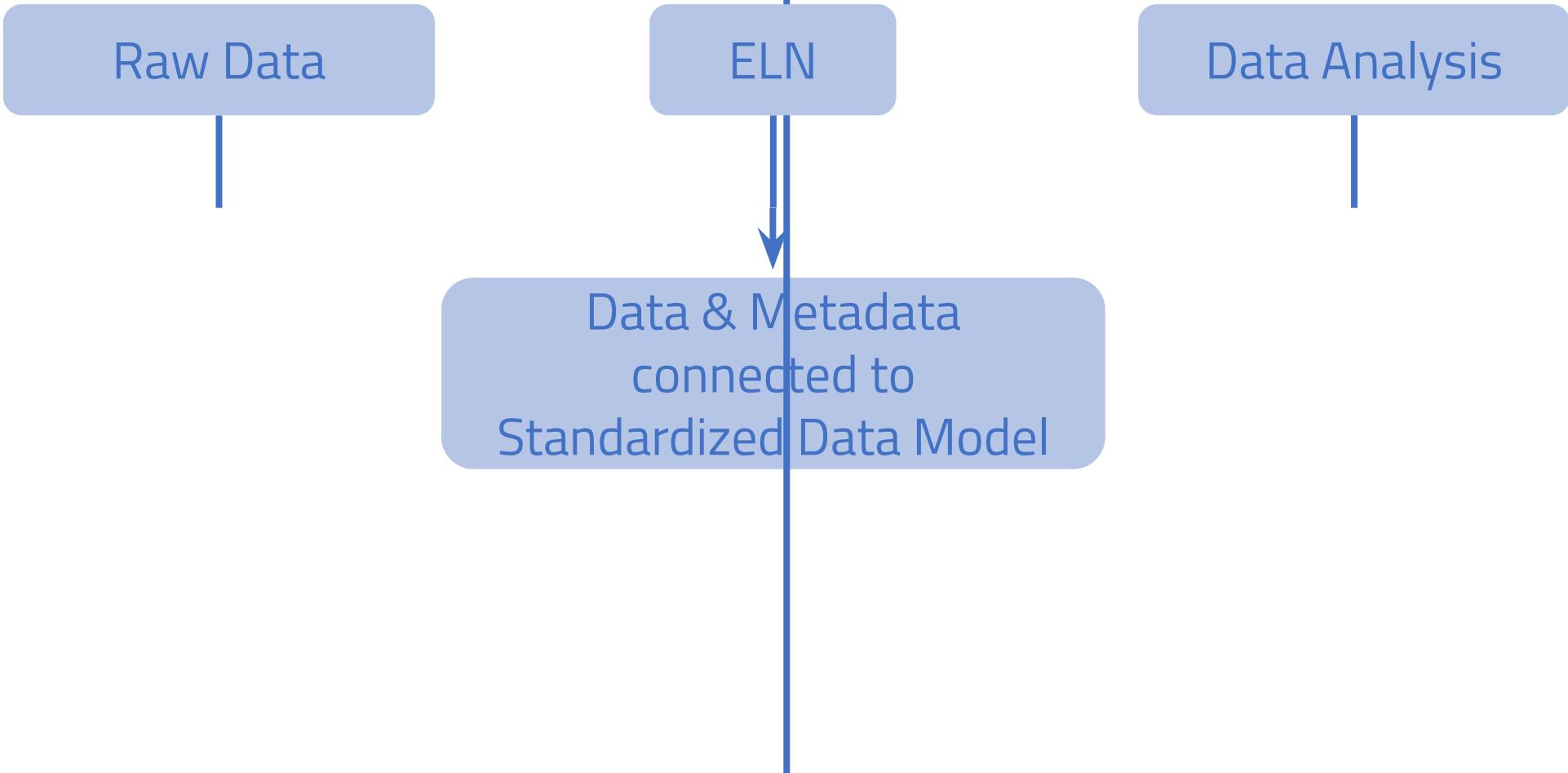
- Post processing software
- User-tailored scripts



[https://de.freepik.com/vektoren-kostenlos/site-statistik-konzeptillustration\\_7140739.htm](https://de.freepik.com/vektoren-kostenlos/site-statistik-konzeptillustration_7140739.htm)

