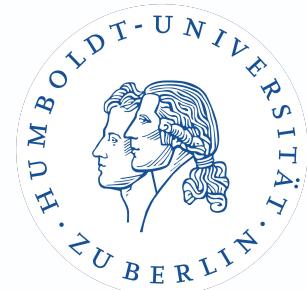


FAIRmat Tutorial 13: NOMAD for Experimental Data Management in Synthesis

Andrea Albino, Hampus Näsström, Sarthak Kapoor, Sebastian Brückner

May 15th 2024, 13:00 – 16:00h, Zoom

github.com/FAIRmat-NFDI/AreaA-Examples/tutorial13



NOMAD for Experimental Data Management in Synthesis

Part I Introduction

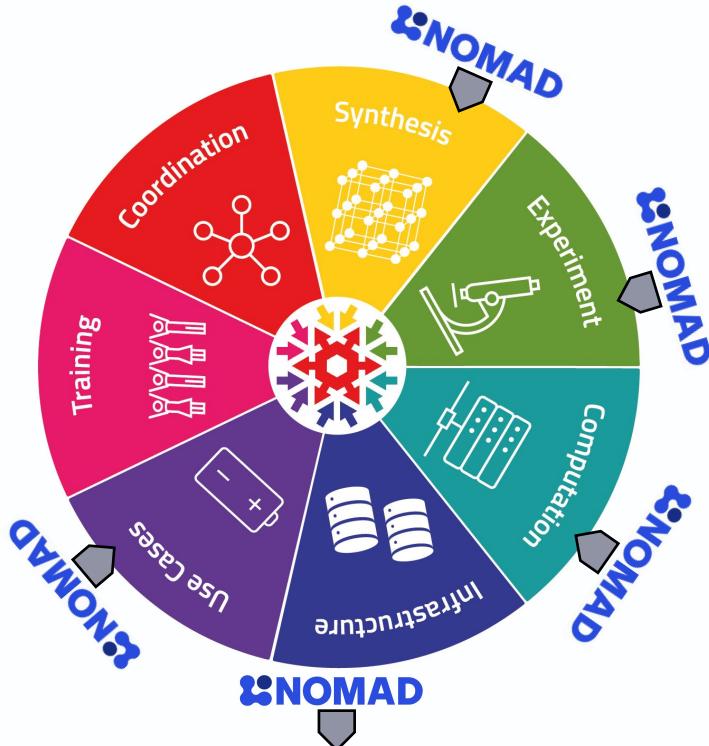
by

Sebastian Brückner

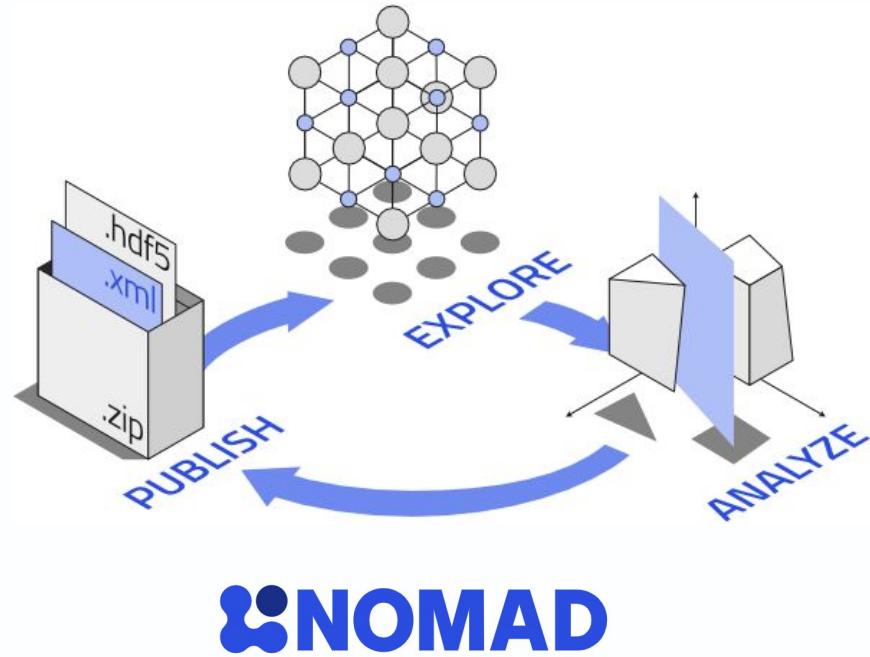


What are NFDI, FAIRmat, NOMAD

FAIRmat is the **NFDI** 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

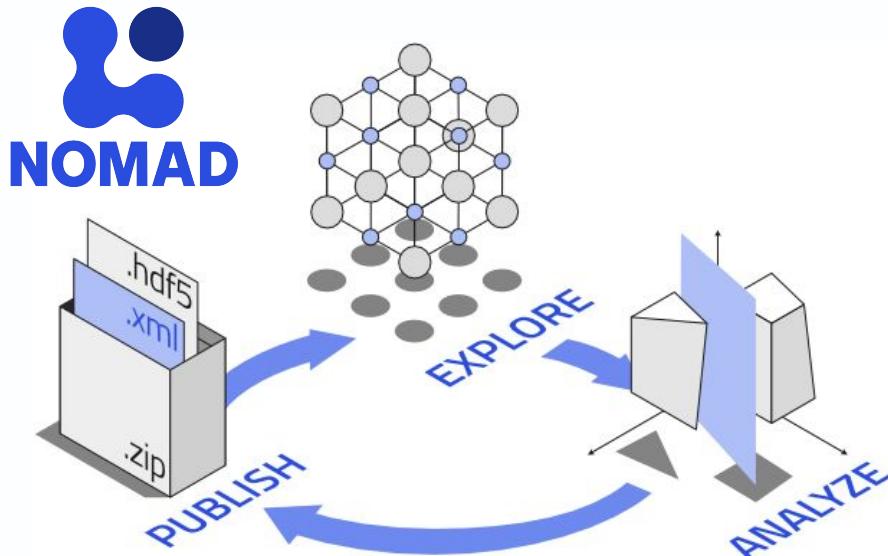




NOMAD comes in 2 flavours

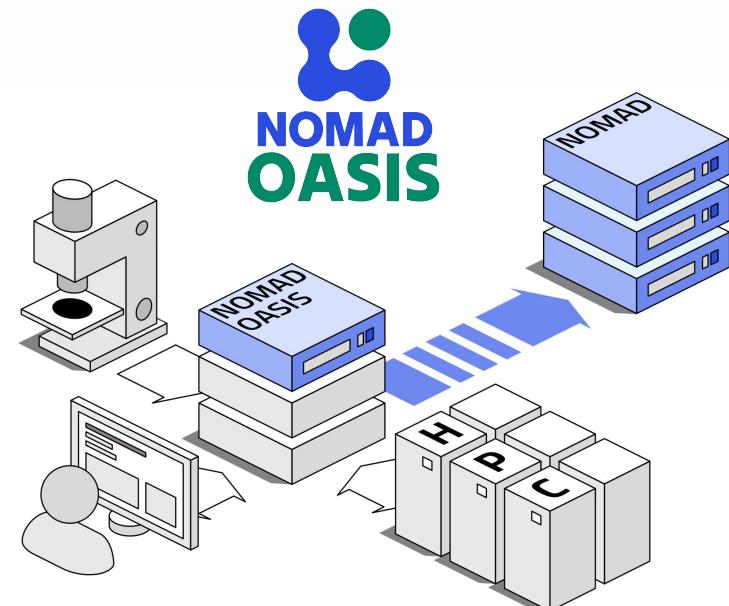
Publish your data and analysis

Archive Repository



Manage your lab

Local management tool | ELN

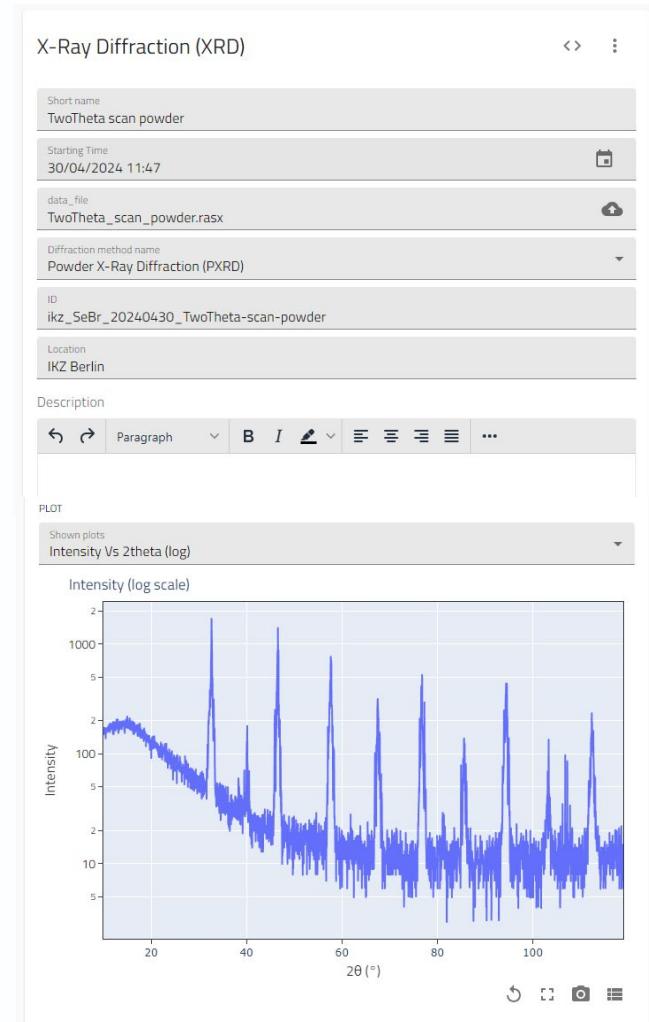
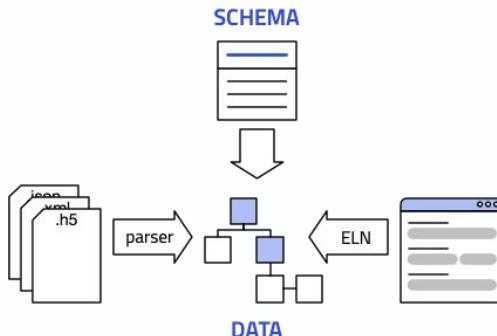




NOMAD Oasis features

Data Schema:

- Formal description of the data model
 - defines the data structure
- Defines ELN functionality, plots, etc.
 - Tutorial part 2&3



