Generating and publishing datasets using AiiDA and Materials Cloud

Kristjan Eimre
EPFL, Switzerland

CECAM workshop: Machine Learning Interatomic Potentials and Accessible Databases
Grenoble
11.09.2024

Swiss National Science Foundation
MARVEL
NATIONAL CENTRE OF COMPETENCE IN RESEARCH
MAX DRIVING THE EXASCALE TRANSITION
PAUL SCHERRER INSTITUT PSI
EPFL

Outline

Automated Interactive Infrastructure and Database for Computational Science

- Recent usability improvements

Web platform for seamlessly sharing data and resources

- Materials Cloud 3D crystals database updates
- Archive-OPTIMADE integration

AiiDA

http://www.aiida.net

Computational science infrastructure that provides

- scalable workflow engine
- built-in support for HPC
- flexible plugin system
- automatic full data provenance

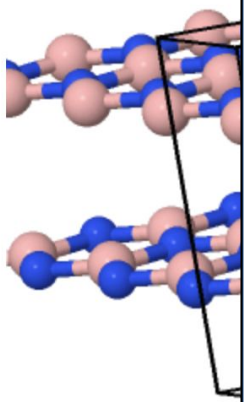
Language: implemented and API in python

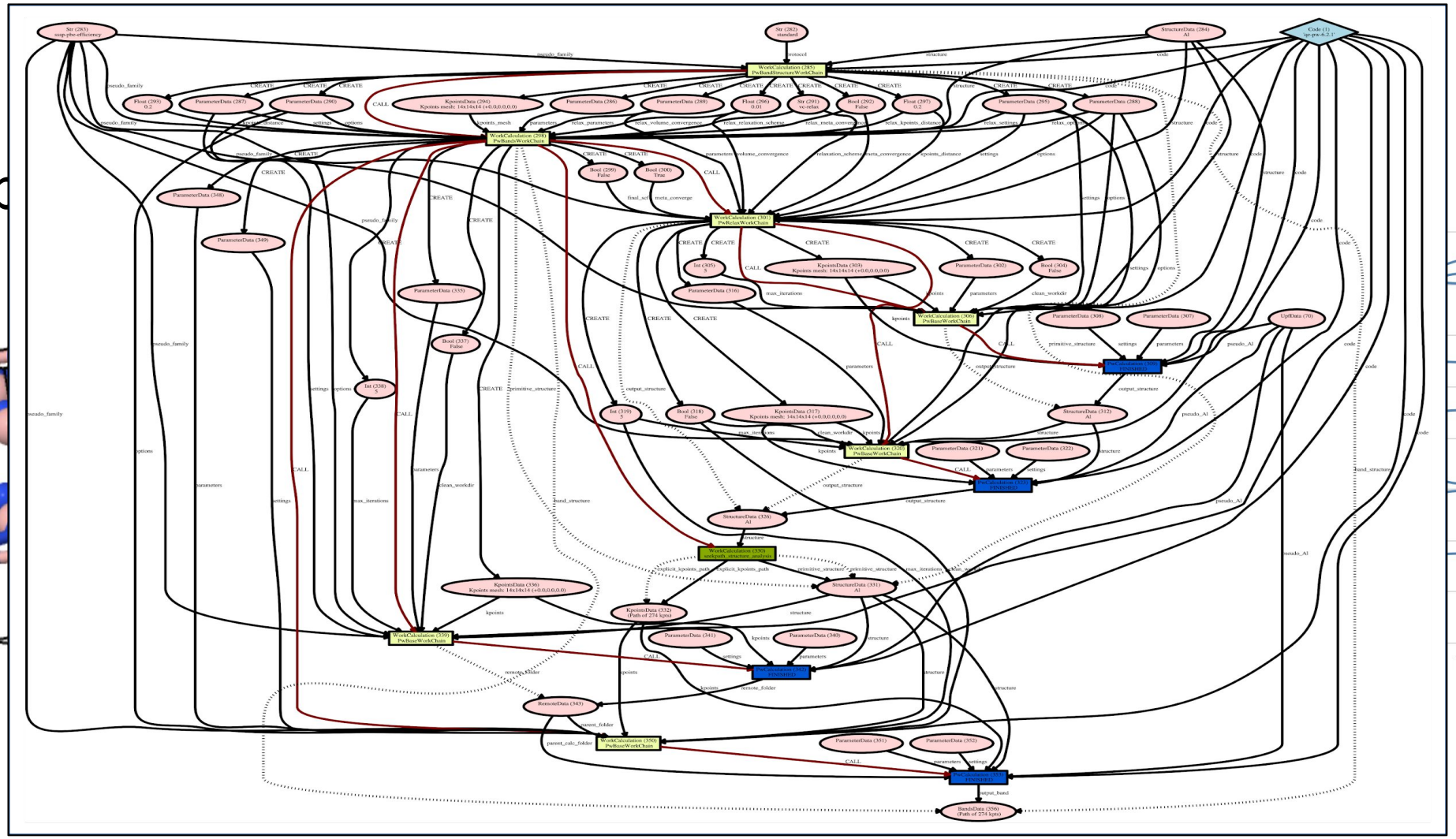
Open source: https://github.com/aiidateam/aiida-core

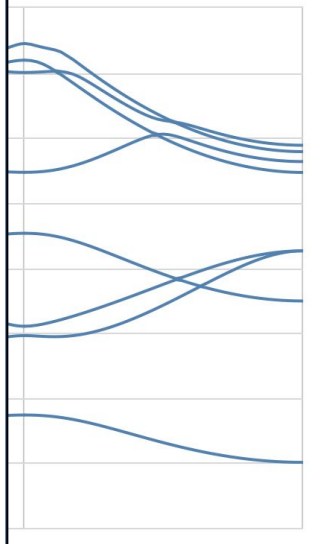
G. Pizzi et al., Comp. Mat. Sci. 111, 218-230 (2016)
S.P. Huber et al., Scientific Data 7, 300 (2020)

Workflow engine

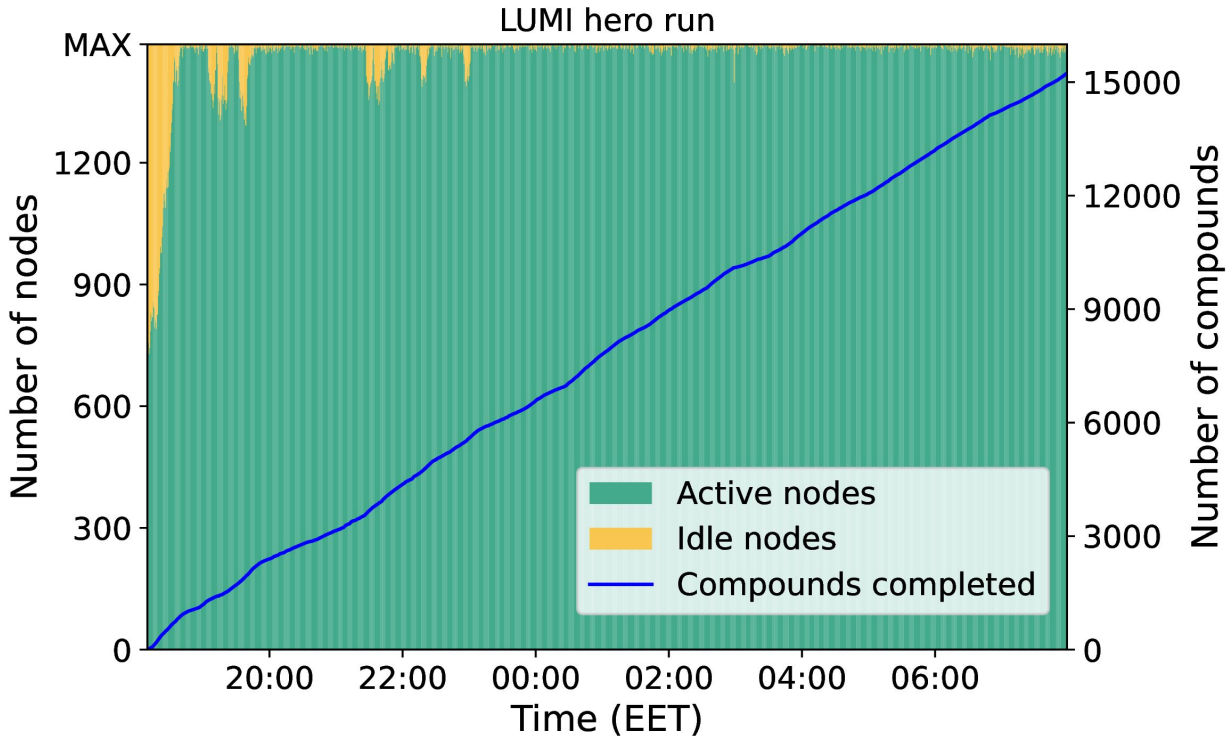
Examp







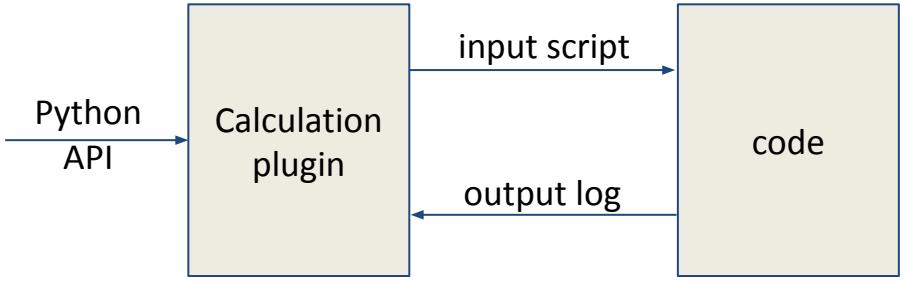
Scalable workflow engine



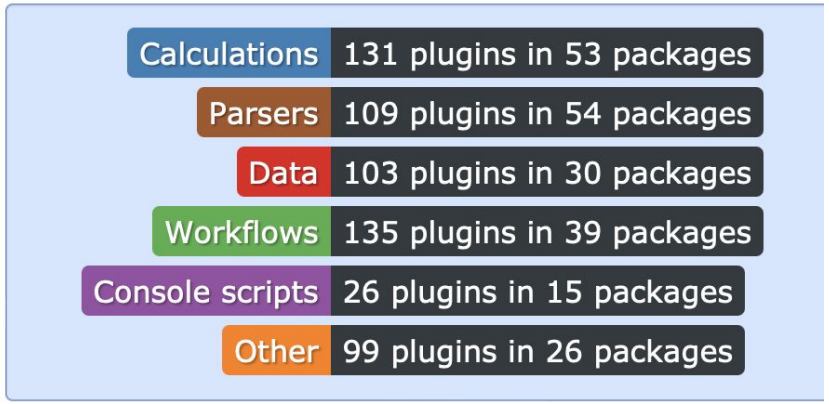
- Geometry relaxations of 15,324 inorganic compounds in 13 hours
- Full partition of LUMI-C: 1,500 nodes with 128 cores each
- 7887 issues dealt with on the fly

Plugins

https://aiidateam.github.io/aiida-registry/



Registered plugin packages: 91



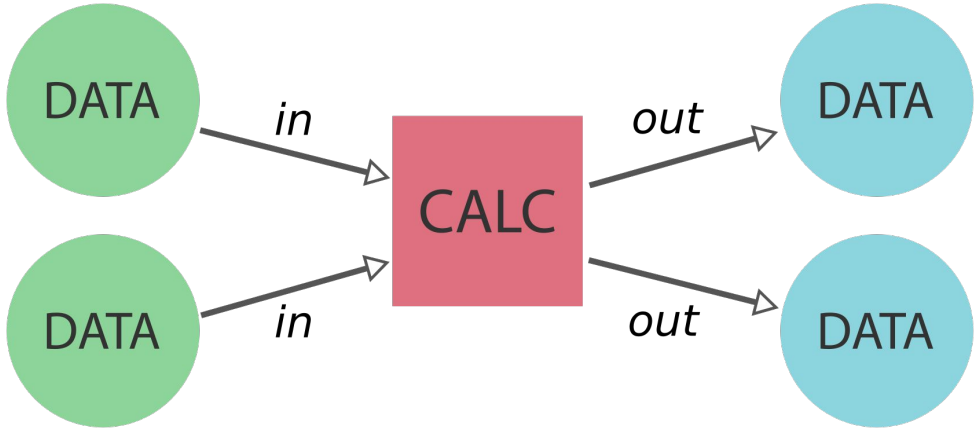
- Plugins collected in the AiiDA plugin registry
- 130+ codes currently supported
- Many are community-contributed

Data provenance

What you get as a bonus: full data provenance!

