FAIRmat

Data Management

4. IKZ Winterschool – Machine Learning

Markus Scheidgen
FAIRmat/NOMAD/IRIS/Physik HU

Agenda

- Research Data Management (RDM)
- Managing Data with NOMAD (30 min)
- Modeling data (30 min)
- Working with APIs (30 min)

https://de.freepik.com/vektoren-kostenlos/illustration-des-datenanalysediagramms_2807756.htm

Research Data Management (RDM)

Research Data Management (RDM)

- **Definition**: The process of organizing, preserving, and sharing research data throughout its lifecycle.
- **Goals**: Ensuring the reliability, integrity, and accessibility of research data for current and future use.
- **Activities**: Planning for data collection and documentation, implementing data storage and backup systems, developing policies and procedures for sharing and preserving data.
- **Importance**: Ensuring the quality, reusability and preservation of data to support scientific discovery and validation.

RDM from Three Perspectives

Findable, accessible, interoperable, and re-usable (FAIR) data. Published data to augment traditional paper. Communities should establish data **standards**.

Data as investment and Intellectual Property. Institutes should provide **policies** and **resources** to support researchers to manage their data.

Acquire, organise, and prepare data for analysis. Smooth collaboration with others. Researchers should use **tools** to simplify and unify their work with data.

RDM Tools for Individual Researchers

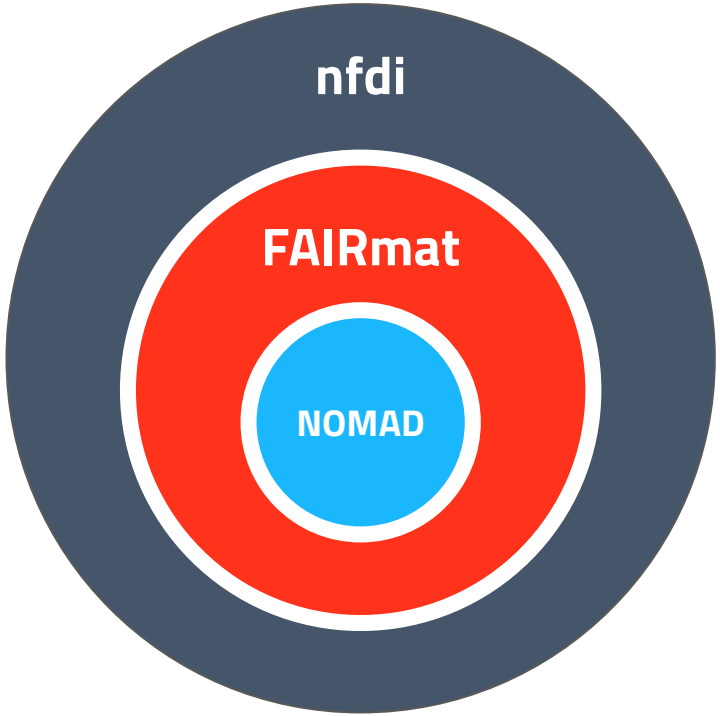
- File management and backup software
- Data documentation and annotation tools (e.g. Excel)
- Data visualization tools
- Scripting and programming languages
- Databases

- NOMAD
- ELN
- LIMS
- Data repositories

*Developed by communities to establish **standards.***
*Required by institutes as part of their **policies** and provided **resources.***
*Used by individuals as day-to-day **tools.***

Managing Data with NOMAD

What are NFDI / FAIRmat / NOMAD



nfdi: Nationale Forschungsdaten Infrastructure, link
(national research data infrastructure)

FAIRmat: NFDI consortium for FAIR materials science data, link
(FAIR: findable, accessible, interoperable, re-usable)

NOMAD: A web-based service and software for managing FAIR materials science data, link
FAIRmat uses NOMAD to build a federated infrastructure of connected NOMAD installations

FAIRmat values

FAIR

Findable, Accessible, Interoperable, Re-usable

FAIR principles can transform the field of condensed-matter physics and the chemical physics of solids.

Open access

Use open processes to support a wide community

FAIRmat advocates for an urgently needed culture shift towards data sharing, and stands for open access to scientific materials data and tools.

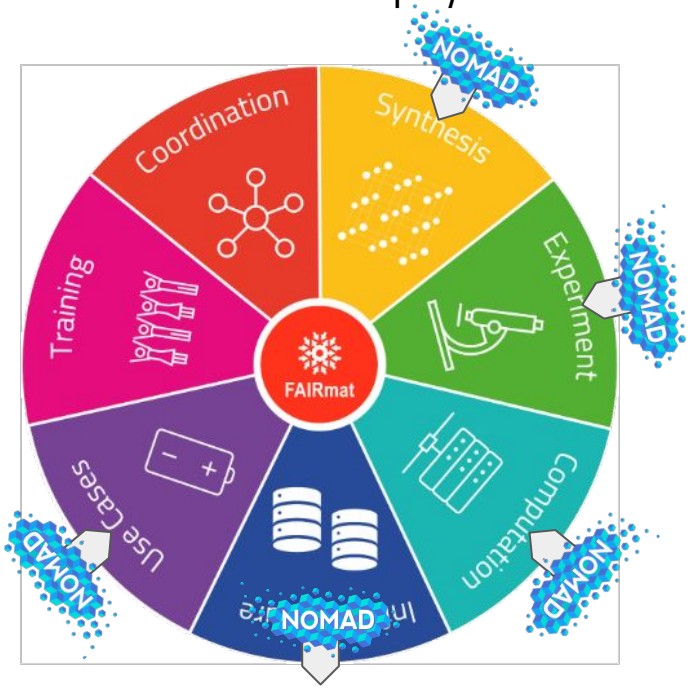
Bottom-up approach

Embracing the community

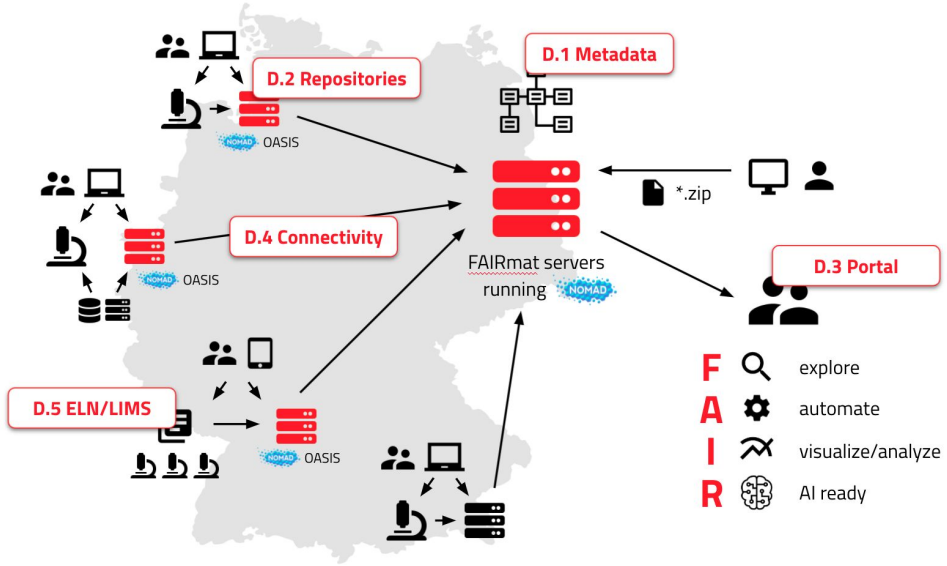
FAIRmat follows an approach that is driven by the needs of scientists and already enjoys strong support from the community.