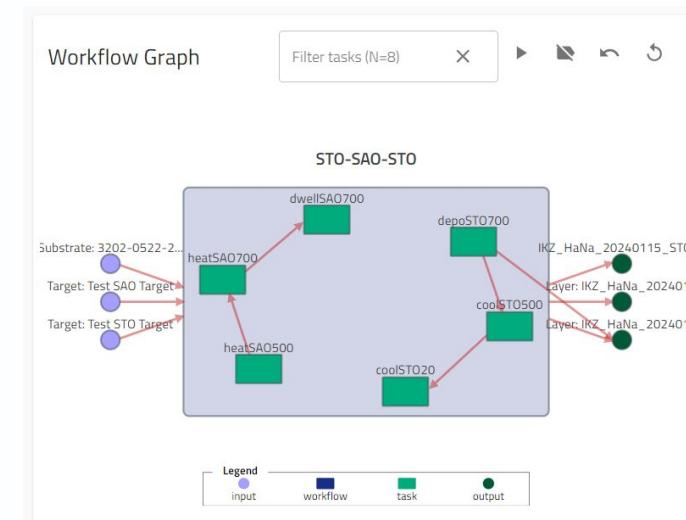


NOMAD Oasis features

General data model → NOMAD's Base Sections

- Entity-Activity model for Materials Science → **Tutorial part 2**
 - Allows to build general tools: Entity-References/History Card, Workflow-Visualizer, general search apps

History			
2/2 activities			☰
Name	Entry type	Entry creation time	
XRD 0240115 STO-SAO-STO-PLD	ELNXRayDiffraction	13.5.2024, 10:49:35	→
STO-SAO-STO	IKZPulsedLaserDeposition	5.4.2024, 14:59:56	→





NOMAD Oasis features

NOMAD Oasis plugin mechanism

Python code, automatized data processing, file parsing, custom Apps → part 3

The screenshot shows the NOMAD Oasis web interface. At the top, there is a navigation bar with 'PUBLISH', 'EXPLORE', 'ANALYZE', and 'ABOUT' dropdowns, a user profile icon, and links for 'Welcome Andrea Albino', 'LOGOUT', and 'UNITS'. Below the navigation is a breadcrumb trail 'Your uploads / Upload'. The main area has tabs for 'OVERVIEW' and 'FILES'. Under 'OVERVIEW', there is a section for an 'unnamed upload' with an ID 'y6POeeDuRB24pz5JhdB7Nw'. It includes a file list, search, and filter icons. Below this is a step-by-step guide: '1 Prepare and upload your files'. The text says: 'Here you can upload files. Top-level .zip/.tar files will be uncompressed automatically. For more information, see our documentation on [uploading files](#) or view the [supported codes](#). Optionally, you can also create an entry from built-in or uploaded schemas. Please take a look at our documentation on [schemas](#)'. A large blue button at the bottom says 'DROP FILES HERE OR CLICK TO OPEN DIALOG' with a cloud icon. To the right is a 'CREATE FROM SCHEMA' button. An arrow points from the text 'Drop files here or click to open dialog' to the 'DROP FILES HERE OR CLICK TO OPEN DIALOG' button.

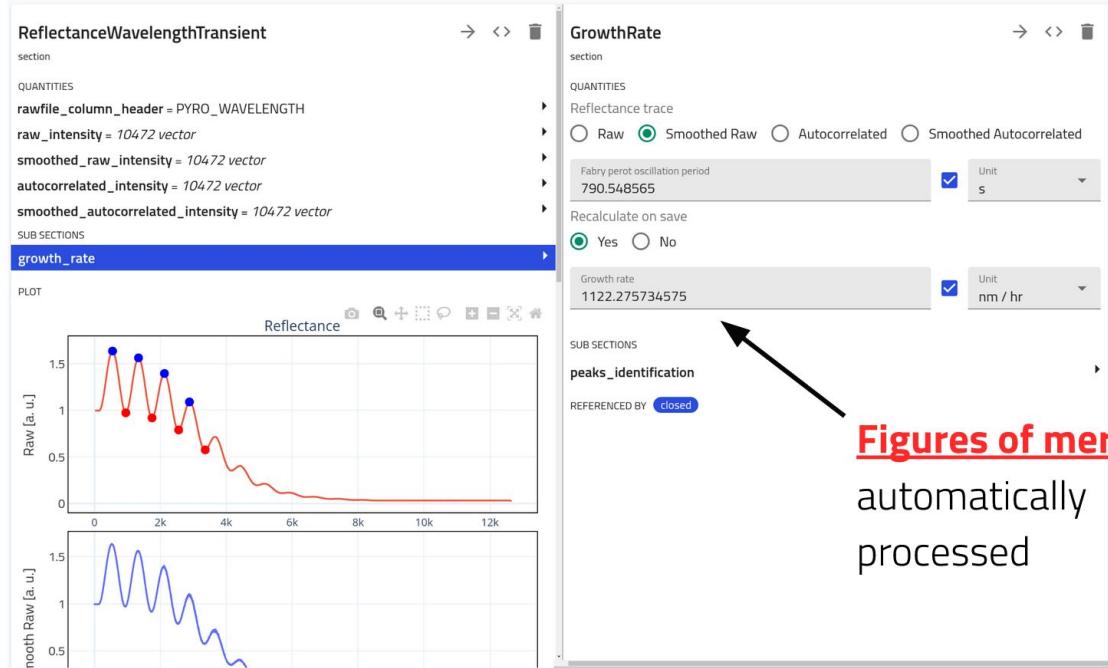




NOMAD Oasis features

NOMAD Oasis plugin mechanism

Python code, automatized data processing, file parsing, custom Apps → part 3



Figures of merit
automatically
processed



NOMAD Oasis features

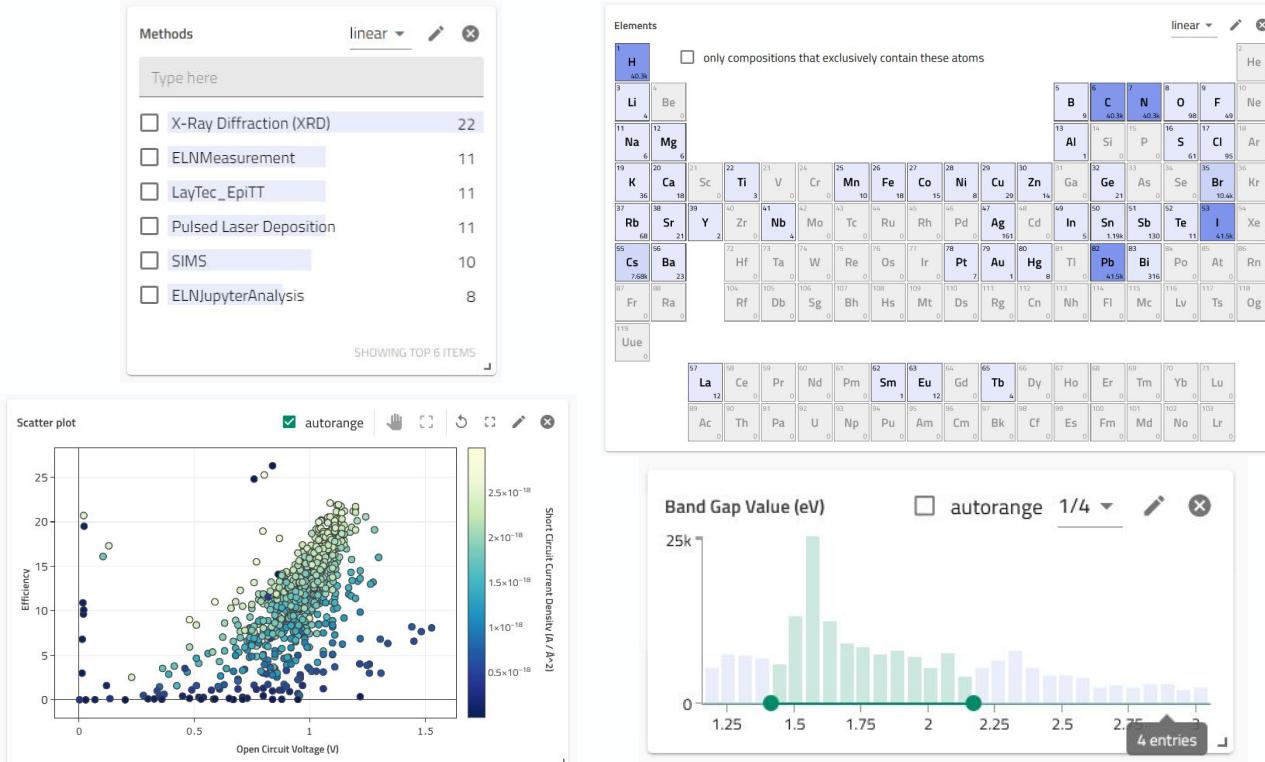
Explore Data: Custom Search Apps

The screenshot shows a search interface for 'PLD Layer'. At the top, there's a navigation bar with 'PUBLISH', 'EXPLORE', 'ANALYZE', and 'ABOUT' dropdowns. Below that is a logo for 'PLD Layer' and a 'FILTERS' sidebar with options like 'Material', 'Elements / Formula', 'Electronic Lab Notebook', 'User Defined Quantities', 'Author / Origin / Dataset', and 'Visibility / IDs / Schema'. A large search dropdown is open, showing categories: 'EXPERIMENT', 'USE CASES', and 'ALL'. The 'EXPERIMENT' category contains links for 'PLD Layers', 'All experiments', 'Measurements | Processes', and 'Samples'. The 'USE CASES' category contains a link for 'Solar Cells'. The 'ALL' category contains a link for 'Entries'. To the right of the search dropdown, there's a small periodic table with elements Scandium (Sc), Titanium (Ti), Yttrium (Y), Zirconium (Zr), Hafnium (Hf), and Rhenium (Rf) highlighted.



