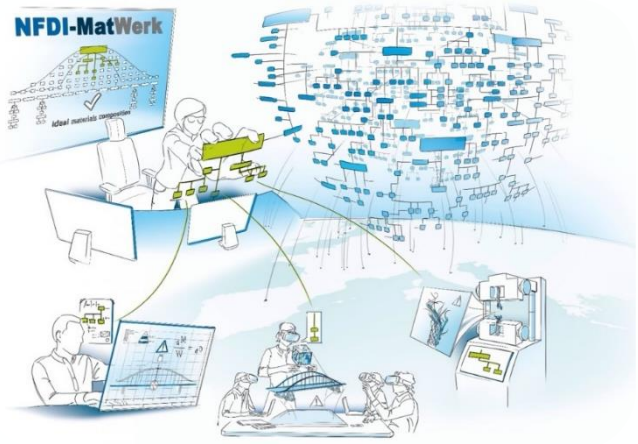


Kerzel, U. [1]; Hickel, T.[2]; Hunke, S.[3]; Korte-Kerzel, S.[4]; Kruzikova, P.[2]; Mathews, P.[5]; Mouhib, F.-Z.[4]; Ost, P.[6]; Rejiba, K. [4], Siemer, N.[5]

[1]RWTH Aachen; [2]Bundesanstalt für Materialforschung und -prüfung (BAM), Berlin; [3]ITC RWTH Aachen University; [4]Institut für Metallkunde und Materialphysik, RWTH Aachen University; [5]Max Planck Institut for Systainable Materials, Düsseldorf; [6]KIT, Karlsruhe

NFDI MatWerk

NATIONAL RESEARCH DATA INFRASTRUCTURE FOR MATERIALS SCIENCE & ENGINEERING

Model driven data space exploration: Interoperable infrastructure for experimental and computational data

MSE 2024

24.09.2024

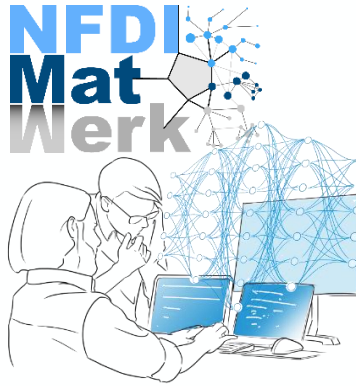
Funded by DFG Deutsche Forschungsgemeinschaft German Research Foundation

Project number 460247524

DFG Deutsche Forschungsgemeinschaft

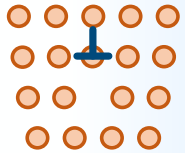
SFB 1394

Defect Phases: Motivation

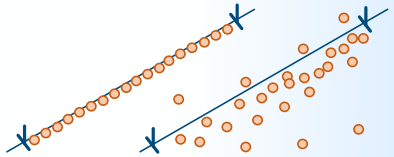


Textbook

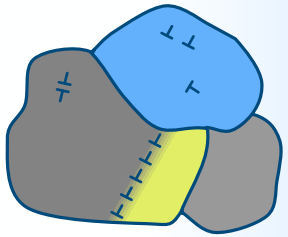
Lattice defects



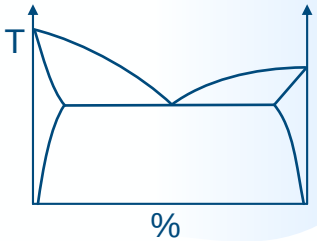
Alloying



Microstructure evolution



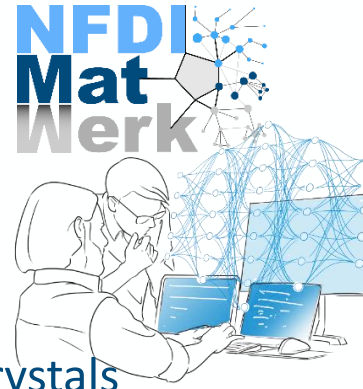
Phase composition and stability

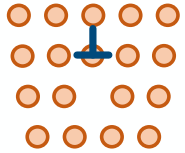
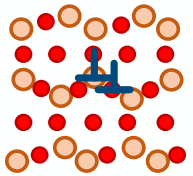
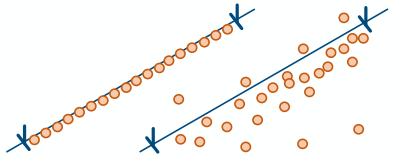
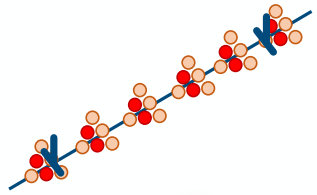
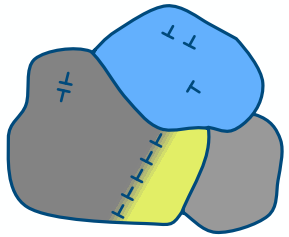
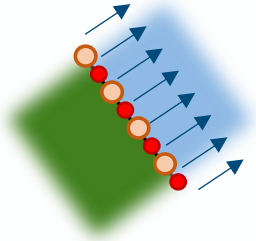
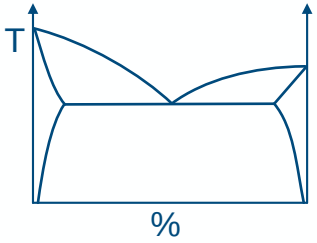
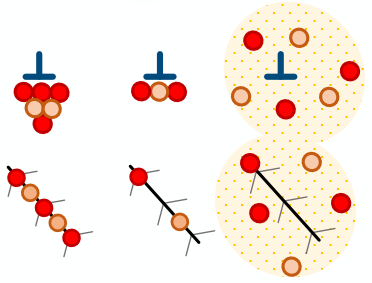


New materials

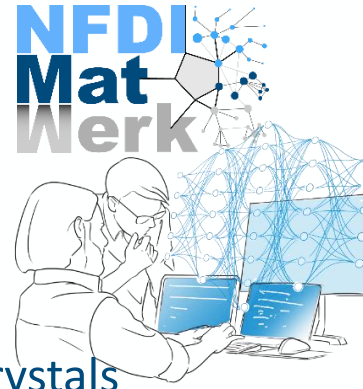
- ✓ Sustainable energy conversion
- ✓ Safe mobility
- ✓ Recycling potential
- ✓ Green manufacturing

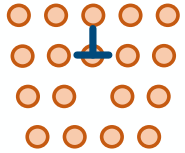
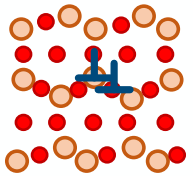
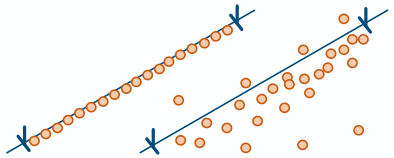
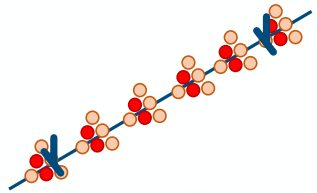
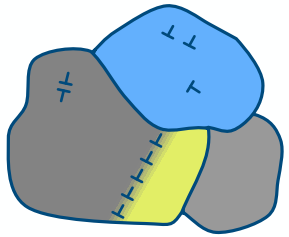
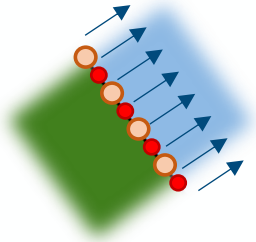
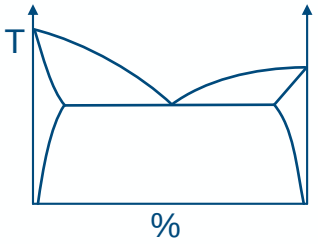
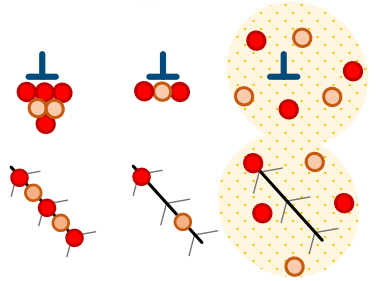
Defect Phases: Motivation



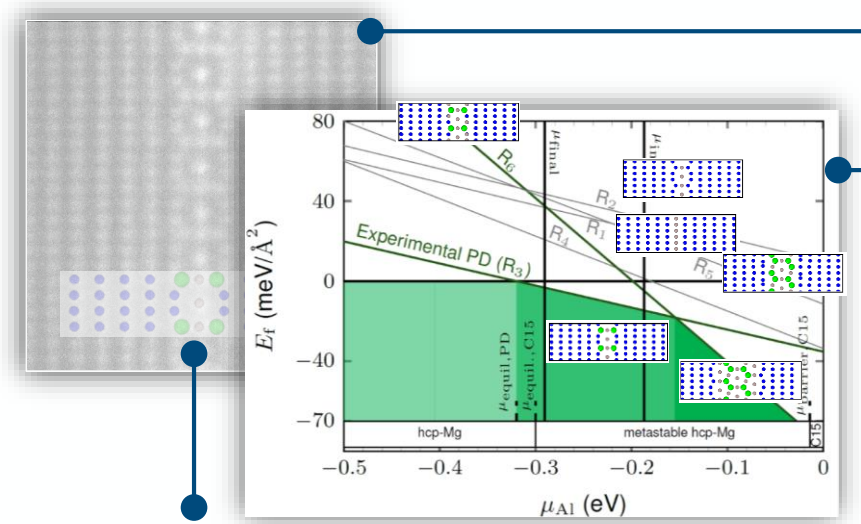
	Textbook		Research	
Lattice defects		⟷		Defects in ordered crystals
Alloying		⟷		Chemistry of defects
Microstructure evolution		⟷		Effects on processes
Phase composition and stability		⟷		Defect chemistry and stability

Defect Phases: Motivation



	Textbook	Missing:	Research	
Lattice defects		• Physical Links • Coherent Framework		Defects in ordered crystals
Alloying				Chemistry of defects
Microstructure evolution				Effects on processes
Phase composition and stability				Defect chemistry and stability

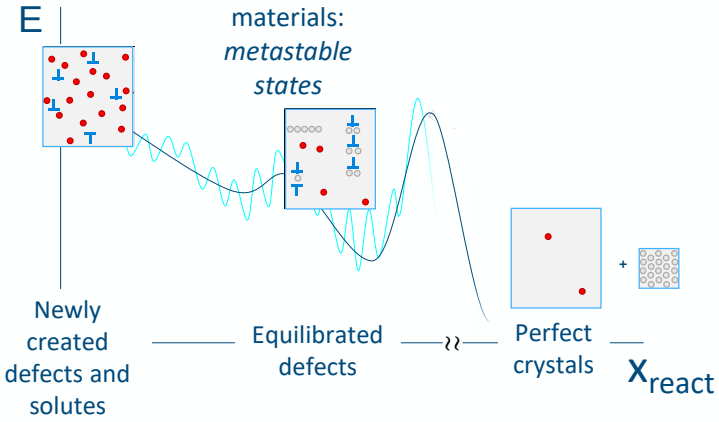
Defect phases and defect phase diagrams

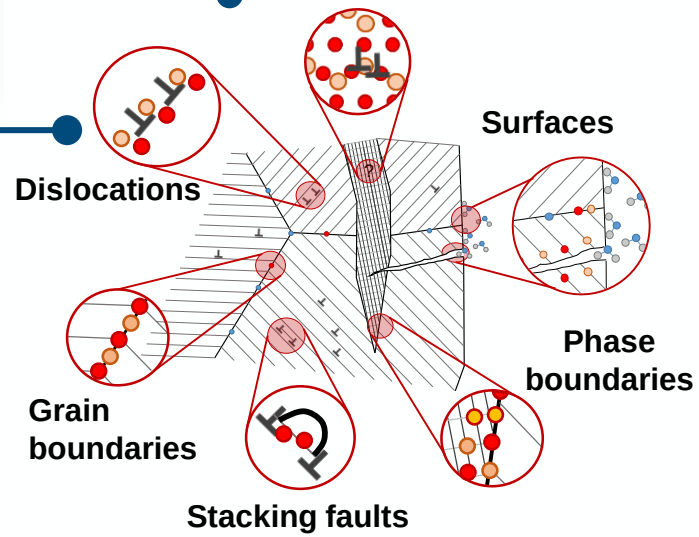


What is their atomic scale structure?

How can we predict them?

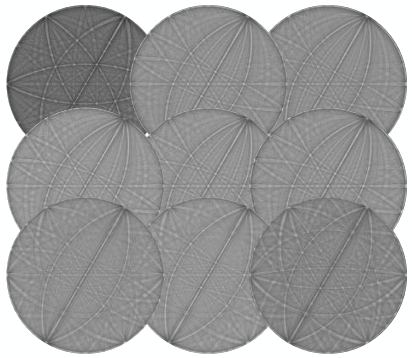
How do they compete?





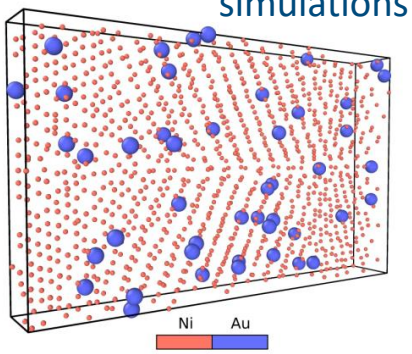
Korte-Kerzel et al. (2022) Int Materials Reviews 67(1), 89-117

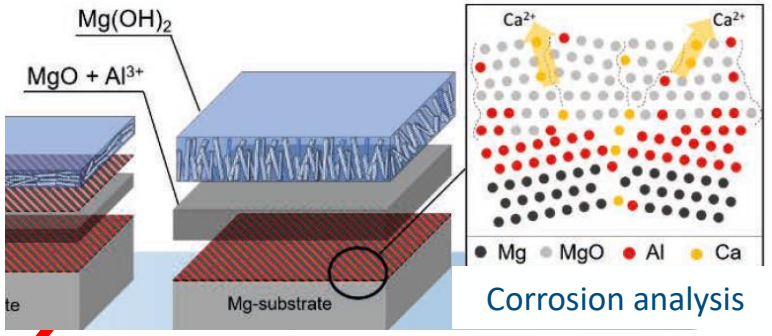
Challenge: Multidisciplinary Approach: SFB 1394