# Research Data Management

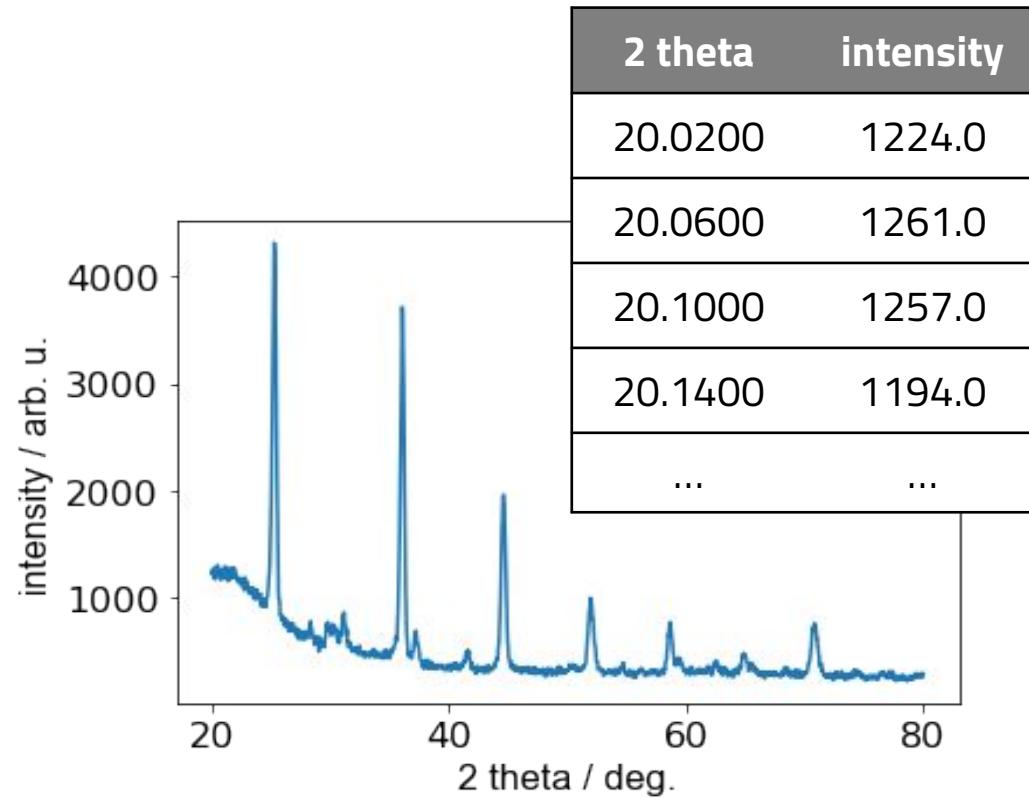
Data Modelling



# Data & Metadata

## Data:

actual content of information



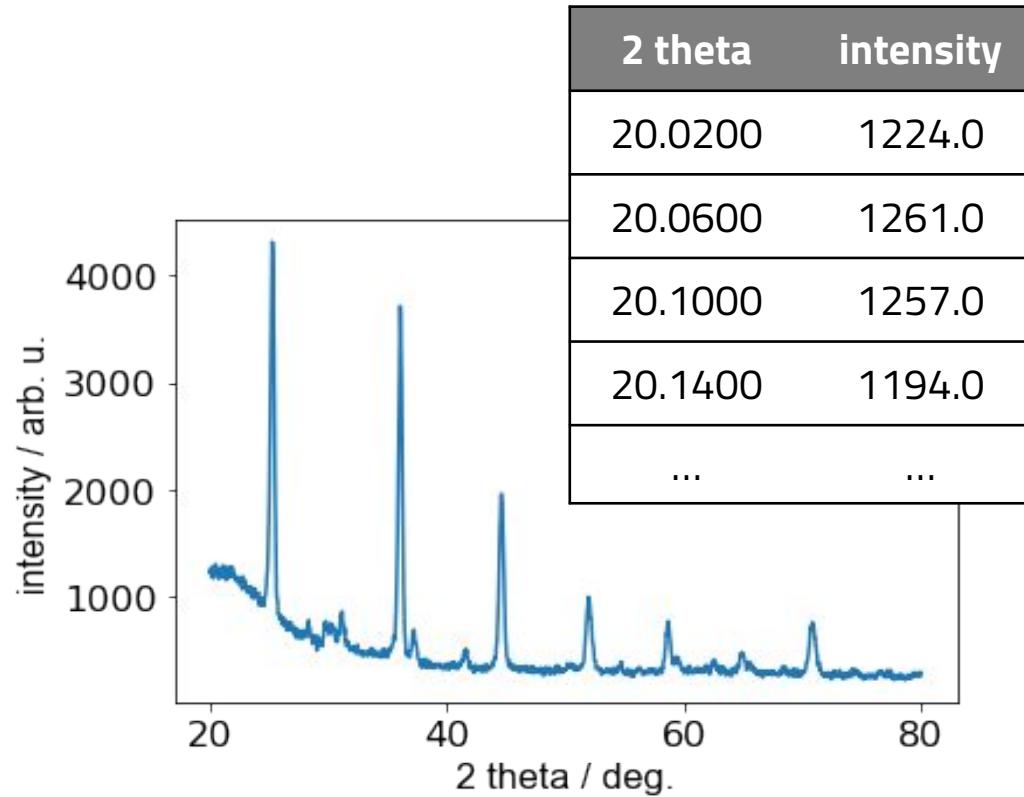
# Data & Metadata

### Data:

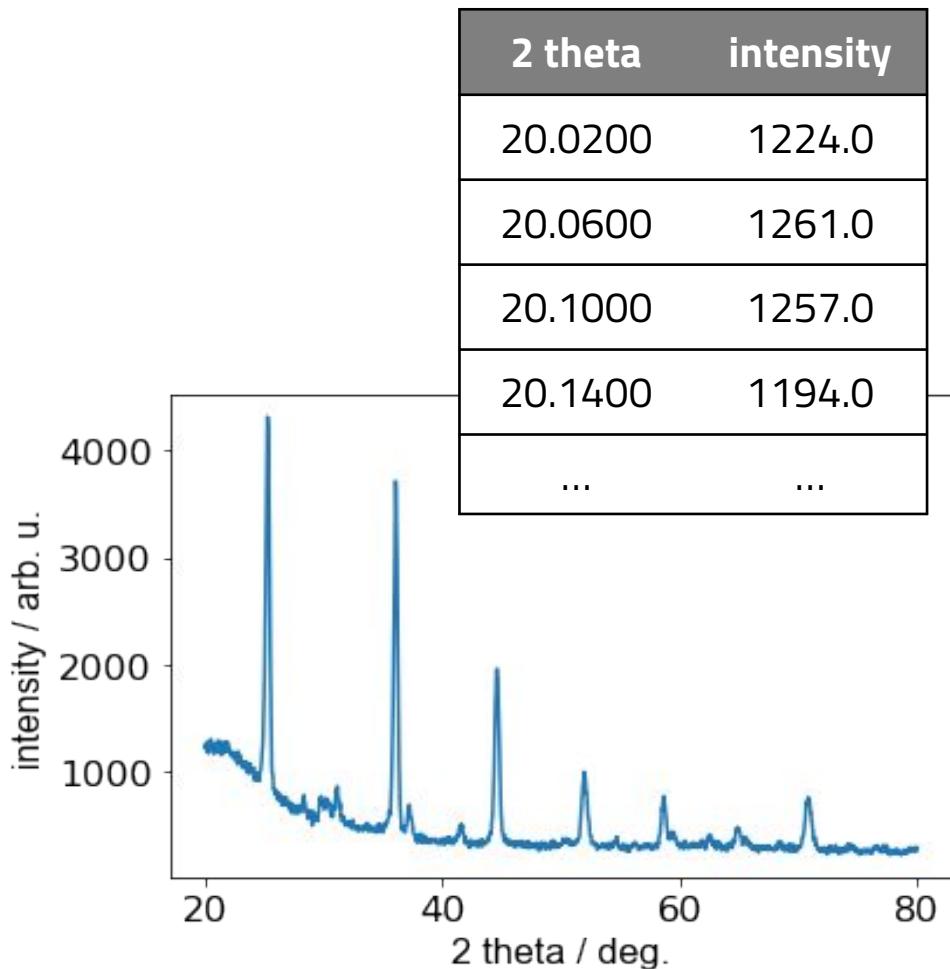
actual content of information

### 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:

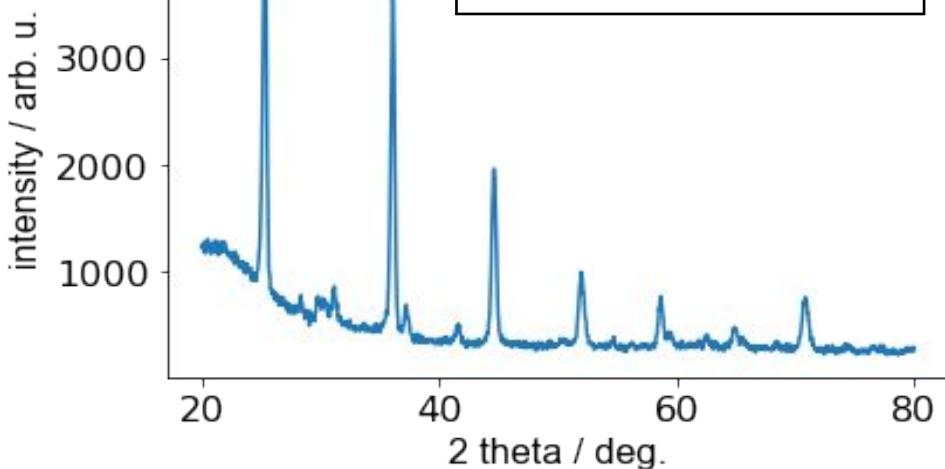
#### 2 theta:

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



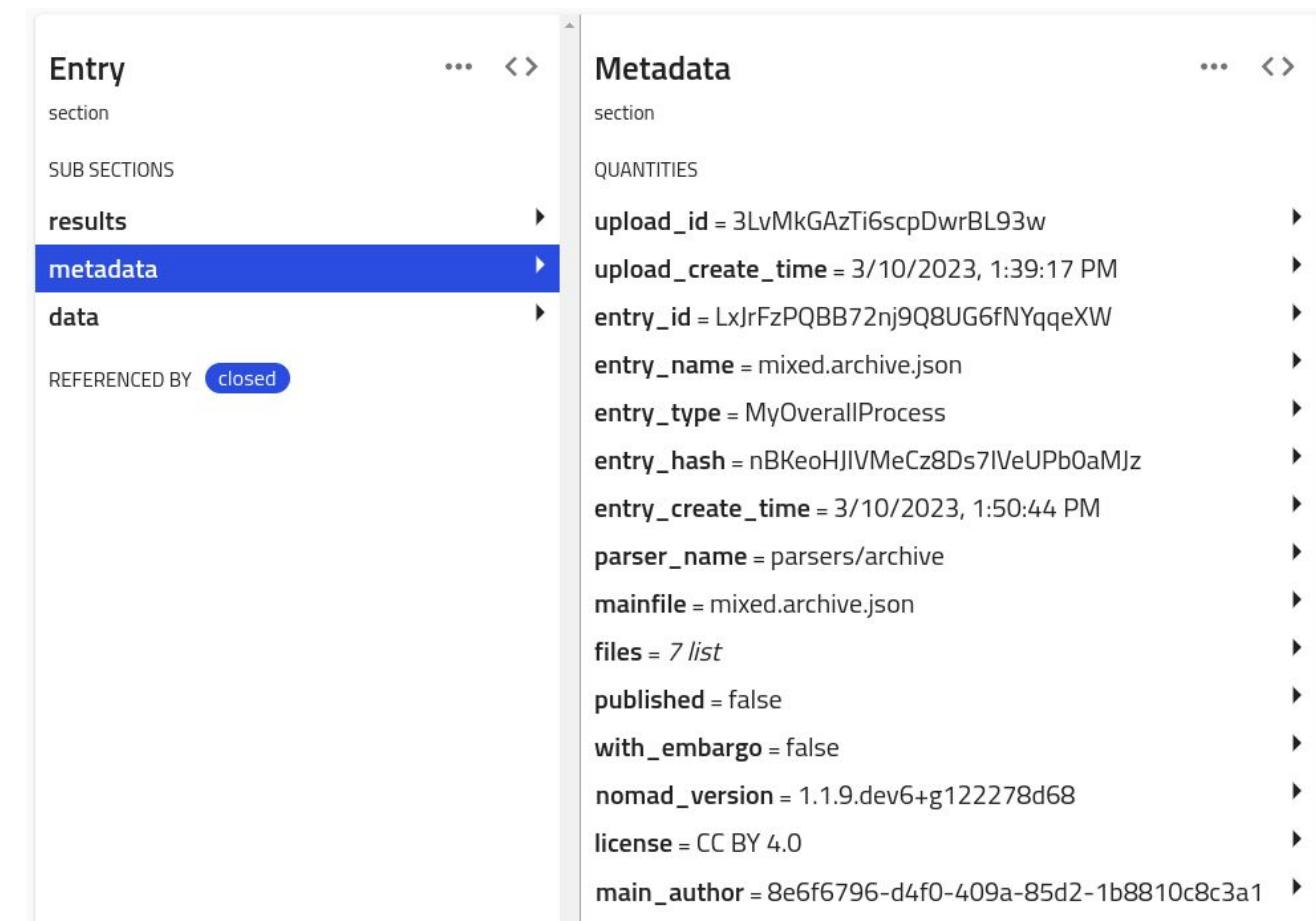
## Data:

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:

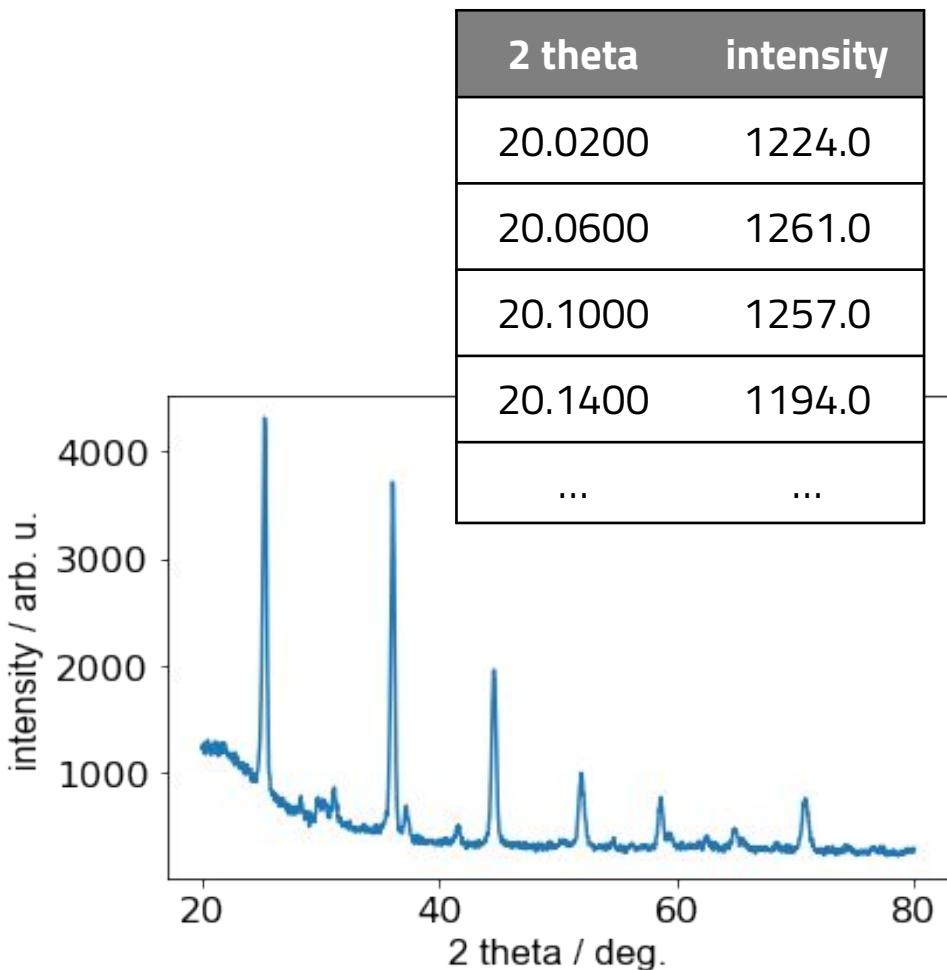
provides information about containers of data



# Data & Metadata

## Data:

actual content of information



## Descriptive Metadata:

provides important context  
about data

Date of the experiment?