NOMAD Oasis features

Explore Data: Custom Search Apps

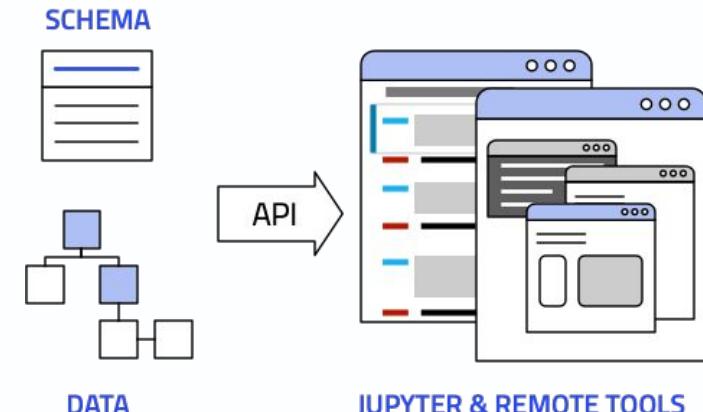
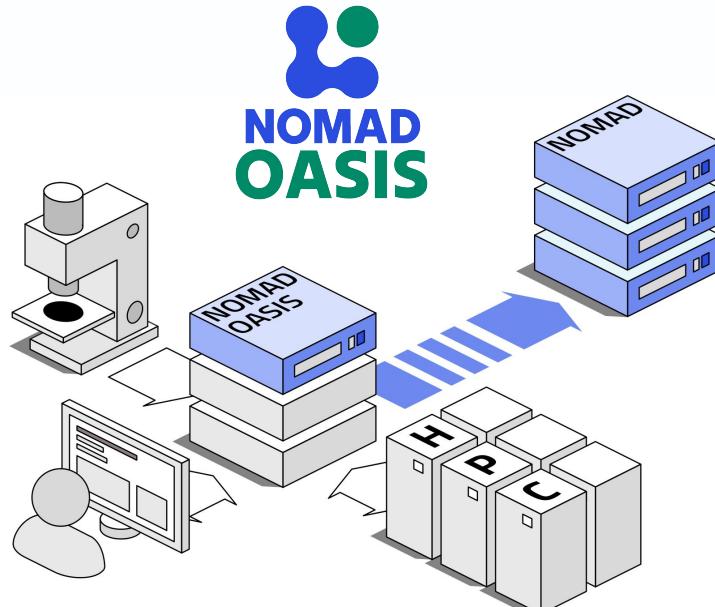




NOMAD Oasis features

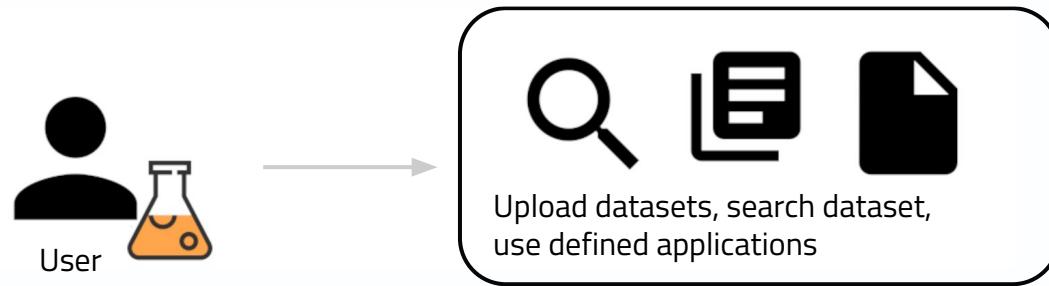
NOMAD Oasis custom image

A deployed NOMAD instance which includes the custom built plugins →part 4





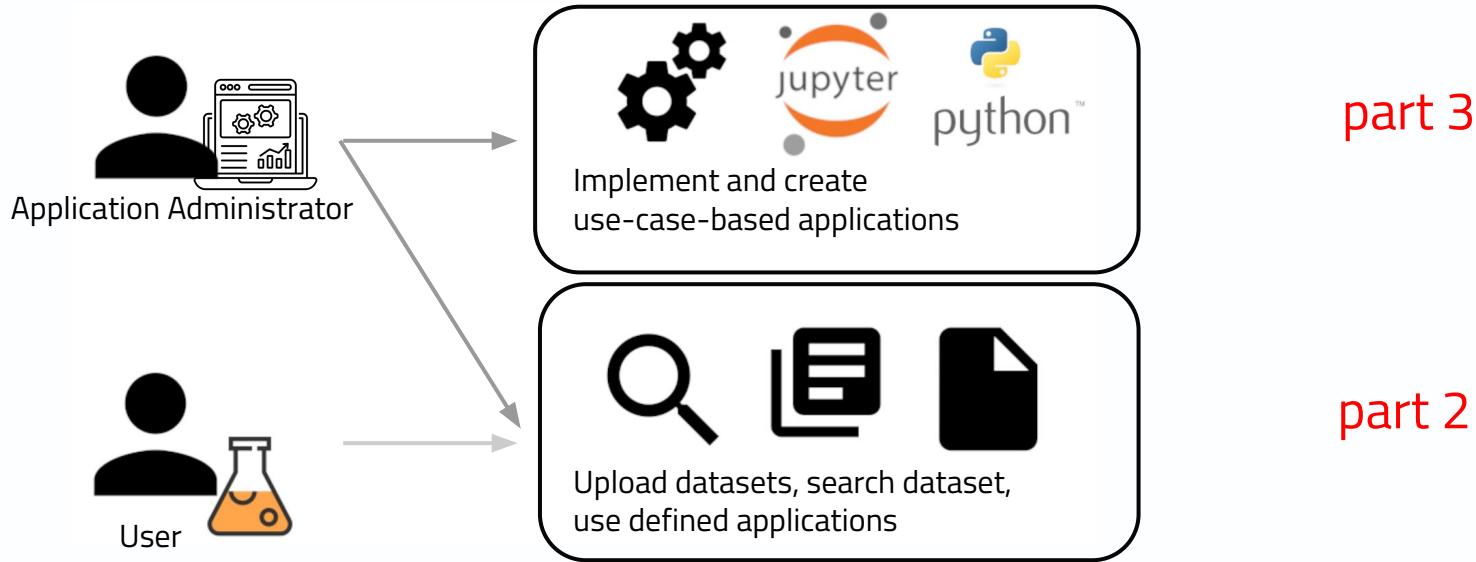
RDM Infrastructure: Roles & Responsibilities



part 2

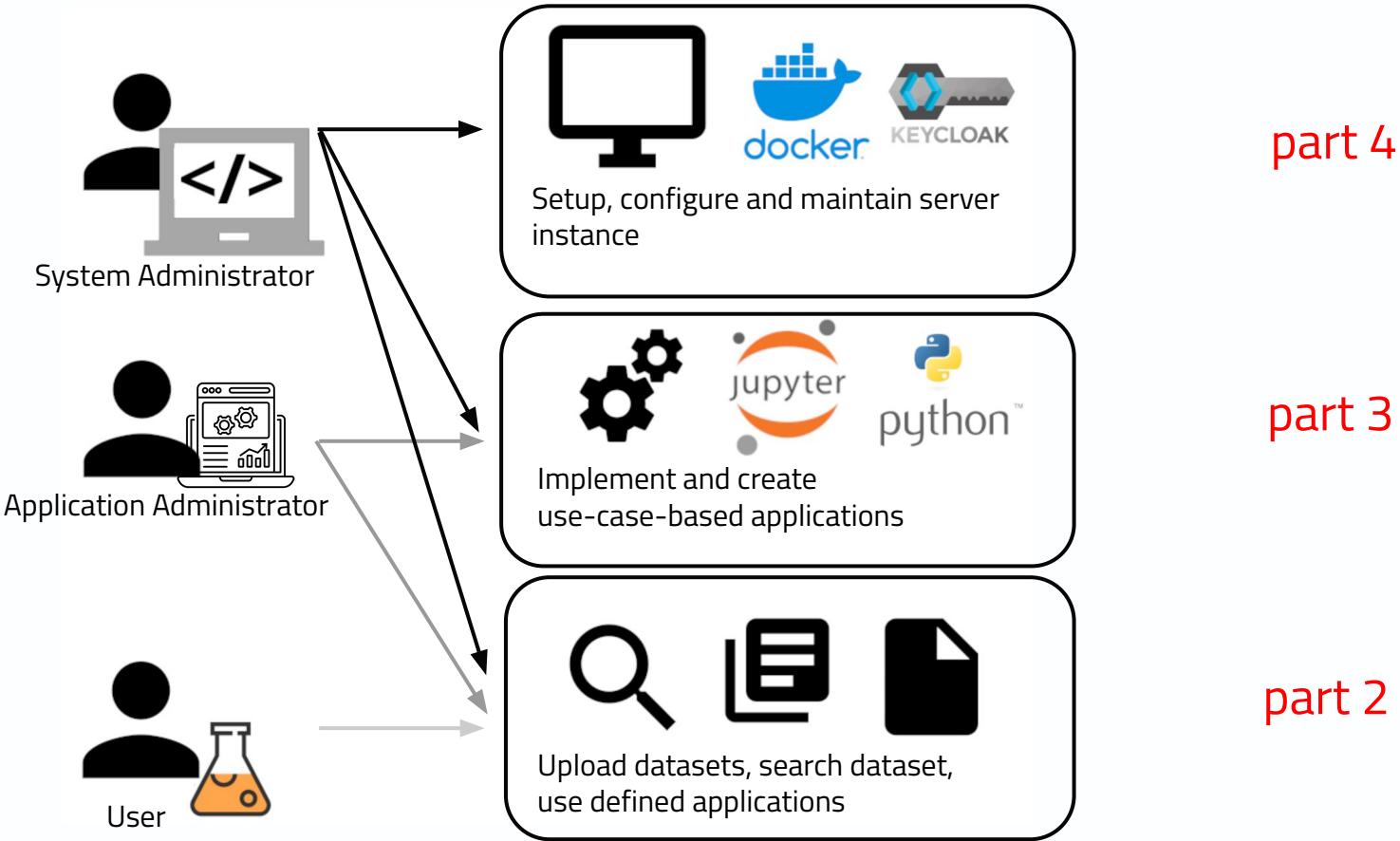


RDM Infrastructure: Roles & Responsibilities





RDM Infrastructure: Roles & Responsibilities



part 4

part 3

part 2



Data Stewardship + FAIRmat

FAIRmat Area A Team:



Sebastian Brückner



Andrea Albino



Hampus Näsström



Sarthak Kapoor



Whole FAIRmat team
(~30 developers and
domain experts)



Tutorial Schedule

Time	Section	Content	Target audience	Requirements & Skills
13:00 – 13:10	1. Introduction	<ul style="list-style-type: none">• FAIRmat & NOMAD (Oasis)• RDM Roles & Responsibilities	All users	
13:10 – 13:20	2.A NOMAD's Base Sections	<ul style="list-style-type: none">• Data modelling and structure• Schema types	All users	
13:20 – 13:50	2.B Built-in Schemas for ELN	<ul style="list-style-type: none">• Basic NOMAD usage• Exploring built-in Schemas	All users	NOMAD account
13:50 – 14:00	Break			
14:00 – 15:00	3. Customization - Schema & Plugin development	<ul style="list-style-type: none">• NOMAD plugin development	Data Stewards & Data Scientists	+ basic Python, + GitHub account
15:00 – 15:30	4. Deploying your NOMAD plugins	<ul style="list-style-type: none">• Write a NOMAD Oasis image with plugins and deploy it	System Admins	+ (Computer with Docker)
15:30 – 16:00	Open Q&A			



Resources:

- Tutorial 13 homepage: github.com/FAIRmat-NFDI/AreaA-Examples/tutorial13
- Discord: discord.com/NOMAD
- NOMAD – documentation, discord, forum: www.nomad-lab.eu
- FAIRmat – tutorials, Youtube, events, domain experts: www.fairmat-nfdi.eu