Large scale analysis & Machine Learning

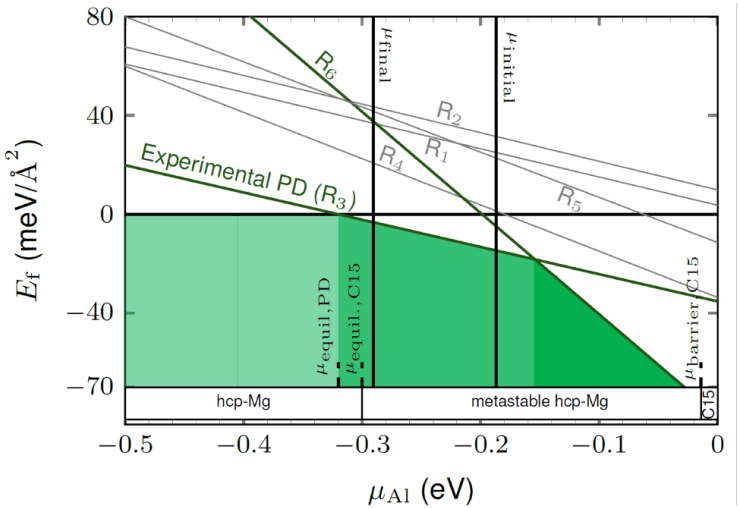
Modelling & Atomistic simulations

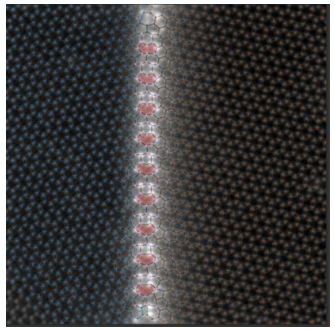




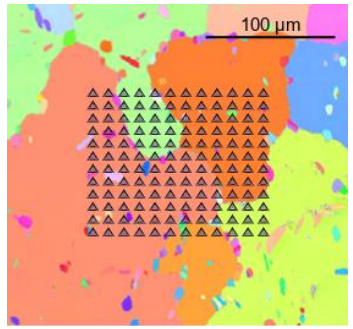
Corrosion analysis

Properties of defect phases



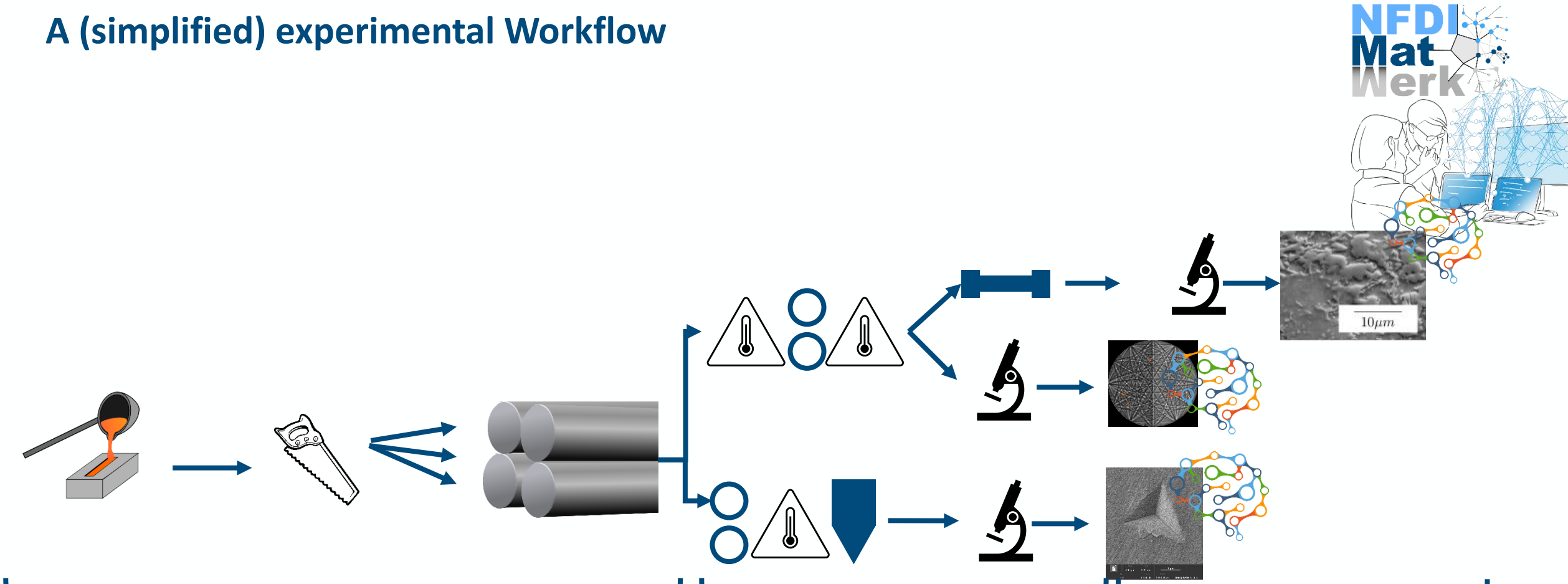


Atomic scale characterizations



Microstructure and property analysis

A (simplified) experimental Workflow

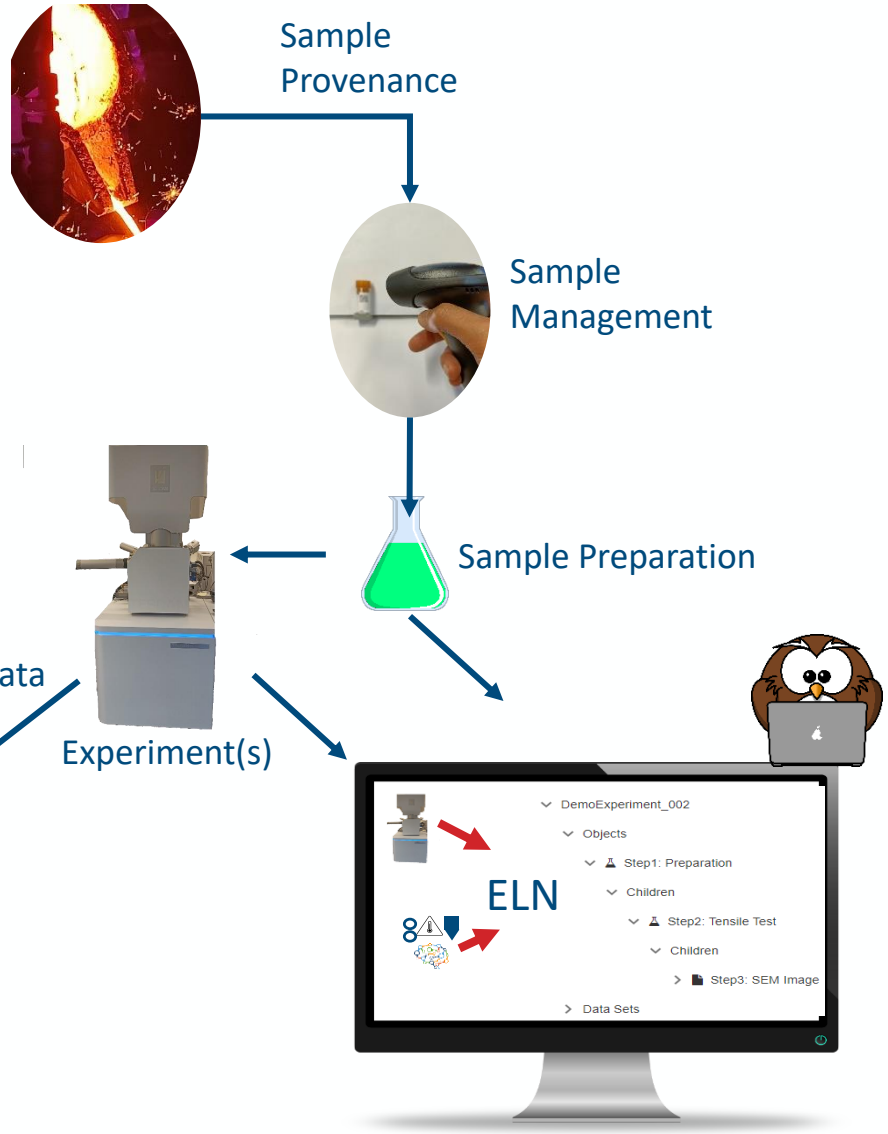


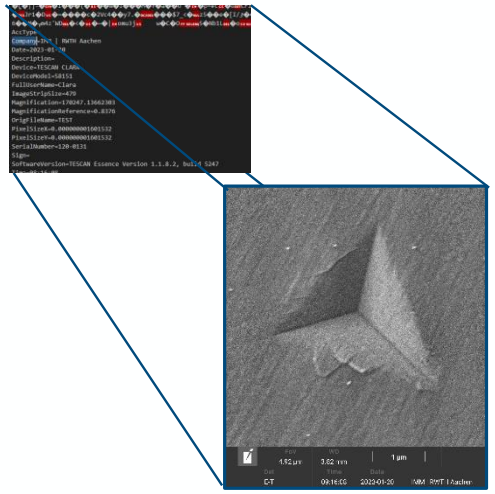
Sample creation
(may be split into several child samples)

Experimental paths
(e.g. heat treatment, tensile test, indents,….)

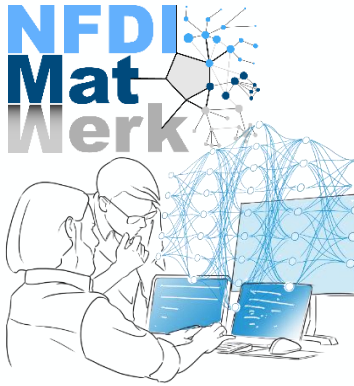
Analysis paths
(e.g. imaging, mathematical / statistical / ML analysis, …)

Where do we need to be?





automatic metadata extraction & storage



- Everything in one place
- Publish data (zenodo, etc)
- Easy onboarding / hand-over
- Track work across many years
- Build a data repository → Machine Learning & AI

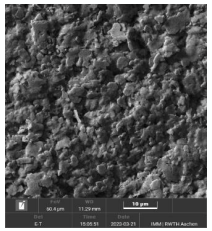
What do we need to track (experiment)?

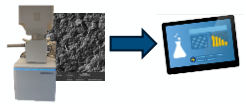
Experimental Samples:
- The samples themselves (physically)
  → sample management
- How they are created
- Preparation, mechanical tests,
- …

Scientific Instruments
- Make & Model & Version
- Software version
- Settings during experiment
- …

"Big Data":
- Integration with ELN
- Accessible from everywhere (Desktop, cluster, conference, research/field trip, …)

Code
- Control code
- Analysis code
- Software environment & libraries
- Virtual Machine / Container?





Experimental Data
- Raw data
- Processed & analysed data
- Automatic metadata extraction

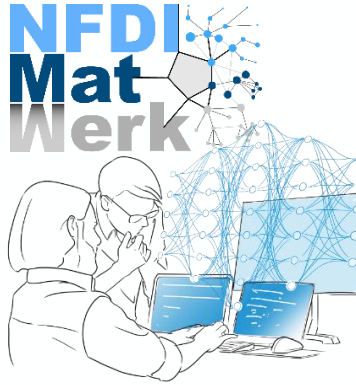
Description of Experiments, Results, …
- Description of experiments
- Annotations / comments
- Link to samples, devices, data, methods

→ Electronic Lab Notebooks (ELN)

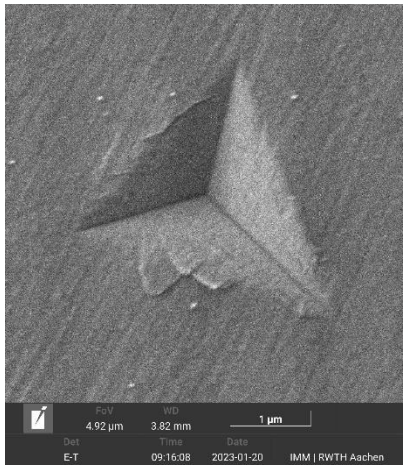
Chemicals & Consumables:
- What is used in sample preparation?

Standard Operating Procedures / Protocols
- How are experiments done?
- How are samples prepared?
- How are instruments operated & settings?

API / Python Access
- Interact with analysis code
- Batch processing
- Search / filter / retrieve data & results



Tracking Samples & Measurements



(Modern) Instruments store
many details about measurements:

- Settings
- Conditions
- Results

→ extract (meta-) data automatically & store in RDM

Metadata Fields

Name:
DemoUpload_001_0

Company:
IMM | RWTH Aachen

PixelSizeX:
0.000000001601532

PixelSizeY:
0.000000001601532

SessionID:
2a75e90a-bdf2-4161-ad38-607291f62585

Device:
S8151

DeviceModel:
S8151

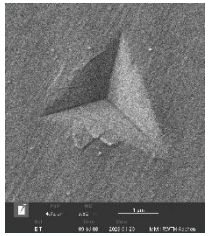
DateTime:
2023-01-20T08:16:08

Date:
2023-01-20

Time:
08:16:08

SoftwareVersion:
TESCAN Essence Version 1.1.8.2, build 5247

Tracking Samples & Measurements



Metadata Fields

Name:
DemoUpload_001_0

Company:
IMM | RWTH Aachen

PixelSizeX:
0.000000001601532

PixelSizeY:
0.000000001601532

SessionID:
2a75e90a-bdf2-4161-ad38-607291f62585

Device:
S8151

DeviceModel:
S8151

DateTime:
2023-01-20T08:16:08

Date:
2023-01-20

Time:
08:16:08

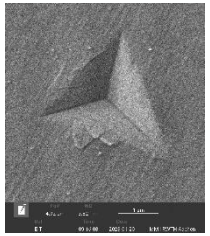
SoftwareVersion:
TESCAN Essence Version 1.1.8.2, build 5247

Challenge: Many different instruments

- No global standard for open data
- Vendor-specific file-formats,
  some more “open” than others

→ Parsers needed for each file format
→ painstaking and difficult work,
esp. if many different experiments/labs
work together.

Tracking Samples & Measurements



Metadata Fields

Name:
DemoUpload_001_0

Company:
IMM | RWTH Aachen

PixelSizeX:
0.000000001601532

PixelSizeY:
0.000000001601532

SessionID:
2a75e90a-bdf2-4161-ad38-607291f62585

Device:
S8151

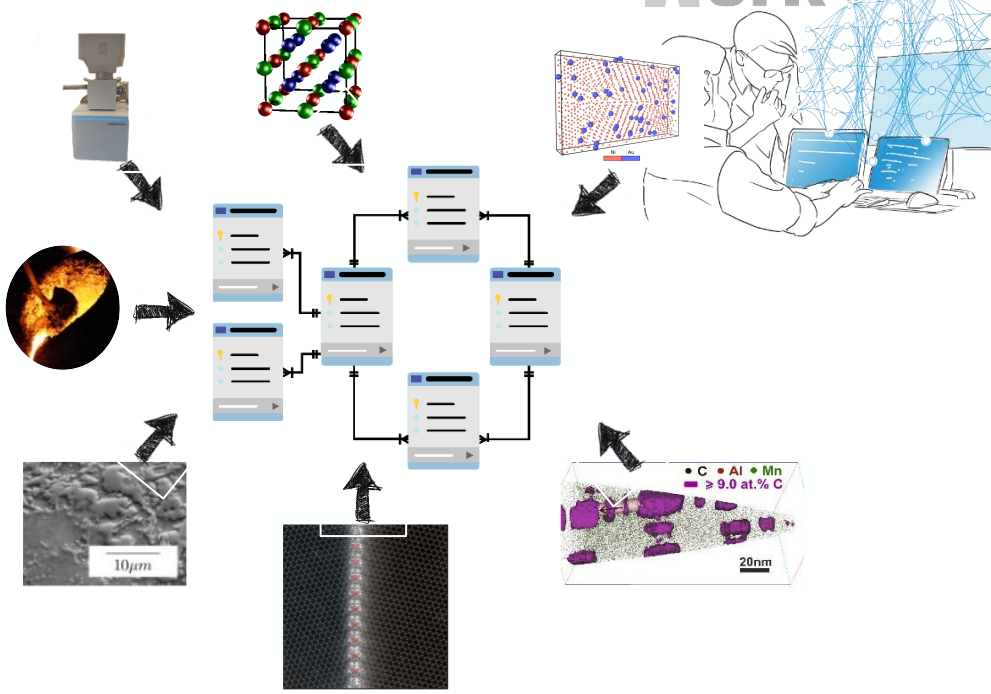
DeviceModel:
S8151

DateTime:
2023-01-20T08:16:08

Date:
2023-01-20

Time:
08:16:08

SoftwareVersion:
TESCAN Essence Version 1.1.8.2, build 5247



Challenge: Many different instruments

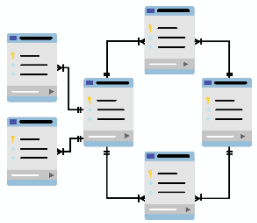
- No global standard for open data
- Vendor-specific file-formats, some more “open” than others

→ Parsers needed for each file format
→ painstaking and difficult work, esp. if many different experiments/labs work together.

Challenge: Detailed Metadata Schema

- Samples
- Instruments
- Methods & Procedures
- Consumables, Preparation material, …
- Data - Measurements & Results
- Simulation

Available Solution Space



Requirements

- Relational data model
  - Defined metadata schema
  - Hierarchical (parent/child) relations
  - Controlled vocabulary
  - Complex queries

- Electronic lab notebook
  - Integrated in relational model
  - Standard operating procedures (SOP)
  - Inventory of samples, devices, SOP
    → included in workflow
  - Additionally: free-text comments, images, …

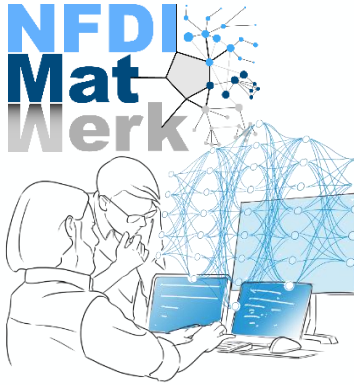
- Big Data storage

- Automated data ingestion + metadata from exp. devices

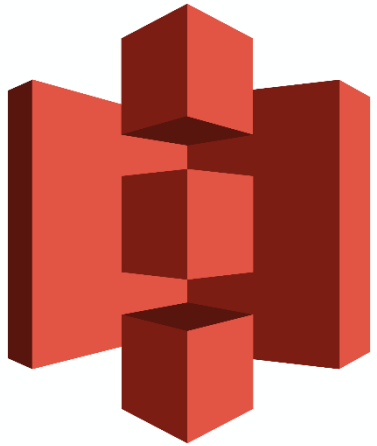
- Share data (within collaboration, publicly)

- Convenient access for analysis (e.g. integrated, “hooks”, …)