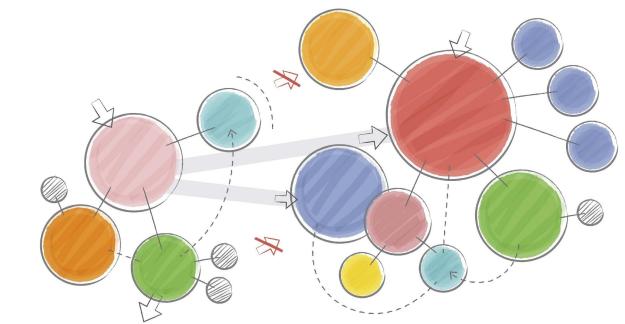
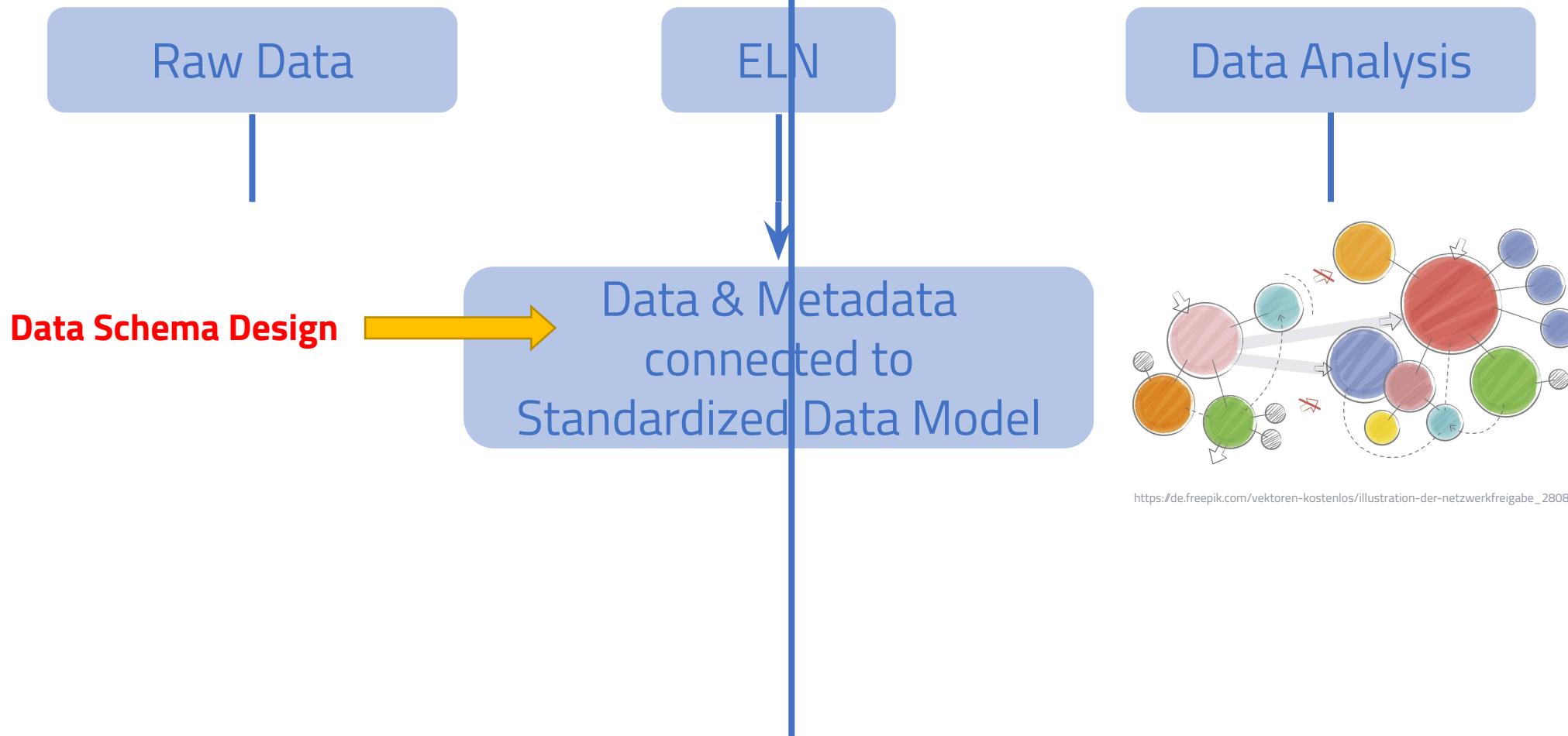
Temperature?

Instrument specifications?

Specimen specifications?

Humidity in the room?