What is FAIRmat and NOMAD?

FAIRmat is the NDFI consortium to build a FAIR federated data infrastructure for solid state physics



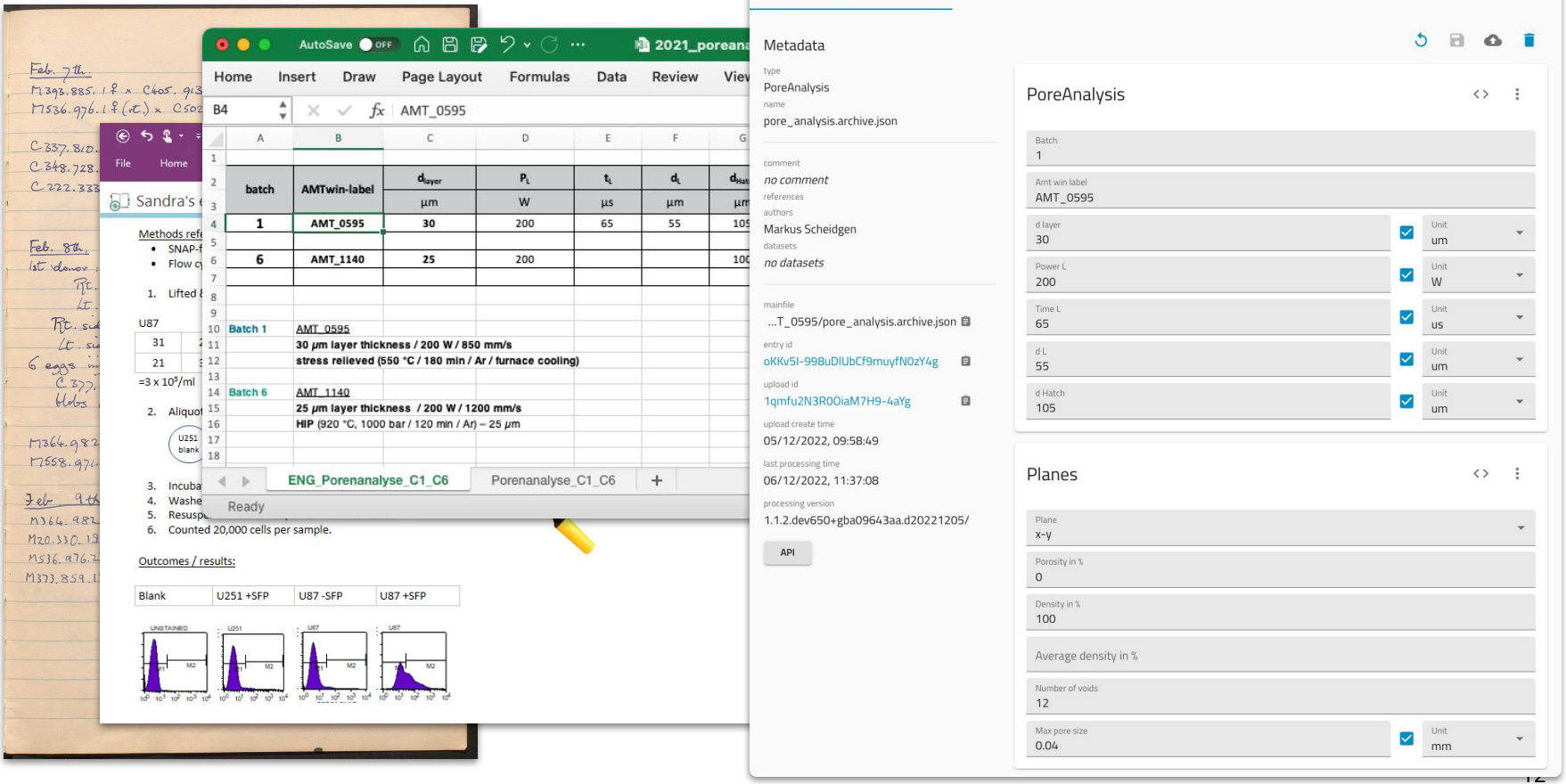
NOMAD is a web-based software for FAIR research data management in materials science