- Sample management



openBIS

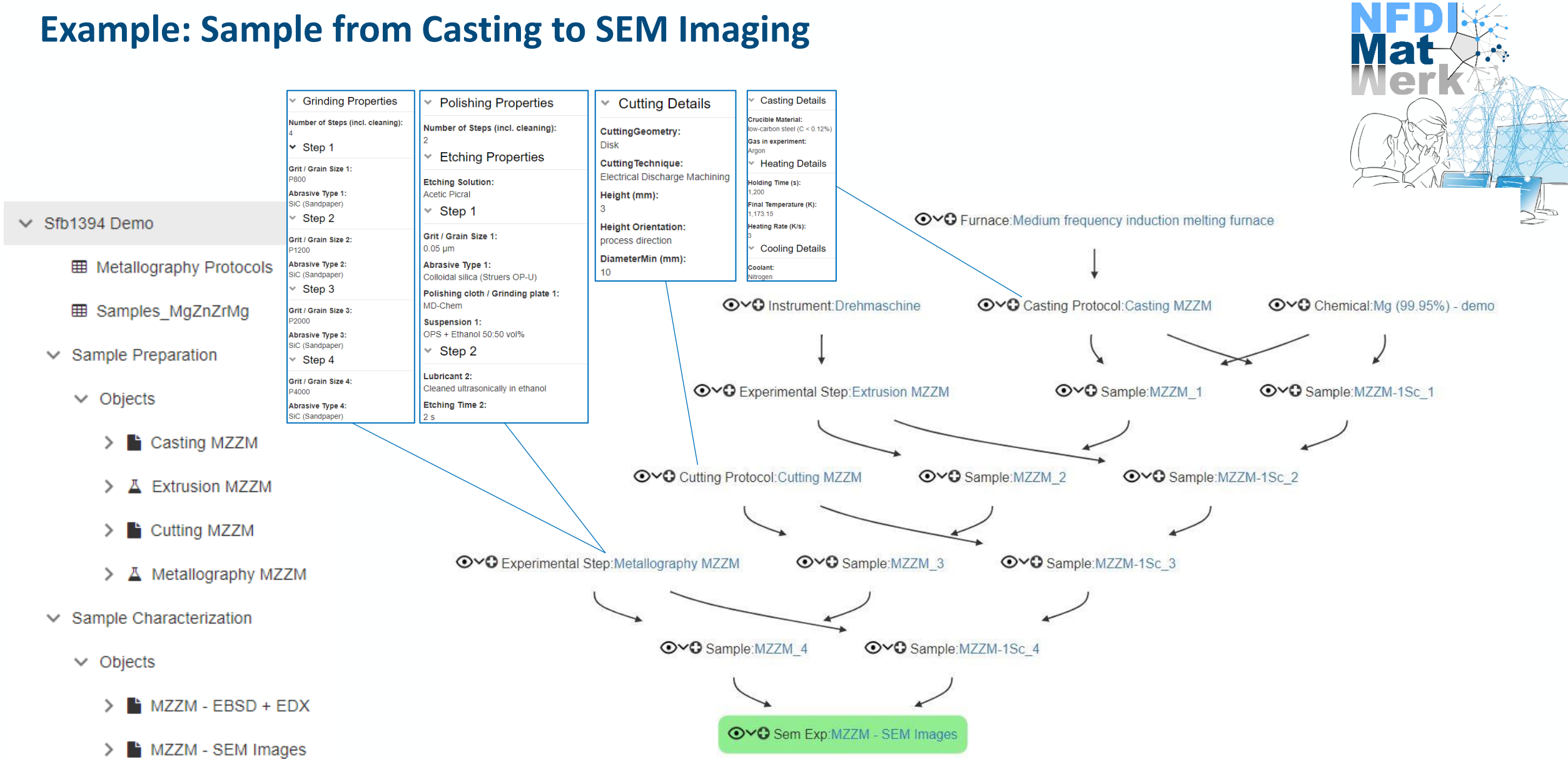


S3 Object Store
By Coscine @ RWTH

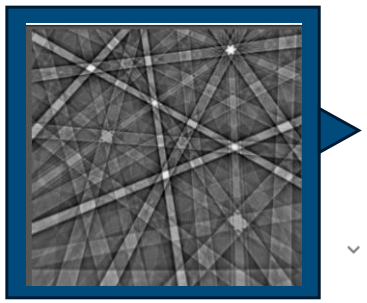
Coscine

C. Oelsen (Wikipedia), BSD

Example: Sample from Casting to SEM Imaging



Example: ELN Entry (EBSD Simulation)



Global Search

Filter

- Crc1394 A03
- Crc1394 A04
- Crc1394 A05
- Crc1394 A06
- Crc1394 A07n
  - EBSD Simulation MgAlCa
    - Objects
      - MgAlCa_Contcar_1_MonteCarlo
        - Children
          - MgAlCa_Contcar_1_Masterpattern
            - Children
              - MgAlCa_Contcar1_Screenpattern
                - Data Sets
                  - EMSoft Clarity nml file
                  - MgAlCa_Contcar1_EMSoft_ScreenPattern
                  - MgAlCa_Contcar1_Screenpatten_20kV
                  - MgAlCa_Contcar1_Screenpattern_logfile
            - Data Sets
              - MgAlCa_Contcar1_EMSoft_MasterPattern
              - MgAlCa_Contcar1_Masterpattern_20kV
              - MgAlCa_Contcar_1_Masterpattern_log
        - Data Sets

Default Experiment: EBSD Simulation MgAlCa

+ New | Edit | Upload | More ...

General info

Name:
EBSD Simulation MgAlCa

Default object type:
SIMULATION_EXP

Show in project overview:
true

Grant:
CRC1394

Start date:
2023-01-01 00:00:00 +0100

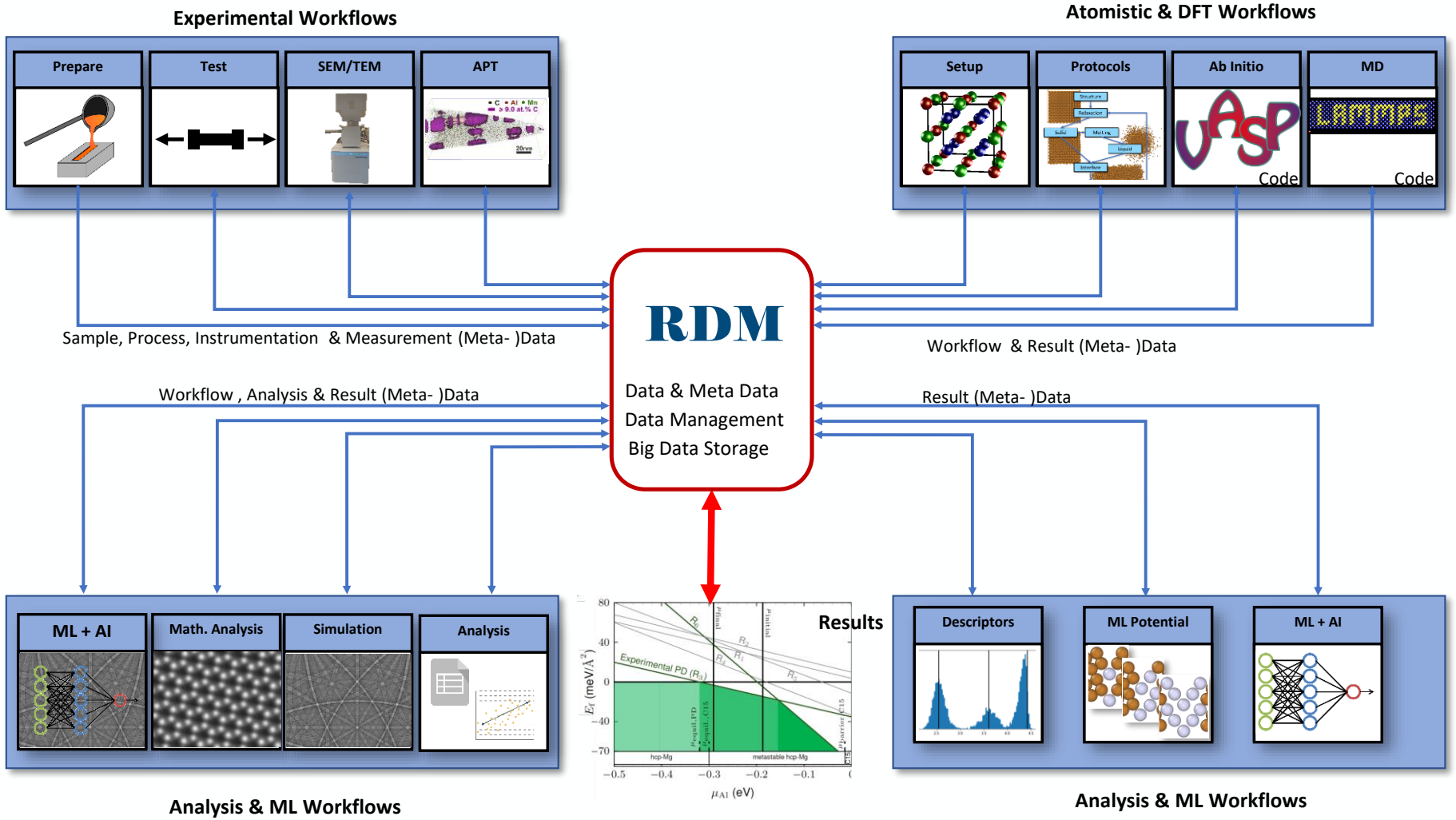
Objects

+ New Simulation Exp | More ...

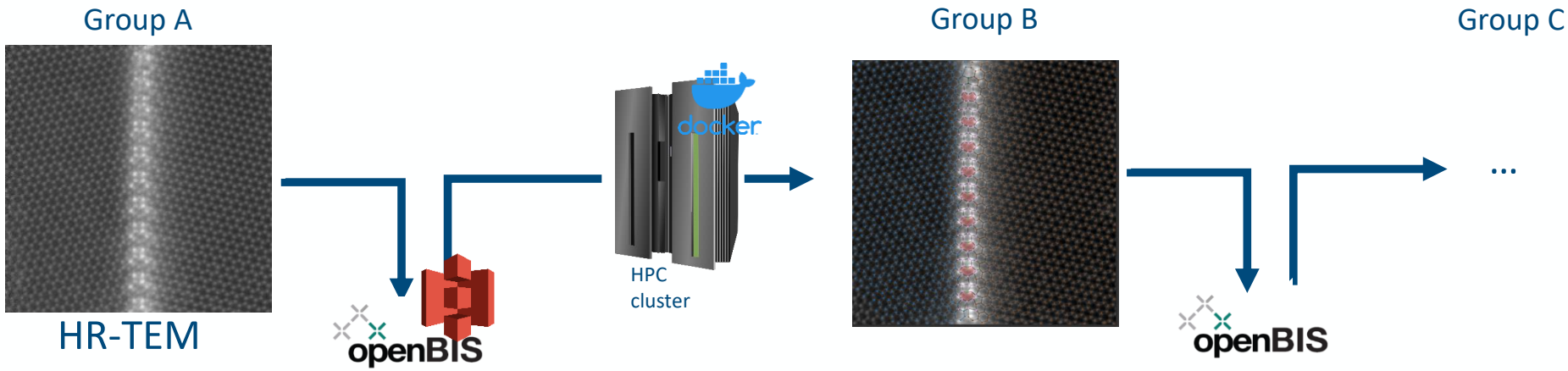
1-3 of 3 | Rows per page: 10 | COLUMNS | FILTERS | EXPORTS

Code	Name	Permid / Default Barcode	Identifier	Sim. type	EBSD Simulation Type	JobID on the compute cluster	Shou proj over
Operator: AND / Filter							
SIM504	MgAlCa_Contcar_1_MonteCarlo	20240228132412428-1823	/CRC1394/CRC1394_A07N/SIM504	EBSD	Monte-Carlo	39550220	(emp
SIM505	MgAlCa_Contcar_1_Masterpattern	20240228143732933-1835	/CRC1394/CRC1394_A07N/SIM505	EBSD	Master	39550220	(emp
SIM506	MgAlCa_Contcar1_Screenpattern	20240229081240732-	/CRC1394/CRC1394_A07N/SIM506	EBSD	Screen	41307050	true

Data Flow and Exchange

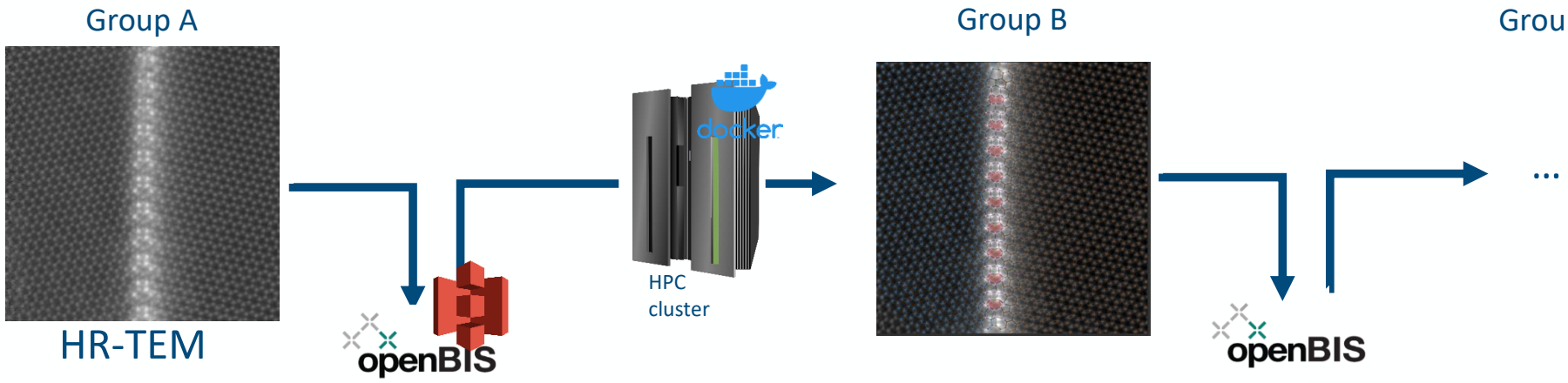


Workflows & Defect Phase Diagrams



Collaborate across groups, RDM as central “data hub”

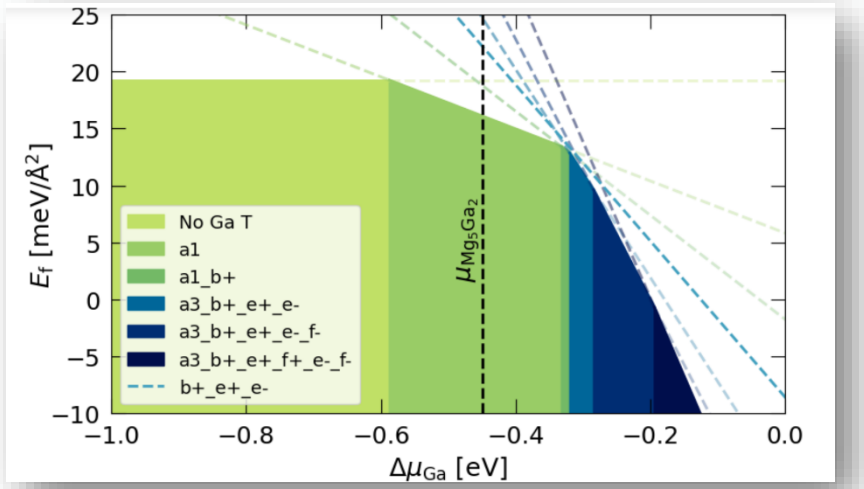
Workflows & Defect Phase Diagrams



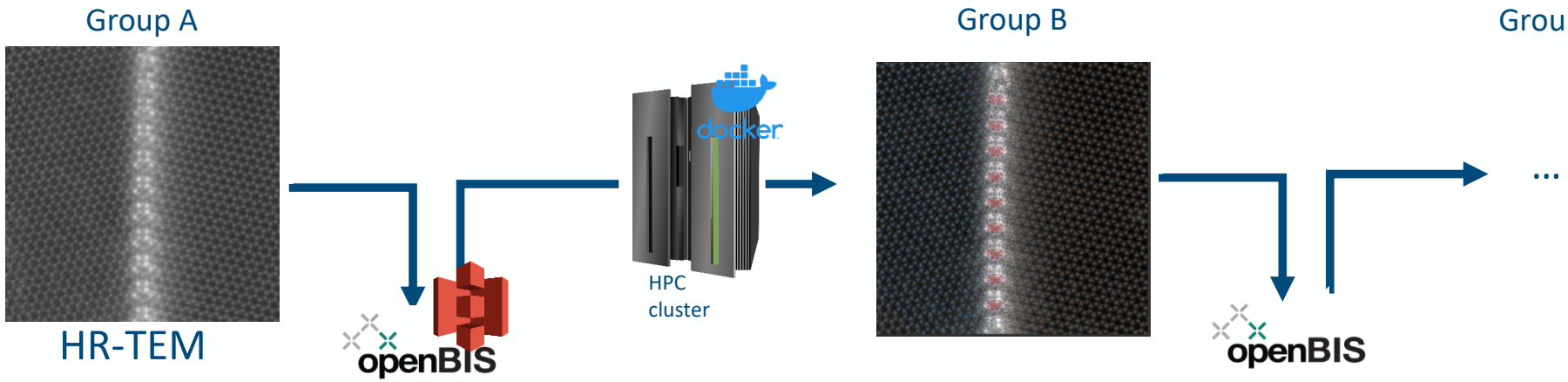
Collaborate across groups, RDM as central “data hub”

```python
final_mu_Ga = np.array([start_mu, end_mu]) # chemical potentials for the defined range

# input chemical potential: 'mu_x', formation energies: ' energies', colors: 'colors' and labe
DPD = construct_DPD(mu_x=final_mu_Ga-mu_Ga, energies=formation_energies_list, output_energy_un
                    labels=labels_list)
# plotting the DPD using 'plot()'
DPD.plot(xlim=[-1,0], ylim=[-10,25], alpha_fill=1, legend=True)
```