NOMAD



NOMAD for Experimental Data Management in Synthesis

Part II
NOMAD's Base Sections
by
Andrea Albino



Outline

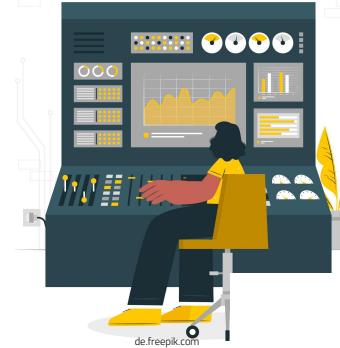
- 🧪 Data & Metadata
- 🧪 Base Classes
- 🧪 Implementing Structured Data
- 🧪 Hands On Tutorial



Research Data Management

Raw Data

- Log files from instruments
- Recipe files from process software



de.freepik.com

ELN

- Manually recorded data and metadata



de.freepik.com

Data Analysis

- Post processing software
- User-tailored scripts



de.freepik.com



Research Data Management

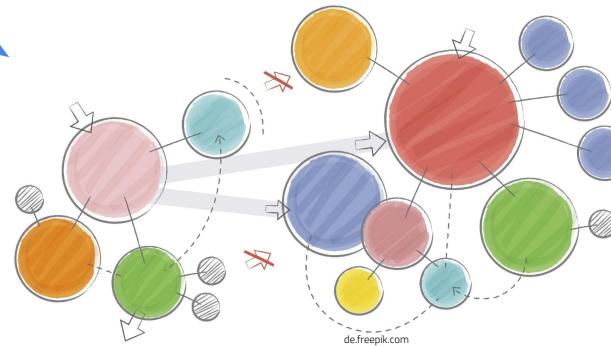
Raw Data

Combine independent information spaces
towards an organic and scalable dataset

ELN

Data Model to connect
Data & Metadata

Data Analysis

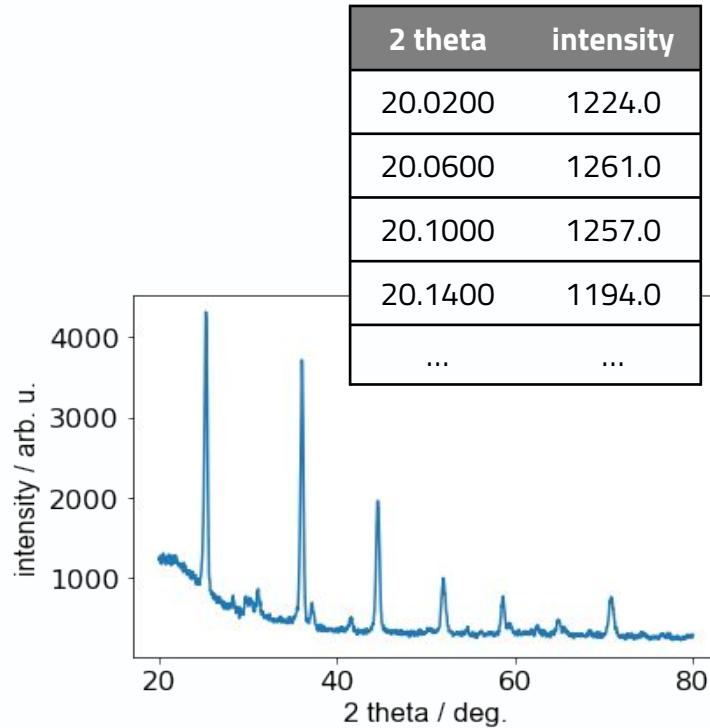




Data & Metadata

Data:

actual content of information

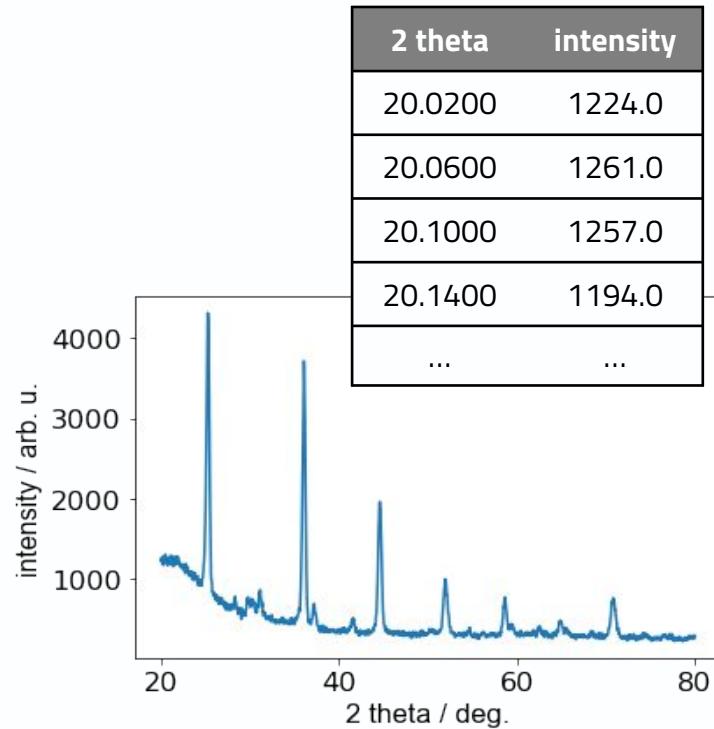




Data & Metadata

Data:

actual content of information



Metadata:

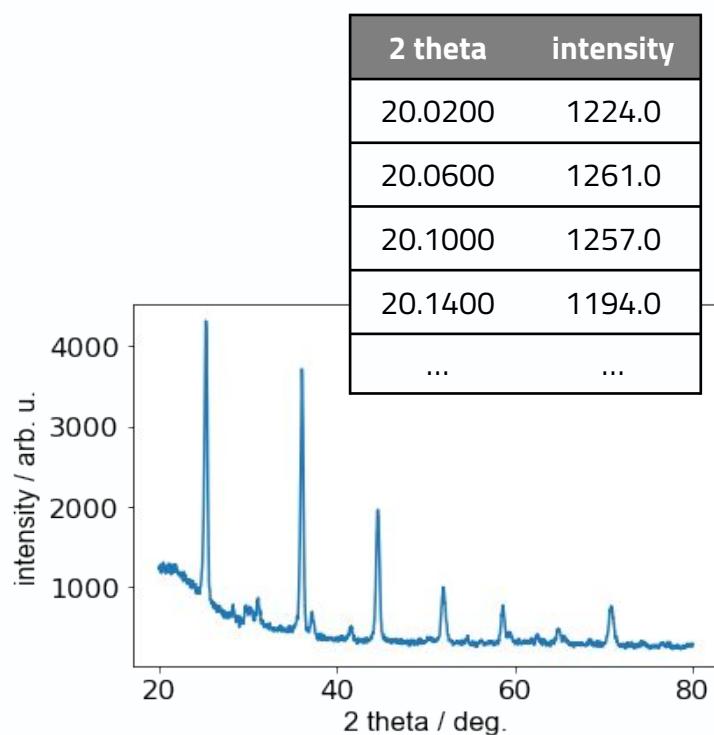
data that provides information about other data



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:

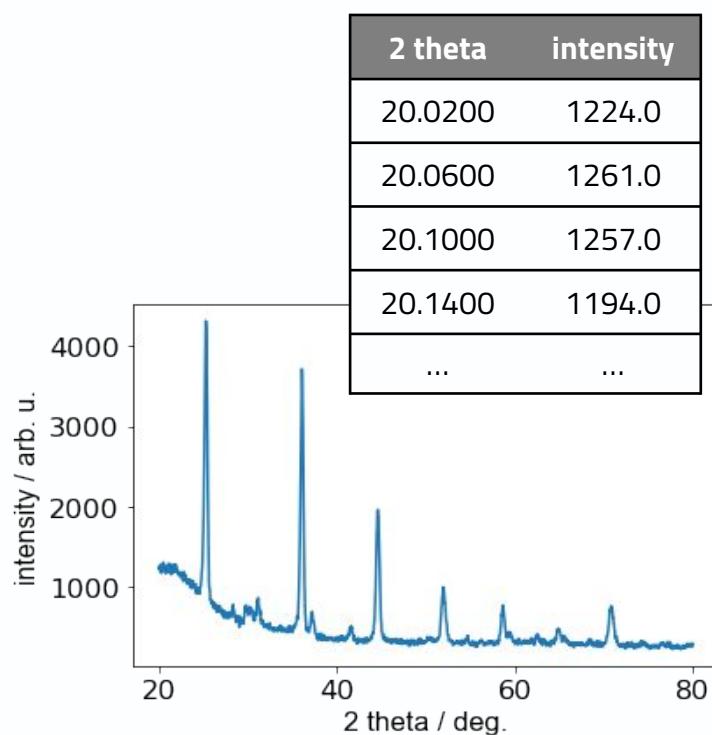
description: The count at each 2-theta value.
unit: dimensionless
type: float



Data & Metadata

Data:

actual content of information



Structural Metadata:

provides information about containers of data

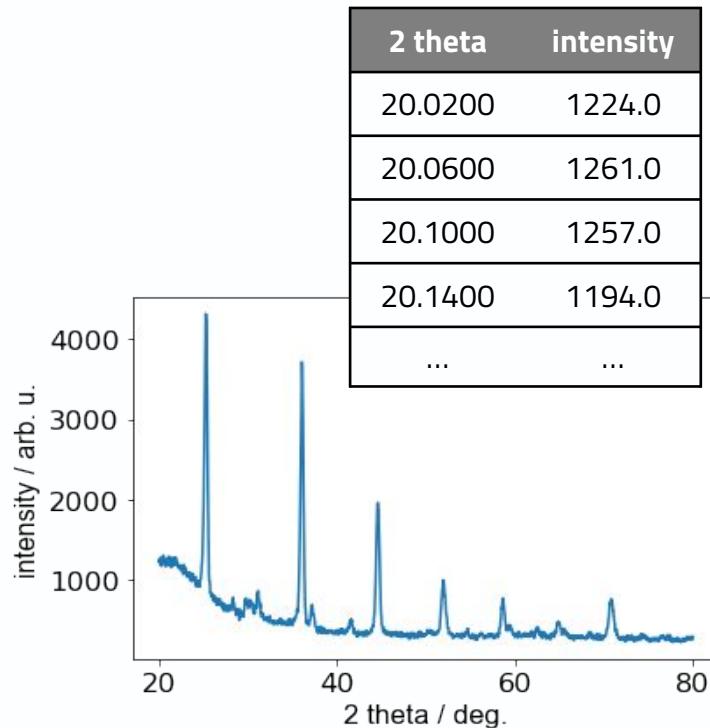
Entry	Metadata
section	section
SUB SECTIONS	QUANTITIES
results	upload_id = 3LvMkGAzTi6scpDwrBL93w
metadata	upload_create_time = 3/10/2023, 1:39:17 PM
data	entry_id = LxJrFzPQBB72nj9Q8UG6fNYqqeXW
REFERENCED BY	entry_name = mixed.archive.json
	entry_type = MyOverallProcess
	entry_hash = nBKeoHJIvMeCz8Ds7lVeUPb0aMJz
	entry_create_time = 3/10/2023, 1:50:44 PM
	parser_name = parsers/archive
	mainfile = mixed.archive.json
	files = 7 list
	published = false
	with_embargo = false
	nomad_version = 1.1.9.dev6+g122278d68
	license = CC BY 4.0
	main_author = 8e6f6796-d4f0-409a-85d2-1b8810c8c3a1



Data & Metadata

Data:

actual content of information



Descriptive Metadata:

provides important context about data

Date of the experiment?