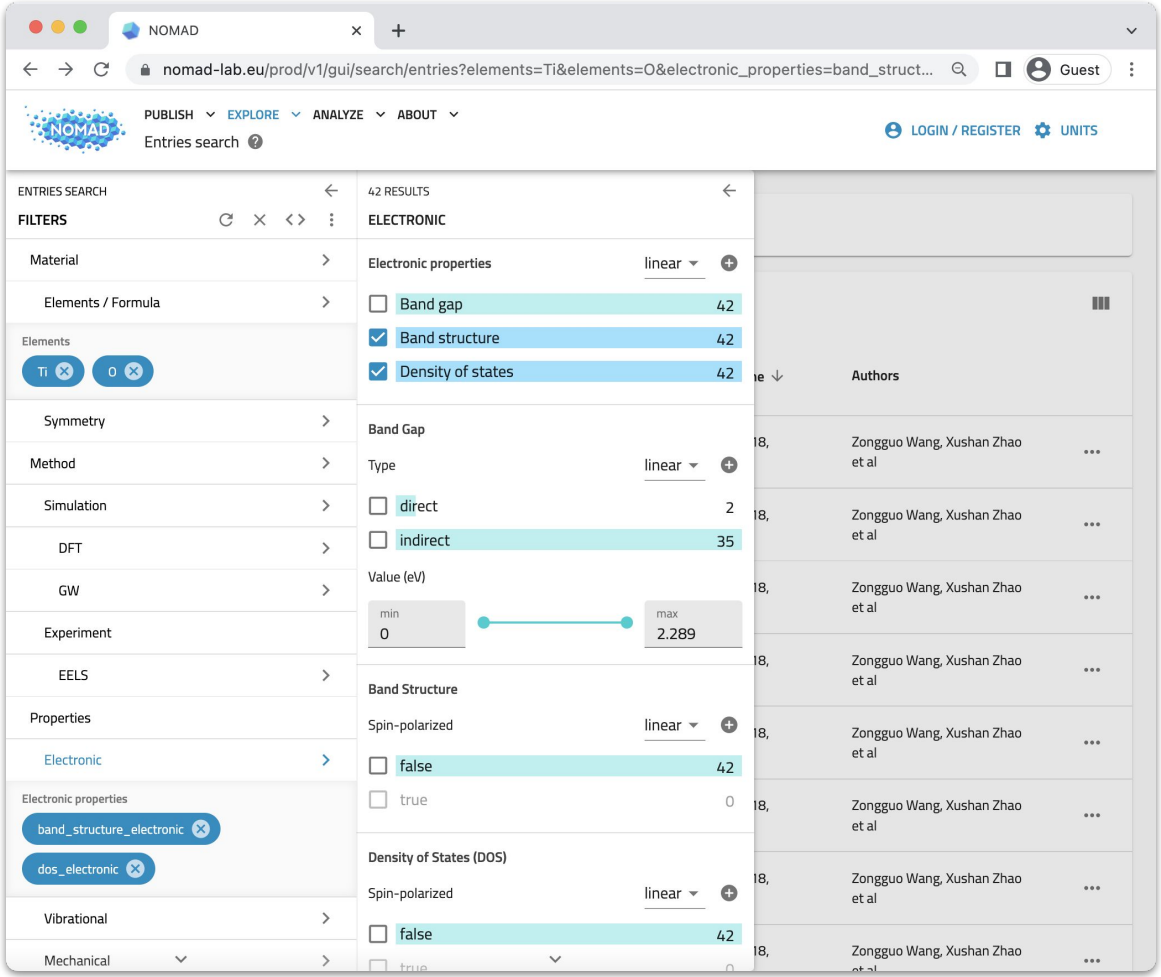


The FAIRmat team

Open positions: fairmat-nfdi.eu

Unstructured data, **structu**




NOMAD
nomad-lab.eu/prod/v1/gui/search/entries?elements=Ti&elements=O&electronic_properties=band_struct...
Guest
PUBLISH
EXPLORE
ANALYZE
ABOUT
Entries search
LOGIN / REGISTER
UNITS
ENTRIES SEARCH
FILTERS
Material
Elements / Formula
Elements
Ti
O
Symmetry
Method
Simulation
DFT
GW
Experiment
EELS
Properties
Electronic
Electronic properties
band_structure_electronic
dos_electronic
Vibrational
Mechanical
42 RESULTS
ELECTRONIC
Electronic properties
linear
Band gap 42
Band structure 42
Density of states 42
Band Gap
Type
linear
direct 2
indirect 35
Value (eV)
min
0
max
2.289
Band Structure
Spin-polarized
linear
false 42
true 0
Density of States (DOS)
Spin-polarized
linear
false 42
Authors
Zongguo Wang, Xushan Zhao et al

NOMAD

nomad-lab.eu/prod/v1/gui/search/entries/entry/id/PGutRVteRyGYiUymc1xgdw/e9SAHYm9B05xMrMe2H...

Guest

PUBLISH EXPLORE ANALYZE ABOUT

Entries search / Entry

LOGIN / REGISTER UNITS

e9SAHYm9B05xMrMe2HNpCcLF2T18

material id

A_1EkzAi2iFORWMpo9ddaPtRFnxF

upload id

PGutRVteRyGYiUymc1xgdw

upload create time

08/02/2020, 09:38:05

last processing time

28/12/2021, 07:45:59

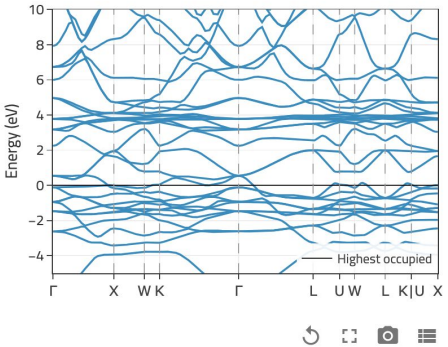
processing version

1.0.0/37587aa6

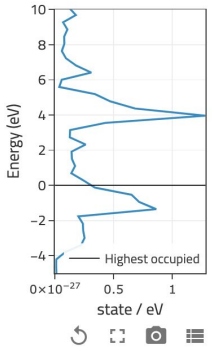
API

Electronic properties

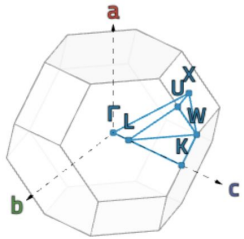
Band structure



Density of states



Brillouin zone



Band gaps

Ch.	Value (eV)	Type
0	0	no gap

FAIRmat

NOMAD

nomad-lab.eu/prod/v1/gui/search/entries/entry/id/PGutRVteRyGYiUymc1xgdw/e9SAHYm9B05xMrMe2HNpCcLF2T18/archive/run/system/atoms/lattice_vectors

Guest

PUBLISH EXPLORE ANALYZE ABOUT

Entries search / Entry / Processed data

LOGIN / REGISTER UNITS

OVERVIEW RAW DATA PROCESSED DATA LOGS

search

code specific all defined definitions

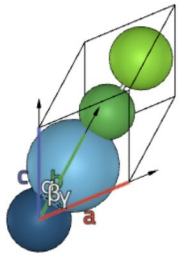
Entry
section
SUB SECTIONS
- results
- metadata
- workflow
- run

Run
section
SUB SECTIONS
- calculation
- method
- program
- system

System
section
SUB SECTIONS
- atoms
- prototype
- symmetry

QUANTITIES
- chemical_composition = LaMgNiOs
- chemical_composition_hill = LaMgNiOs
- chemical_composition_reduced = LaMgNiOs
- configuration_raw_gid = 9nKX-vTV8ie36M9I3fldft-…
- is_representative = true
- type = bulk

Atoms
section



QUANTITIES
- labels = *4 list*
- lattice_vectors = *3 × 3 matrix*
- periodic = *3 list*
- positions = *4 × 3 matrix*
- species = *4 vector*