Expectation from Simulation

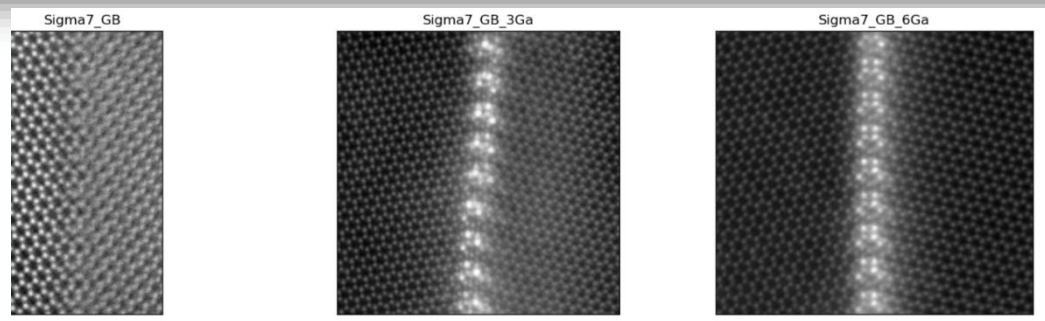


Workflows & Defect Phase Diagrams

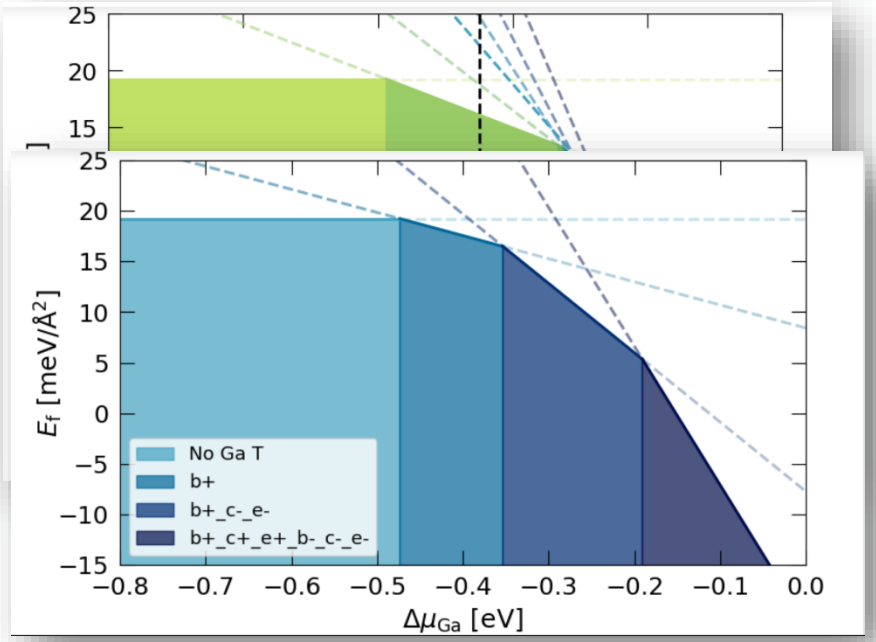


Collaborate across groups, RDM as central “data hub”

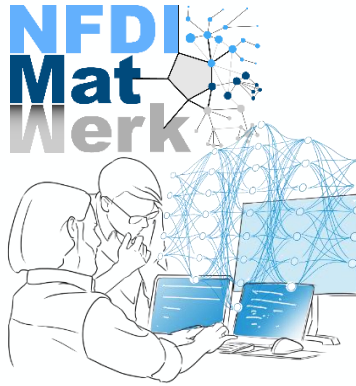
```python
final_mu_Ga = np.array([start_mu, end_mu]) # chemical potentials for the defined range

# input chemical potential: 'mu_x', formation energies: ' energies', colors: 'colors' and labe
DPD = construct_DPD(mu_x=final_mu_Ga-mu_Ga, energies=formation_energies_list, output_energy_un
                    labels=labels_list)
# plotting the DPD using 'plot()'
DPD.plot(xlim=[-1,0], ylim=[-10,25], alpha_fill=1, legend=True)
```



Combined result
Simulation + Experiment



Key Take-Aways



Building an interoperable infrastructure for simulation & experimental data.

- “Just” an electronic lab notebook (ELN) is not enough:
  track samples & provenance, methods, instruments, …
  → ELN + Laboratory Information System (LIMS): openBIS

- A lot of (meta-) data can be captured, but:
  - Large variety of scientific instruments & methods
  - Each instruments comes with their own open/closed data format
    → many parsers need to be written & adapted to extract data & metadata and transform to open data formats for analysis and data publication

PyIron

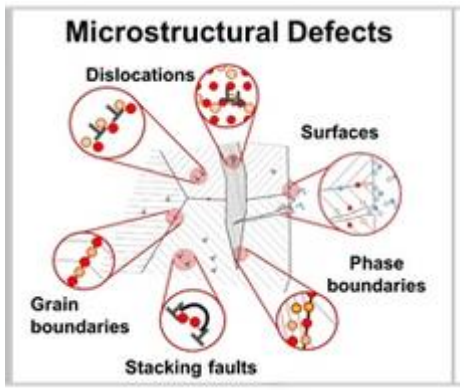
Metadata Schema

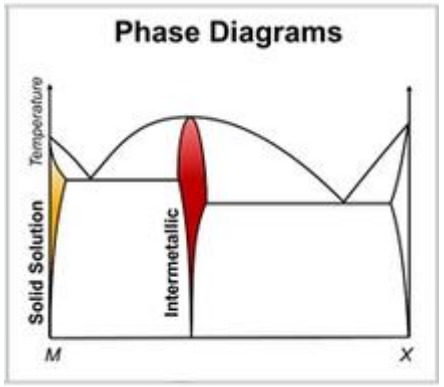
- Development of extensive metadata schema for data, instruments, methods, …
- Development of tailored tools for connecting instruments to RDM
  → prerequisite to building workflows.
- Combine simulation & experimental data → Defect Phase Diagrams

OpenBIS Tools

BACKUP

Data sheet for IUC04: Model driven data space exploration





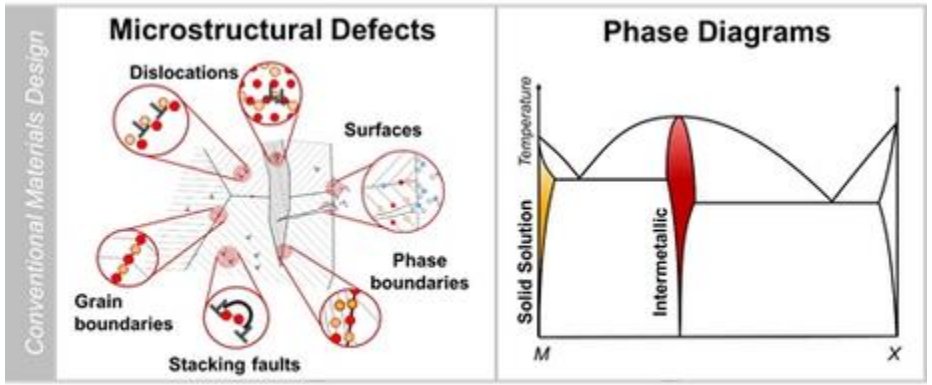
24.09.2024

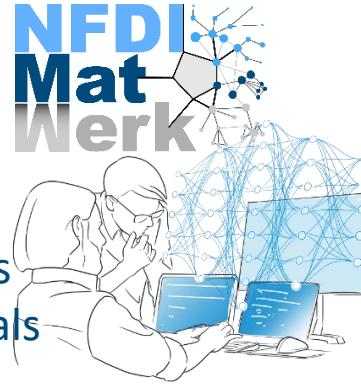
Main TA	TA-WSD
Related TA	TA-OMS
PPs	• 02 / Defect phases in structural materials (CRC 1394) • 05 / HoMMage – Hysteresis design of magnetic materials for efficient energy conversion (CRC/TRR 270)
Material/Data	Mg-Al-Ca alloys / Thermodynamic and structural data of defects
Success Scenario	User can automatically generate defect phase diagrams that can be used to predict the performance of materials
Added value	General framework of model-driven thermodynamic databases that combine computation and experiment
Requirements	• Workflows combining theoretical and exp. structural data of defects • Adaptive databases for high dim. data structures containing sparse data • Visualization (of thermodin. dataspace from microstructure, chemical or mechanical perspective) • Multiscale simulation (of multiphysics data) • Electronic lab book (for defect data)

SFB 1394

Structural and Chemical Atomic Complexity

- From Defect Phase Diagrams to Material Properties





Main Success Scenario:

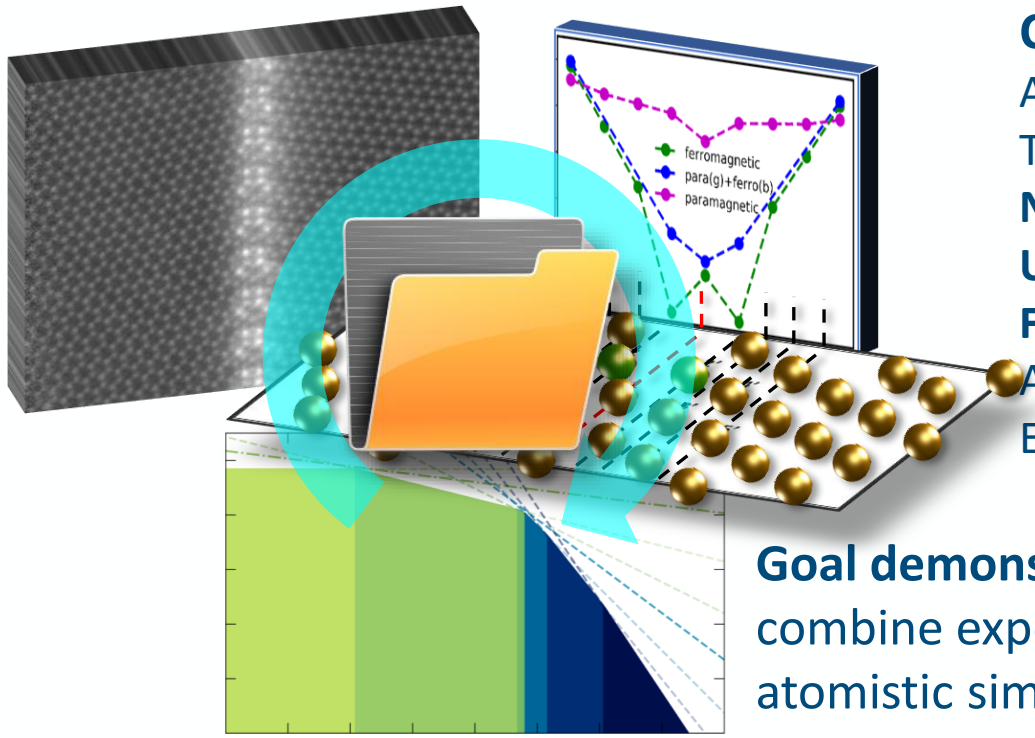
- User can automatically generate defect phase diagrams
- that can be used to predict the performance of materials

Topics

- defect phase diagrams are novel → only a limited amount of thermo-chemo-structural data is available.
- attributed to certain defect types, characters and states.
- model-driven (guided probing) collection of the relevant experimental and computational data
- post-processing with newly established simulation protocols.
- multidisciplinary perspective (i.e., atomic configurations, chemical potentials, and of materials properties).

Main requirements

- Workflows combining theoretical and experimental structural data of defects
- Adaptive databases for high dimensional data structures containing sparse data
- Visualization (of thermodynamic dataspace from microstructure, chemical or mechanical perspective)
- Multiscale simulation (of multiphysics data)
- Electronic lab book (for defect data)



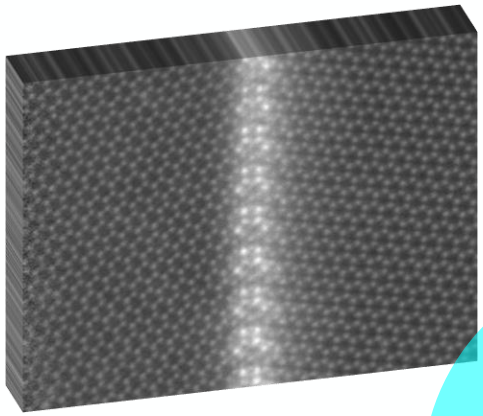
Contributors:

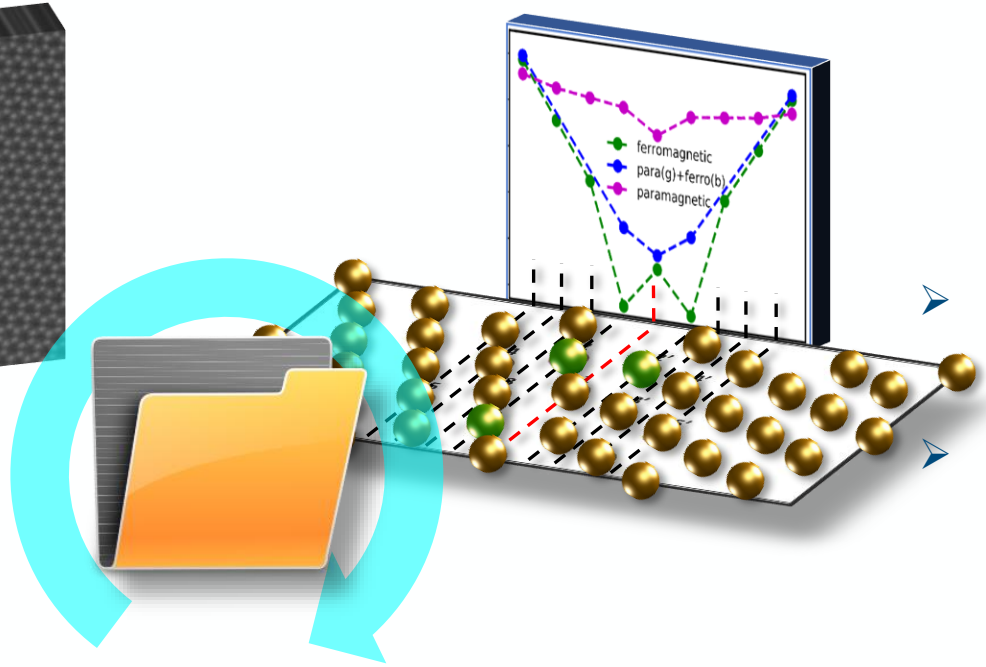
Anika Lenze (IWM),
Tilmann Hickel (MPIE),
Niklas Siemer (MPIE),
Ulrich Kerzel (RWTH),
Fatim-Zahra Mouhib (RWTH),
Amirreza Moghaddam (RWTH),
Ebrahim Norouzi (FIZ),
Philipp-Joachim Ost (KIT),
Elias Vitali (KIT),
Matthias Grönewald (TU Darmstadt),
Sarath Menon (MPIE),
Erik Bitzek (MPIE),
Prince Mathews (MPIE),
Steffen Brinkmann (FZ Jülich)

Goal demonstrator: Provide an environment that allows one to combine experimental data from electronmicroscopy and atomistic simulations for defect phase diagrams

Main steps:

- Ample number of planar defects observed in Ca and Al rich Mg samples
- HAADF-TEM images yield data on chemical & structural configurations

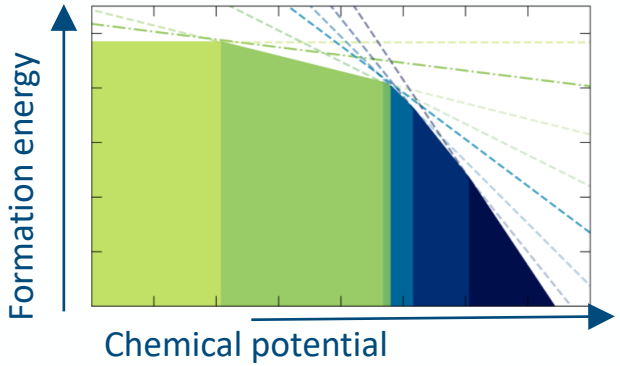




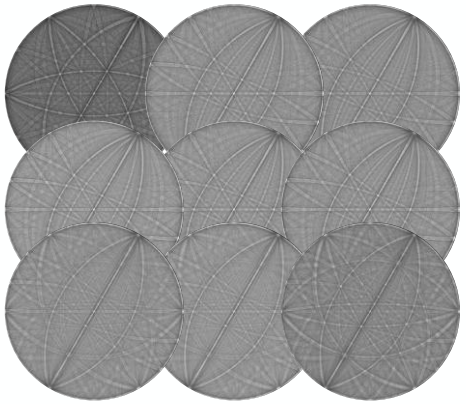
- Atomic coordinates for structural models of grain boundaries
- Segregation energy profiles for Al and Ca solutes