AiiDA stores:

- data transformations/calculations
- inputs
- outputs
- inter-connections

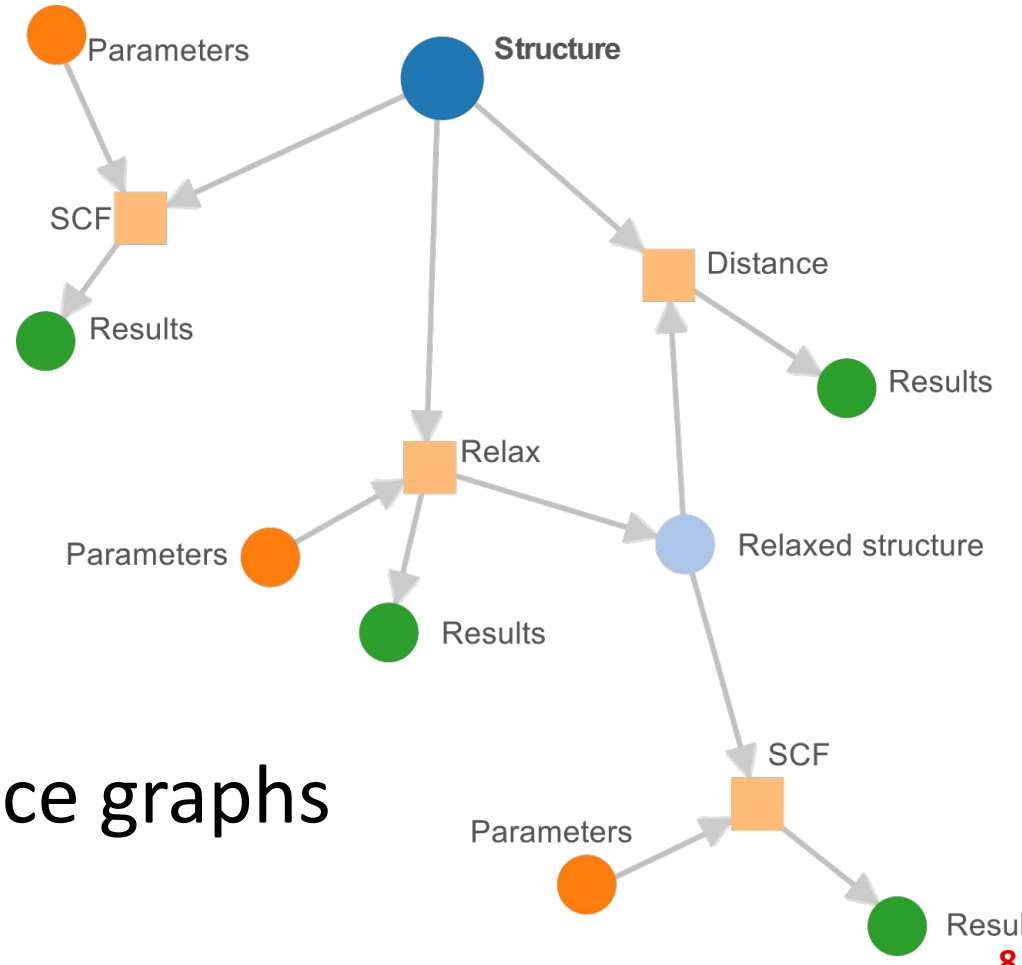


Data provenance

What you get as a bonus: full data provenance!

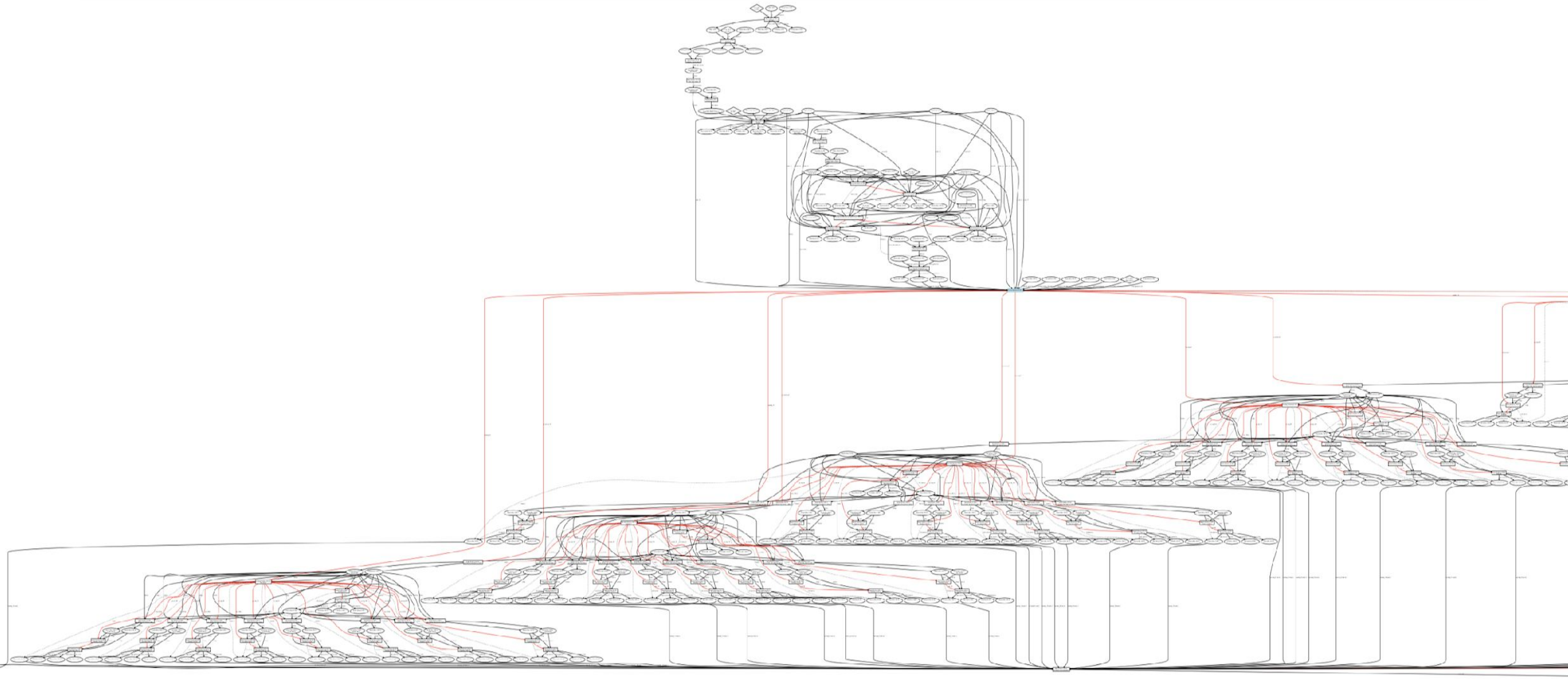
AiiDA stores:

- data transformations/calculations
- inputs
- outputs
- inter-connections



Provenance graphs

Data provenance



AiiDA improvements

Recent effort to improve the usability and reduce the learning curve

AiiDA improvements: installation

Traditional (Complete) installation

- Requires
  - PostgreSQL
  - RabbitMQ

```
$ pip install -q aiida-core
$ verdi profile setup core.psql_dos
Report: enter ? for help.
Report: enter ! to ignore the default and set no
Profile name: new_profile
Set as default? [Y/n]: n
Email Address (for sharing data) [kristjan.eimre@
First name [Kristjan]:
Last name [Eimre]:
Institution [EPFL]:
Use RabbitMQ? [Y/n]:
PostgreSQL engine [postgresql_psycopg2]:
PostgreSQL hostname [localhost]:
PostgreSQL port [5432]:
PostgreSQL username:
```

verdi presto (v2.6.1, July 1st)

```
$ pip install -q aiida-core
$ verdi presto
Report: Option `--use-postgres` not enabled: config
Report: RabbitMQ server not found (Failed to connec
Report: See https://aiida-core.readthedocs.io/en/st
Report: Initialising the storage backend.
Report: Storage initialisation completed.
Success: Created new profile `presto-3`.
Success: Configured the localhost as a computer.
$ verdi status
  ✔ version:     AiiDA v2.6.2
  ✔ config:      /home/kristjan/.aiida
  ✔ profile:     presto-3
  ✔ storage:     SqliteDosStorage[/home/kristjan/.ai
  ● broker:      No broker defined for this profile:
  ● daemon:      No broker defined for this profile:
$
```

Not suitable for high-throughput

AiiDA improvements: aiida-shell

aiida-shell: run any executable through AiiDA without a plugin

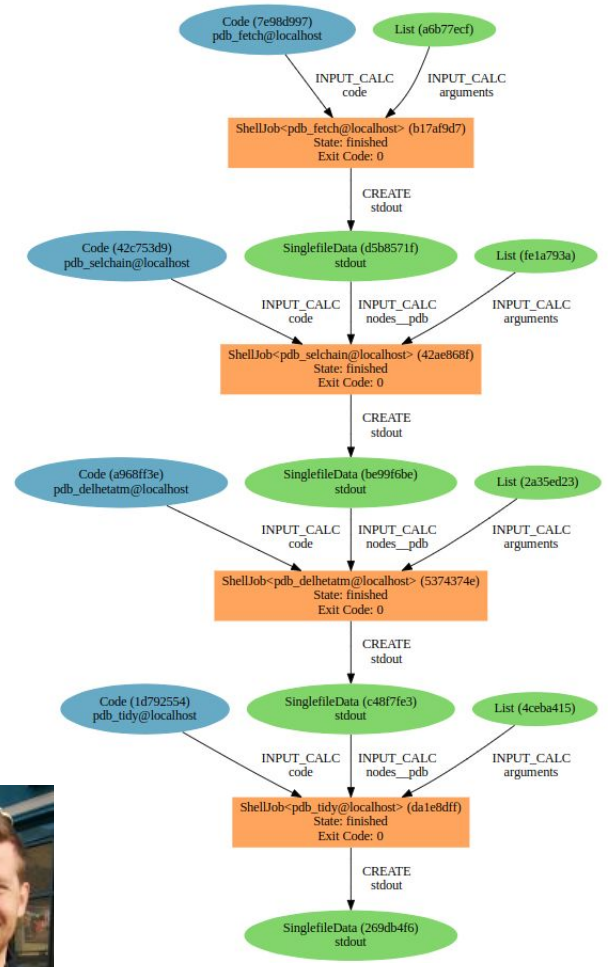
Example shell commands (manipulating Protein Data Bank files):

```
$ pdb_fetch 1brs | pdb_selchain -A,D | pdb_delhetatm | pdb_tidy > 1brs_AD_noHET.pdb
```