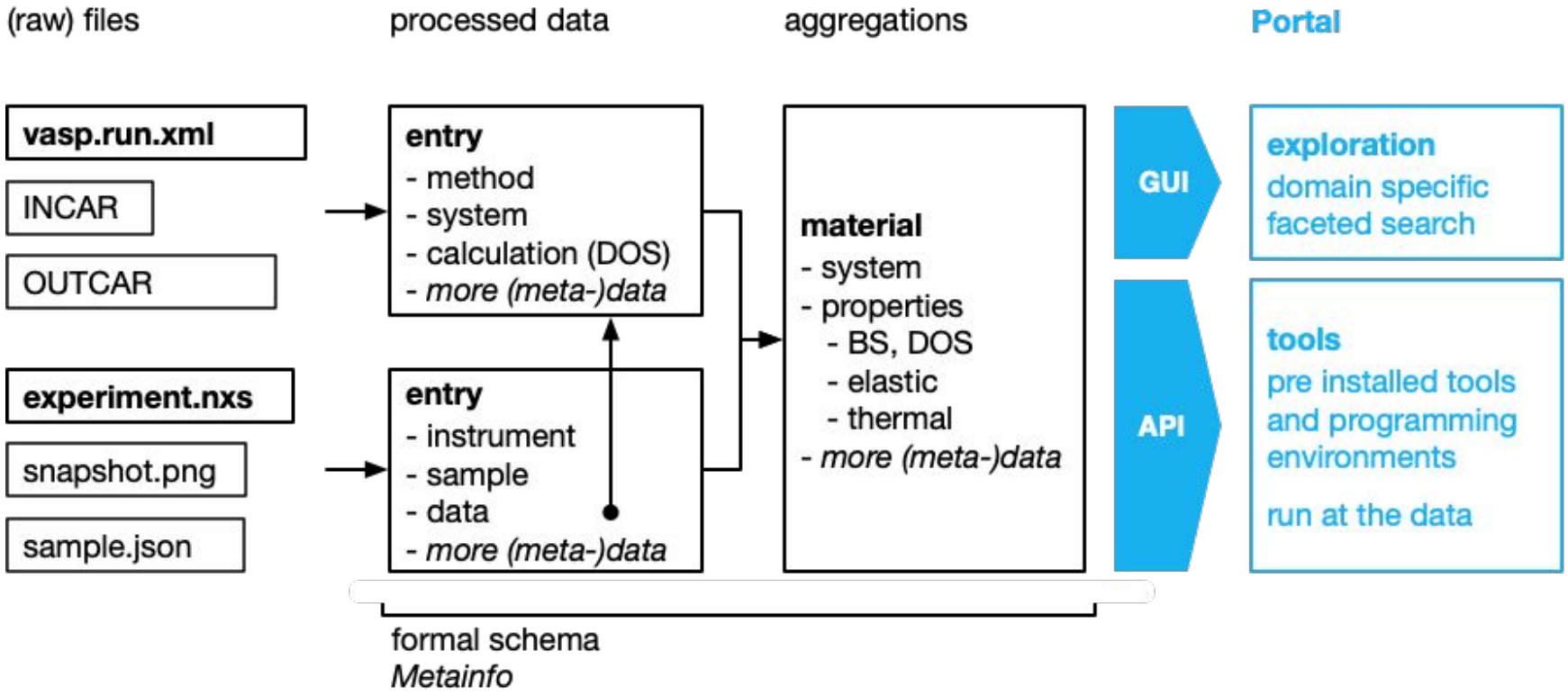
lattice_vectors
quantity
VALUE

0	3.34494	3.34494
3.34494	0	3.34494
3.34494	3.34494	0

(3 × 3)
Å

FAIRmat

Core functionality: Processing data files to extract (meta-)data



NOMAD

nomad-lab.eu/util/north-oasis/gui/user/uploads...

PUBLISH EXPLORE ANALYZE ABOUT

Your uploads / Upload / Entry

Welcome Markus Scheidgen LOGOUT UNITS

OVERVIEW FILES DATA LOGS

SiO2onSi.elli...

entry NX

acquisiti... NX

program

version

definition

experiment...

experiment...

instrument

plot NX

delta_50d...

delta_60d...

delta_70d...

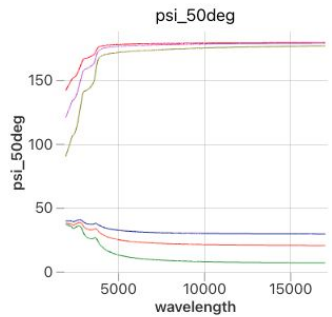
psi_50deg

entry Display Inspect

NX Spectrum X Linear Y Linear

n 1088

x D0



Entry References

Referencing the following entries

Not referencing other entries.

Referenced by the following entries

Ellipsometry workflow example

nomad-lab.eu/util/north-oasis/north/user/mscheid...

jupyterhub Ellipsometry workflow example (autosaved)

Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3 (ipykernel)

Run Markdown

Ellipsometry workflow example

In this notebook, an ellipsometry data set of 2 nm SiO2 on Si is analyzed using the analysis tool pyElli

1. Create NeXus file from measurement data

The metadata of the experiment are listed in a YAML file (**eln-data.yaml**, which is automatically created when saving the metadata entered into the electronic lab notebook (ELN) within NOMAD) according to the application definition NXellipsometry. The name of the data file (here **test-data.dat**) needs to be specified in the ELN and, hence, is defined as an entry 'filename' in the YAML file. Using the **ellips** reader and the application definition in NXDL format, a NeXus file (**SiO2onSi.ellips.nxs**) is created. Both the data and metadata files must be stored in this repository.

Note: When creating or modifying the YAML file without using the ELN, make sure that all required fields are provided; recommended and optional fields may be provided if known and meaningful.

In [ ]:
```
from nexusutils.dataconverter.convert import convert
```

In [ ]:
```
convert(input_file=["eln_data.yaml"],
        reader='ellips',
        nxdl='NXellipsometry',
        output='SiO2onSi.ellips.nxs')
```

2. Inspect the NeXus file with h5web

In [ ]:
```
from jupyterlab_h5web import H5Web
```

In [ ]:
```
H5Web('SiO2onSi.ellips.nxs')
```

This is the end of the general template. Continue to fill the notebook based on **your own** post-processing of the *.nxs file.

3. Analyze Ψ and Δ values using a transfer-matrix solver

Import the analysis tool pyElli:

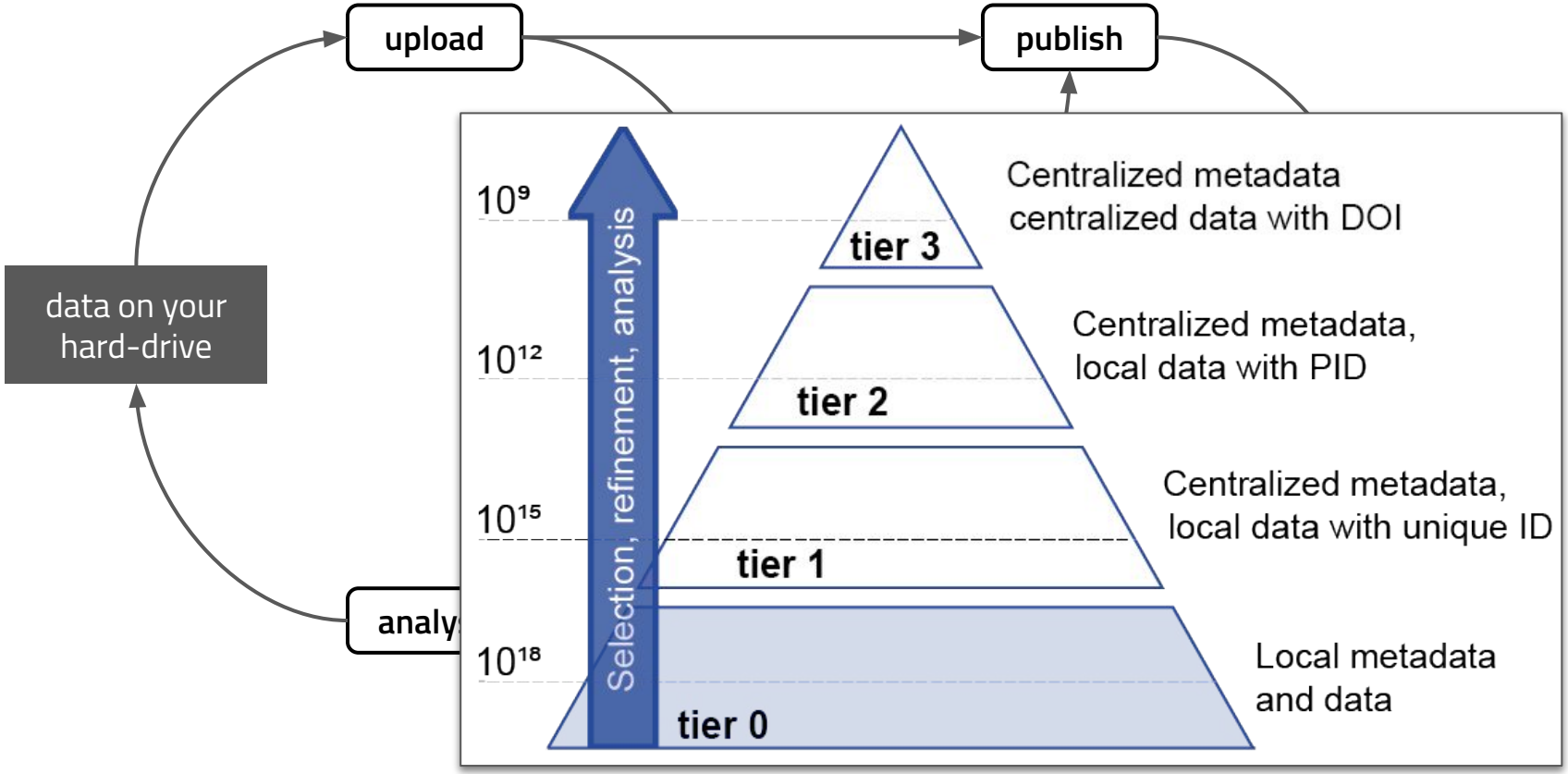
In [ ]:
```
import elli
from elli.fitting import ParamsHist, fit
from elli.importer.nexus import read_nexus_psi_delta
from elli.dispersions import TableSpectraRay
```