Central storage solution required

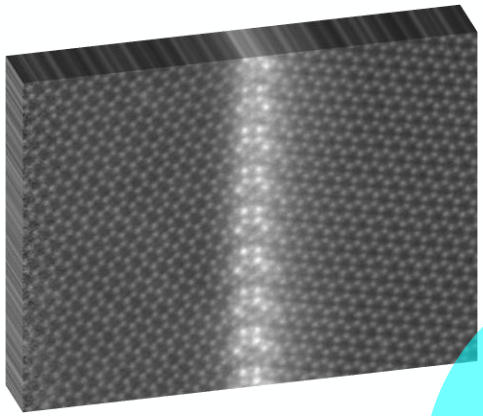
- Various raw data formats
- HDF5 files of simulations
- S3 storage system (CoSciene)
- Electronic Lab Notebook (OpenBIS) & experimental data/metadata management



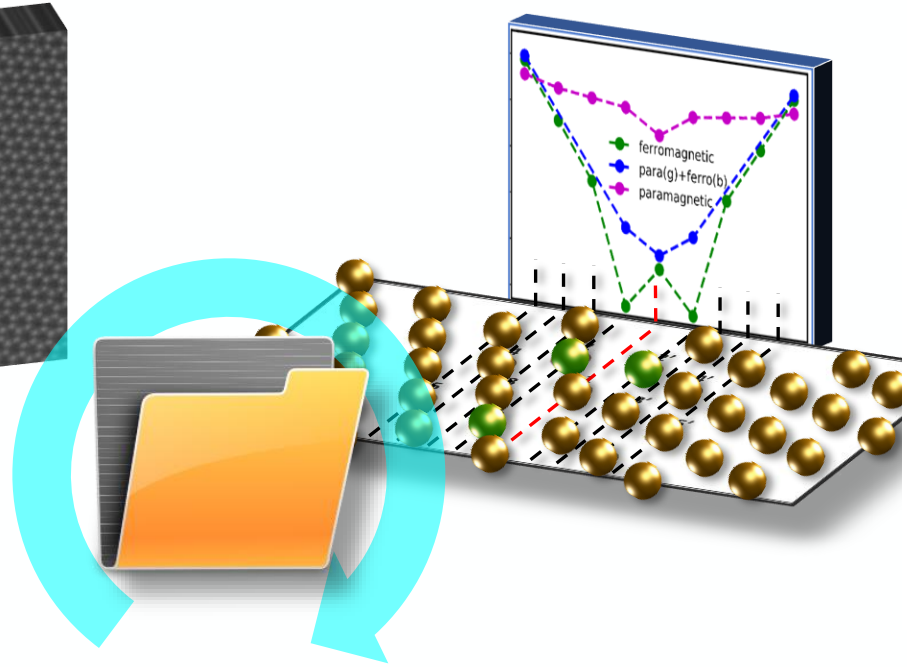
- Defect formation energies for variety of chemical & structural configurations
- Dependence on chemical potential
- Defect phase diagrams

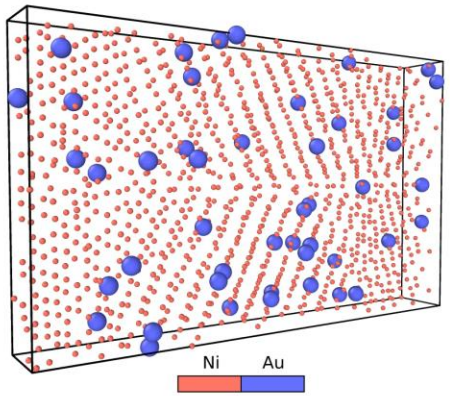


Large scale analysis

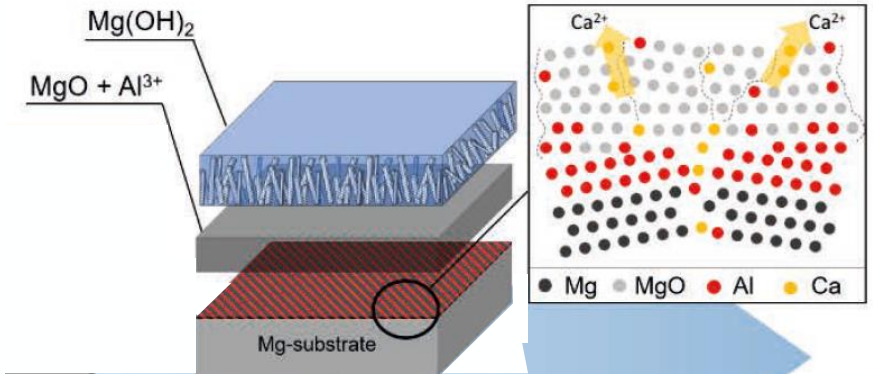


Atomic scale characterizations

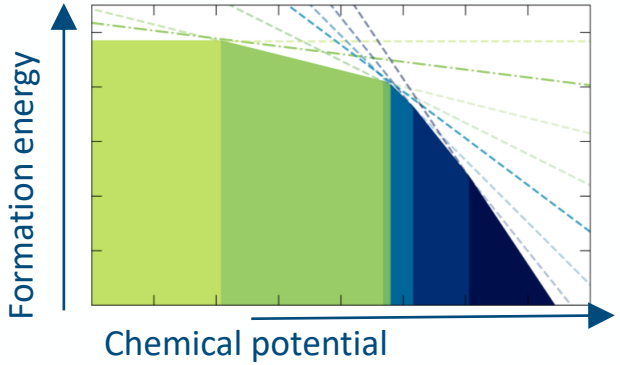


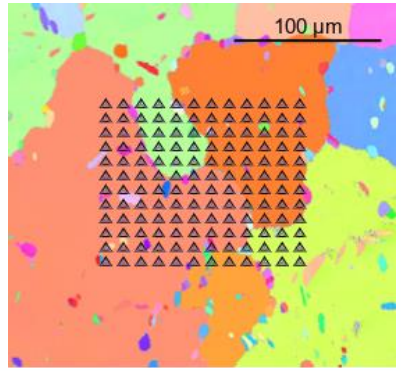


Modelling & Atomistic simulations



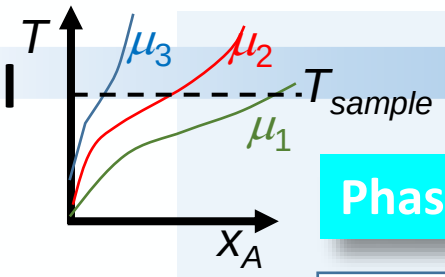
Corrosion analysis



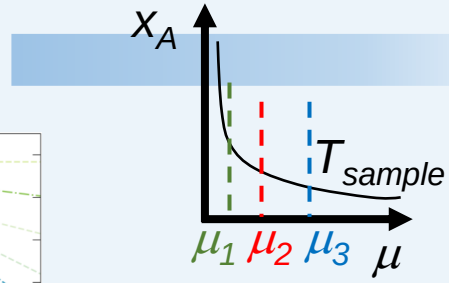


Microstructure and property analysis

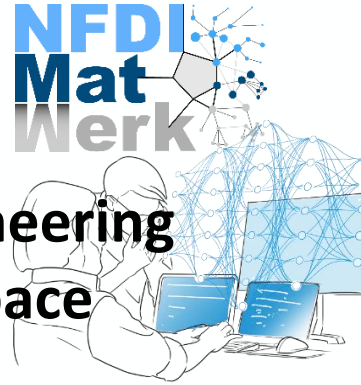
Experimental space



μ-space

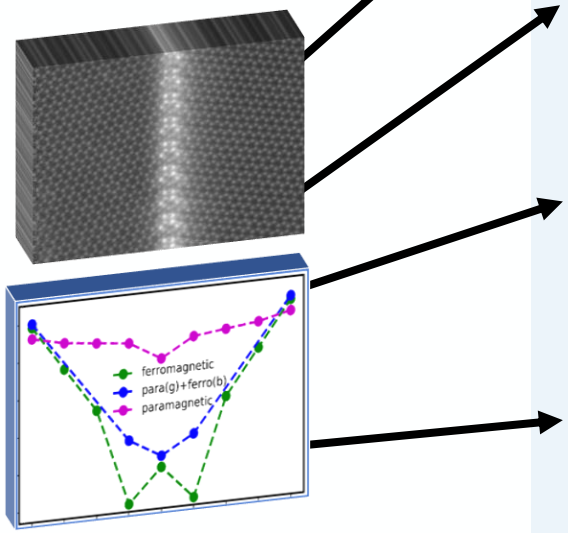


Engineering space

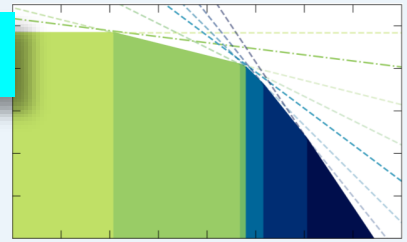


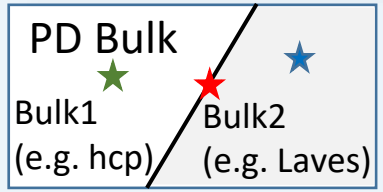
Global and µm-scale sample conditions

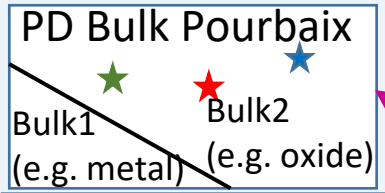
Samples/ Experiments

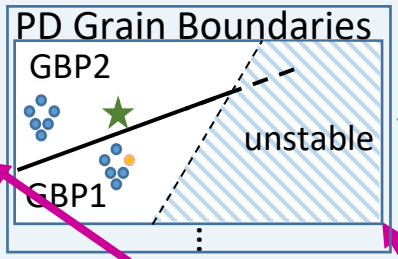


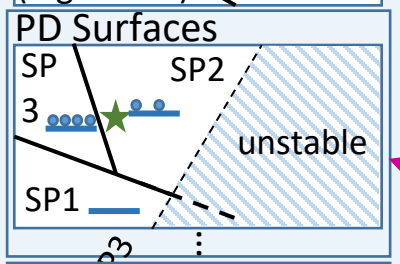
Phase diagrams

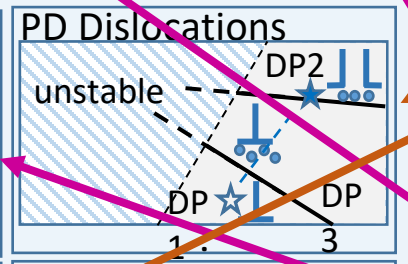


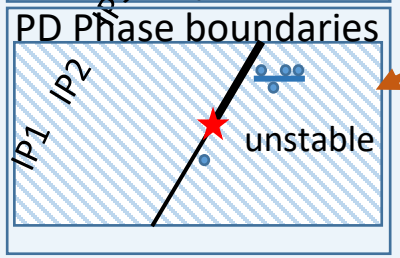


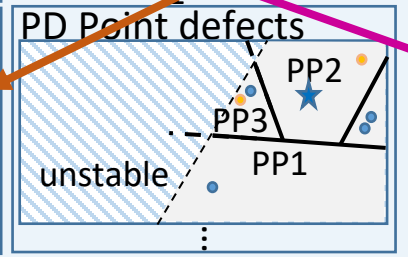




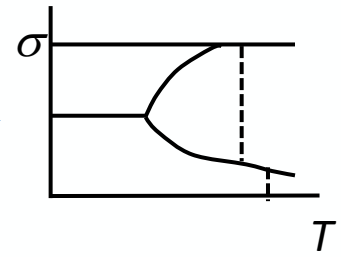




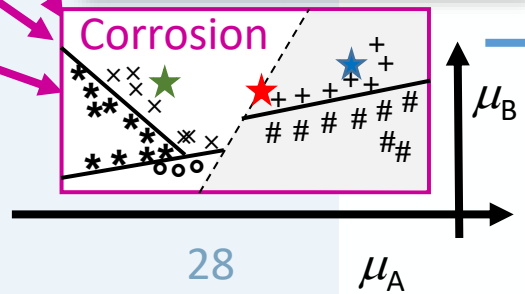


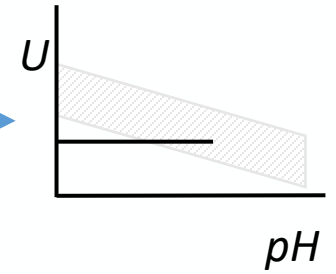


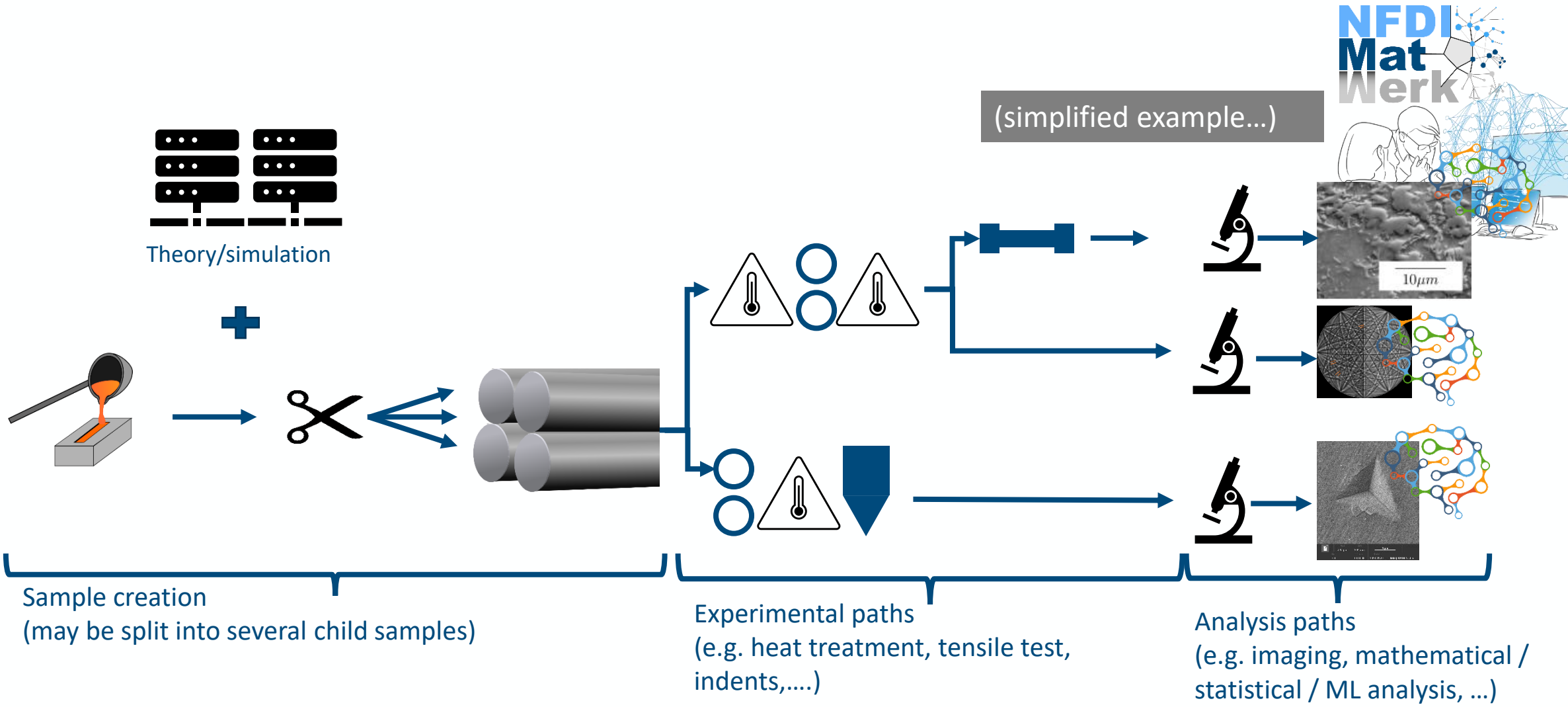
Guided probing of mechanisms and properties