Temperature?

Instrument specifications?

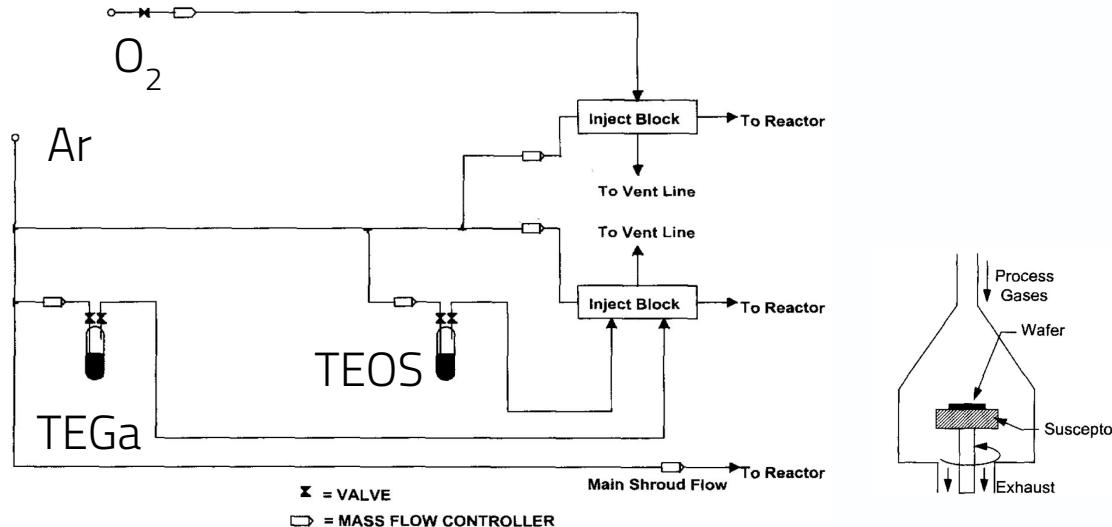
Specimen description?

Humidity in the room?



Fitting Data in “Base Classes”

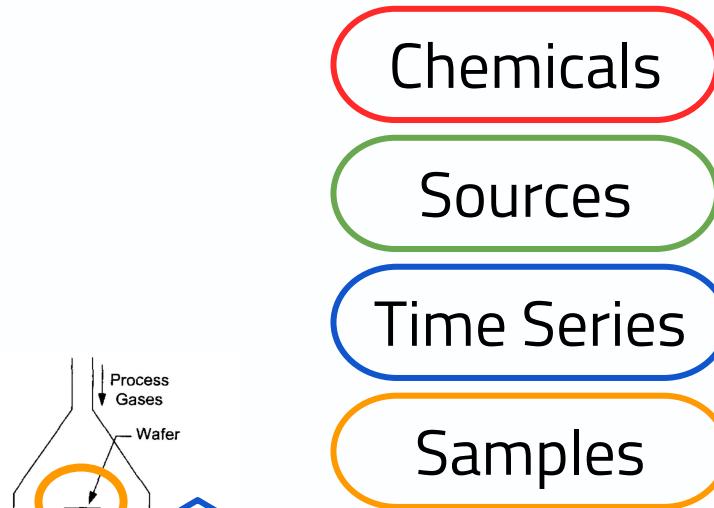
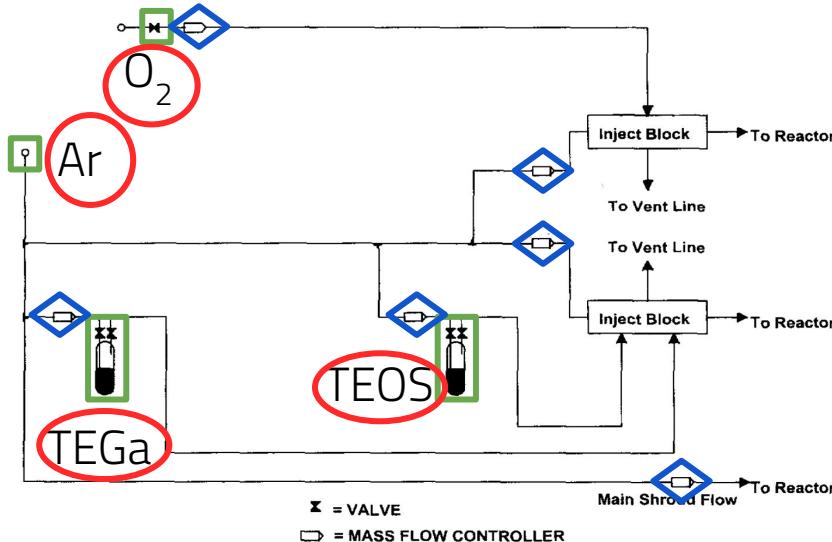
Boiling down the data to elemental building blocks:





Fitting Data in “Base Classes”

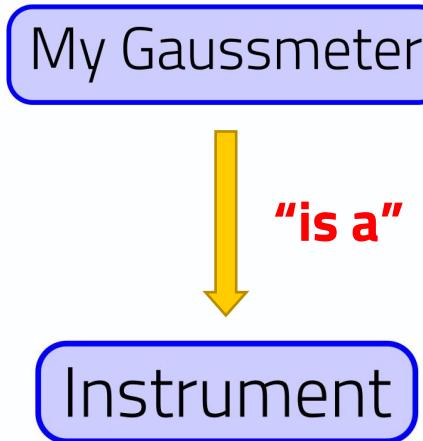
Boiling down the data to elemental building blocks:





Using Base Classes

Inheritance



"My Gaussmeter"
inherits
the properties of
"Instrument"



Using Base Classes

Inheritance

&

Composition

My Gaussmeter



“is a”

Instrument

“My Gaussmeter”
inherits
the properties of
“Instrument”

Experiment



“has a”

My Gaussmeter

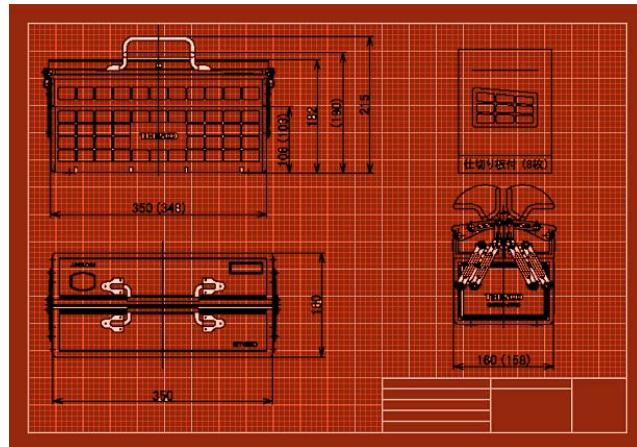
“Experiment”
is composed by
an “Instrument”
(a “User”, a “Sample”, etc.)



Schema and Templates

Schema:

formal description of data,
data types, and data file structure,
such as XML files



Blueprint of a toolbox

Template:

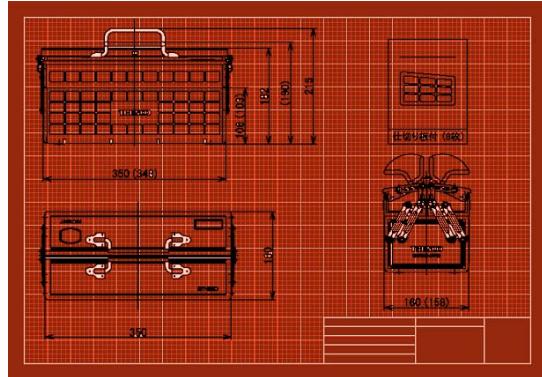
physical object, or instance,
generated from a schema



Toolbox tailored on specific set of tools



Schema and Templates



Schema



Template

+



Data



Structured
Archive file





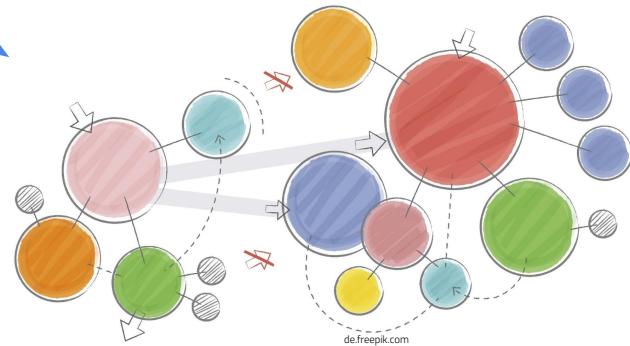
Research Data Management

Raw Data

ELN

Data Analysis

Deliver unified analysis across data
and reusable ML solution at scale



Data Model to connect
Data & Metadata



Search
Visualization
Analysis



Implementing Structured Data

	Type	Implementation	Advantage
0	Unstructured	-	-
1	Built-in Classes	available in NOMAD	out-of-the-box solution
2	Custom Schema	text file (.yaml format)	fast & flexible deployment
3	User Defined Plugin	Python code	automation of parsing
4	Community Standard	Git repositories	FAIR data



Implementing Structured Data

Type	Implementation	Advantage
0 Unstructured	-	-
1 Built-in Classes	available in NOMAD	out-of-the-box solution
2 Custom Schema	text file (.yaml format)	fast & flexible deployment
3 User Defined Plugin	Python code	automation of parsing
4 Community Standard	Git repositories	FAIR data

Hands-on Tutorial by Sarthak Kapoor



Implementing Structured Data

	Type	Implementation	Advantage
0	Unstructured	-	-
1	Built-in Classes	available in NOMAD	out-of-the-box solution
2	Custom Schema	text file (.yaml format)	fast & flexible deployment
3	User Defined Plugin	Python code	automation of parsing
4	Community Standard	Git repositories	FAIR data

Hands-on Tutorial by Hampus Näsström



Implementing Structured Data

Type	Implementation	Advantage
0 Unstructured	-	-
1 Built-in Classes	available in NOMAD	out-of-the-box solution
2 Custom Schema	text file (.yaml format)	fast & flexible deployment
3 User Defined Plugin	Python code	automation of parsing
4 Community Standard	Git repositories	FAIR data