3.1. Load data from NeXus file

We load the data from the generated NeXus file and select the angle of incidence we want to analyze. You may set ANGLE to

FAIRmat

What is NOMAD Oasis?



Manage data from many sources

computer programs simulations instruments automatized workflows	**supported files** input/output simulation codes nexus/HDF5 files other formats	PARSERS one parser per code/format	**(meta)data** structured, human and machine processable data based on a well-defined schema
human activity handling and use of samples, instruments, … manual workflows	**forms** specialised data entry fields rich text, images, tables JSON	GUI based on schema annotations	**(meta)data** structured, human and machine processable data based on a well-defined schema
databases reference data LIMS/ELN collaboration	**tabular data** CSV, Excel import from other databases JSON, XML, YAML, …	MAPPING based on a schema mapping	**(meta)data** structured, human and machine processable data based on a well-defined schema well-defined schema well-defined schema

Data Modeling

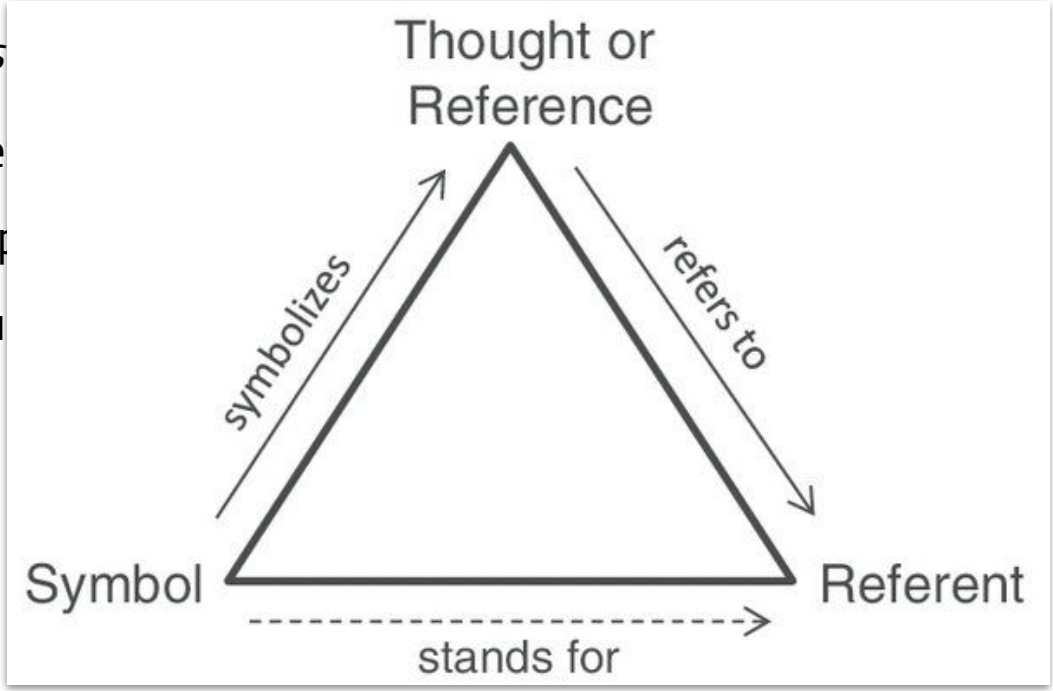
https://de.freepik.com/vektoren-kostenlos/3d-modellierungskonzeptillustration_9558648.htm

Models in general

Model: a purposeful abstract representation of something.

"The best material model of a cat is

- Mathematical model, e.g. Ne
- Physical model, e.g. a paper p
- Conceptual model, e.g. langu



Terminology: (Meta-)Data, Formats, and Schemas

<table>
<tr><td colspan="2">data schemas</td><td rowspan="2">databases / formats</td></tr>
<tr><td>meta-data
- when, who, where, environment
- settings, sample preparation
- workflow steps
- scales, dimensions, units</td><td>data
- pixels
- matrices
- numbers</td></tr>
<tr><td colspan="3">real life phenomena or simulation
- sample under a microscope
- solution of a mathematical model</td></tr>
</table>