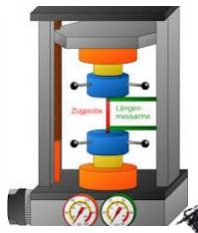
Mechanism-property diagrams

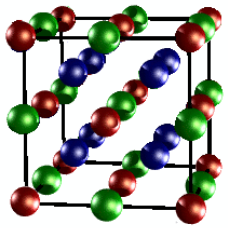




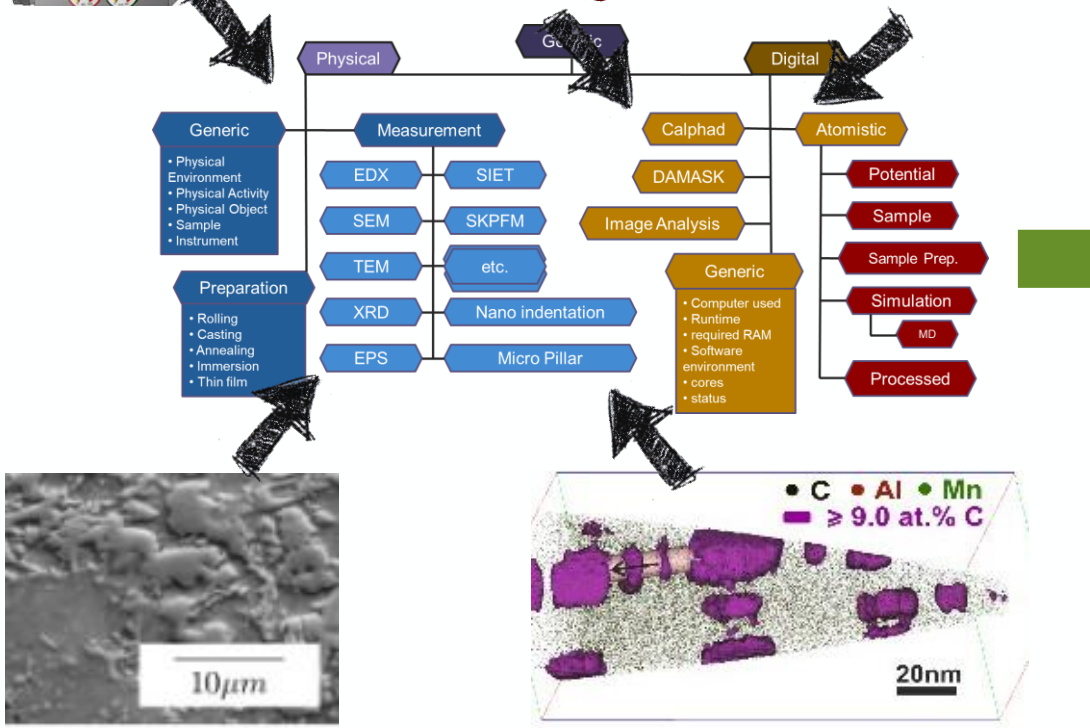


- Tracking of samples and (chain of) experiments & analyses (incl. hierarchies)
- Capture data in variety of environments (from workshop to computing cluster)
- Capture and extract metadata from instruments & procedures





pyron



NFDI MatWerk

Generic Fields

```
ID*              :   (string) # ID for the object, be it sample, experiment, sim...
External/alias ID :   (string) # To associate an additional ID
User*            :   (string)
Date*            :   (date)   # Default is today.
Affiliation      :   (string)
DOIs             :   (string) # To associate publications produced using this object.
Comments         :   (string)
```

Generic Fields | **Physical Environment**

```
Temperature [°C]                    :   (string)
Relative Humidity [%]               :   (string)
Environmental gas                   :   (string)
```

Generic Fields | **Physical Environment** | **Physical Activity** | **Experiment XY**

```
Operator          :   (string)   # The person who did the
Instrument ID     :   (string)   # Which device was used
```

zenodo

DOI:

DOI 10.5281/zenodo.6513745

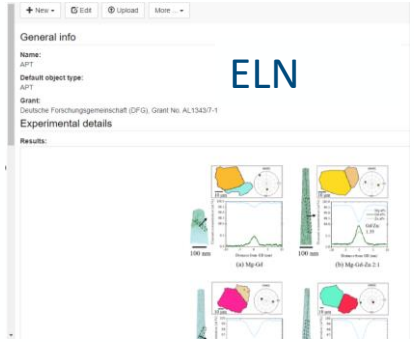
License (for files):

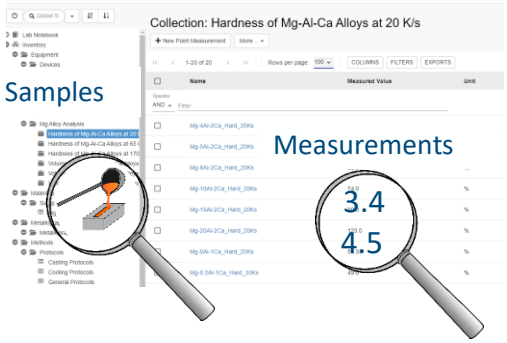
Creative Commons Attribution 4.0 International

Established common terminology &
metadata schema - implemented on Coscine

Coscine

Fair data objects
Long-term storage

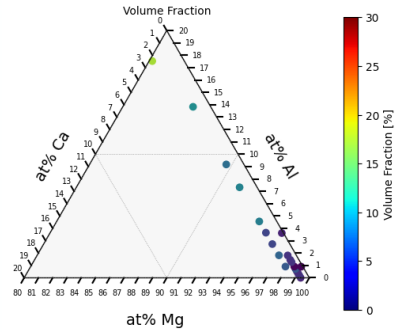


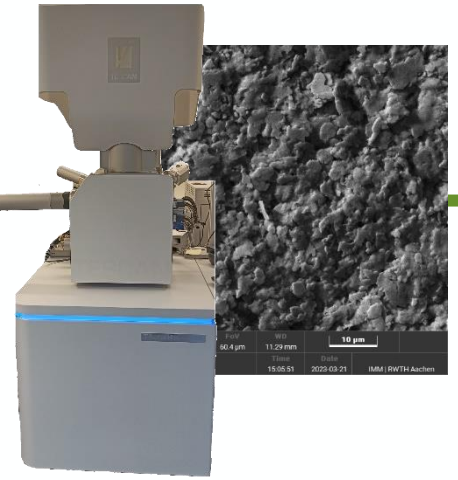


jupyter

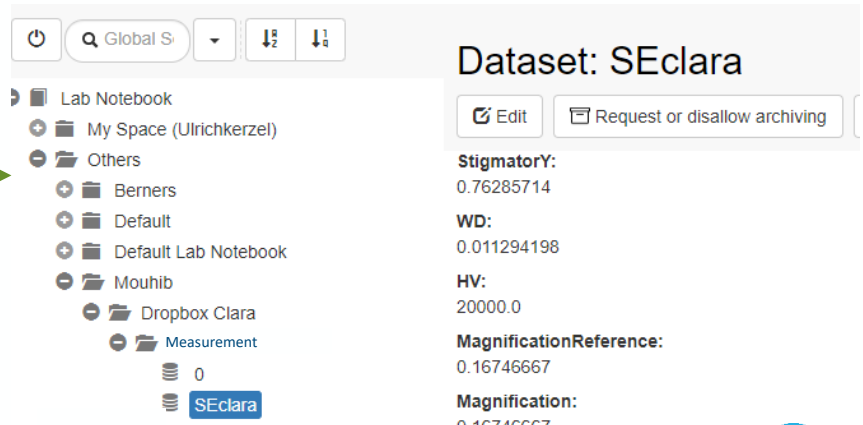
	Name	Element X	AT% X	Element Y	AT% Y	Element Z	AT% Z	Measured Value	Unit
0	Mg-0Al-1Ca	Mg	99.39	Al	0.0	Ca	0.61	3.39	%
1	Mg-0.2Al-1Ca	Mg	99.21	Al	0.18	Ca	0.61	4.92	%
2	Mg-0.4Al-1Ca	Mg	99.03	Al	0.36	Ca	0.61	4.16	%
3	Mg-0.6Al-1Ca	Mg	98.85	Al	0.54	Ca	0.61	3.88	%
4	Mg-0.8Al-1Ca	Mg	98.67	Al	0.72	Ca	0.61	8.2	%

Analysis

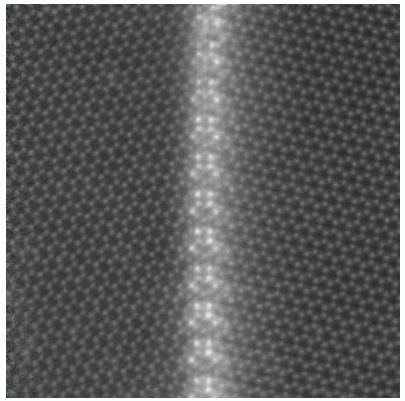




Automatic upload &
Metadata extraction



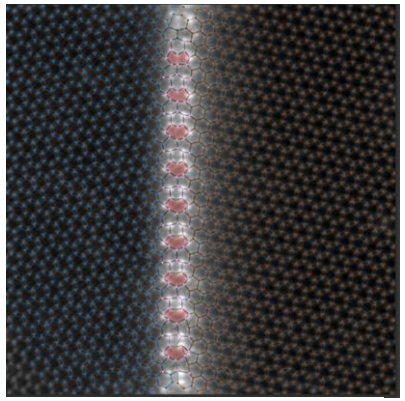
Market - place



HR-TEM

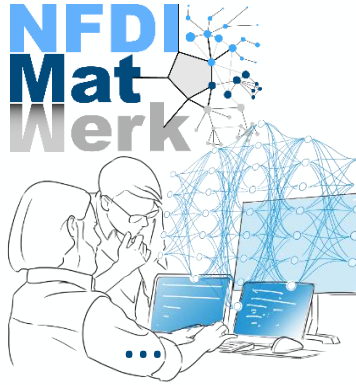
Coscine openBIS

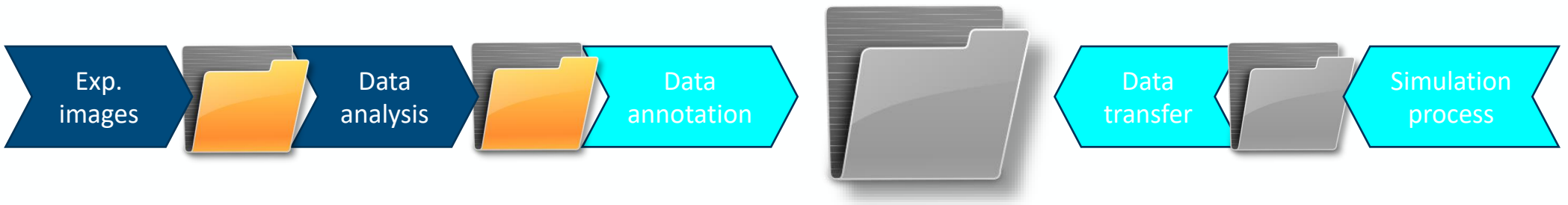
HPC cluster

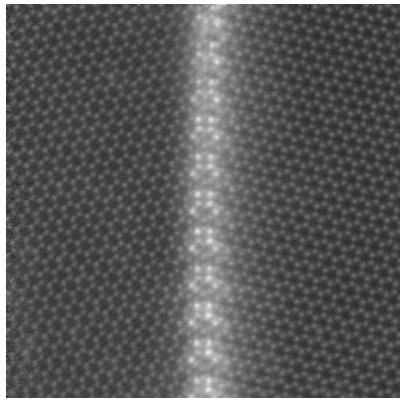


Mathematical pattern ID
("Bump Fitting")

Coscine openBIS

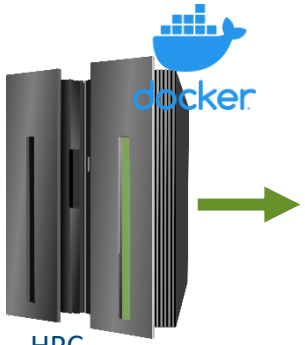




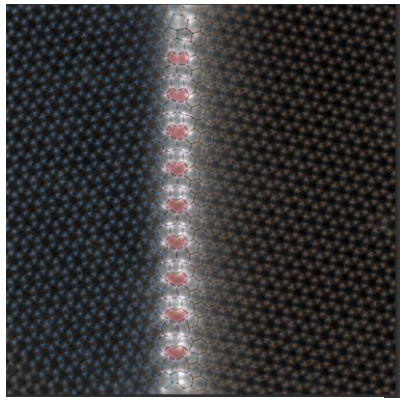


HR-TEM

Coscine openBIS

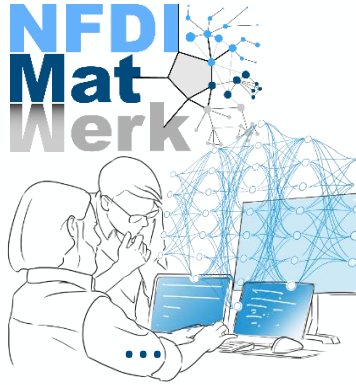


HPC cluster

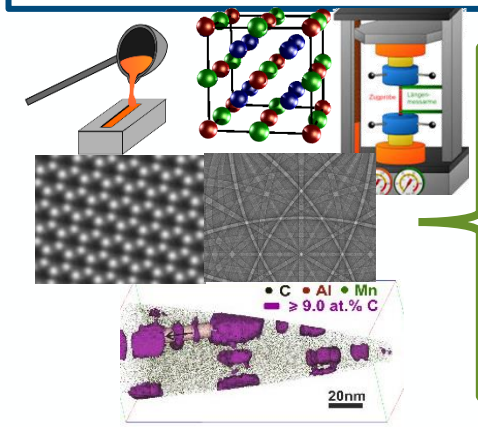


Mathematical pattern ID
("Bump Fitting")

Coscine openBIS



Workflows for DPD (across CRC)