GitHub repos will be showed in this Tutorial

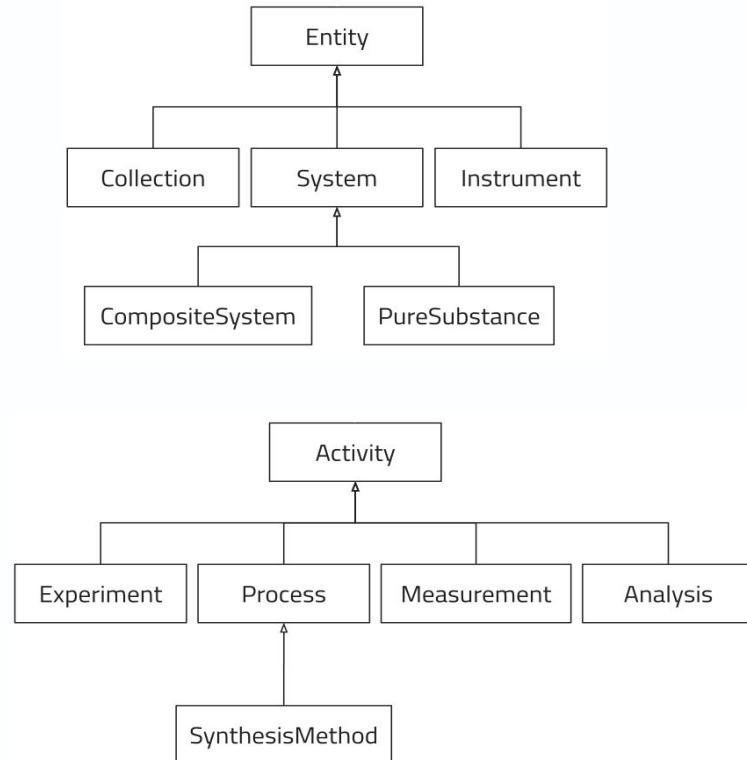


0. Built-in Classes

nomad-lab.eu/docs/howto/customization/base_sections

The screenshot shows the Nomad Lab interface with the following details:

- Top navigation bar: PUBLISH, EXPLORE, ANALYZE, ABOUT, Welcome Andrea Alibio, LOGOUT, UNITS.
- Left sidebar: Your uploads / Upload, OVERVIEW, 1. Prepare and upload files or view documents, 2. Process, 3. Edit automatically, 4. Publish, 5. Publish.
- Main area: NOMAD Measurements, X-Ray Diffraction (XRD) is highlighted.
- Bottom dialog box: Title 'unnamed upload', Subtitle 'upload', Description 'Here you can upload files or view documents.', List of built-in schemas: NOMAD Measurements, X-Ray Diffraction (XRD), Basic ELN, Analysis ELN, Basic ELN, Collection ELN, Experiment ELN, Generic Sample ELN, Instrument ELN, Material Processing ELN, Measurement ELN, Substance ELN. A sub-section 'built-in schema' shows 'X-Ray Diffraction (XRD)'.
- Buttons at the bottom: CANCEL, CREATE.





1. Custom Schema

YAML is a Markup Language defining hierarchy with indentation

```
definitions:  
  - name: 'Tutorial 13 sintering schema'  
    sections:  
      - TemperatureRamp:  
        base_sections:  
          - nomad.datamodel.metainfo.basesections.ProcessStep  
        quantities:  
          - initial_temperature:  
            type: np.float64  
            unit: celsius  
            description: "initial temperature set for ramp"  
            annotations:  
              - eln:  
                component: NumberEditQuantity  
                defaultDisplayUnit: celsius  
      - Sintering:  
        base_sections:  
          - nomad.datamodel.metainfo.basesections.Process  
          - nomad.datamodel.data.EntryData  
        sub_sections:  
          - steps:  
            - repeats: True  
            - section: '#/TemperatureRamp'
```

- extensible
- structured
- plain text
- human readable



1. Custom Schema

YAML is a Markup Language defining hierarchy with indentation

```
definitions:  
  - name: 'Tutorial 13 sintering schema'  
    sections:  
      - TemperatureRamp:  
        base_sections:  
          - nomad.datamodel.metainfo.basesections.ProcessStep  
        quantities:  
          - initial_temperature:  
            type: np.float64  
            unit: celsius  
            description: "initial temperature set for ramp"  
            m_annotations:  
              - eln:  
                component: NumberEditQuantity  
                defaultDisplayUnit: celsius  
      - Sintering:  
        base_sections:  
          - nomad.datamodel.metainfo.basesections.Process  
          - nomad.datamodel.data.EntryData  
        sub_sections:  
          - steps:  
            - repeats: True  
            - section: '#/TemperatureRamp'
```



1. Custom Schema

YAML is a Markup Language defining hierarchy with indentation

```
definitions:  
  - name: 'Tutorial 13 sintering schema'  
    sections:  
      - TemperatureRamp:  
        base_sections:  
          - nomad.datamodel.metainfo.basesections.ProcessStep  
        quantities:  
          - initial_temperature:  
            type: np.float64  
            unit: celsius  
            description: "initial temperature set for ramp"  
            m_annotations:  
              - eln:  
                component: NumberEditQuantity  
                defaultDisplayUnit: celsius  
      - Sintering:  
        base_sections:  
          - nomad.datamodel.metainfo.basesections.Process  
          - nomad.datamodel.data.EntryData  
        sub_sections:  
          - steps:  
            - repeats: True  
            - section: '#/TemperatureRamp'
```

Check Tutorial 8 for YAML schemas:
youtube.com/@TheNOMADLaboratory



2. User Defined Plugin

Screenshot of a user interface for managing uploads, likely a plugin for a larger application. The top navigation bar includes PUBLISH, EXPLORE, ANALYZE, and ABOUT dropdowns, a user profile icon, and links for LOGOUT and UNITS.

The main area shows "Your uploads / Upload". It has two tabs: OVERVIEW and FILES. The FILES tab is currently active, showing a toolbar with search, user, cloud, refresh, and other icons.

In the OVERVIEW tab, there is a list item for an unnamed upload with an ID: y6POeeDuRB24pz5JhdB7Nw. Below this, a step-by-step guide is provided:

- 1 Prepare and upload your files

Instructions state: Here you can upload files. Top-level .zip/.tar files will be uncompressed automatically. For more information, see our documentation on [uploading files](#) or view the [supported codes](#). Optionally, you can also create an entry from built-in or uploaded schemas. Please take a look at our documentation on [schemas](#).

A large blue button with a cloud icon and the text "DROP FILES HERE OR CLICK TO OPEN DIALOG" is present. An arrow points from this button to a file selection dialog window below it, which shows the path " > /".

Below the dialog, four file icons are shown with their extensions: .xlsx, .dat, .hdf5, and .xrdml.



2. User Defined Plugin

ReflectanceWavelengthTransient

section

QUANTITIES

```
rawfile_column_header = PYRO_WAVELENGTH
raw_intensity = 10472 vector
smoothed_raw_intensity = 10472 vector
autocorrelated_intensity = 10472 vector
smoothed_autocorrelated_intensity = 10472 vector
```

SUB SECTIONS

growth_rate

PLOT

The plot shows Reflectance on the y-axis (0 to 1.5) versus wavelength on the x-axis (0 to 12k). It displays two traces: a red line for raw intensity and a blue line for smoothed raw intensity. Blue dots mark specific peaks in the raw trace.

A zoomed-in view of the raw intensity plot from approximately 0 to 4k on the x-axis, focusing on the first few peaks. Blue dots highlight the peaks used for analysis.

GrowthRate

section

QUANTITIES

Reflectance trace

Raw Smoothed Raw Autocorrelated Smoothed Autocorrelated

Fabry perot oscillation period
790.548565 Unit s

Recalculate on save
 Yes No

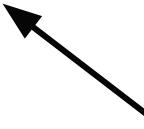
Growth rate
1122.275734575 Unit nm / hr

SUB SECTIONS

peaks_identification

REFERENCED BY closed

Figures of merit
automatically
processed



43



2. User Defined Plugin

Check Tutorial 12 for Plugin development:
youtube.com/@TheNOMADLaboratory





2. User Defined Plugin

Check Tutorial 12 for Plugin development:
youtube.com/@TheNOMADLaboratory



Deploy and start using your plugin

Hands-on Tutorial by Hampus Näsström



3. Community Plugins

Materials synthesis classes:

github.com/FAIRmat-NFDI/nomad-material-processing

Characterization classes:

github.com/FAIRmat-NFDI/nomad-measurements

Post-processing and analysis classes:

github.com/FAIRmat-NFDI/nomad-analysis

NOMAD for Experimental Data Management in Synthesis

Part II
Built-in schemas for ELN
by
Sarthak Kapoor

[step-by-step documentation](#)



Summary Part II

Type	Implementation	Advantage
0 Unstructured	-	-
1 Built-in Classes	available in NOMAD	out-of-the-box solution
2 Custom Schema	text file (.yaml format)	fast & flexible deployment
3 User Defined Plugin	Python code	automation of parsing
4 Community Standard	Git repositories	FAIR data

Hands-on Tutorial by Sarthak Kapoor



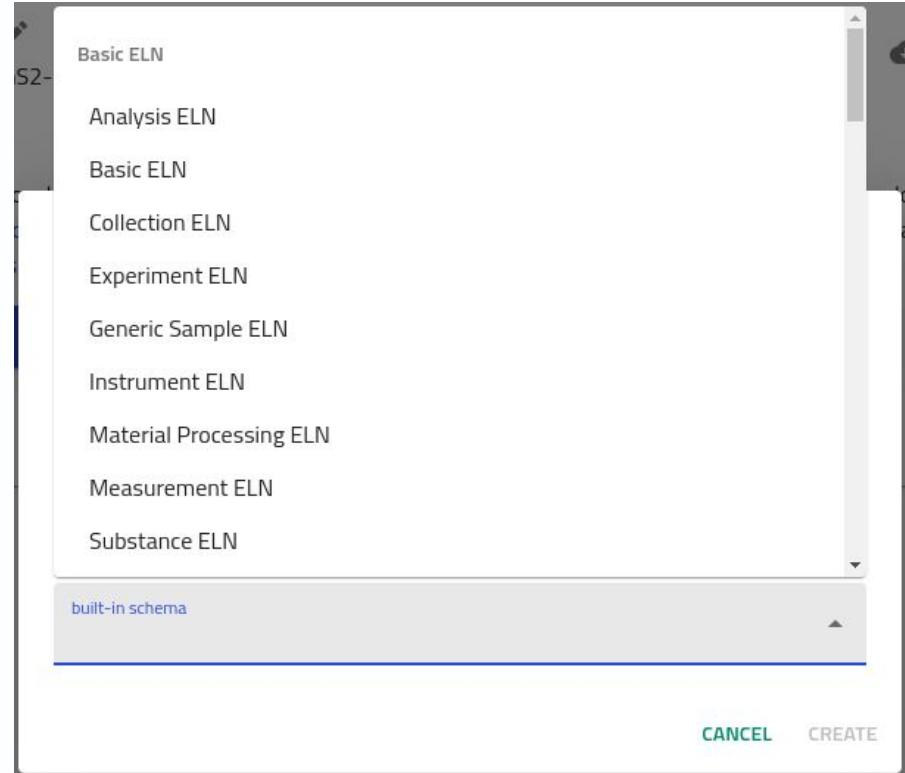
Summary Part II

- Built-in classes

Highlights:

- Creating Archives
- Referencing Archives
- Collaborate with colleagues
- Navigate Archives

A recap of the Part II:
[step-by-step documentation](#)





Summary Part II

- Built-in classes
- Inheriting classes and extending in YAML

Check Tutorial 8 for YAML schemas:

youtube.com/@TheNOMADLaboratory

[github.com/FAIRmat-NFDI/
AreaA-Examples/tutorial8](https://github.com/FAIRmat-NFDI/AreaA-Examples/tutorial8)

```
definitions:  
  .. name: 'Tutorial-13-sintering-schema'  
  .. sections:  
    .. TemperatureRamp:  
      .. base_sections:  
        .. - nomad.datamodel.metainfo.basesections.ProcessStep  
      .. quantities:  
        .. initial_temperature:  
          .. type: np.float64  
          .. unit: celsius  
          .. description: "initial temperature set for ramp"  
          .. m_annotations:  
            .. eln:  
              .. component: NumberEditQuantity  
              .. defaultDisplayUnit: celsius  
        .. Sintering:  
          .. base_sections:  
            .. - nomad.datamodel.metainfo.basesections.Process  
            .. - nomad.datamodel.data.EntryData
```



Summary Part II

- Built-in classes
- Inheriting classes and extending in YAML
- Using Plugins developed in FAIRmat

XRD plugin:

- Rigaku (.rasx)
- Malvern Panalytical (.xrdml)
- Bruker (.brml)

[github.com/FAIRmat-NFDI/
nomad-measurements](https://github.com/FAIRmat-NFDI/nomad-measurements)

NOMAD for Experimental Data Management in Synthesis

Break!

NOMAD for Experimental Data Management in Synthesis

Part III Schema & Plugin development by Hampus Näsström

[step-by-step documentation](#)



Introduction Part III

Deploy and start using your plugin.

[step-by-step documentation](#)



Introduction Part III

Check Tutorial 12 for Plugin development:
youtube.com/@TheNOMADLaboratory

Available kinds of plugin:

- Apps
- Normalizers
- Parsers
- Schema packages

nomad-lab.eu/oasis/docs/howto/plugins/plugins#plugin-entry-points



Introduction Part III

step-by-step documentation

If you want to know more about software development tools:

- what is Git
- what is VSCode, i. e., an Integrated Development Environment (IDE)
- what is Pip
- creating a Python package
- uploading a package to PyPI
- what is cruft



Introduction Part III

	Type	Implementation	Advantage
0	Unstructured	-	-
1	Built-in Classes	available in NOMAD	out-of-the-box solution
2	Custom Schema	text file (.yaml format)	fast & flexible deployment
3	User Defined Plugin	Python code	automation of parsing
4	Community Standard	Git repositories	FAIR data

Hands-on Tutorial by Hampus Näsström



Introduction Part III

	Type	Implementation	Advantage
0	Unstructured	-	-
1	Built-in Classes	available in NOMAD	out-of-the-box solution
2	Custom Schema	text file (.yaml format)	fast & flexible deployment
3	User Defined Plugin	Python code	automation of parsing
4	Community Standard	Git repositories	FAIR data

We appreciate Issues or Pull Requests!



Introduction Part III

- Materials synthesis classes:

github.com/FAIRmat-NFDI/nomad-material-processing

- Characterization classes:

github.com/FAIRmat-NFDI/nomad-measurements

- Post-processing and analysis classes:

github.com/FAIRmat-NFDI/nomad-analysis

Use cases inheriting from Community Plugins:

github.com/FAIRmat-NFDI/AreaA-data_modeling_and_schemas/IKZ_plugin

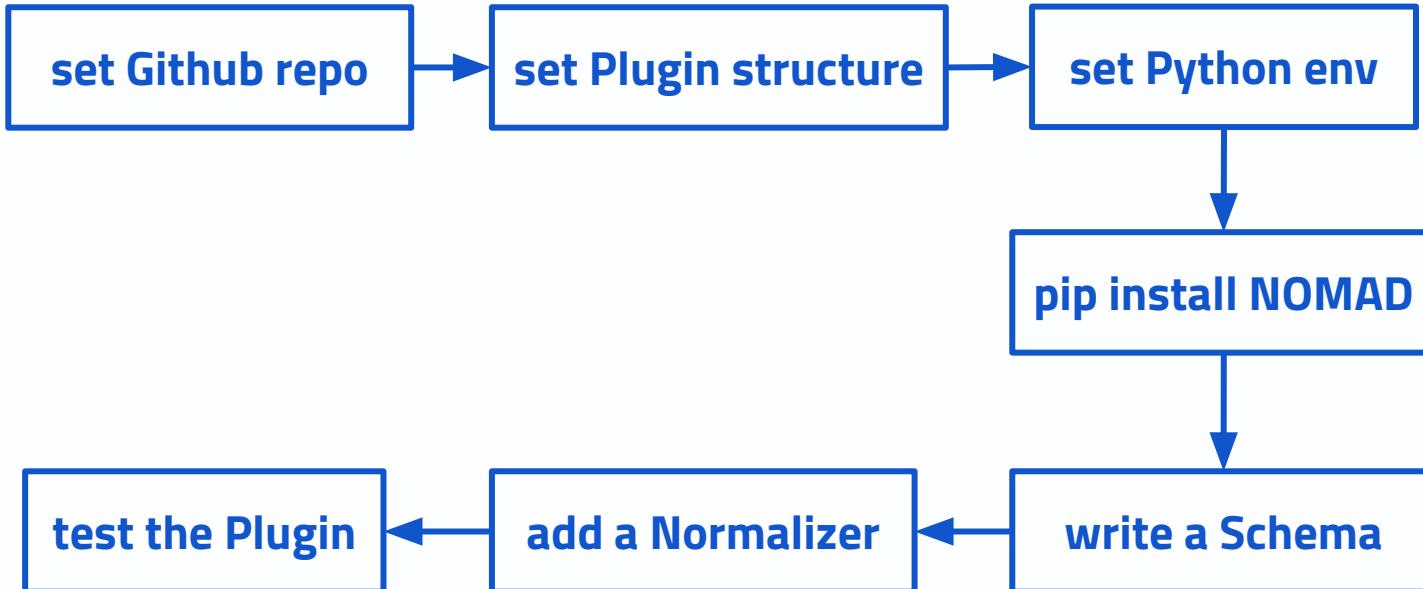
NOMAD for Experimental Data Management in Synthesis

Part III Schema & Plugin development by Hampus Näsström

[step-by-step documentation](#)