OO structure models, e.g. URL class diagrams

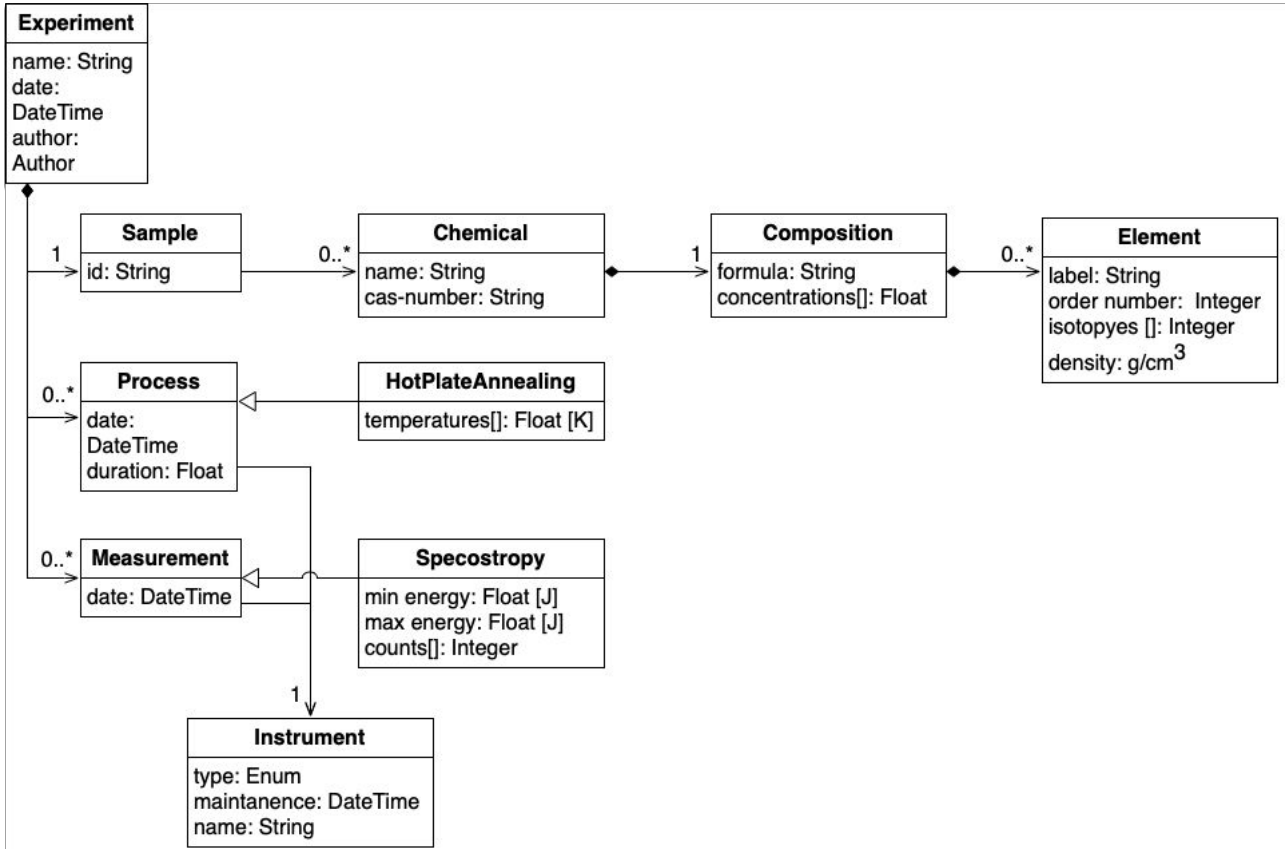
Classes (or concepts, definitions) and
Objects (or instances, occurrences)

- Classes define sets of possible objects by defining shared object properties.
- Objects always instantiate exactly one class and can (only) have the class defined properties.

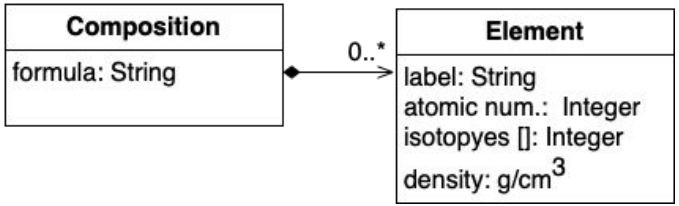
Relationships

- Reference
- Containment
- Dependency
- Generalisation (specialization)
- [Instantiation]

OO structure models, e.g. UML class diagrams



NOMAD Schema




```python
import numpy as np
from nomad.metainfo import MSection, Quantity, SubSection

class Element(MSection):
    label = Quantity(type=str)
    atomic_number = Quantity(type=int)
    density = Quantity(type=np.float64, unit='g/cm**3')
    isotopes = Quantity(type=int, shape=['*'])

class Composition(MSection):
    formula = Quantity(type=str)
    concentrations = Quantity(np.float64, shape=['*'])
    elements = SubSection(section=Element, repeats=True)
```

```yaml
definitions:
  sections:
    Element:
      quantities:
        label:
          type: str
        atomic_number:
          type: int
        density:
          type: np.float64
          unit: g/cm**3
        isotopes:
          type: int
          shape: ['*']
    Composition:
      quantities:
        composition:
          type: str
        concentrations:
          type: float
          shape: ['*']
      sub_sections:
        elements:
          section: Element
          repeats: true
```

NOMAD Data

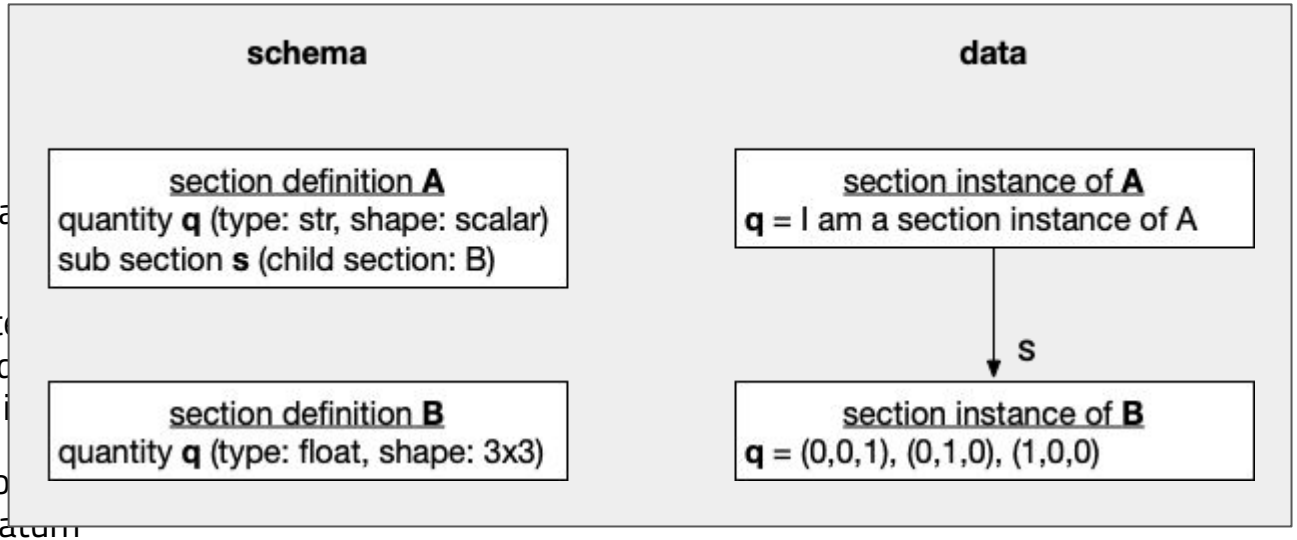
```python
import json
from nomad.units import ureg

water = Composition(formula='H2O')
water.elements.append(
    Element(
        label='H',
        atomic_number=1,
        isotopes=[1, 2, 3],
        density=0.000082
    ))
water.elements.append(
    Element(
        label='O',
        atomic_number=8,
        isotopes=[16],
        density=0.001308
    ))

print(water.elements[0].density.to(ureg('kg/m**3')))
print(json.dumps(water.m_to_dict(), indent=2))
```

```json
{
  "formula": "H2O",
  "elements": [
    {
      "label": "H",
      "atomic_number": 1,
      "density": 8.2e-05,
      "isotopes": [
        1,
        2,
        3
      ]
    },
    {
      "label": "O",
      "atomic_number": 8,
      "density": 0.001308,
      "isotopes": [
        16
      ]
    }
  ]
}
```

Types of defin



Definitions: All have a name a

Section:

- Defines a block of relat
- Can be instantiated and
- *groups*, *classes*, *entities* i