Run via aiida-shell:

```
from aiida_shell import launch_shell_job

results, node = launch_shell_job('pdb_fetch', '1brs')
results, node = launch_shell_job('pdb_selchain', '-A,D {pdb}', {'pdb': results['stdout']})
results, node = launch_shell_job('pdb_delhetatm', '{pdb}', {'pdb': results['stdout']})
results, node = launch_shell_job('pdb_tidy', '{pdb}', {'pdb': results['stdout']})
```



https://github.com/sphuber/aiida-shell

Dr. Sebastiaan P. Huber

AiiDA improvements: WorkGraph

Workflow: (x + y) * z extending to **(x + y) * z + a**

Calculations

```
@calcfunction
def add(x, y):
 return Int(x + y)

@calcfunction
def multiply(x, y):
 return Int(x * y)
```

WorkChain

```
class AddMultiplyWorkChain(WorkChain):

    @classmethod
    def define(cls, spec):
        super().define(spec)
        spec.input('x')
        spec.input('y')
        spec.input('z')
        spec.outline(
            cls.add,
            cls.multiply,
            cls.results,
        )
        spec.output('result')

    def add(self):
        self.ctx.sum = add(self.inputs.x,
self.inputs.y)

    def multiply(self):
        self.ctx.product =
multiply(self.ctx.sum, self.inputs.z)

    def results(self):
        self.out('result', self.ctx.product)
```

WorkGraph

```
wg = WorkGraph("add_multiply")
wg.tasks.new(add, name="add")
wg.tasks.new(multiply, name="multiply",
    x=wg.tasks["add"].outputs["result"])
```

https://github.com/aiidateam/aiida-workgraph

Dr. Xing Wang

AiiDA improvements: WorkGraph

Workflow: (x + y) * z extending to **(x + y) * z + a**

```
class AddMultiplyAddWorkChain(WorkChain):

    @classmethod
    def define(cls, spec):
        super().define(spec)
        spec.expose_inputs(AddMultiplyWorkChain,
namespace='add_multiply')
        spec.input('a')
        spec.outline(
            cls.add_multiply,
            cls.add2,
            cls.results,
        )
        spec.output('result')

    def add_multiply(self):
        future =
self.submit(AddMultiplyWorkChain,
**self.exposed_inputs(AddMultiplyWorkChain,
'add_multiply'))
        return ToContext(add_multiply=future)

    def add2(self):
        self.ctx.sum =
add(self.ctx.add_multiply.outputs.result,
self.inputs.a)

    def results(self):
        self.out('result', self.ctx.sum)
```

WorkChain

```
class AddMultiplyWorkChain(WorkChain):

    @classmethod
    def define(cls, spec):
        super().define(spec)
        spec.input('x')
        spec.input('y')
        spec.input('z')
        spec.outline(
            cls.add,
            cls.multiply,
            cls.results,
        )
        spec.output('result')

    def add(self):
        self.ctx.sum = add(self.inputs.x,
self.inputs.y)

    def multiply(self):
        self.ctx.product =
multiply(self.ctx.sum, self.inputs.z)

    def results(self):
        self.out('result', self.ctx.product)
```

WorkGraph

```
wg = WorkGraph("add_multiply")
wg.tasks.new(add, name="add")
wg.tasks.new(multiply, name="multiply",
   x=wg.tasks["add"].outputs["result"])
```

```
wg.tasks.new(add, name="add2",
  x=wg.tasks["multiply"].outputs["result"])
```

https://github.com/aiidateam/aiida-workgraph

Dr. Xing Wang

AiiDA improvements

- verdi storage backup <destination>
  - incremental backup of the AiiDA profile
  - supports remote paths directly

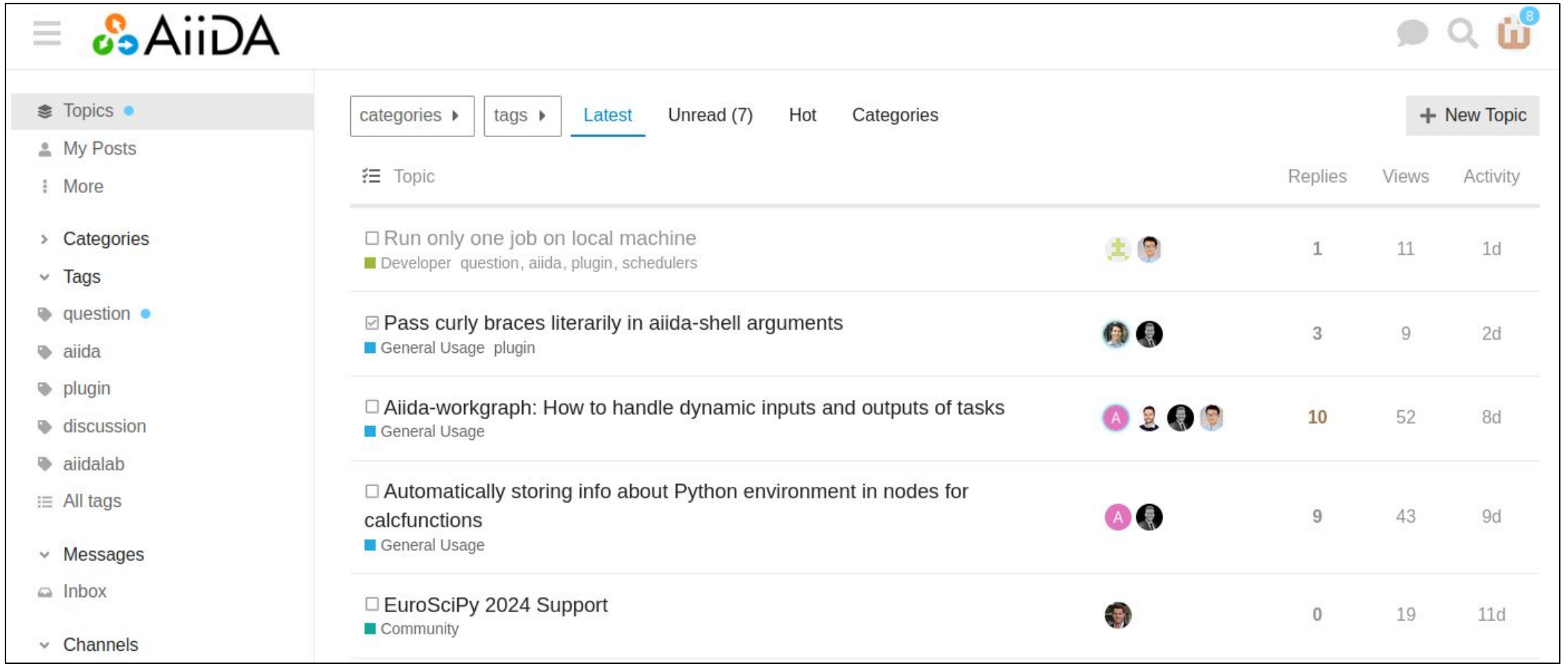
- verdi process dump <uuid/pk>
  - dumps all input and output scripts of the selected aiida process into a human-readable folder structure

```
➤ verdi process dump 7727b19f-5c77-4d2a-bbb0-71a8cf87b4b7
Success: Raw files for WorkChainNode <485293> dumped into
➤ cd dump-PwBaseWorkChain-485293/
➤ tree
.
├── 01-create_kpoints_from_distance
│   └── inputs
│       └── source_file
├── 02-iteration_01-PwCalculation
│   ├── inputs
│   │   ├── _aiidasubmit.sh
│   │   └── aiida.in
│   ├── node_inputs
│   │   └── pseudos
│   │       ├── Ge
│   │       │   └── ge_pbesol_v1.4.uspp.F.UPF
│   │       └── P
│   │           └── P.pbesol-n-rrkjus_psl.1.0.0.UPF
│   └── outputs
│       ├── _scheduler-stderr.txt
│       ├── _scheduler-stdout.txt
│       ├── aiida.out
│       └── data-file-schema.xml
└── README.md

9 directories, 10 files
```