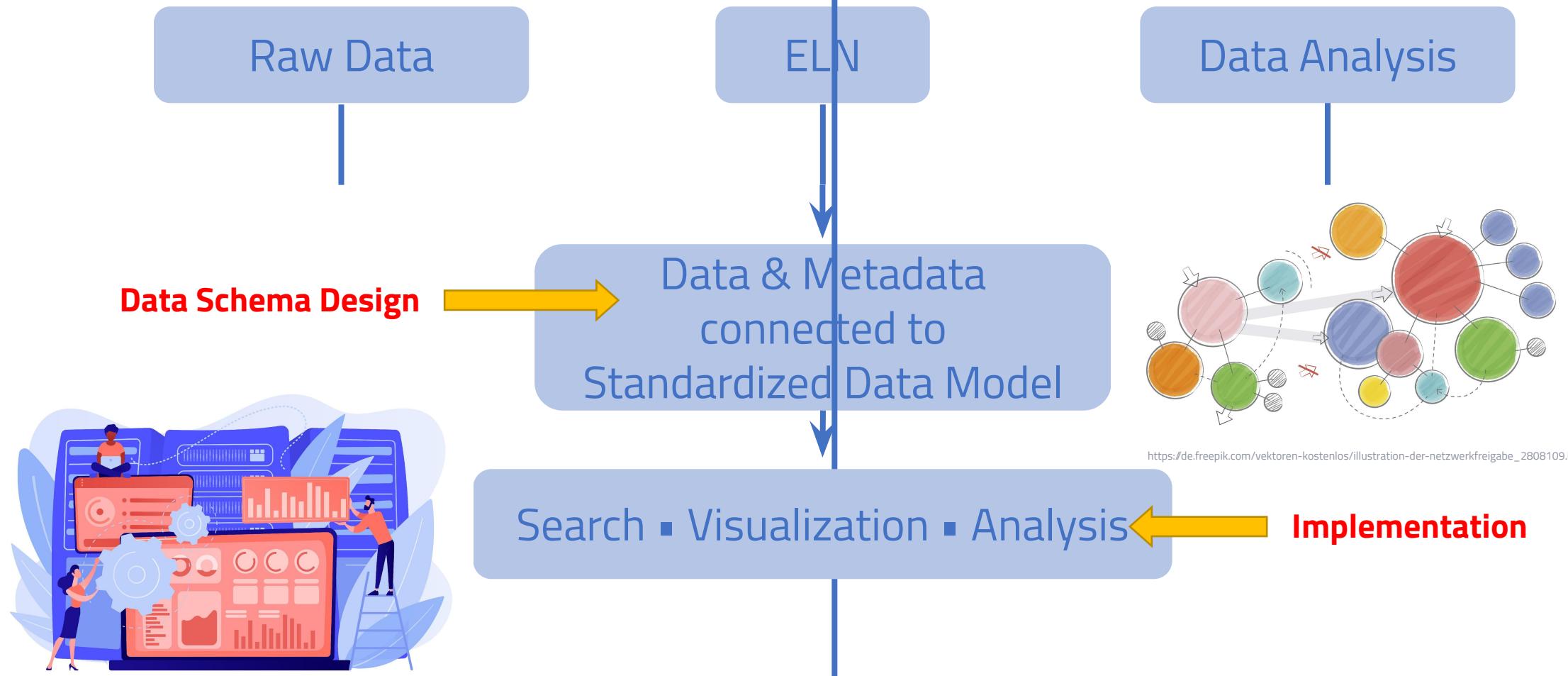
# Research Data Management



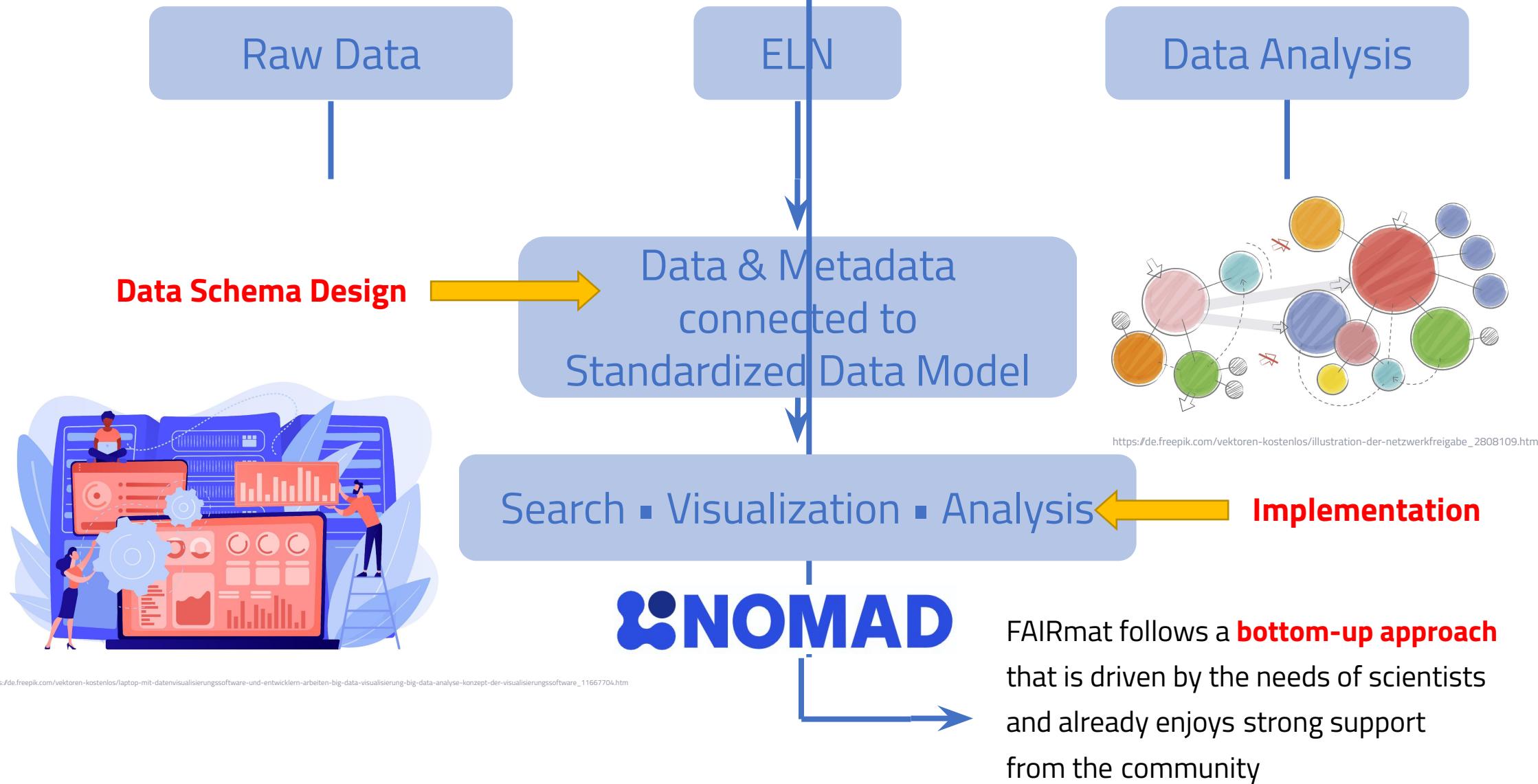
[https://de.freepik.com/vektoren-kostenlos/illustration-der-netzwerkfreigabe\\_2808109.htm](https://de.freepik.com/vektoren-kostenlos/illustration-der-netzwerkfreigabe_2808109.htm)

# Research Data Management

Data Modelling



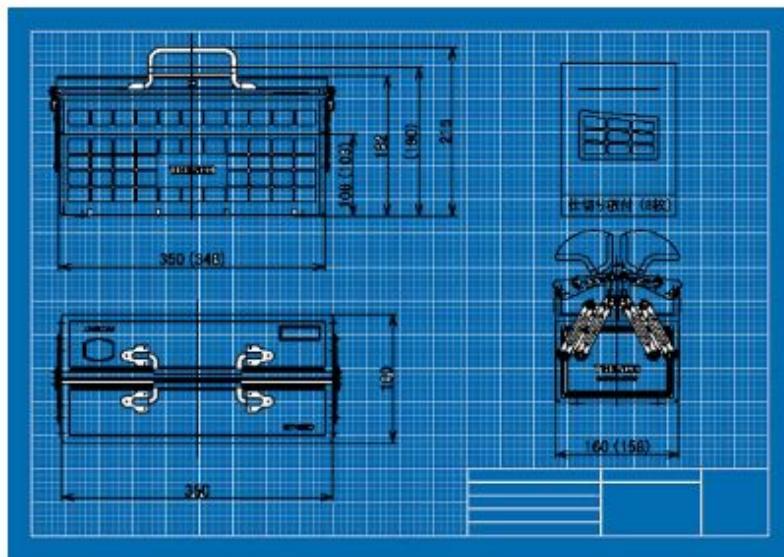
# Research Data Management



# 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

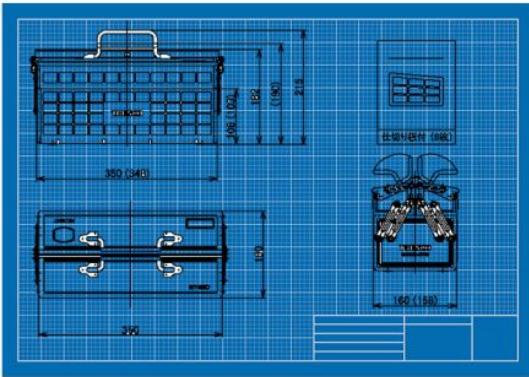
## Template:

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



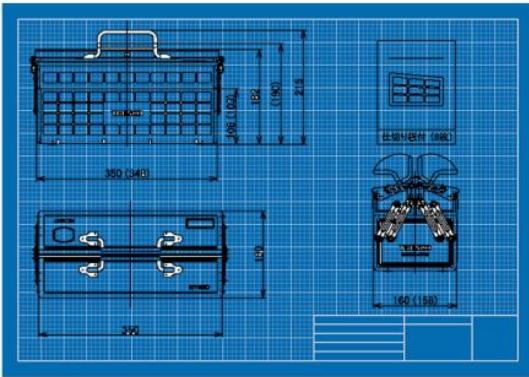
Structured Archive File



Data



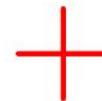
# Schema and Template Concepts



Schema



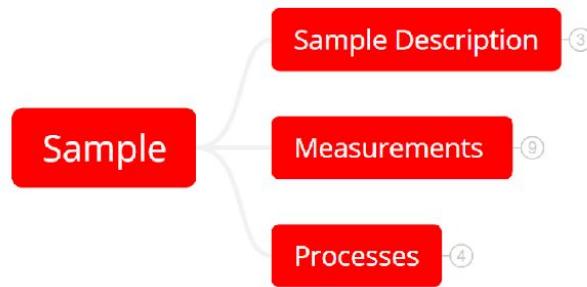
Template



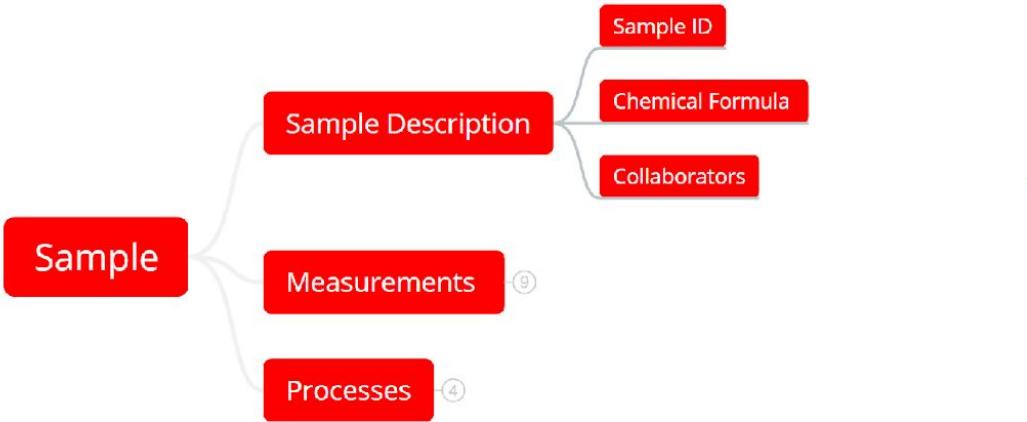
Structured Archive File



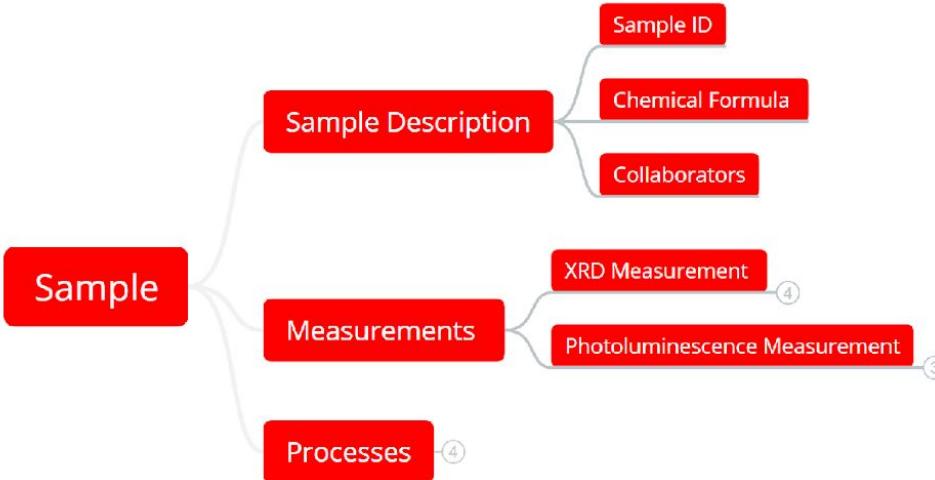
Data



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

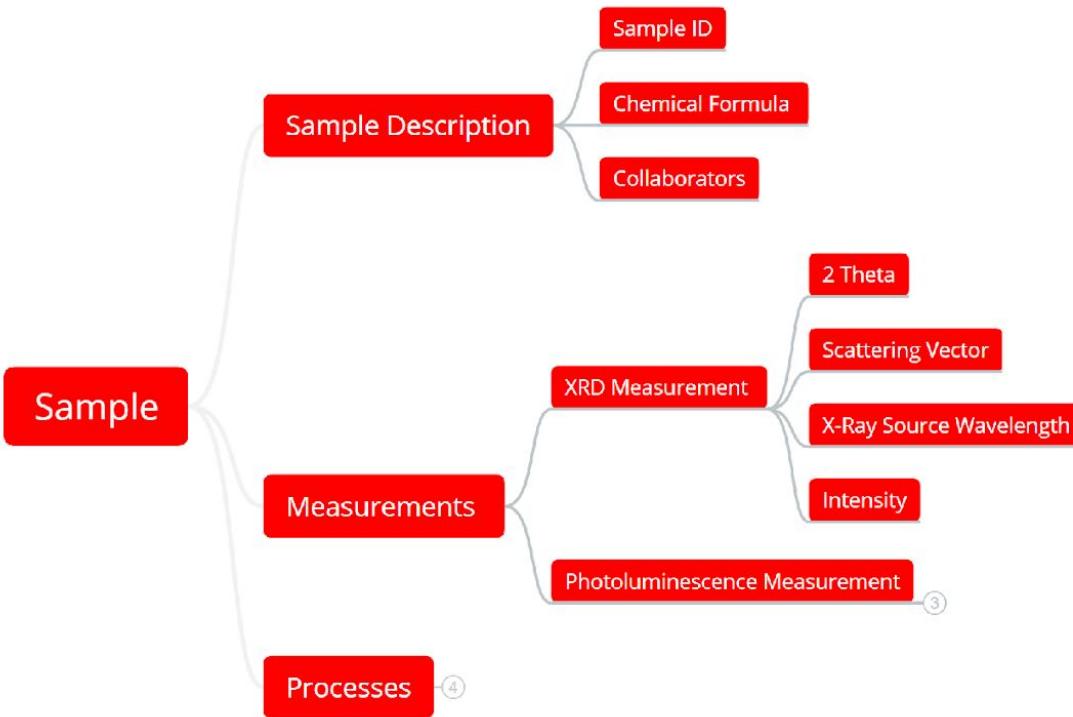


Example of file formats: **XML, JSON, YAML, or HDF5**



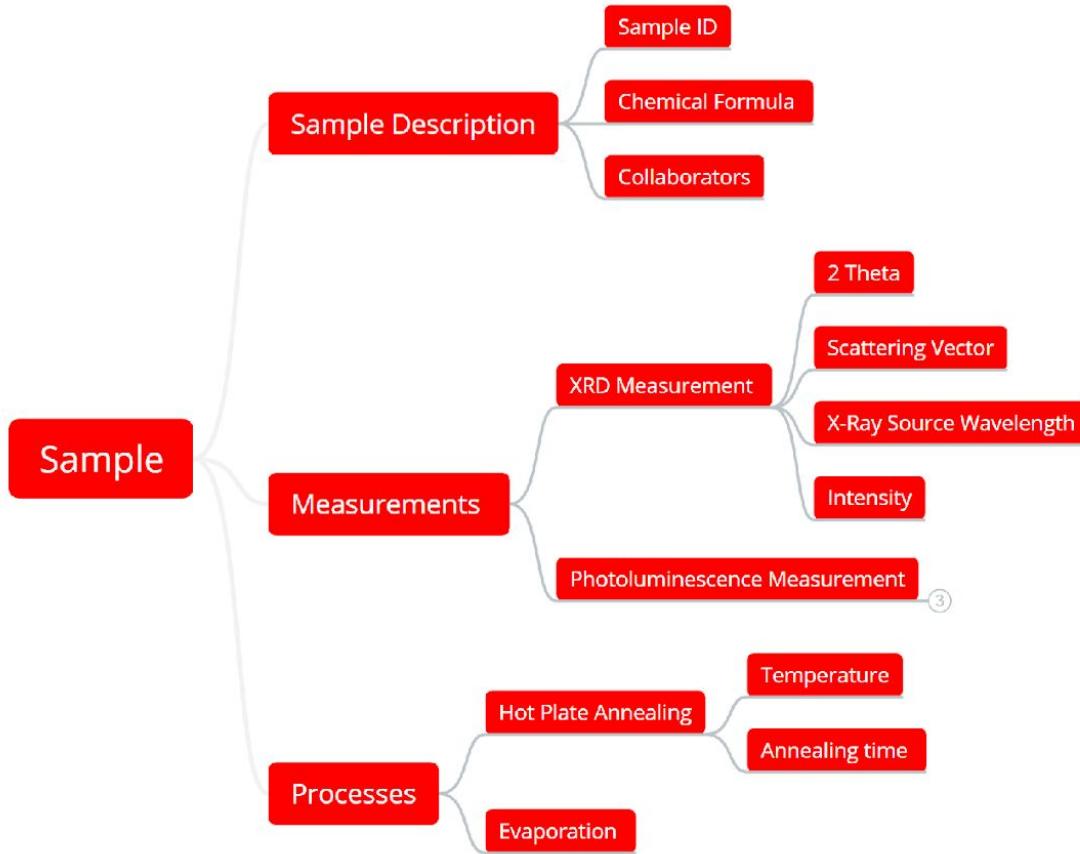
Example of file formats: **XML, JSON, YAML, or HDF5**