Quantity (properties of sectio

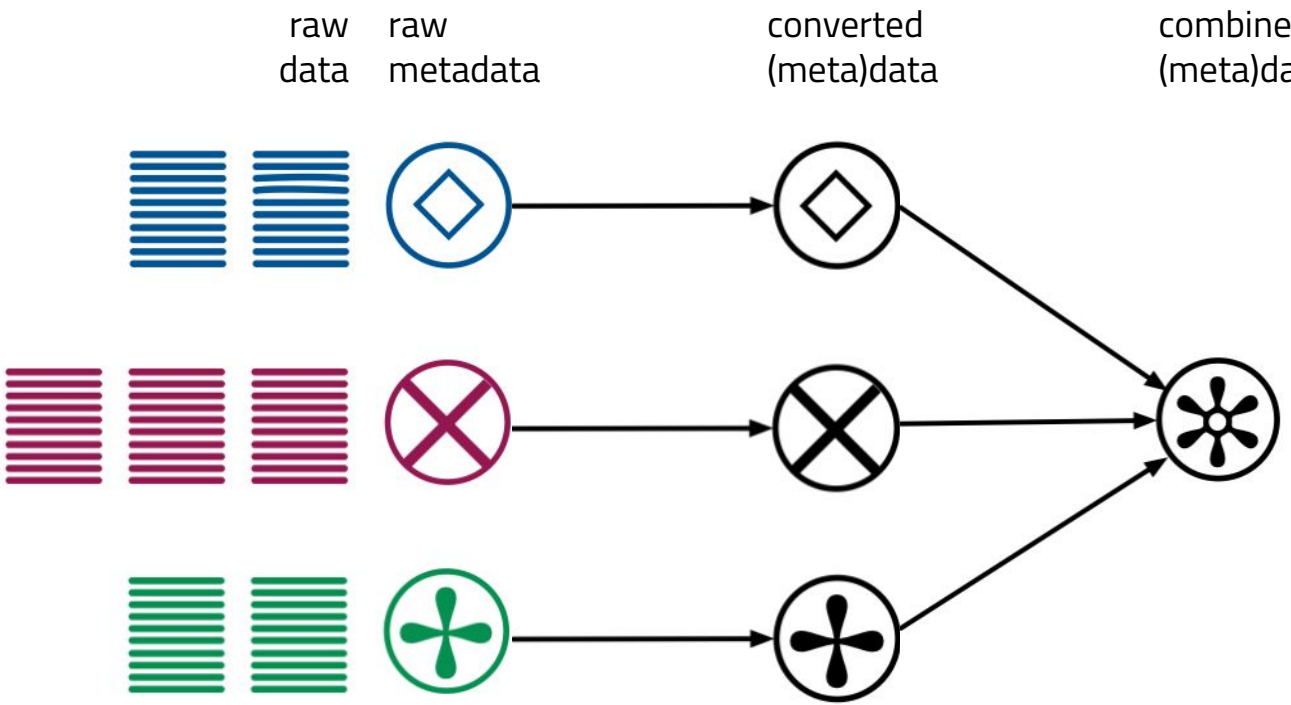
- Defines an individual datum
- Has a type: primitive (str, bool, float), reference (e.g. another section or quantity), enum, datetime, URL, file
- Has a shape: list, scalar, vector, matrix
- *fields*, *properties*, *attributes*, *references* in other schema systems

Sub section (properties of sections):

- Defines a *parent-child* (or whole-part) relationship between instances of a *parent* section definition and a *child* section definition
- Is a property of the parent section definition and relates to a child section definition
- *child*, *content* in other schema systems

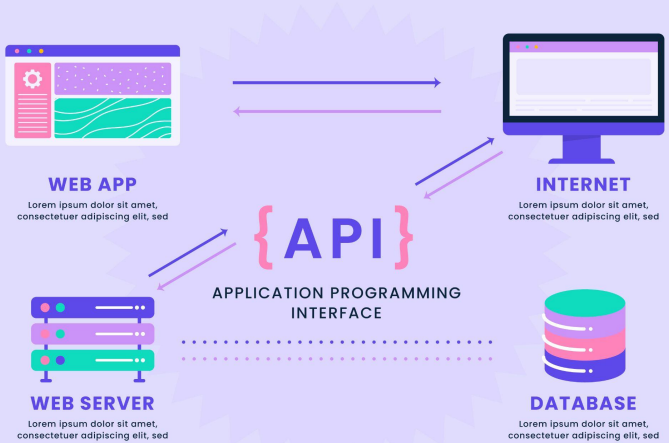
Example NOMAD *Schema* and *Processed Data*

Heterogeneous (Meta)data: Converting / Parsing



Related

- database schemas
- dataclasses in programming, e.g. pydantic
- ontologies and semantic web
- JSON schema, JSON-ld, linked-ml, XML schema, …
- neXus



Working with APIs

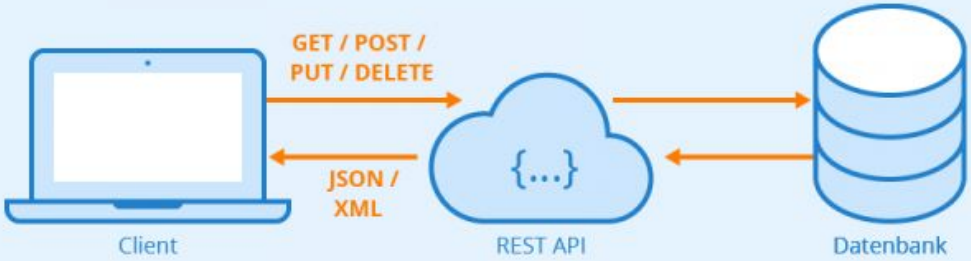
https://de.freepik.com/vektoren-kostenlos/flaches-design-api-infografik_25001333.htm

RESTful APIs

- **A**pplication **P**rogramming **I**nterface
  - A contract that defines the possible communication between two software components independent of the implementation of either component
- **RE**presentational **S**tate **T**ransfer-ful APIs
  - REST is a popular architectural style for building web-services
  - based on HTTP, URLs, (and JSON)
  - application **state** is **represented** (as JSON) in resources (the R in URL) that can be **transferred** from and to the server
  - no in memory sessions, each requests is handled independently
  - other styles exist, e.g. SOAP, graphQL

RESTful APIs

- **A**pplication **P**rogramming
  - A contract that define
    components indepen
- **RE**presentational **S**tate **Tr**
  - REST is a popular arc
  - based on HTTP, URLs
  - application **state** is **re**
    can be **transferred** fr
  - no in memory session
  - other styles exist, e.g



HTTP, URLs, JSON

- **H**yper **T**ext **T**ransfer **P**rotocol
  - application-layer network protocol
  - simple operations to modify *resources* GET,
- URL/URIs (**U**niform **R**esource **L**ocator/**I**dentifier) internet
  - protocol://network-location/p/a/t/h?query=
  - https://docs.google.com/presentation/d/1p oFLbzpH_V4WtUCk/edit?usp=sharing
- **J**ava**s**cript **O**bject **N**otation


```
{
  "employee": {
    "name": "John Doe",
    "age": 35,
    "address": {
      "street": "123 Main St",
      "city": "Anytown",
      "state": "CA",
      "zip": "12345"
    },
    "phoneNumbers": [
      {
        "type": "home",
        "number": "555-555-1234"
      },
      {
        "type": "work",
        "number": "555-555-5678"
      }
    ],
    "email": "john.doe@example.com",
    "is_active": true,
    "hire_date": "2022-01-01T00:00:00Z"
  }
}
```