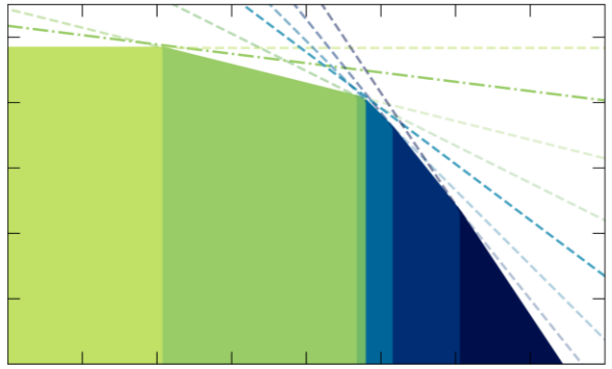
openBIS

pyron

Coscine

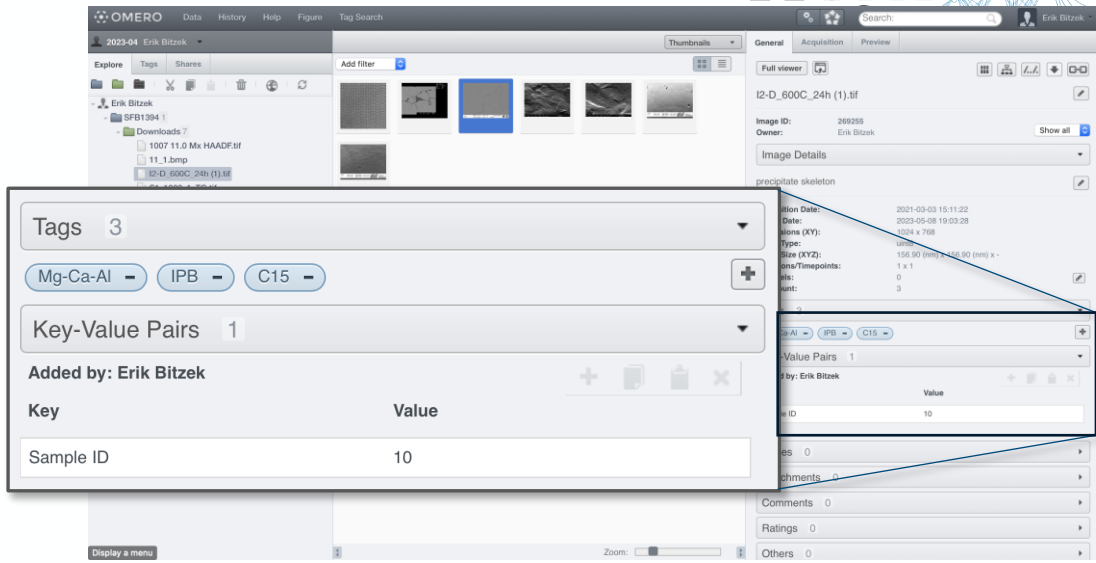
jupyter

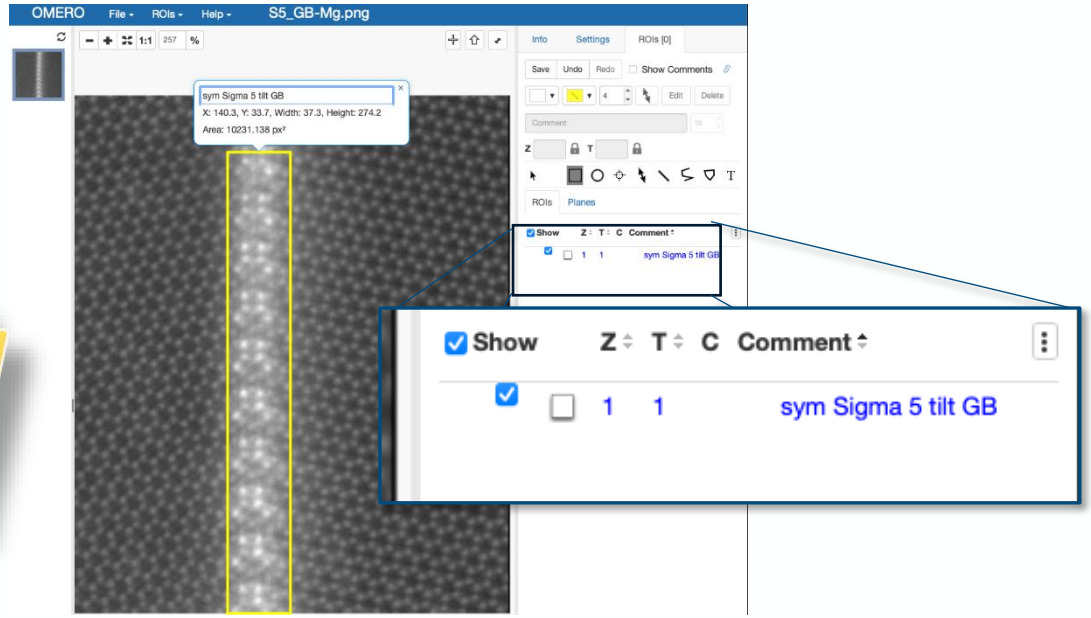
Formation energy



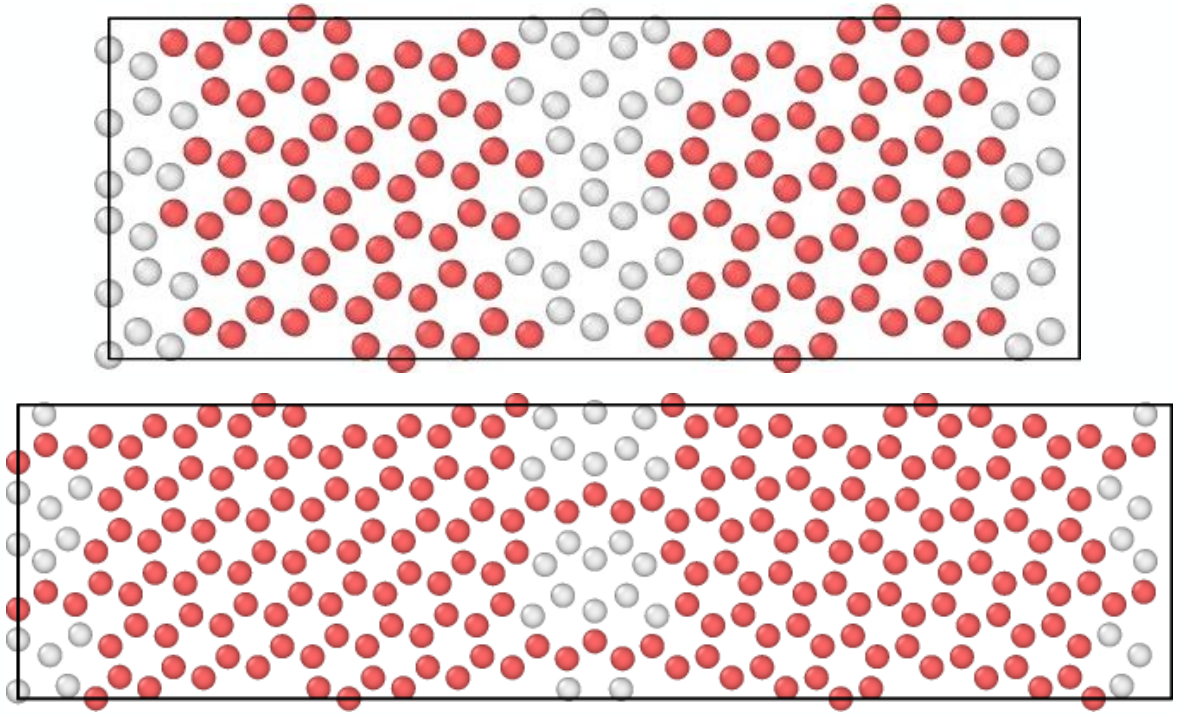
Chemical potential

- Over 150 image file formats supported, including all major microscope formats
- Extracts automatically metadata, additional metadata can be provided, metadata templates
- View, **annotate** (**Image details, key-value pairs, annotated ROIs,** can be **searched** and exported)**,** organize, analyze & share data
- Possibility for scripting (e.g. analyzing, publishing),
- API
- Integration in ecosystem with OpenBis demonstrated





- Two sigma 7 grain boundaries (GBs) in hcp Mg:
  - Σ=7 [0001] | 38.2° Sym. plane (-1 5 -4 0)
  - Σ=7 [0001] | 21.78° Sym. Plane (1 2 -3 0)
- Two types of sigma 7 21.78° GBs:



$$\gamma_{GB} = \frac{E_{GB} - E_{bulk}}{2A_{GB}}$$

T – type

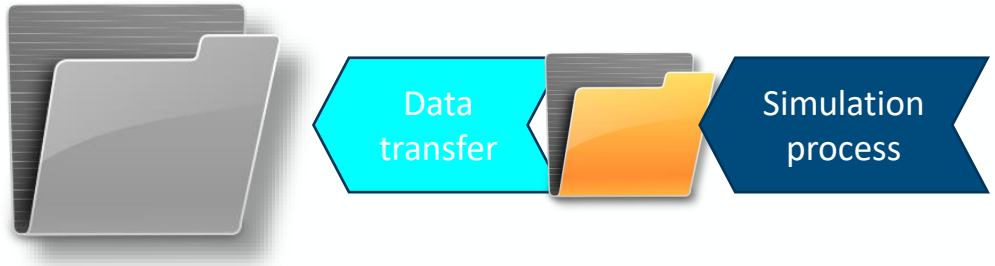
GB Energy = 308.6 mJ/m^2

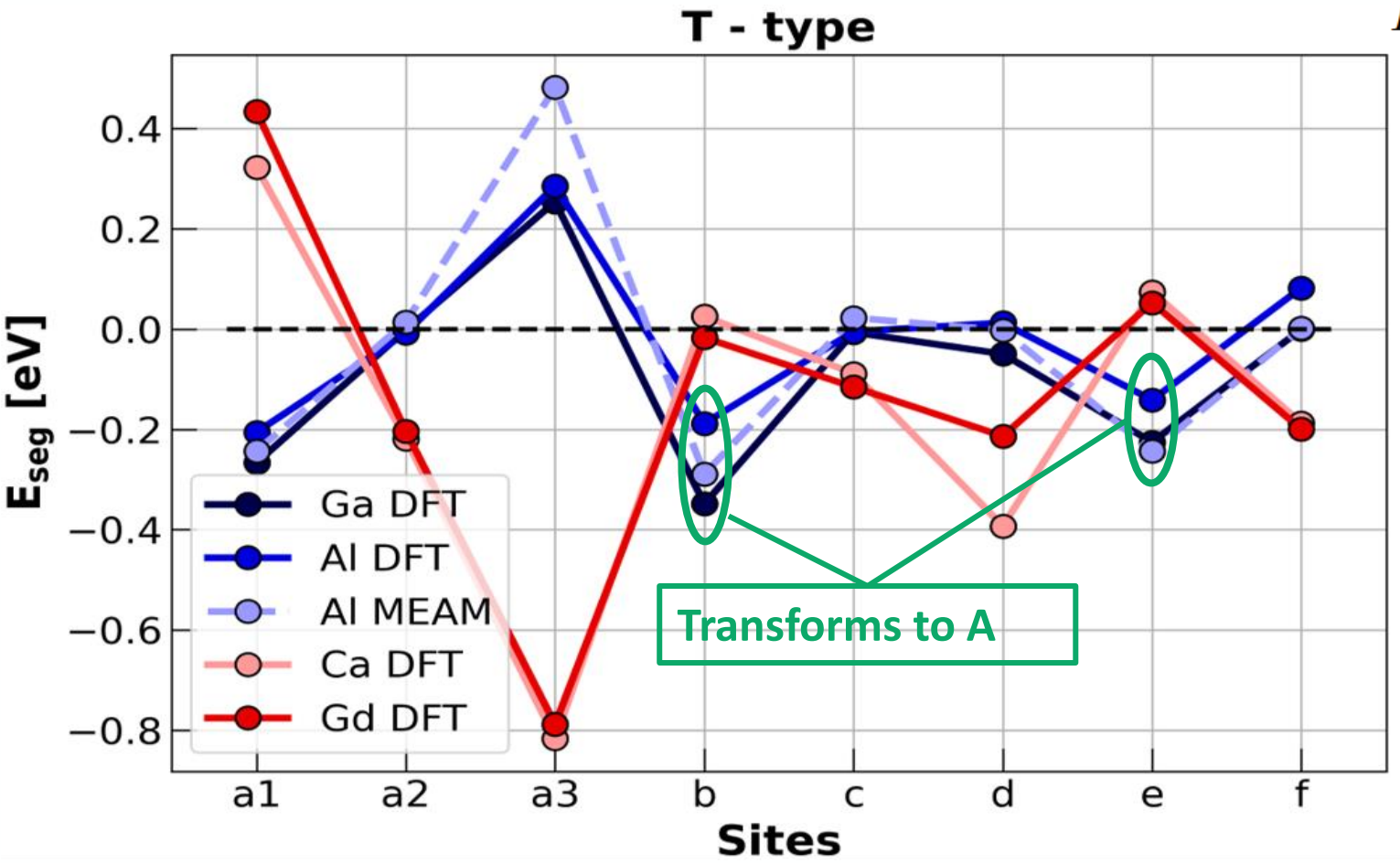
Excess Vol. = 0.22271 A

A – type

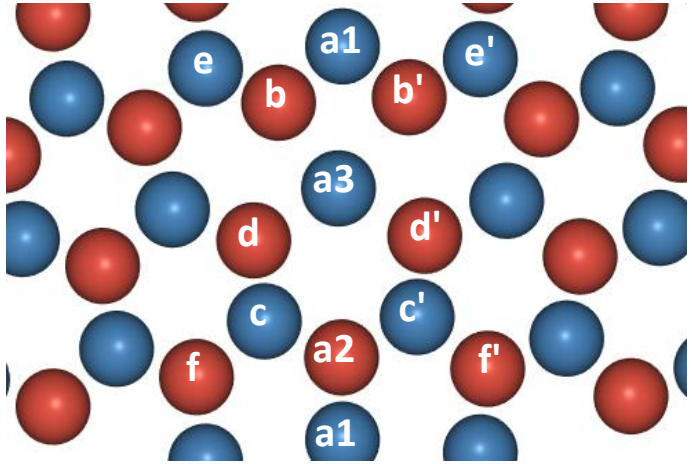
GB Energy = 311.2 mJ/m^2

Excess Vol. = 0.27256 A



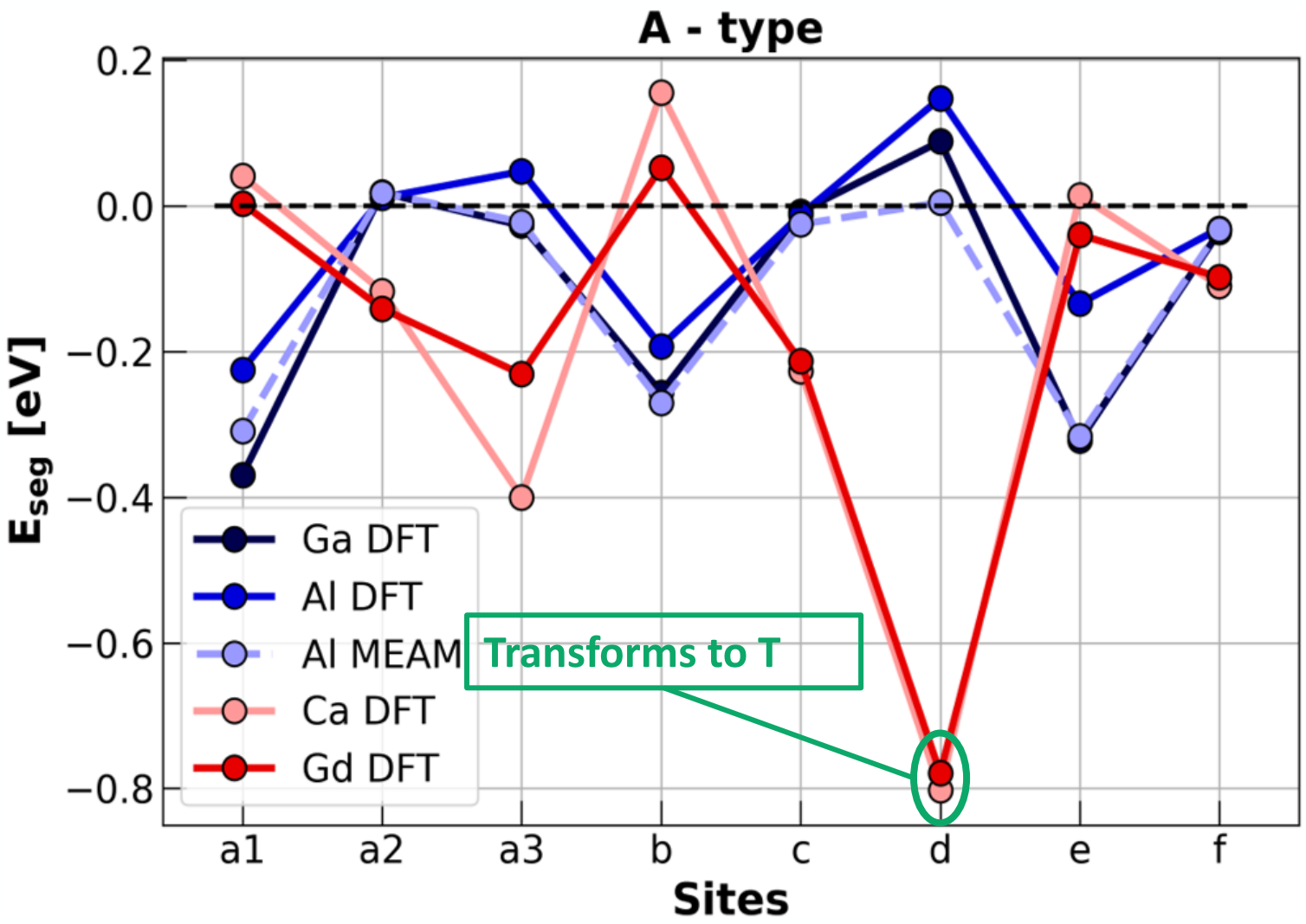


$$E_{\text{seg}} = [E(\text{Mg}_{N-1}\text{Y}) - E(\text{Mg}_N)]_{\text{GB}} - [E(\text{Mg}_{M-1}\text{Y}) - E(\text{Mg}_M)]_{\text{bulk}}$$

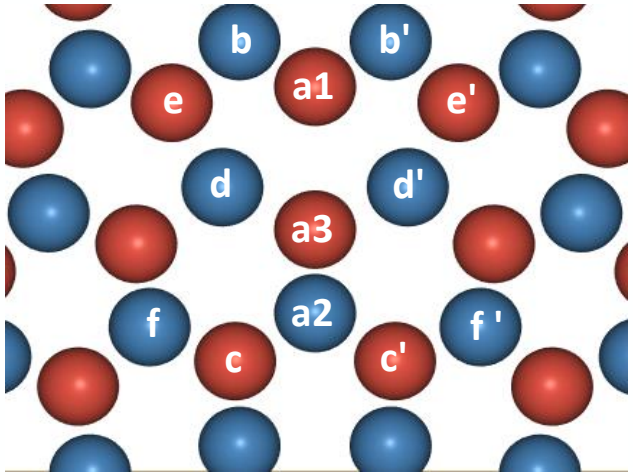


- Ga and Al show similar segregation behavior
- Ca shows different segregation preference
- Mg, Al stabilize A-type; most stable site *a1*
- Mg-Al MEAM potential, segregation behavior of Al similar to Ga DFT

Y.-M. Kim, N.J. Kim, and B.-J. Lee (2009), "Atomistic Modeling of pure Mg and Mg-Al systems", *Calphad*, **33(4)**, 650-657. DOI: 10.1016/j.calphad.2009.07.004.



$$E_{\text{seg}} = [E(\text{Mg}_{N-1}\text{Y}) - E(\text{Mg}_N)]_{\text{GB}} - [E(\text{Mg}_{M-1}\text{Y}) - E(\text{Mg}_M)]_{\text{bulk}}$$



- Ga and Al, smaller than Mg, show similar segregation behavior
- Ca - larger atom compared to Mg, Al; shows different segregation preference
- Ca stabilizes T-type; most stable site *a3*

NFDI MatWerk

Launcher | DPDs-IUC04.ipynb

```
[1]: from pyiron import Project
     import numpy as np
     import matplotlib.pyplot as

     from itertools import combin

     %config inlinebackend.figure

[2]: pr = Project('DPD')

[3]: mu_Mg = -1.5065073075 # ener
     mu_Ga  = -2.9067873625 # ene

[4]: structure = pr.create.struct
     structure_t = pr.create.stru

[5]: en_0Ga = -236.33363255 # ene
     en_0Ga_t = -242.36916212 # e

[6]: all_labels, all_energies, mi
     for i in range(1, 7):
         labels, energies = np.ge
         all_labels.append([label
         all_energies.append(ener
         min_inds.append(np.argmi

[7]: # Grain boundary Area
     A_t = 2*structure.cell[2][2]

[8]: Ef_0Ga_GB_T = (en_0Ga_t - (l

[9]: mu_range = [-1, 0.5]
     end_mu = mu_Ga + 0.5
```