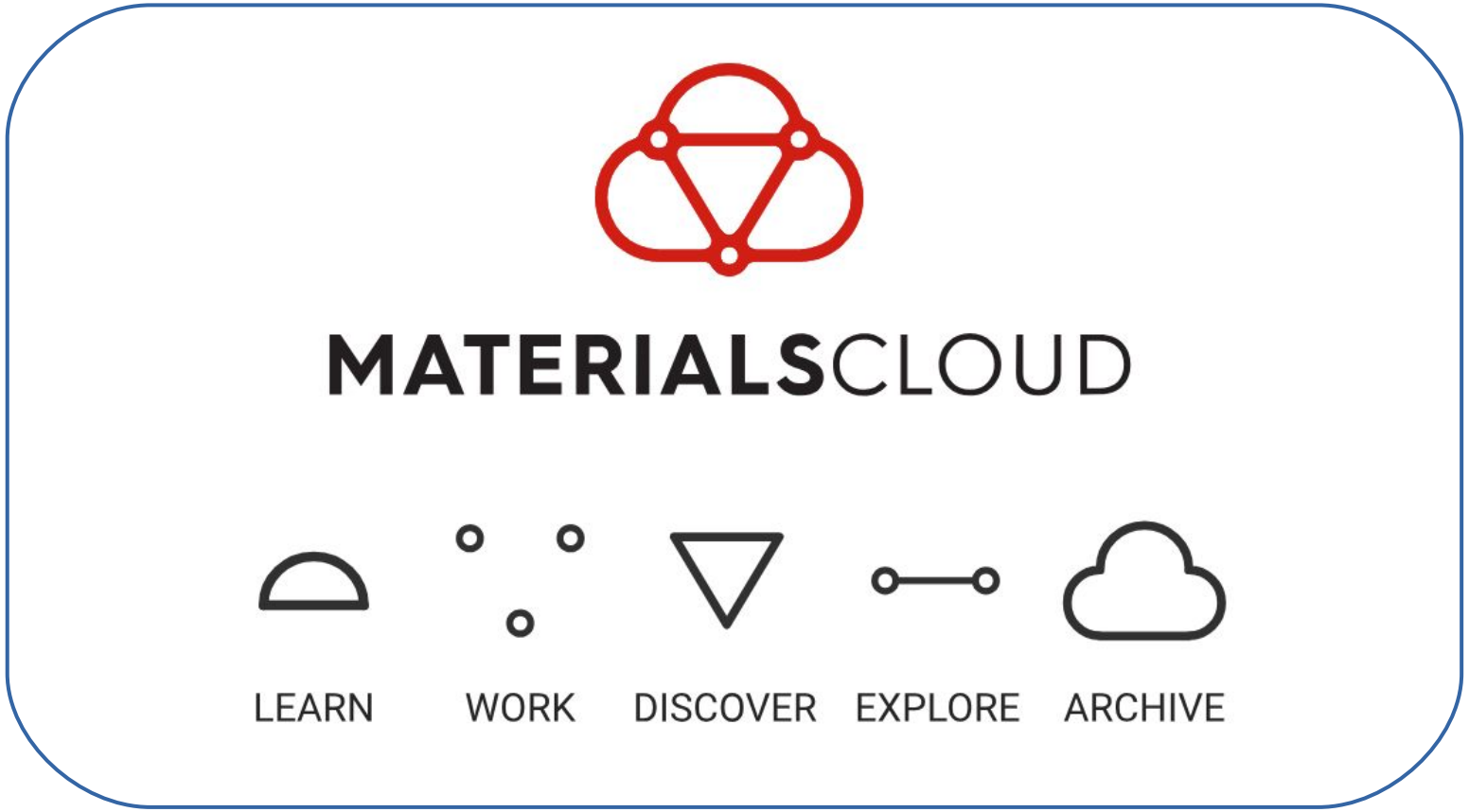
AiiDA discourse forum

https://aiida.discourse.group/



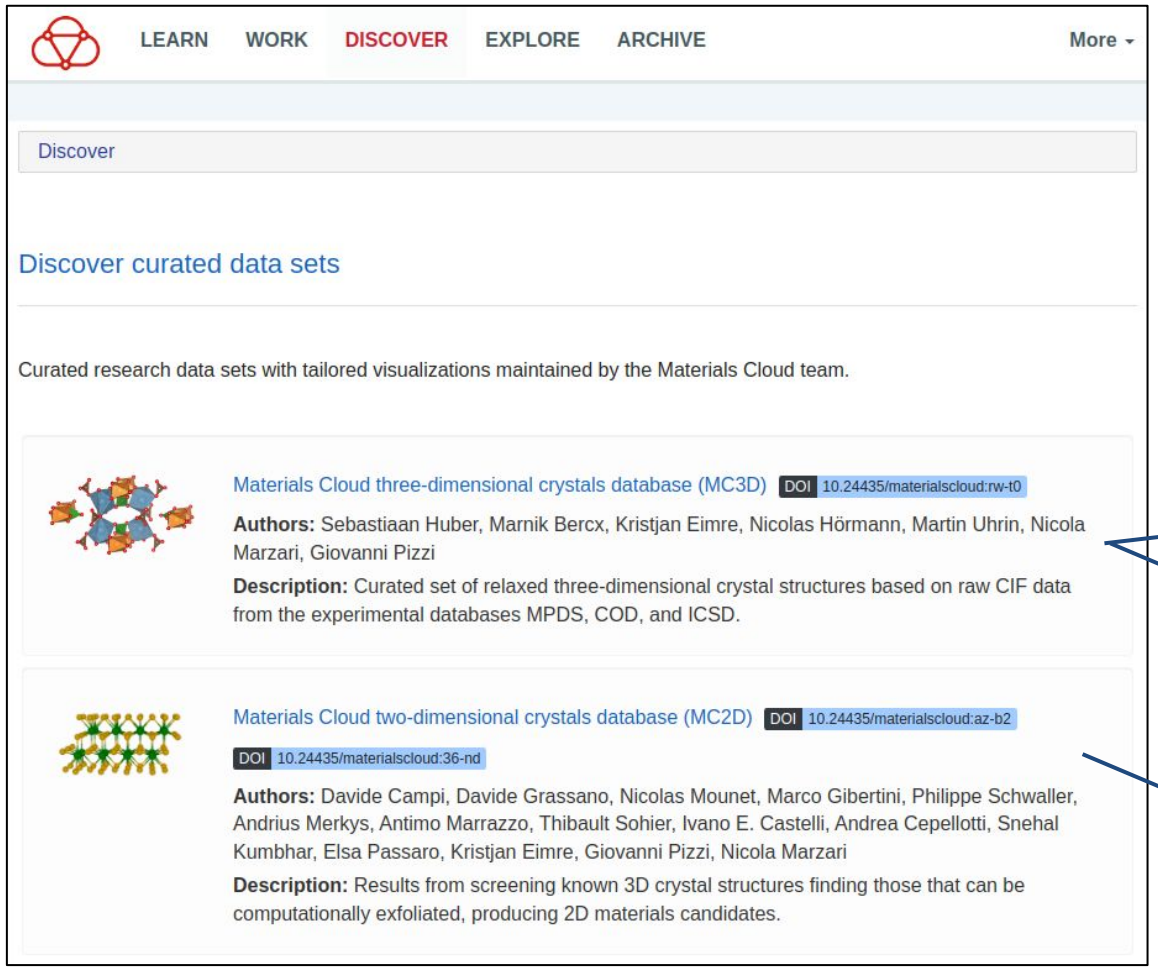
Materials Cloud

https://www.materialscloud.org/

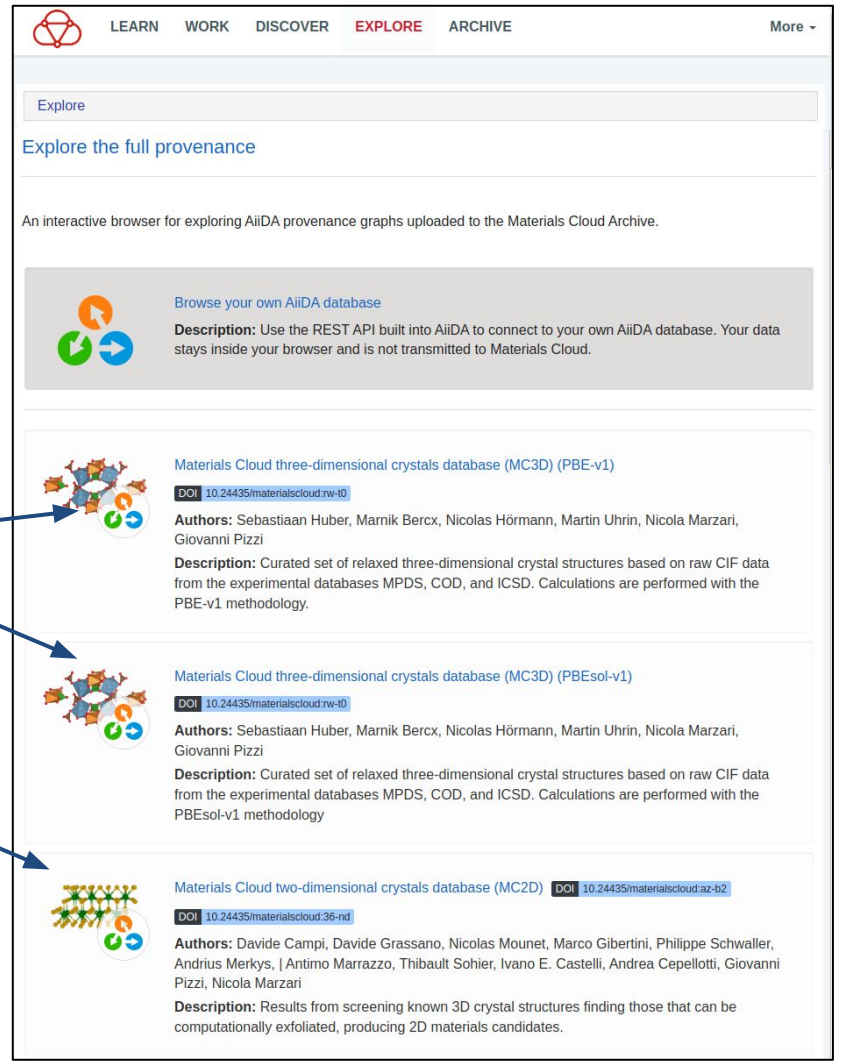


Discover and explore

Discover: curated web interface



Explore: visualize AiiDA provenance



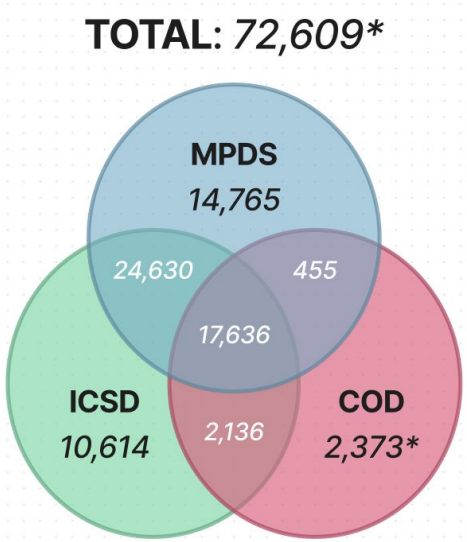
Materials Cloud 3D crystals database

MPDS ICSD FIZ Karlsruhe COD

901,210
Experimental structures

AiiDA

- Filter non-stochiometric, organic, ...
- Check uniqueness



MC3D-source (e.g. mc3d-123)

AiiDA QUANTUMESPRESSO

MC3D/PBE-v1

Relaxation with `PBE-v1` methodology.

34,487 relaxed structures

MC3D/PBEsol-v1

Relaxation with `PBEsol-v1` methodology.