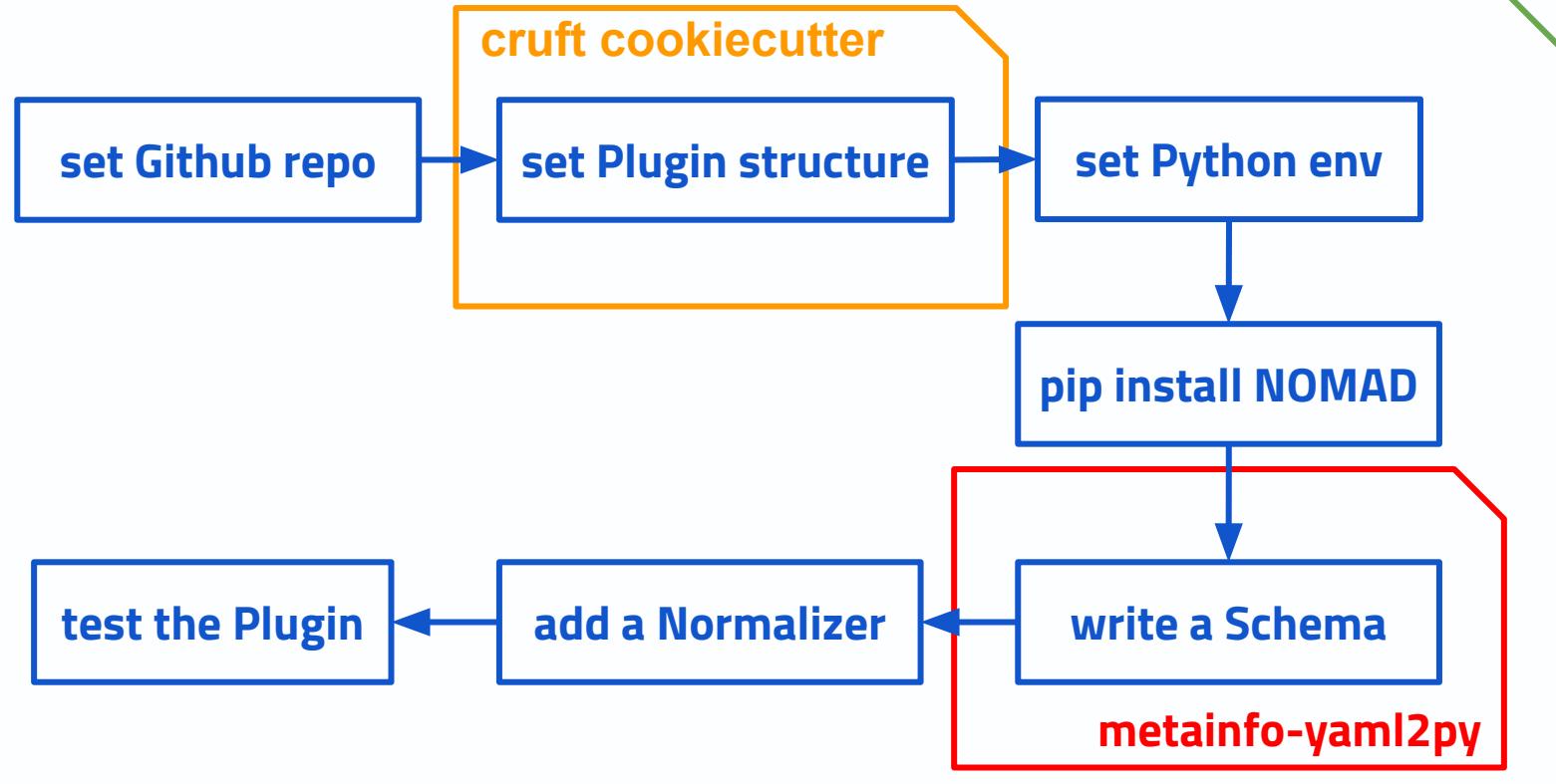
Summary Part III





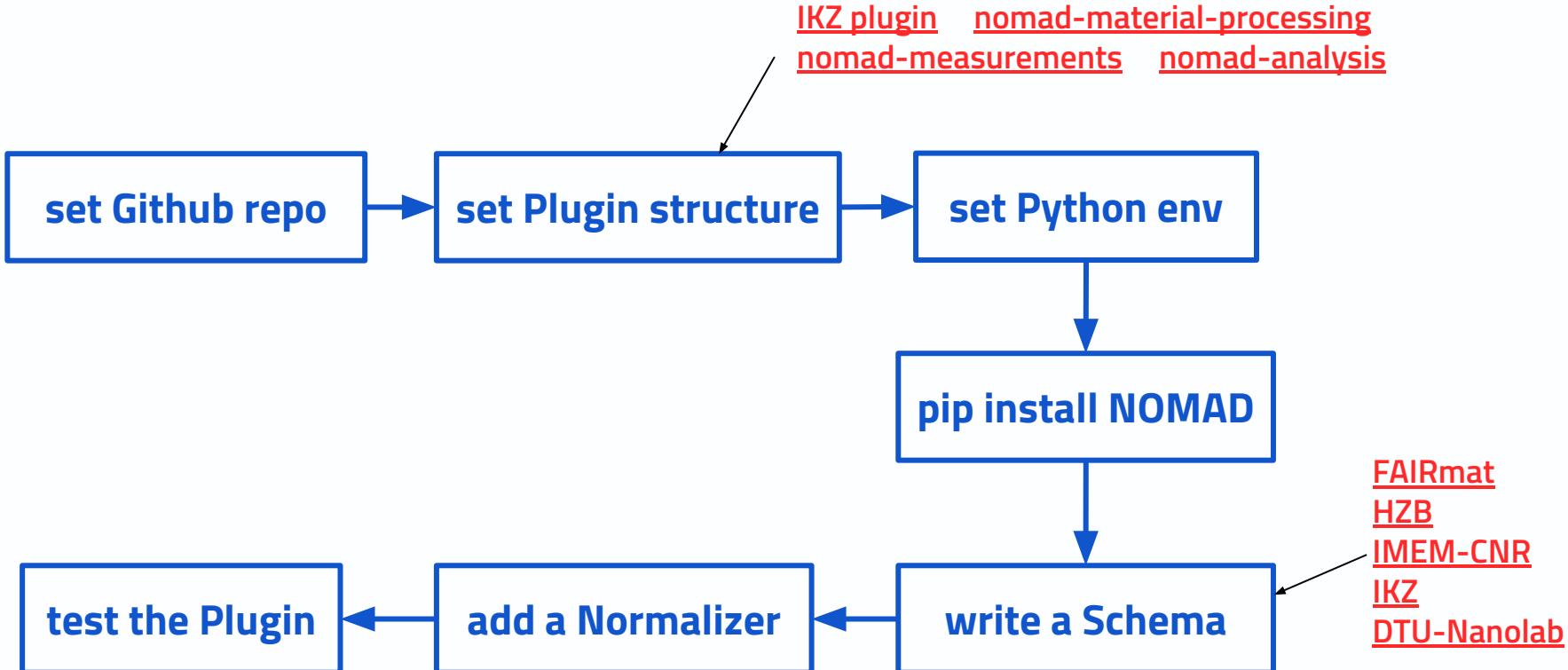
Summary Part III

Github codespaces: VSCode





Summary Part III



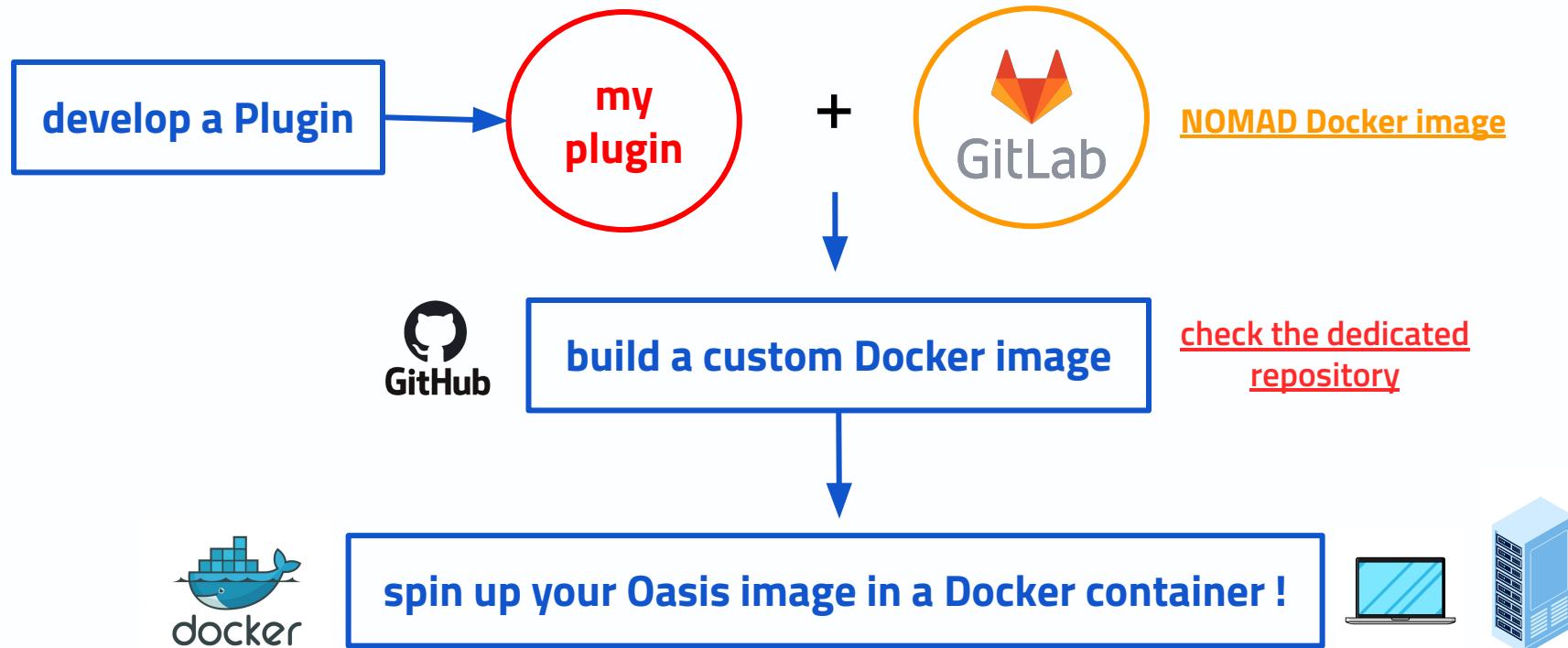
NOMAD for Experimental Data Management in Synthesis

Part IV
Deploying your NOMAD plugins
by
Hampus Näsström

[step-by-step documentation](#)



Introduction Part IV



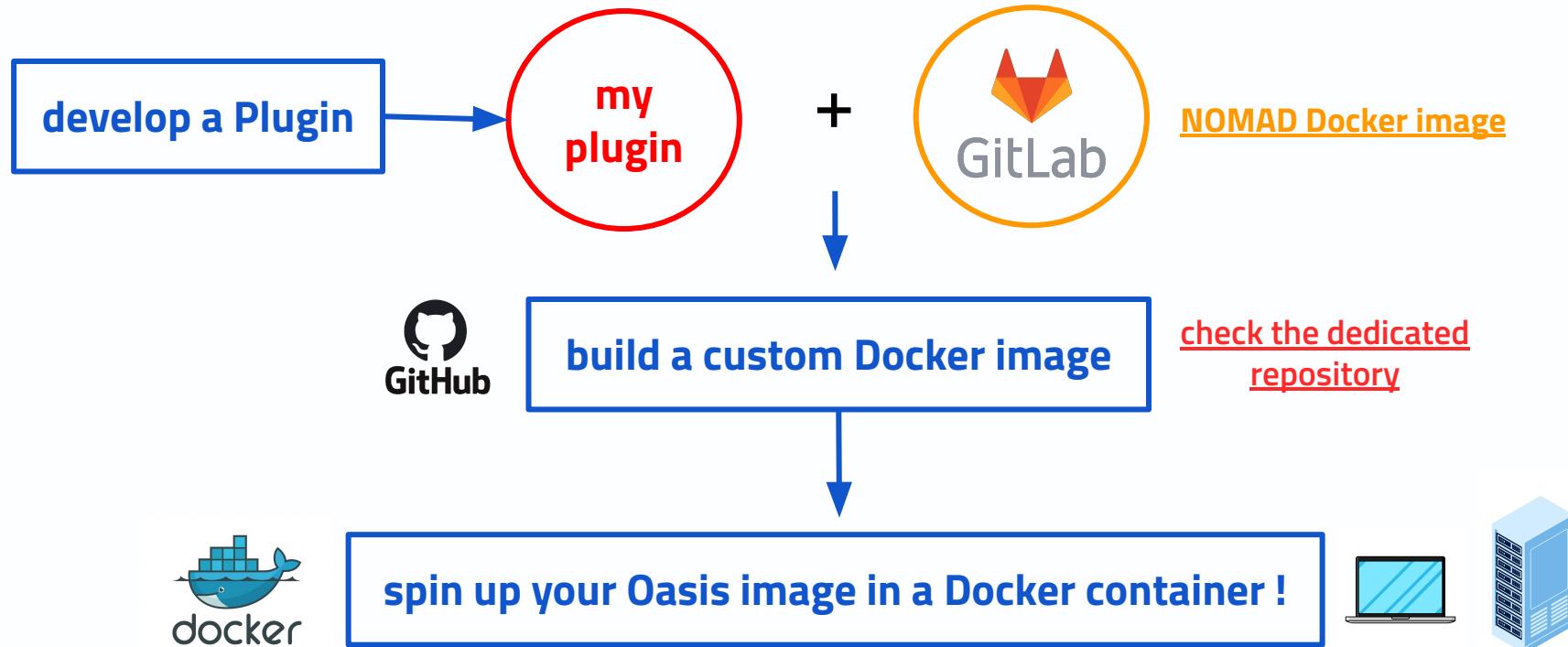
NOMAD for Experimental Data Management in Synthesis

Part IV
Deploying your NOMAD plugins
by
Hampus Näsström

[step-by-step documentation](#)



Summary Part IV





Wrapping up Tutorial 13

- Exploring built-in Schemas in the ELN
- NOMAD plugin development
- Write a NOMAD Oasis image with plugins and deploy it

[Invitation to Discord server](#) [ELNs Discord channel](#) [AreaA-Examples/tutorial13](#)