# 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.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

```

# Using Schemas in Nomad

The screenshot shows the NOMAD interface with the 'My Processes' entry selected. The 'data' section is highlighted in blue. Arrows point from the code snippets to the 'data' field in the interface and to the 'MyProcessesCollection' section below it.

## 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
      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:

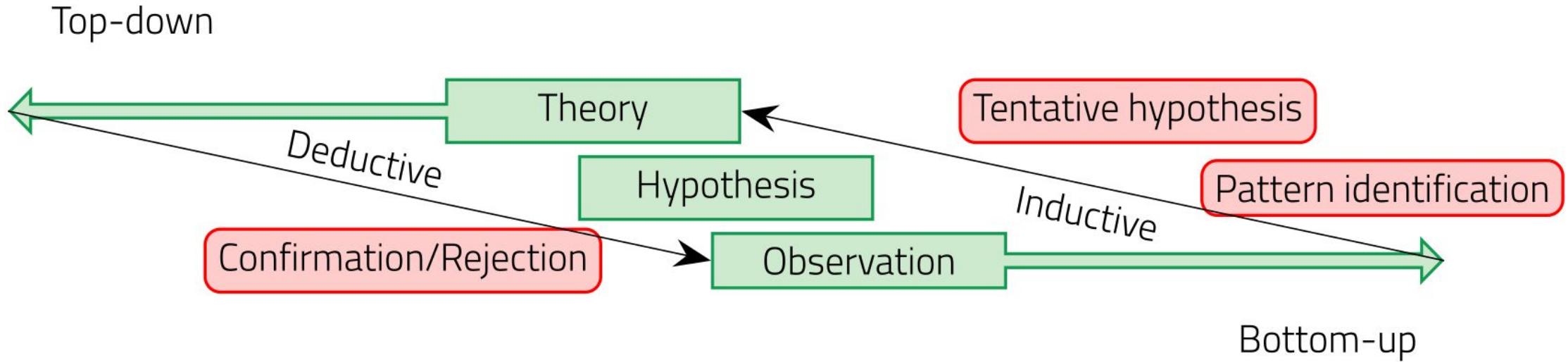
- Left Panel (Schema Definition):**
  - MyProcess:** A section containing a `base_sections` field with a reference to `nomad.datamodel.metainfo.sections`.
  - quantities:** A field containing a `data_file` quantity (type: str) and a `m_annotations` field.
  - sub\_sections:** A field containing a `MyProcessesCollection` section.