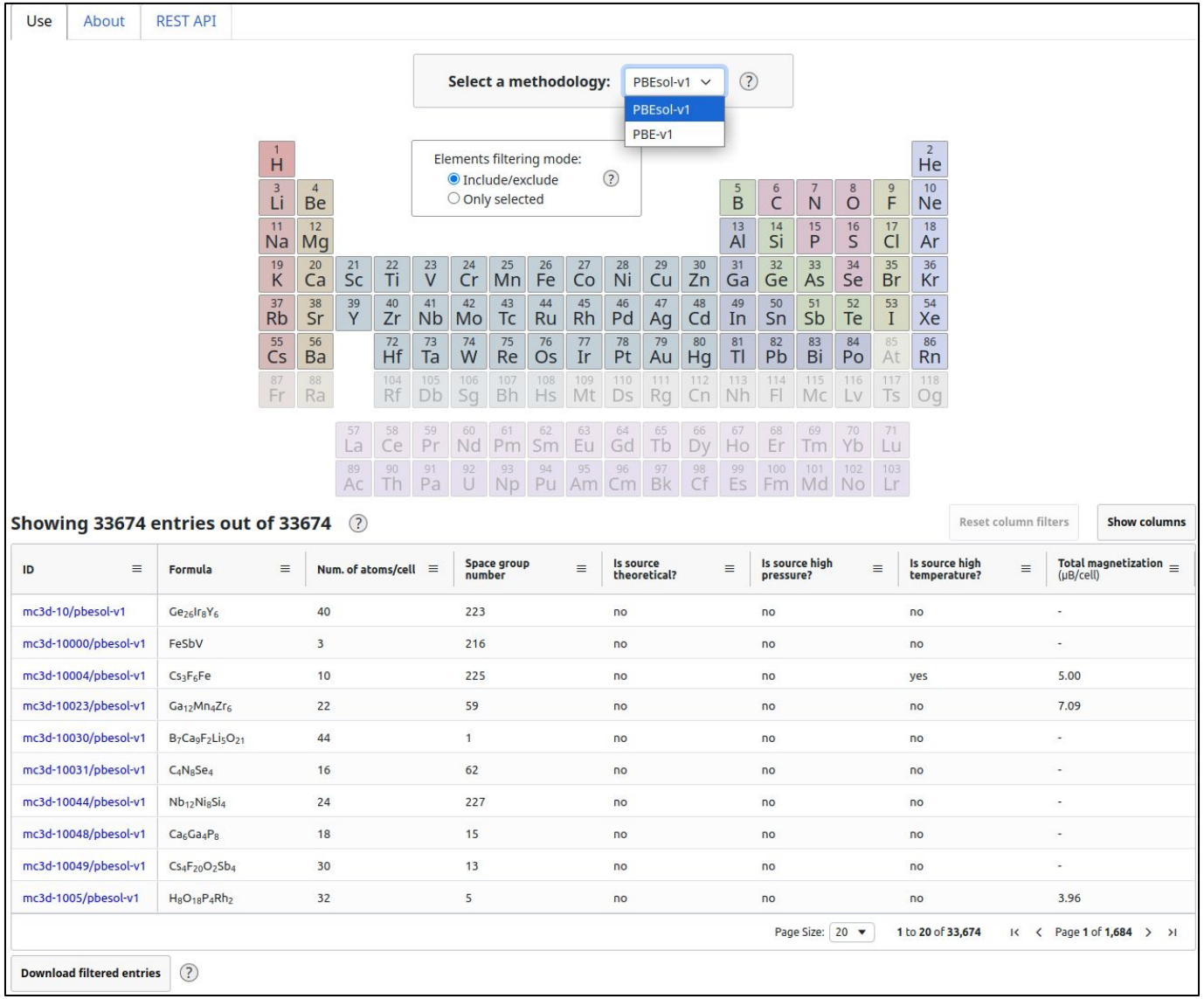
33,674 relaxed structures

Properties/screening:

- thermodyn stability
- spectroscopies
- electrides
- superconductors

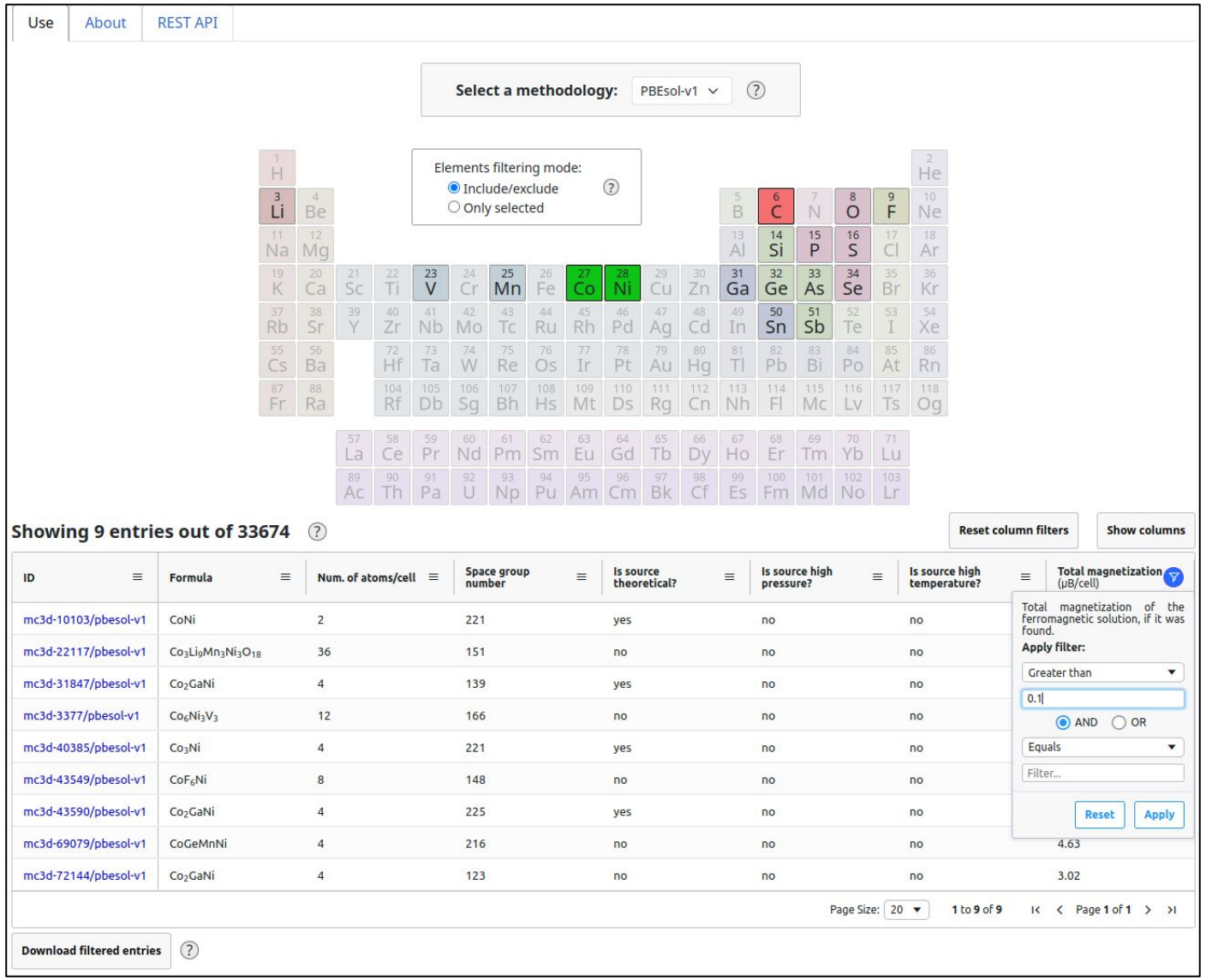
Materials Cloud 3D crystals database

https://mc3d.materialscloud.org/



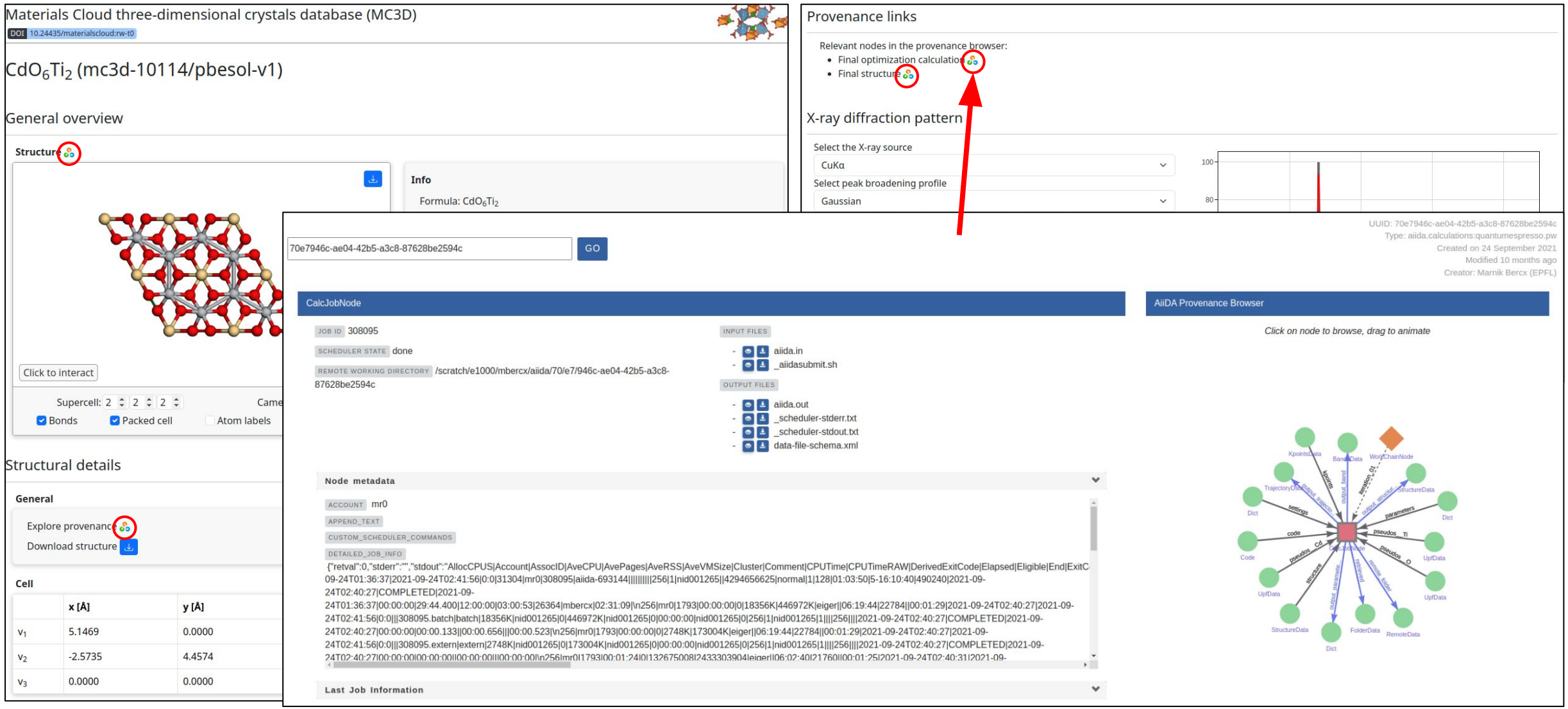
Materials Cloud 3D crystals database

https://mc3d.materialscloud.org/



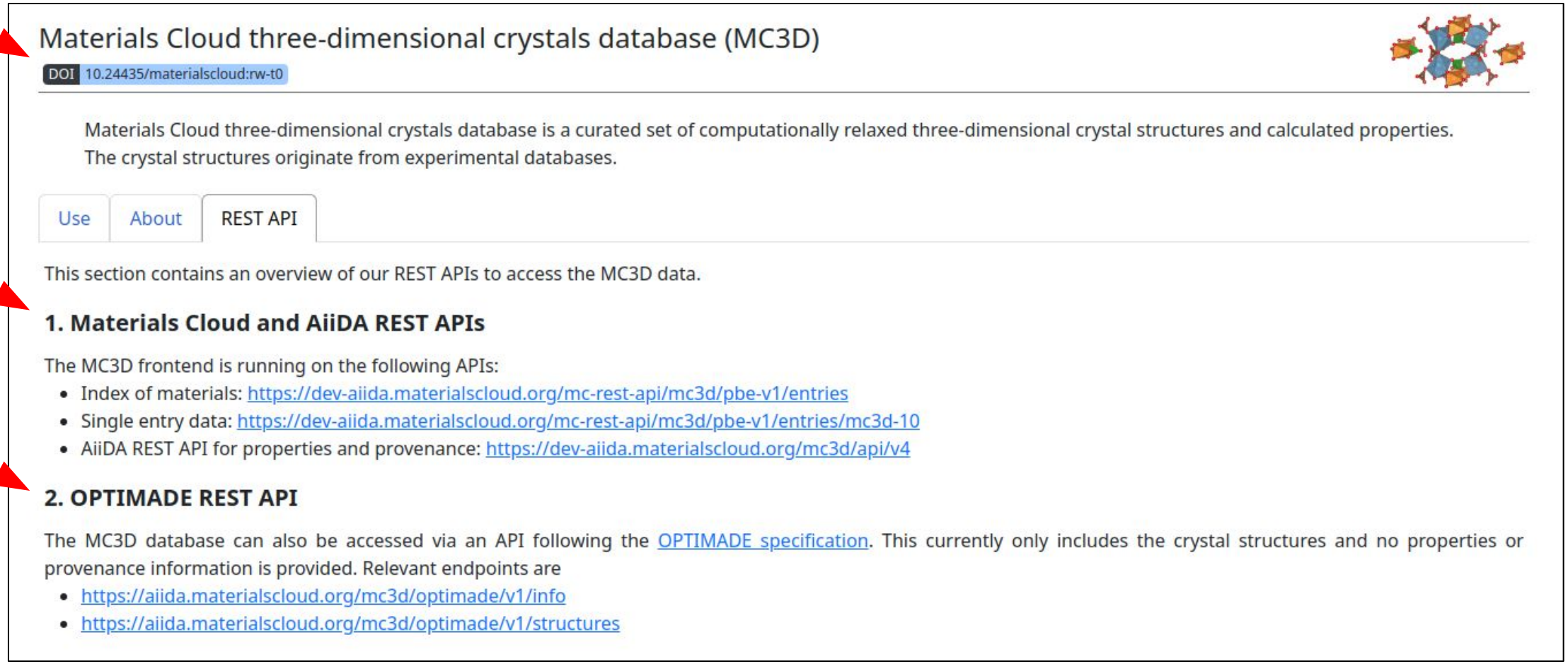
Materials Cloud 3D crystals database

https://mc3d.materialscloud.org/



Materials Cloud 3D crystals database

Accessing data directly



Materials Cloud three-dimensional crystals database (MC3D)

DOI 10.24435/materialscloud:rw-t0

Materials Cloud three-dimensional crystals database is a curated set of computationally relaxed three-dimensional crystal structures and calculated properties. The crystal structures originate from experimental databases.

Use | About | REST API

This section contains an overview of our REST APIs to access the MC3D data.

1. Materials Cloud and AiiDA REST APIs

The MC3D frontend is running on the following APIs:

- Index of materials: https://dev-aiida.materialscloud.org/mc-rest-api/mc3d/pbe-v1/entries
- Single entry data: https://dev-aiida.materialscloud.org/mc-rest-api/mc3d/pbe-v1/entries/mc3d-10
- AiiDA REST API for properties and provenance: https://dev-aiida.materialscloud.org/mc3d/api/v4

2. OPTIMADE REST API

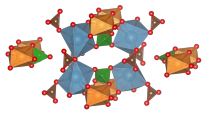
The MC3D database can also be accessed via an API following the OPTIMADE specification. This currently only includes the crystal structures and no properties or provenance information is provided. Relevant endpoints are

- https://aiida.materialscloud.org/mc3d/optimade/v1/info
- https://aiida.materialscloud.org/mc3d/optimade/v1/structures

Archive-OPTIMADE service

Established databases with dedicated infrastructure:

- Materials Cloud 3D crystals database
- Crystallography Open Database
- …



Raw data uploaded to **Materials Cloud Archive**

OPTIMADE API

OPTIMADE clients:

- GUI to browse and discover materials
- import to other applications

Direct programmatic access:

- high throughput;
- machine learning.

In collaboration with
Matthew Evans,
Gian Marco Rignanese, UCLouvain

Archive-OPTIMADE service

https://archive.materialscloud.org

Machine learning-accelerated discovery of A_2BC_2 ternary electrides with diverse anionic electron densities

Zhiqi Wang[1], Yutong Gong[1], Matthew L. Evans[2*], Yujing Yan[1], Shiyao Wang[1], Nanxi Miao[1], Ruiheng Zheng[1], Gian-Marco Rignanese[1,2*], Junjie Wang[1*]

1 State Key Laboratory of Solidification Processing, School of Materials Science and Engineering, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, People's Republic of China.

2 IMCN-MODL, Université catholique de Louvain, Chemin des Étoiles, 8, B-1348 Louvain-la-Neuve, Belgium.

DOI 10.24435/materialscloud:c8-gy [version v1]

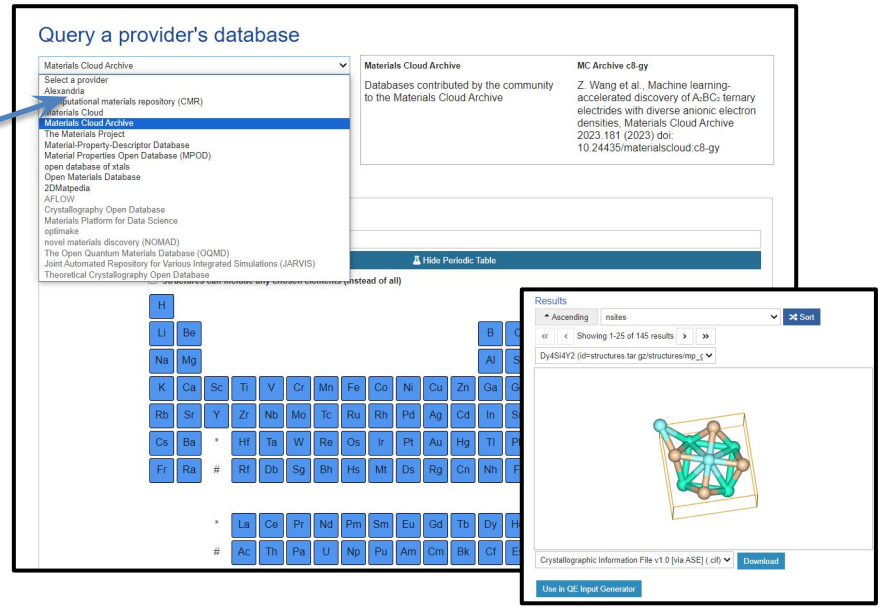
Publication date: Nov 28, 2023

Files

File name	Size	Description
mp_comparison.json.gz MD5	49.0 KiB	A a list of pymatgen `ComputedEntry` containing the results of atomate2 `MPGGARelax` calculations to enable direct comparison with the Materials Project's convex hull (as of 20/11/2023).
data.csv MD5	18.5 KiB	Computed ELF max and stability info for each structure
raw.tar.gz MD5	328.2 MiB	Additional raw data, including bandstructures and phonon dispersion curves for every structure.
structures.tar.gz MD5	40.4 KiB	CIFs of every structure considered, with the exact structures used for initial ELFCar calculations, as well as by the MP compatibility relaxation (in the corresponding sub-folders).
scripts.zip MD5	2.9 KiB	A directory containing Python scripts used for the re-relaxation and stability calculations of the structures, alongside a requirements file with the dependencies required for repeating them, as well as the script required to create this archive from the raw data.
README.txt MD5	1.0 KiB	README
optimade.yaml MD5	1.5 KiB	A config file for the MCloud/OPTIMADE integration that allows ingestion of the data into an OPTIMADE API

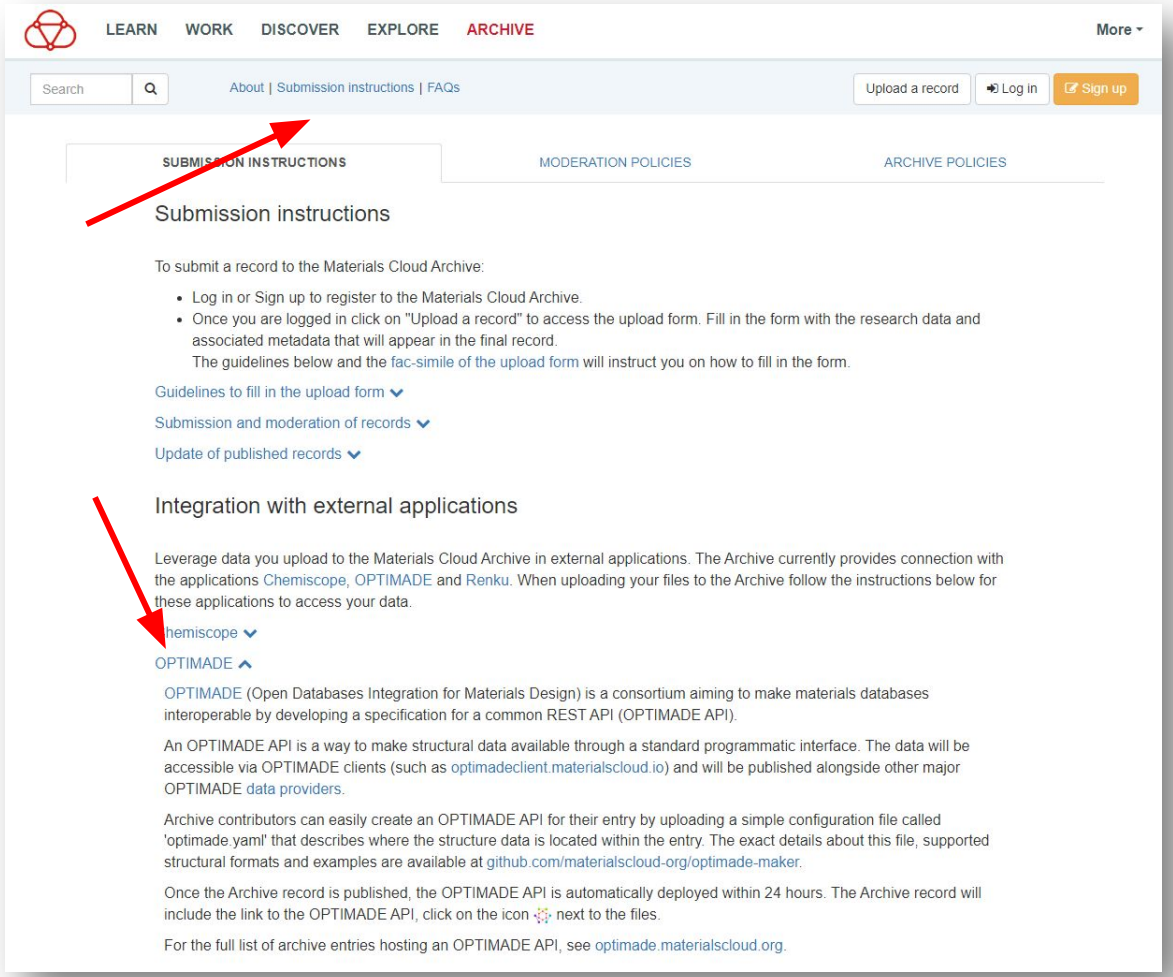
```yaml
entries:
  - entry_type: structures
    entry_paths:
      - file: structures.tar.gz
        matches:
          - structures/mp_gga/*.cif
    property_paths:
      - file: data.csv
    property_definitions:
    - name: elf_max
      title: Maximum value of the ELF
      description: The maximum value of the electron
      unit: dimensionless
      type: float
```