Tools

- browser, e.g. *chrome*
  - GET's URLs by default
  - the developer tools of all browsers have a "network"-tab to observe the HTTP communication performed by the current website
- command-line tools, e.g *curl*, *wget*
  - curl "https://en.wikipedia.org/wiki/Web_service"
- generic HTTP libraries, e.g. *requests*
  - print(requests.get("https://en.wikipedia.org/wiki/Web_service").text)
- application specific libraries, e.g. *nomad-lab*

NOMAD API as an example

- dashboard as documentation and tool
- different APIs, e.g. OPTIMADE
- code-snippets can be copied from many NOMAD UI pages

GET /entries

https://nomad-lab.eu/prod/v1/api/v1/entries**?**

page_size**=**1**&**

q**=**results.material.n_elements__gt__4**&**

order_by**=**publish_time&order=desc**&**

include**=**results.material.elements**&**include=authors.name

open in browser

Pagination

- HTTP is a simple request/response protocol and response need to be limited
- For resources that represent a list of *items*
- Only a few *items* (page) from all existing *items* per request/response
- Different styles of pagination
  - index-based: “get page 5”, “get page 6”, etc., e.g. google search results
  - value-based: “get the page after value XY”, e.g. twitter timeline
- Large datasets can be retrieved by programmatically *paginate* with multiple requests

Requests

```python
import requests
import json

base_url = 'http://nomad-lab.eu/prod/v1/api/v1'

response = requests.post(
    f'{base_url}/entries/query',
    json={
        'query': {
            'results.material.elements': {
                'all': ['Ti', 'O']
            }
        },
        'pagination': {
            'page_size': 1
        },
        'required': {
            'include': ['entry_id']
        }
    })

response_json = response.json()
print(json.dumps(response.json(), indent=2))
```


```
{
  "owner": "public",
  "query": {
    "name": "results.material.elements",
    "value": {
      "all": ["Ti", "O"]
    }
  },
  "pagination": {
    "page_size": 1,
    "order_by": "entry_id",
    "order": "asc",
    "total": 17957,
    "next_page_after_value": "--SZVYOxA2jTu_L-mSxefSQFmeyF"
  },
  "required": {
    "include": [ "entry_id"]
  },
  "data": [
    {
      "entry_id": "--SZVYOxA2jTu_L-mSxefSQFmeyF"
    }
  ]
}
```

Requests: Pagination

```python
import requests

base_url = 'http://nomad-lab.eu/prod/v1/api/v1'
json_body = {
    'query': {
        'results.material.elements': {
            'all': ['Ti', 'O']
        }
    },
    'pagination': {
        'page_size': 10
    },
    'required': {
        'include': [
            'results.material.chemical_formula_hill'
        ]
    }
}
```

```python
formulas = set()

while len(formulas) < 100:
    response = requests.post(
        f'{base_url}/entries/query', json=json_body)
    response_json = response.json()

    for data in response_json['data']:
        formulas.add(
            data['results']['material']['chemical_formula_hill'])

    next_value = response_json['pagination'].get(
        'next_page_after_value')
    if not next_value:
        break
    json_body['pagination']['page_after_value'] = next_value

print(formulas)
```

Requests: NOMAD's archive API

```python
import requests
import json

base_url = 'http://nomad-lab.eu/prod/v1/api/v1'

response = requests.post(
    f'{base_url}/entries/query',
    json={
        'query': {
            'results.material.elements': {
                'all': ['Ti', 'O']
            }
        },
        'pagination': {
            'page_size': 1
        },
        'required': {
            'include': ['entry_id']
        }
    })

response_json = response.json()

first_entry_id = response_json['data'][0]['entry_id']
response = requests.post(
    f'{base_url}/entries/{first_entry_id}/archive/query',
    json={
        'required': {
            'workflow': {
                'calculation_result_ref': {
                    'energy': '*',
                    'system_ref': {
                        'chemical_composition': '*'
                    }
                }
            }
        }
    })
print(json.dumps( response.json(), indent=2))
```

Archive Query

```
from nomad.client import ArchiveQuery
from nomad.metainfo import units

query = ArchiveQuery(
    query={
        'results.method.simulation.program_name': 'VASP',
        'results.material.elements': ['Ti', 'O'],
        'results.properties.geometry_optimization': {
            'final_energy_difference:lt': 1e-22,
        }
    },
    required={
        'workflow': {
            'calculation_result_ref': {
                'energy': '*',
                'system_ref': {
                    'chemical_composition_reduced': '*'
                }}}})
```

Archive Query

```
for result in query.download(100):
    calc = result.workflow[0].calculation_result_ref
    formula = calc.system_ref.chemical_composition_reduced

    if calc.energy.total:
        total_energy = calc.energy.total.value.to(units.eV)
    else:
        total_energy = 'N/A'

    print(f'{formula}: {total_energy}')
```

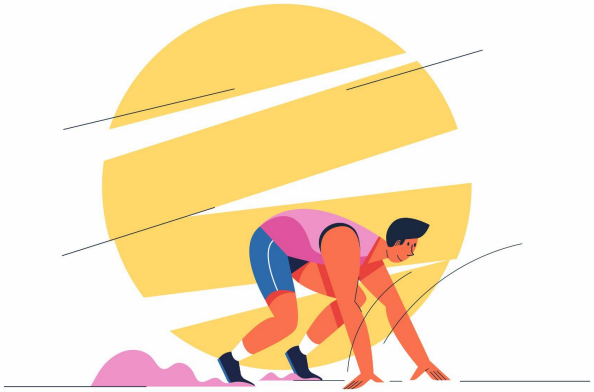
Conclusions

- RDM is important, not just for your institute or your community, but also for yourself
- Tools like NOMAD can help you with RDM and you work
- Data modeling, e.g. with schemas, is the basis for FAIR data
- APIs are an important tool to master to programmatically analyze existing data

- NOMAD
- NOMAD Video Tutorials
- NOMAD Documentation

Questions?

https://de.freepik.com/vektoren-kostenlos/weisser-fragezeichenhintergrund-im-minimalen-stil_8162788.htm



Exercises

https://de.freepik.com/vektoren-kostenlos/athlet-laufmann-oder-sprinter-mann-sporttraining-sportliche-zeichentrickfigur-illustration_18779083.htm

Search and Access data via API

- https://tinyurl.com/ikzwinter
- search: https://nomad-lab.eu/prod/v1/gui/search/entries
- API: https://nomad-lab.eu/prod/v1/api/v1/extensions/docs
- NOMAD Video Tutorials

- Access calculations about ternary systems including two given elements via API
  - using your browser
  - using requests and Python
  - use the archive query, e.g. from the AI toolkit
- Create a NOMAD custom schema and data to “publish” the results of the previous section. You can follow our schema documentation and use the NOMAD v1.1 beta that supports it.