- Right Panel (Data Entry):**
  - Entry:** A section with fields: `section`, `SUB SECTIONS`, `results`, `metadata`, and `data`.
  - MyProcess:** A section with a `data_file` quantity set to `process_data_row.csv`.
  - MyProcessesCollection:** A section containing three entries: `0`, `1`, and `aa`.
  - FILES** and **DATA** tabs are visible at the top 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

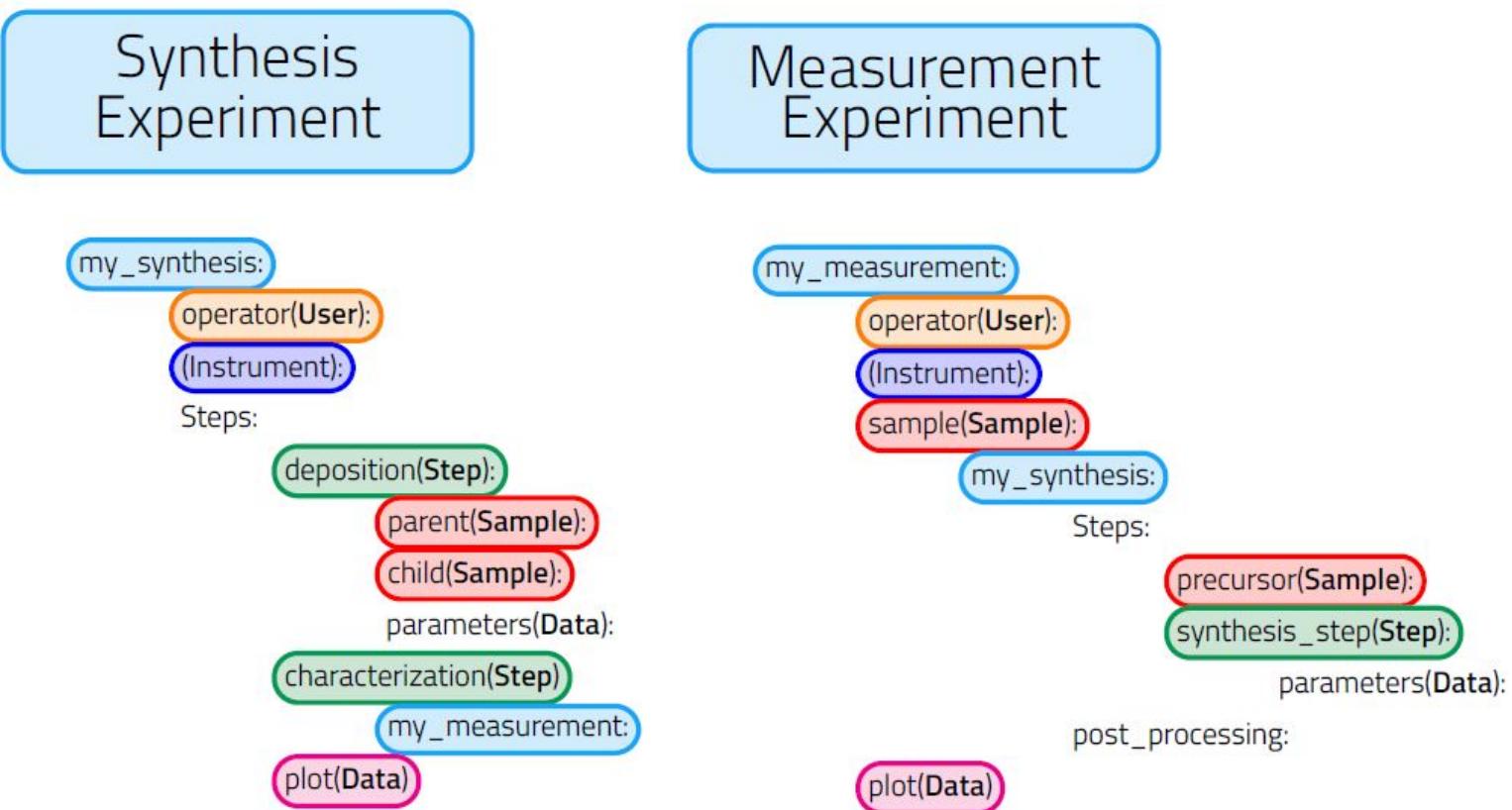
User

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

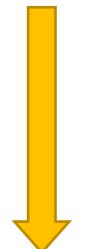


# Representing Hierarchy: Coding a Schema

# 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

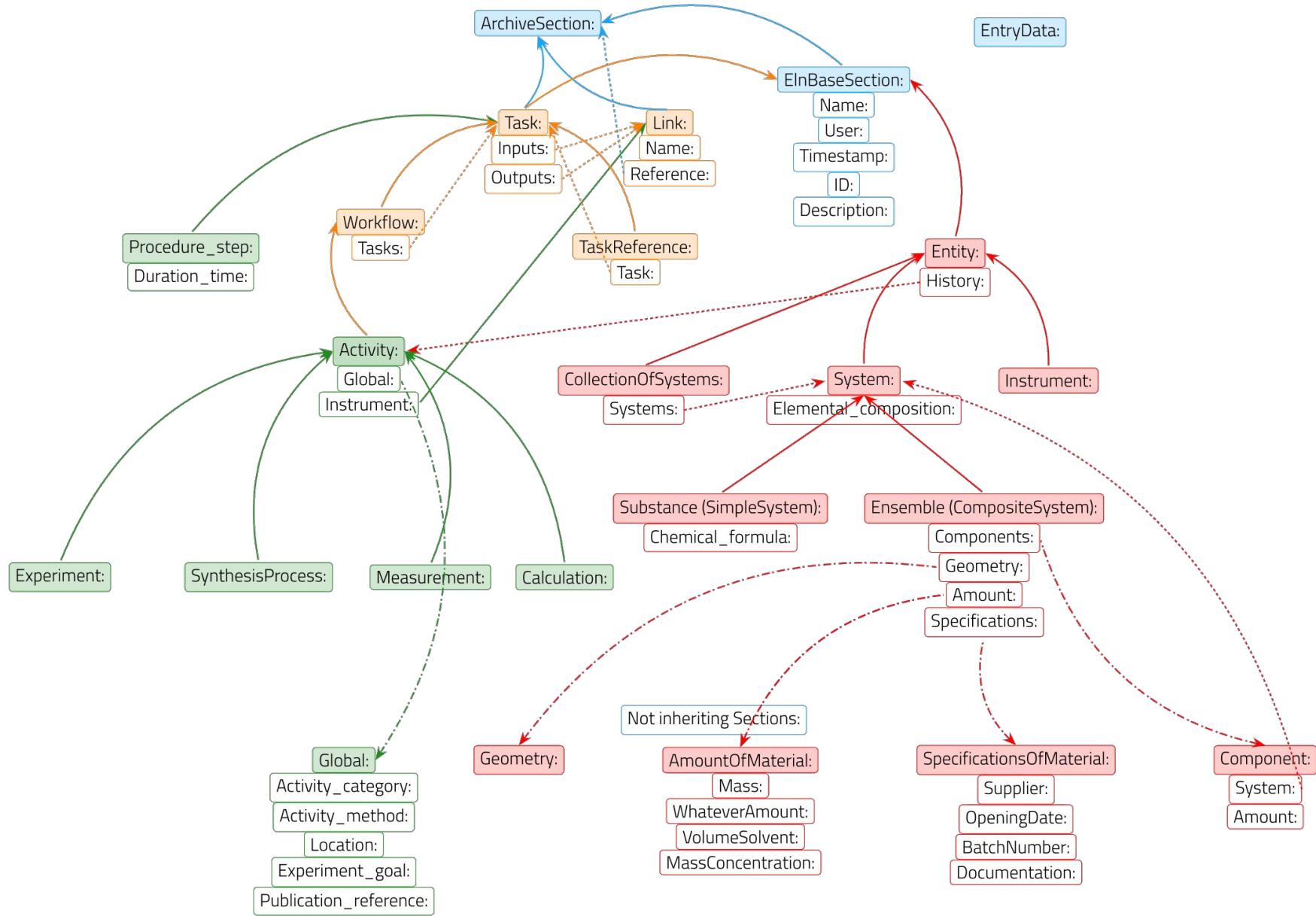
Composition

```
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](mailto: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](https://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!