https://optimadeclient.materialscloud.io



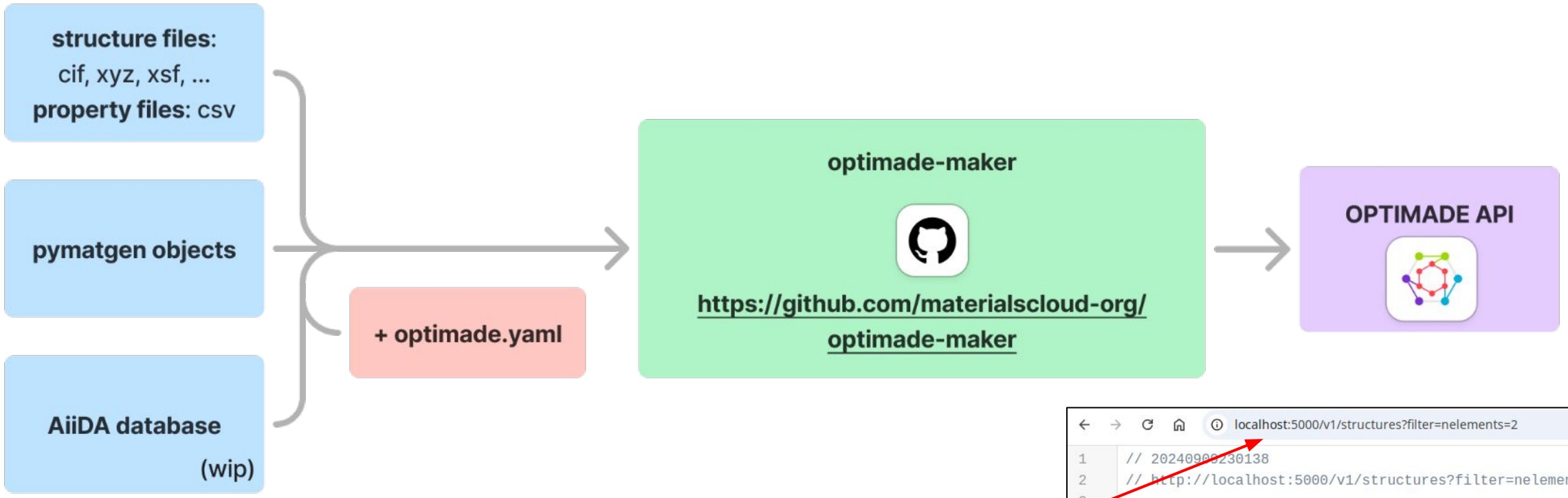
Archive-OPTIMADE service

https://archive.materialscloud.org/



https://optimade.materialscloud.org/

optimade-maker




```
➤ ls
data.csv  optimade.yaml  structures.zip
➤ optimake serve .
2024-09-09 22:50:24 - optimake - INFO - optimade.jsonl doesn't exist. Converting archive.
Parsing structures files: 100%|                                  | 2/2 [00:00<00:00, 77.36
Constructing OPTIMADE structures entries: 2it [00:00, 731.29it/s]
Parsing properties for structures entries: 100%|                 | 1/1 [00:00<00:00, 87.14
INFO:     Started server process [31378]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:5000 (Press CTRL+C to quit)
```

localhost:5000/v1/structures?filter=nelements=2

```
// 20240909230138
// http://localhost:5000/v1/structures?filter=nelements=2

{
  "data": [
    {
      "id": "set2/102",
      "type": "structures",
      "attributes": {
        "immutable_id": "structures.zip/cifs/set2/102.cif",
        "last_modified": "2024-09-09T22:50:24.651087",
        "elements": [
          "C",
          "Sr"
        ],
        "nelements": 2,
```

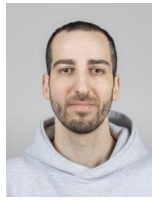
Acknowledgements

People involved in AiiDA, AiiDALab, and Materials Cloud

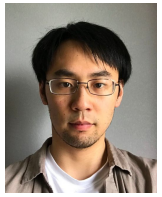

Giovanni Pizzi (PSI)


Marnik Bercx (PSI)


Miki Bonacci (PSI)


Edan Bainglass (PSI)


Xing Wang (PSI)


Jusong Yu (PSI)


Ali Khosravi (PSI)


Alexander Goscinski (PSI)


Michail Minotakis (PSI)


Yuhao Jiang (PSI)

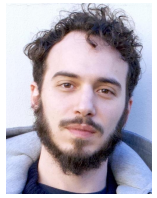

Timo Reents (PSI)

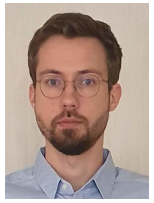

Nataliya Paulish (PSI)


Nicola Marzari (EPFL, PSI)


Sebastiaan P. Huber (Microsoft)


Chris Sewell (EPFL)


Francisco F. Ramirez (EPFL)


Kristjan Eimre (EPFL)


Daniel Hollas (U. Bristol)


Leopold Talirz (Schott)


Aliaksandr Yakutovich (Empa)


Carlo Pignedoli (Empa)


Valeria Granata (EPFL)


Joost VandeVondele (ETHZ,CSCS)


Thomas Schulthess (ETHZ,CSCS)

MARVEL
NATIONAL CENTRE OF COMPETENCE IN RESEARCH

FNSNF
SWISS NATIONAL SCIENCE FOUNDATION

EPFL

PAUL SCHERRER INSTITUT
PSI

swissuniversities

Empa
Materials Science and Technology

MAX
MATERIALS DESIGN AT THE EXASCALE
EUROPEAN CENTRE OF EXCELLENCE

MarketPlace

INTERSECT

EMMC

PRACE
PARTNERSHIP FOR ADVANCED COMPUTING IN EUROPE

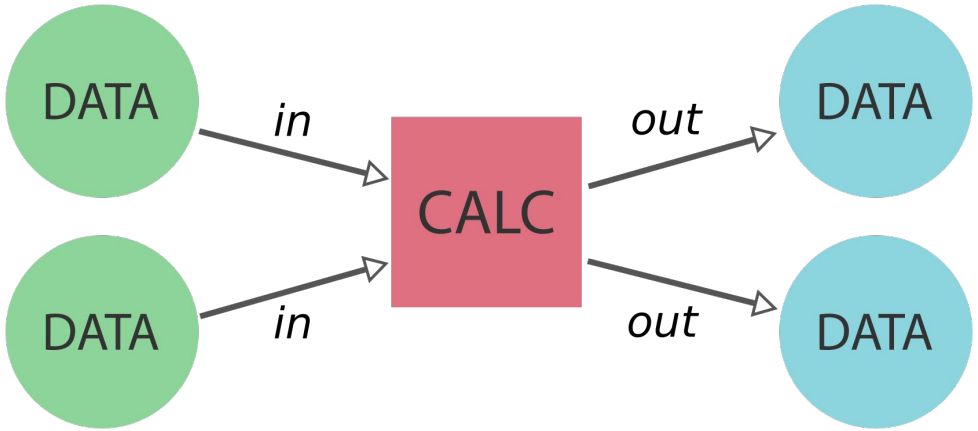
Summary

- Recent AiiDA improvements:
  - verdi presto - easy installation
  - aiida-shell - run executables without AiiDA plugins
  - WorkGraphs - easy combining of AiiDA processes
- Materials Cloud 3D crystals database
  - 'PBEsol-v1' subdatabase
  - Improved web interface
- Materials Cloud Archive - OPTIMADE integration
  - automatic OPTIMADE APIs for contributed raw data
  - optimade-maker

Data provenance

Simple recipe - store:

- data transformations/calculations
- inputs
- outputs
- inter-connections



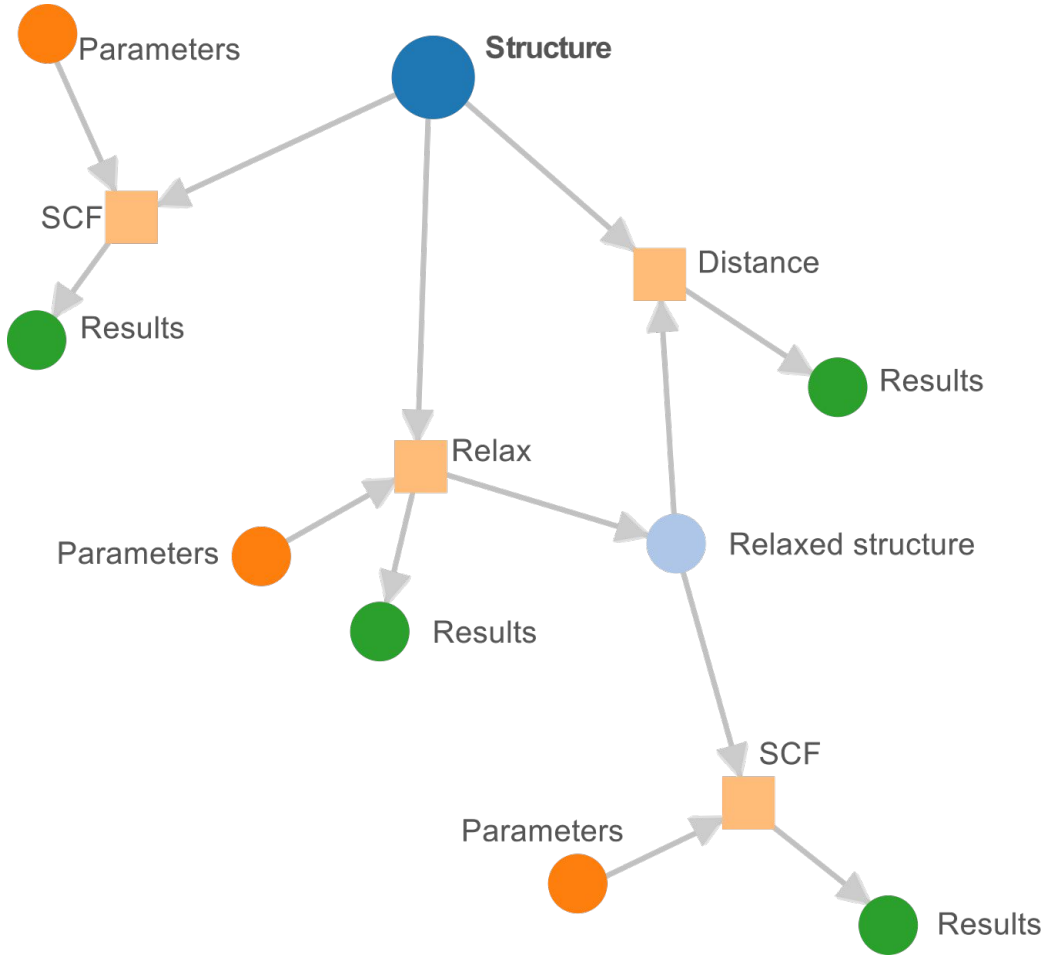
Data provenance

Simple recipe - store:

- data transformations/calculations
- inputs
- outputs
- inter-connections

Provenance graphs

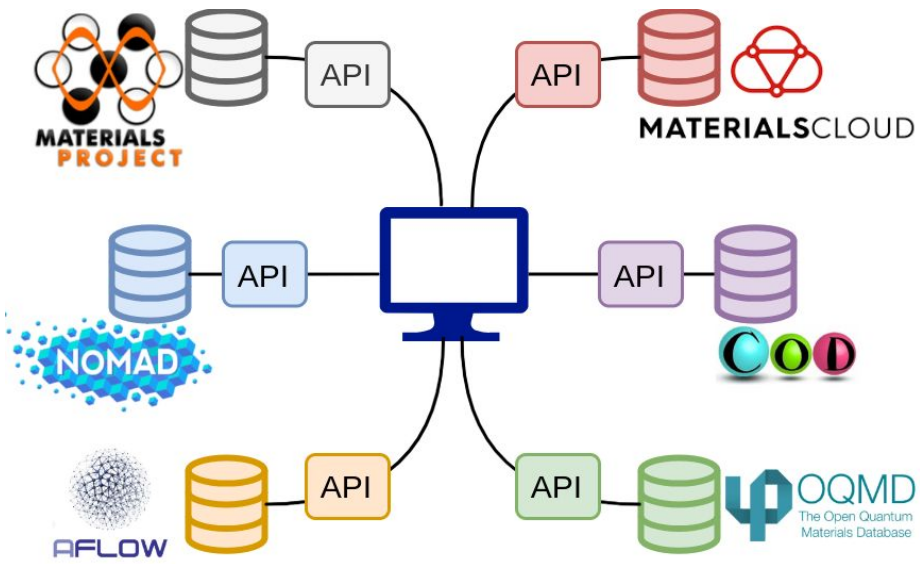
- when data gets reused, a directed graph is created
- that quickly grows in complexity

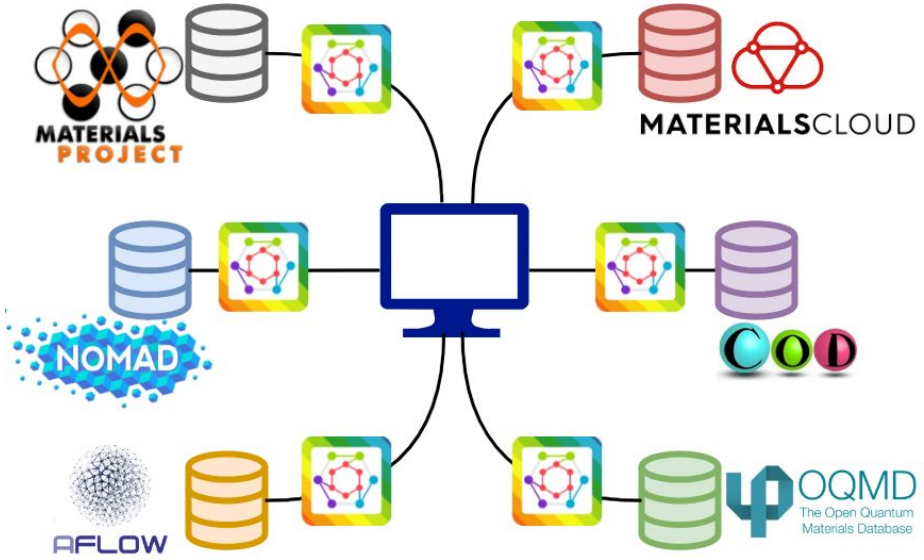


OPTIMADE

OPTIMADE
Open Databases Integration for Materials Design

Consortium to make materials databases interoperable via a **common REST API**, 15+ partners





Andersen et al, Sci. Data 8, 217 (2021)

Evans et al, arXiv 10.48550/arXiv.2402.0057 (2024)

OPTIMADE

OPTIMADE API

base URL specific to the DB

universal OPTIMADE query

https://aiida.materialscloud.org/mc3d/optimade/v1/structures?filter=elements HAS ALL "C", "Si"

Supports queries for/based on

- chemical composition;
- custom properties;
- references;
- & more, with new functionality constantly being added

machine-readable json

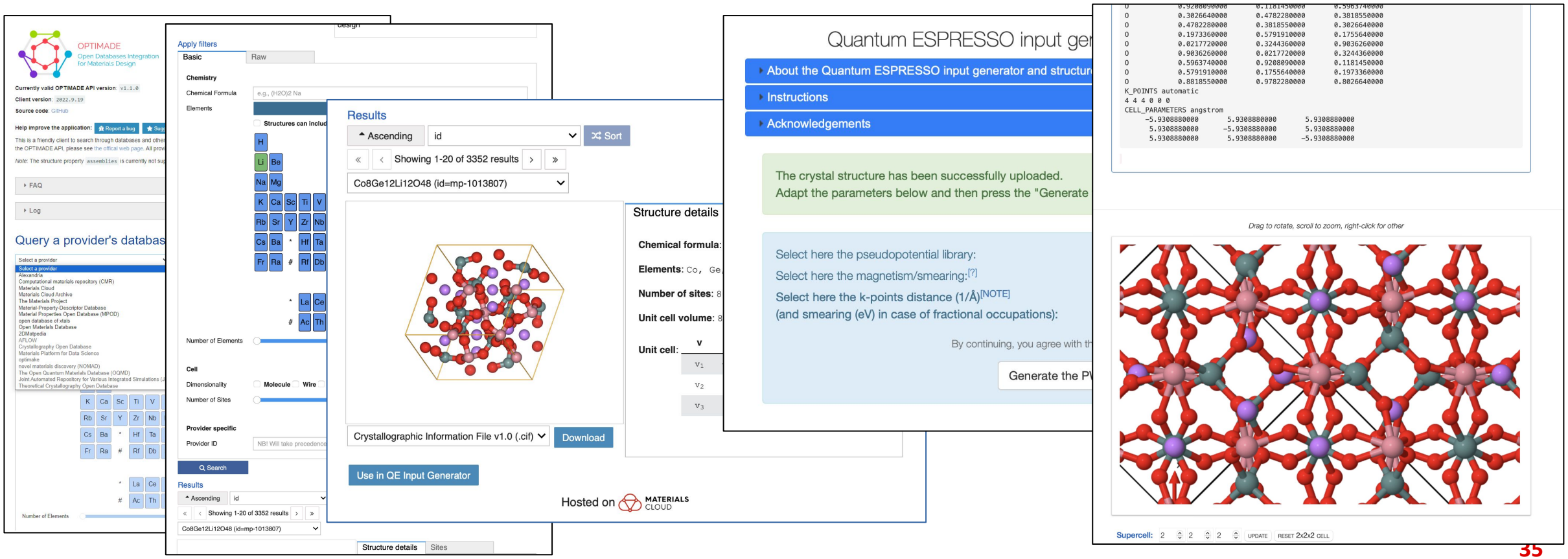

```
{
  "data": [
    {
      "id": "417",
      "type": "structures",
      "links": null,
      "meta": null,
      "attributes": {
        "immutable_id": "6dd18556-a5e6-442d-a0f2-da0ae5beb716",
        "last_modified": "2022-07-01T21:18:58Z",
        "elements": [
          "C",
          "Si"
        ],
        "nelements": 2,
        "elements_ratios": [ … ], // 2 items
        "chemical_formula_descriptive": "C9Si9",
        "chemical_formula_reduced": "CSi",
        "chemical_formula_hill": "C9Si9",
        "chemical_formula_anonymous": "AB",
```

OPTIMADE

OPTIMADE Clients

https://optimadeclient.materialscloud.io/

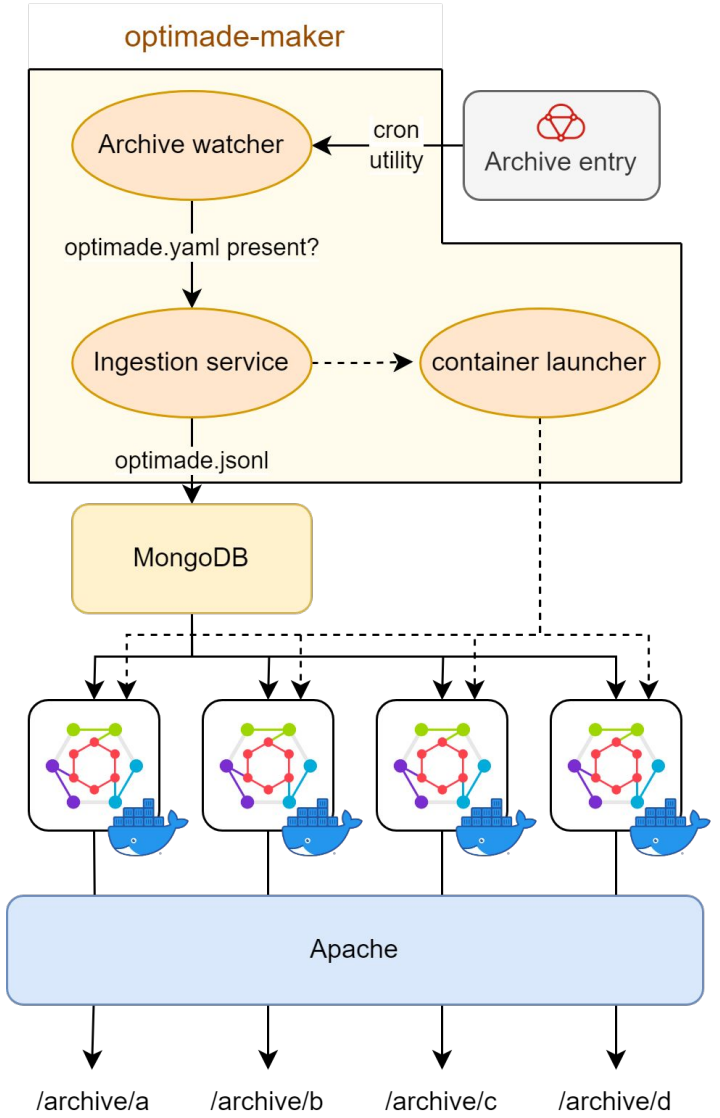
https://www.materialscloud.org/work/tools/qeinputgenerator



Archive-OPTIMADE service

Implementation

- Archive watcher detects optimade.yaml
- Pipeline to populate a MongoDB
- Docker containers for each API
  - OPTIMADE Python tools
- Apache as reverse proxy.
- Tools open source: optimade-maker
  - https://github.com/materialscloud-org/optimade-maker



Materials Cloud OPTIMADE servers

https://www.materialscloud.org/optimade