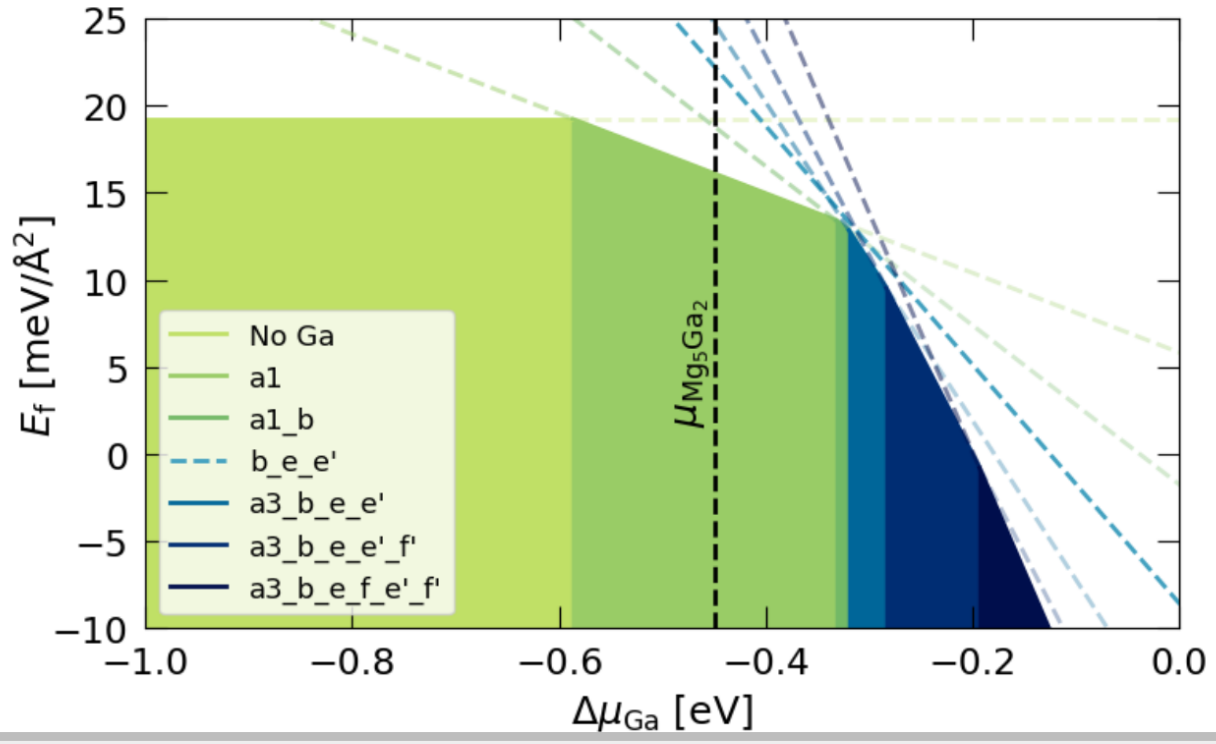
(ipykernel) | Idle

Launcher | DPDs-IUC04.ipynb

Code | Python 3 (ipykernel)

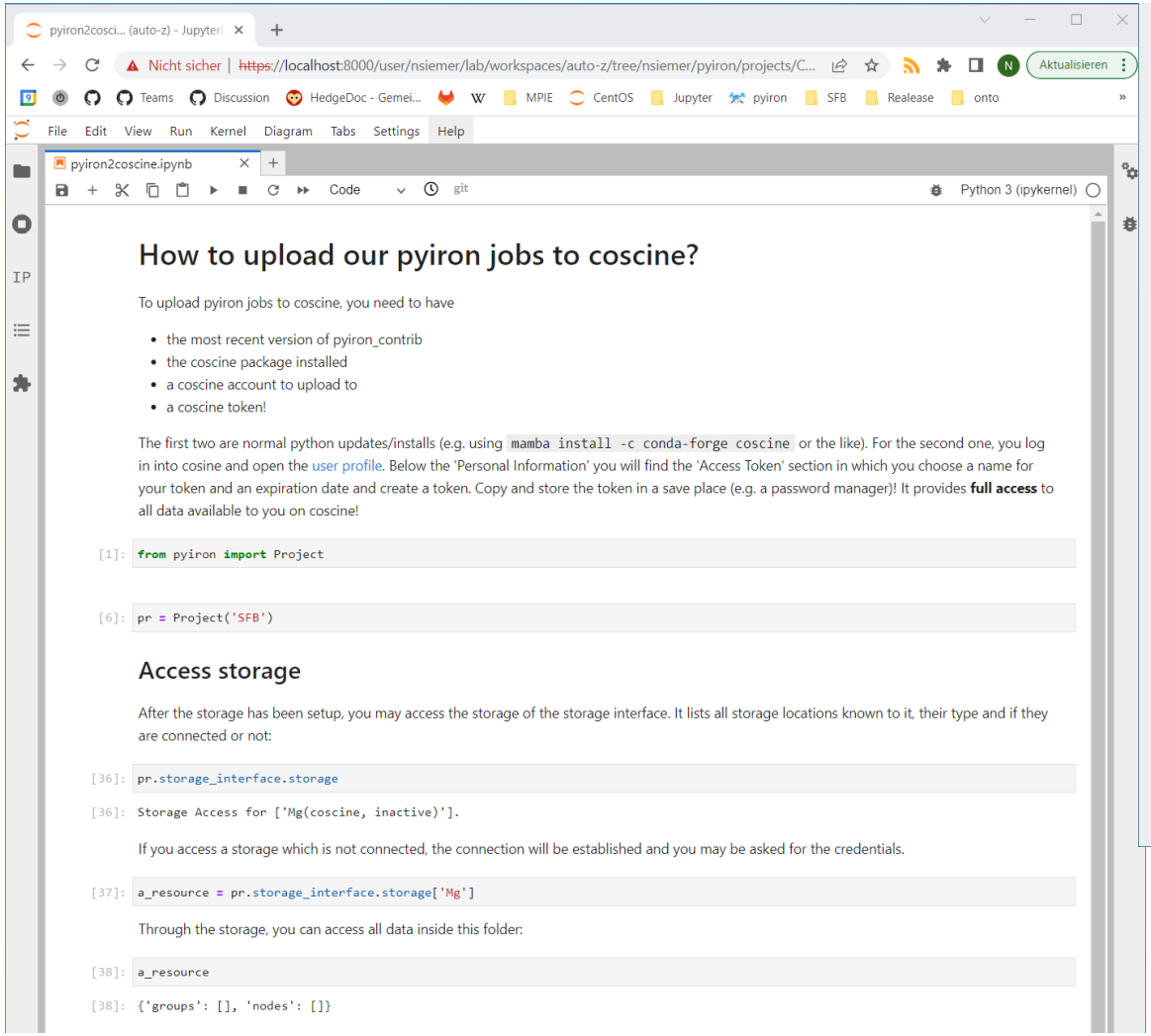
```
[15]: final_mu_Ga = np.linspace(mu_Ga+mu_range[0], mu_Ga+mu_range[-1], 1500)
      DPD = plot_DPD(mu_x=final_mu_Ga-mu_Ga, energies=np.array(all_formation_energies), colors=colors,
                     labels=[all_labels[i][min_inds[i]] for i in range(0,7)])
      DPD.plot(xlim=[-1,0], ylim=[-10,25], alpha_fill=1, legend=True)

      plt.axvline(x=mu_Ga_Mg5Ga_relaxed-mu_Ga, ymax=2, ymin=0, color='k', ls='--', lw=2)
      plt.text(s=r'$\mu_{\rm Mg_5Ga_2}$', x=-0.49, y=2.5, c='k', fontsize=18, rotation=90)
```

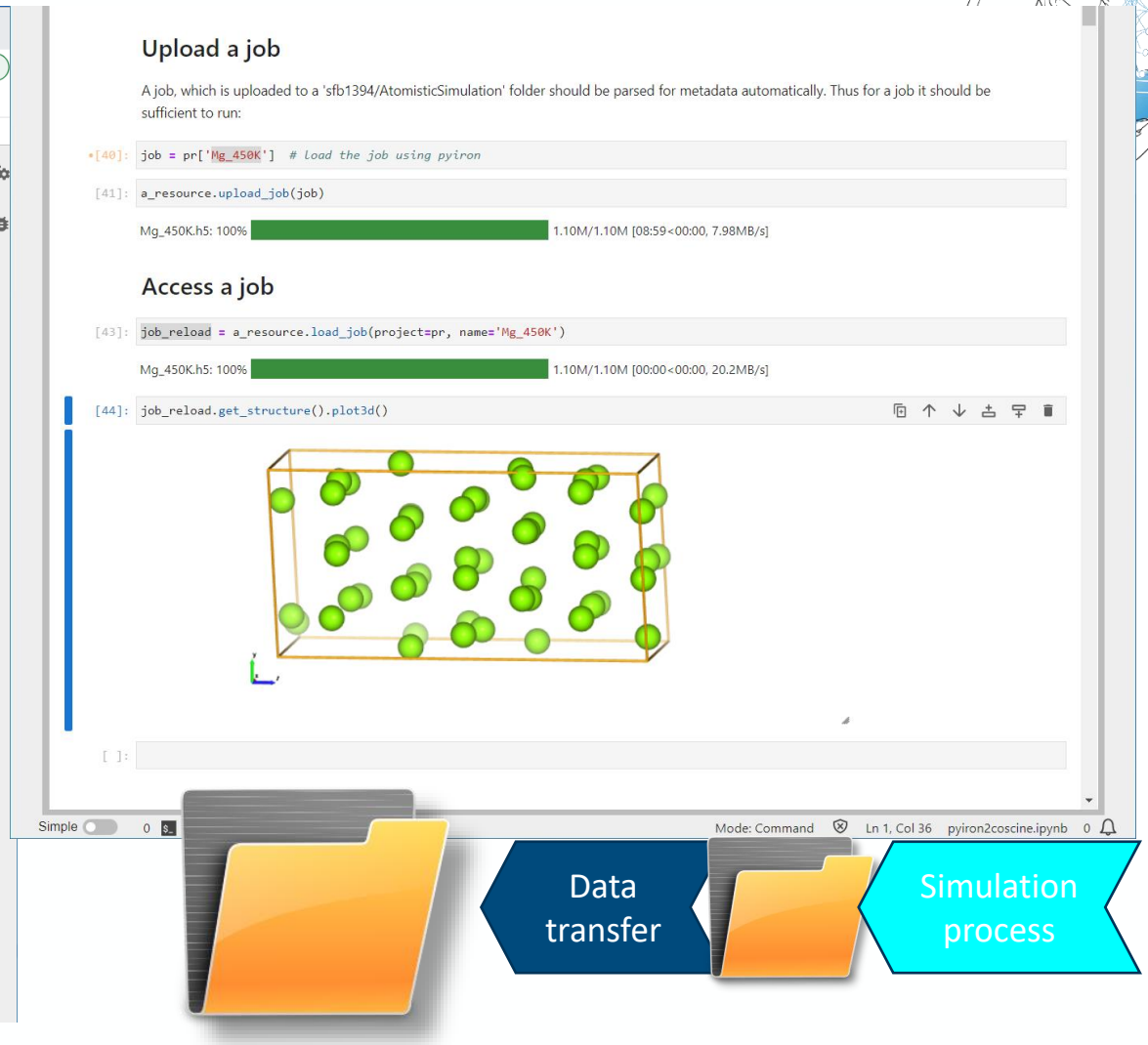
```
[15]: Text(-0.49, 2.5, '$\\mu_{\\rm Mg_5Ga_2}$')
```



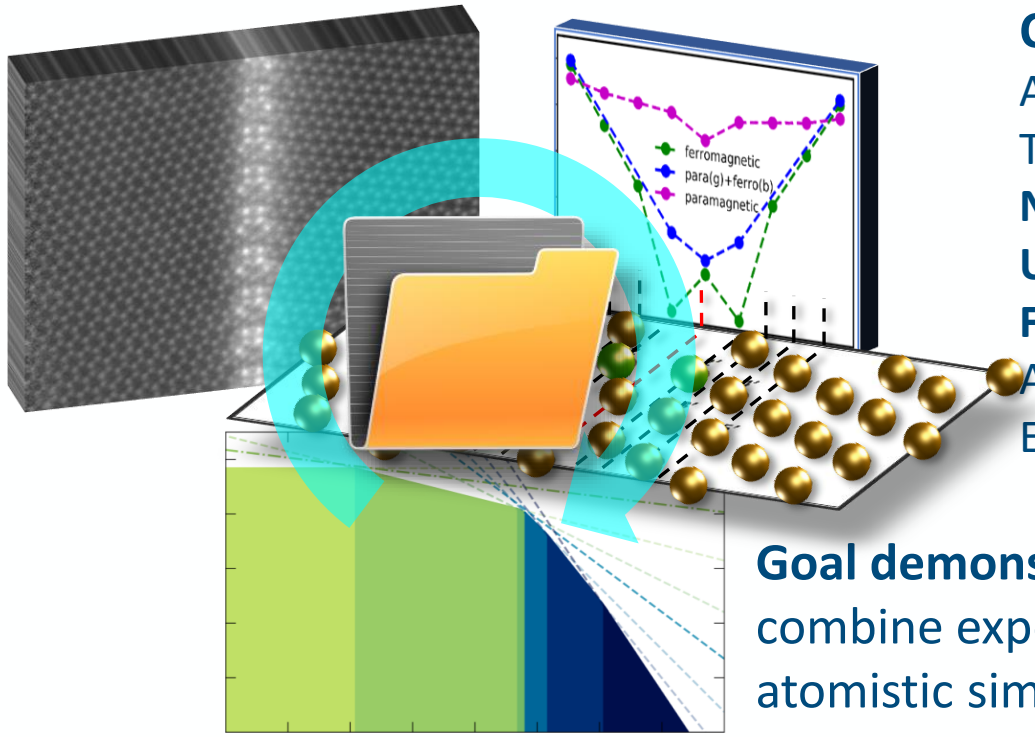
pykernel) | Idle

Mode: Command | Ln 1, Col 1 | DPDs-IUC04.ipynb


pyiron2cosci... (auto-z) - Jupyter
Nicht sicher | https://localhost:8000/user/nsiemer/lab/workspaces/auto-z/tree/nsiemer/pyiron/projects/C...
Aktualisieren
Teams
Discussion
HedgeDoc - Gemei...
MPIE
CentOS
Jupyter
pyiron
SFB
Realease
onto
File Edit View Run Kernel Diagram Tabs Settings Help
pyiron2coscine.ipynb
Code
git
Python 3 (ipykernel)
How to upload our pyiron jobs to coscine?
To upload pyiron jobs to coscine, you need to have
the most recent version of pyiron_contrib
the coscine package installed
a coscine account to upload to
a coscine token!
The first two are normal python updates/installs (e.g. using mamba install -c conda-forge coscine or the like). For the second one, you log in into cosine and open the user profile. Below the 'Personal Information' you will find the 'Access Token' section in which you choose a name for your token and an expiration date and create a token. Copy and store the token in a save place (e.g. a password manager)! It provides full access to all data available to you on coscine!
[1]: from pyiron import Project
[6]: pr = Project('SFB')
Access storage
After the storage has been setup, you may access the storage of the storage interface. It lists all storage locations known to it, their type and if they are connected or not:
[36]: pr.storage_interface.storage
[36]: Storage Access for ['Mg(coscine, inactive)'].
If you access a storage which is not connected, the connection will be established and you may be asked for the credentials.
[37]: a_resource = pr.storage_interface.storage['Mg']
Through the storage, you can access all data inside this folder:
[38]: a_resource
[38]: {'groups': [], 'nodes': []}


Upload a job
A job, which is uploaded to a 'sfb1394/AtomisticSimulation' folder should be parsed for metadata automatically. Thus for a job it should be sufficient to run:
*[40]: job = pr['Mg_450K'] # load the job using pyiron
[41]: a_resource.upload_job(job)
Mg_450K.h5: 100% 1.10M/1.10M [08:59<00:00, 7.98MB/s]
Access a job
[43]: job_reload = a_resource.load_job(project=pr, name='Mg_450K')
Mg_450K.h5: 100% 1.10M/1.10M [00:00<00:00, 20.2MB/s]
[44]: job_reload.get_structure().plot3d()
[ ]:
Simple
Mode: Command
Ln 1, Col 36
pyiron2coscine.ipynb

Data transfer
Simulation process



Contributors:

Anika Lenze (IWM),
Tilmann Hickel (MPIE),
Niklas Siemer (MPIE),
Ulrich Kerzel (RWTH),
Fatim-Zahra Mouhib (RWTH),
Amirreza Moghaddam (RWTH),
Ebrahim Norouzi (FIZ),
Philipp-Joachim Ost (KIT),
Elias Vitali (KIT),
Matthias Grönewald (TU Darmstadt),
Sarath Menon (MPIE),
Erik Bitzek (MPIE),
Prince Mathews (MPIE),
Steffen Brinkmann (FZ Jülich)

Goal demonstrator: Provide an environment that allows one to combine experimental data from electronmicroscopy and atomistic simulations for defect phase diagrams

Main steps: