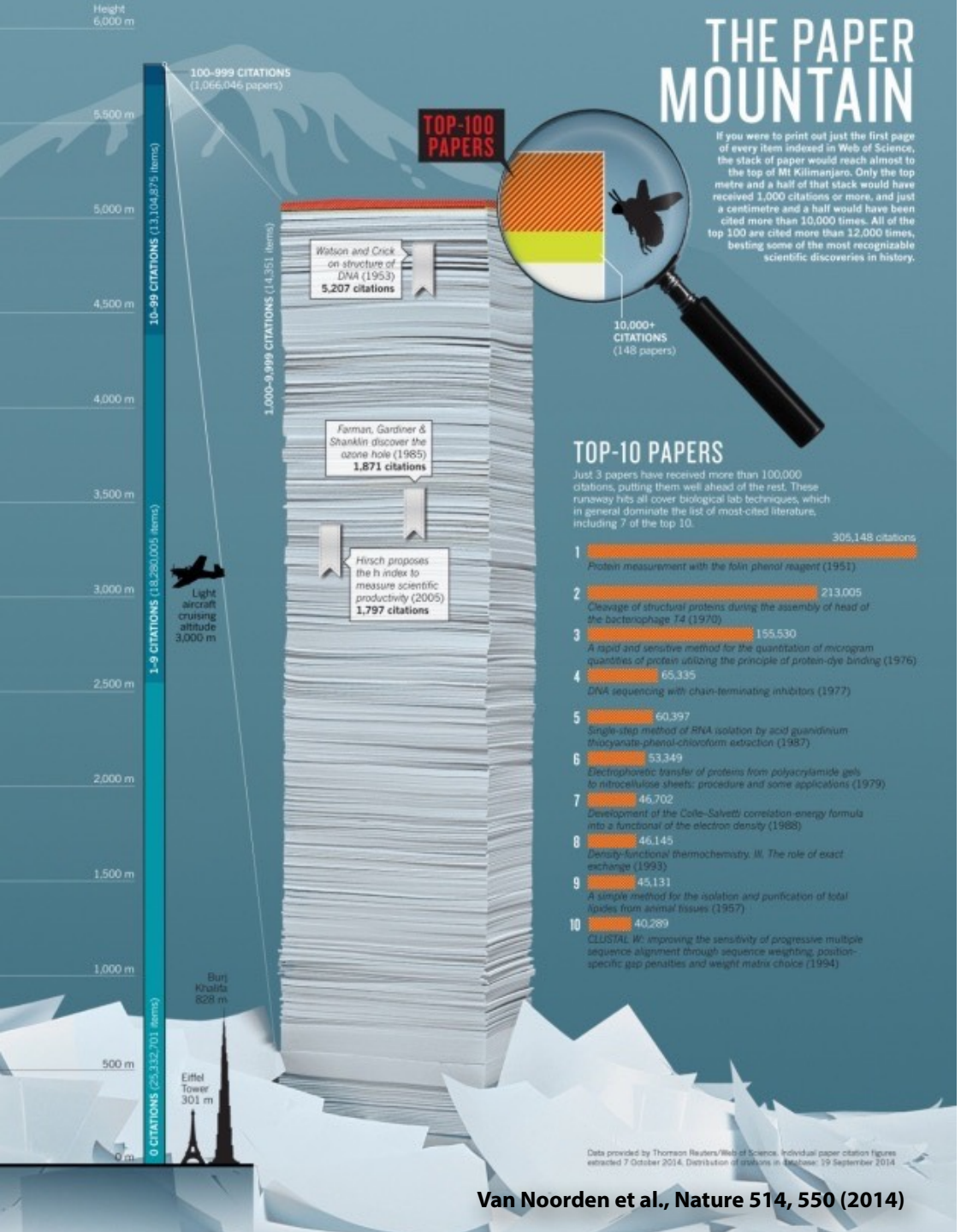
Open Science Platform for Materials Science: AiiDA and the Materials Cloud

Giovanni Pizzi (EPFL)

FNSNF
MARVEL
MAX DRIVING THE EXASCALE TRANSITION
swissuniversities
EPFL ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

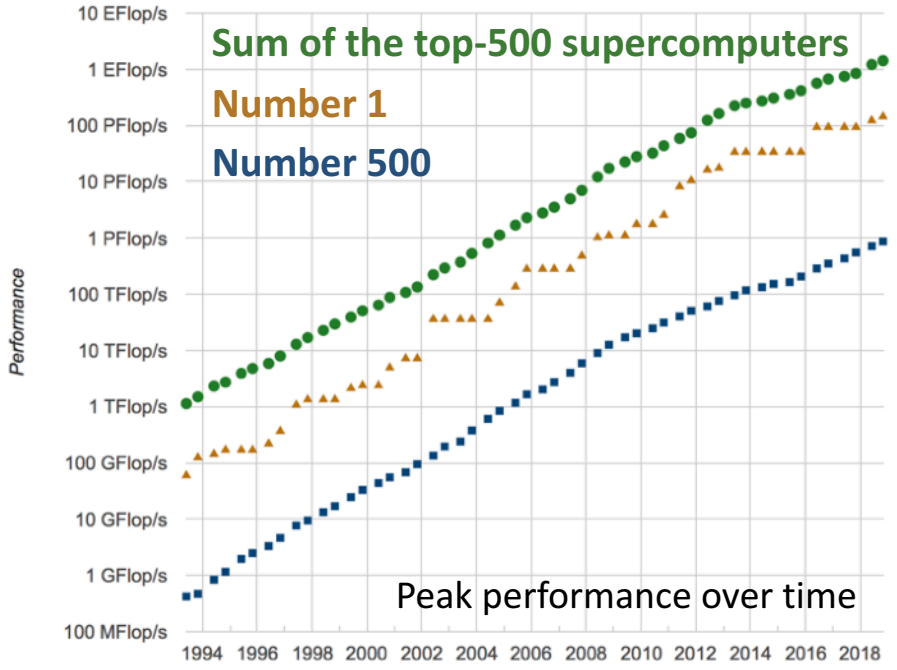


Van Noorden et al., Nature 514, 550 (2014)

Our research: Materials simulations

Nature (2014): 12 papers on **Density Functional Theory** among the top-100 most cited papers in the *entire scientific literature*

Accuracy and predictive power of quantum engines



150,000x increase in the past 20 years

1 month (1998)

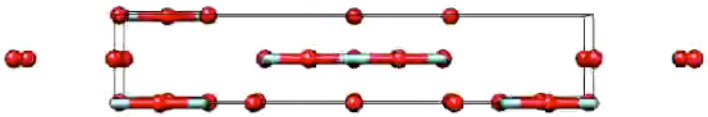
↓

10 seconds (2018)

Result: materials design and discovery via

high-throughput computations

Leverage supercomputers to compute and predict materials' properties



Aim: Compute properties for all of them
(and even new, invented ones)
and **discover novel functional materials**

Open Science Platform: definition

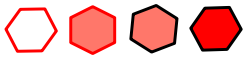
- Our definition of an Open Science Platform [1]:
  - **Open simulation codes**
  - **Open architecture** to manage simulations and **open workflows**
  - Support for **Open Data**, **Data Management Plans** and FAIR-compliant sharing
  - **Straightforward availability** of the tools, with **curated open-data services** enabling turn-key workflows (pseudopotential libraries, …)

[1] Pizzi G. (2018) *Open-Science Platform for Computational Materials Science: AiiDA and the Materials Cloud.* In: Andreoni W., Yip S. (eds), Handbook of Materials Modeling (Springer, Cham).

Our goal

Build an open-science infrastructure
with computational services offered
to scientific, industrial community and beyond

Like a synchrotron, but for
open and reproducible simulations



Our two core infrastructures

AiiDA as the "operating system" to manage, automate and store simulations and their results

and

Materials Cloud as the open-science dissemination portal and cloud simulation platform

OPEN SCIENCE PLATFORM:

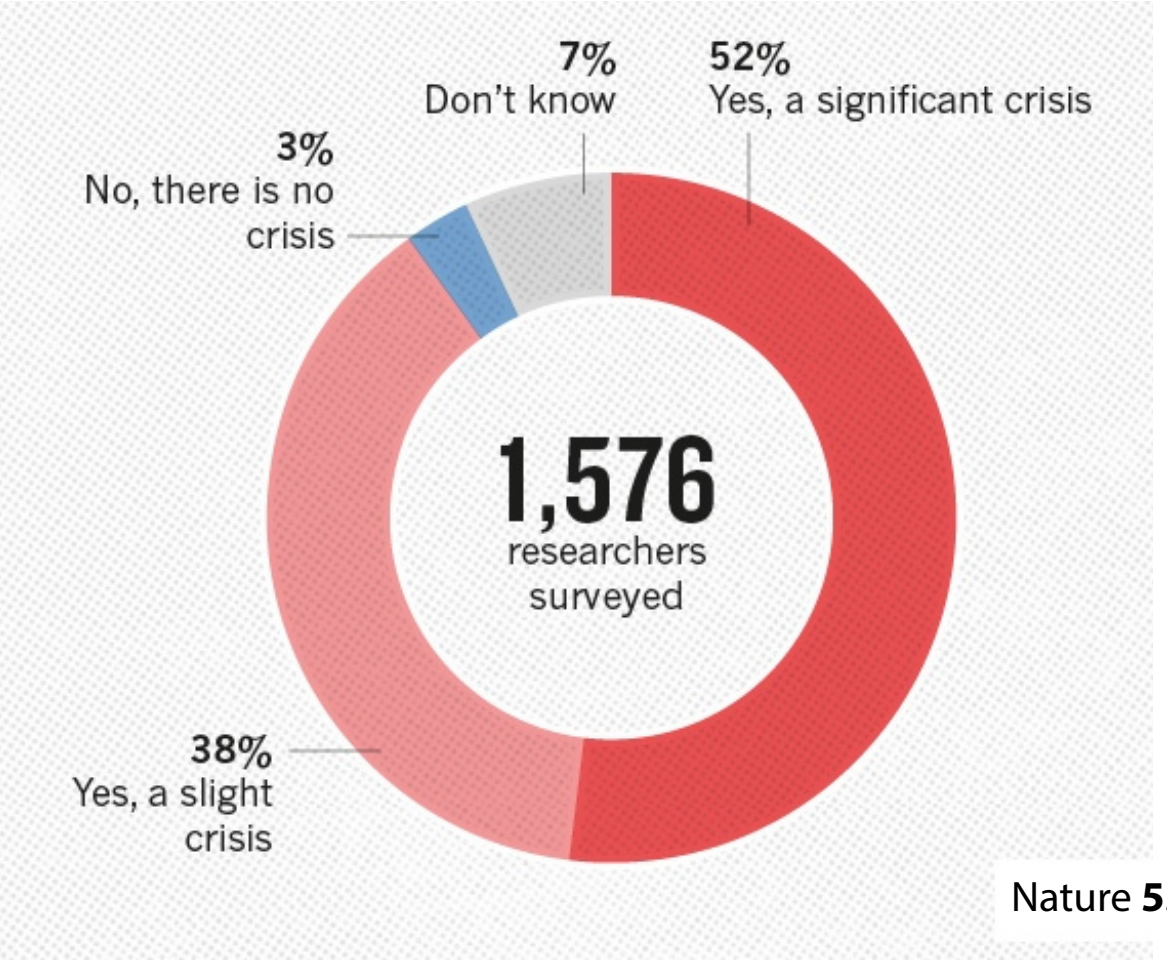
G. Pizzi et al., Comp. Mat. Sci. 111, 218-230 (2016)

How to manage simulations and their provenance?

We need a **tool** to help us
organise research and store provenance

Reproducibility: a cornerstone of the scientific method

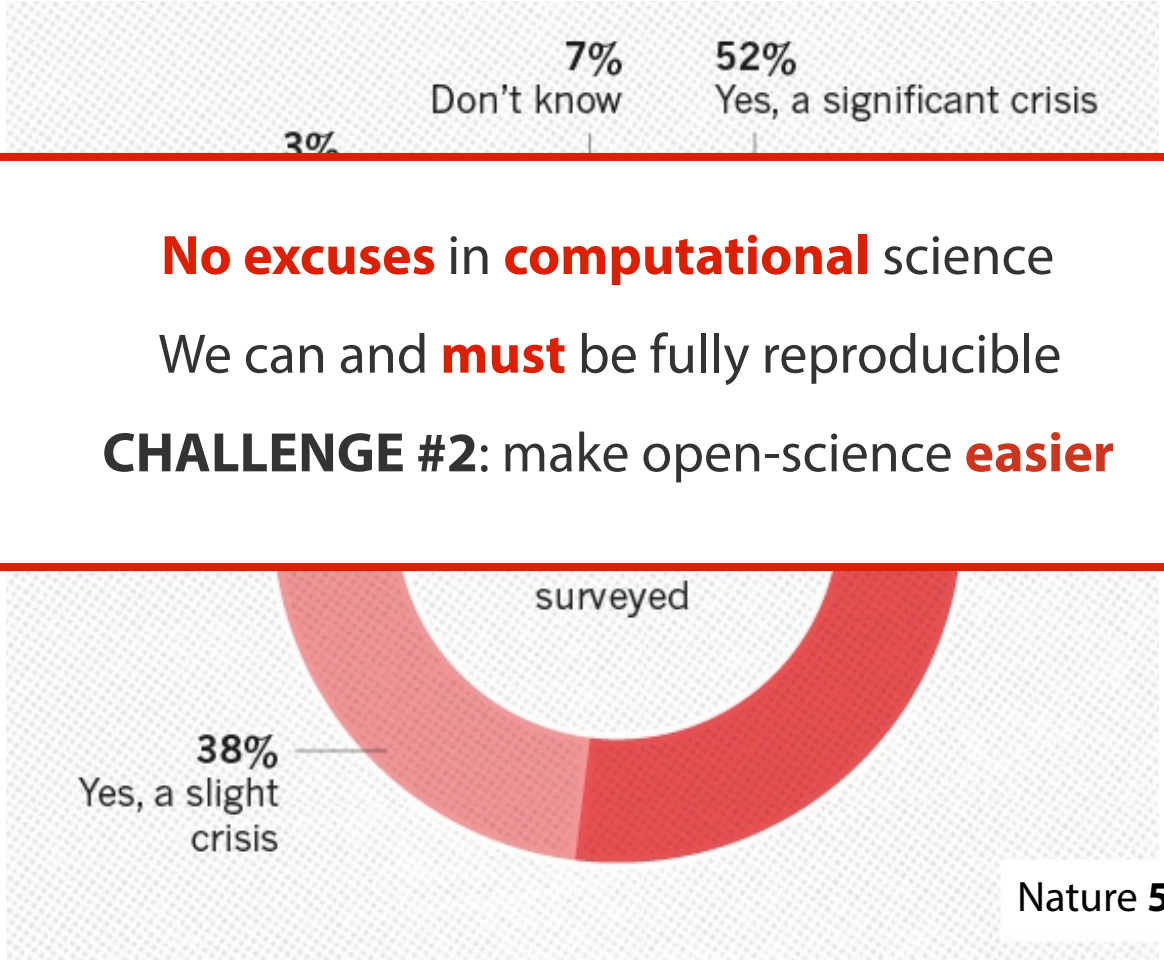
IS THERE A REPRODUCIBILITY CRISIS?



Nature **533**, 452–454 (2016)

Reproducibility: a cornerstone of the scientific method

IS THERE A REPRODUCIBILITY CRISIS?



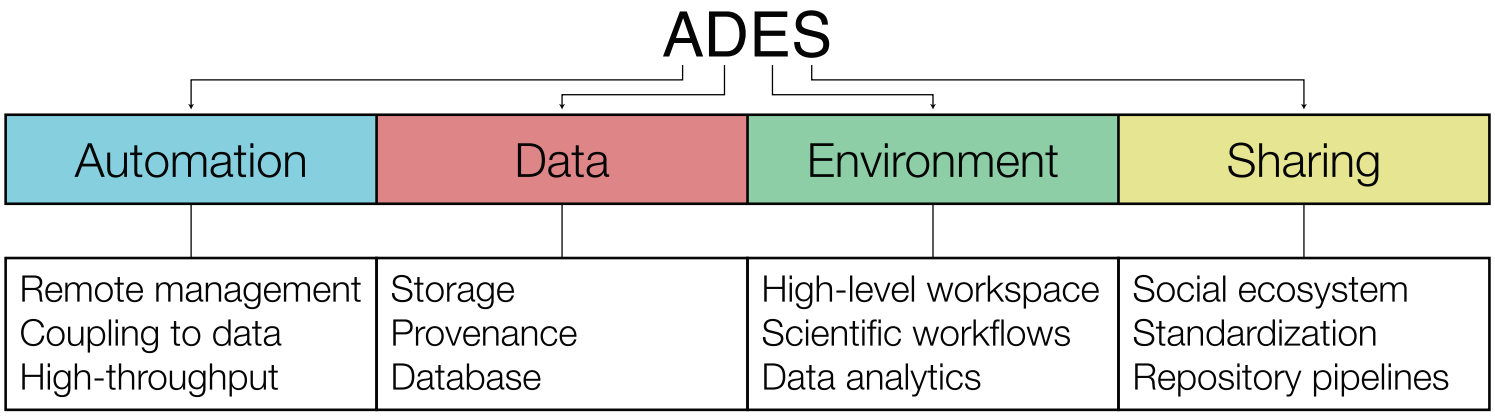
No excuses in **computational** science

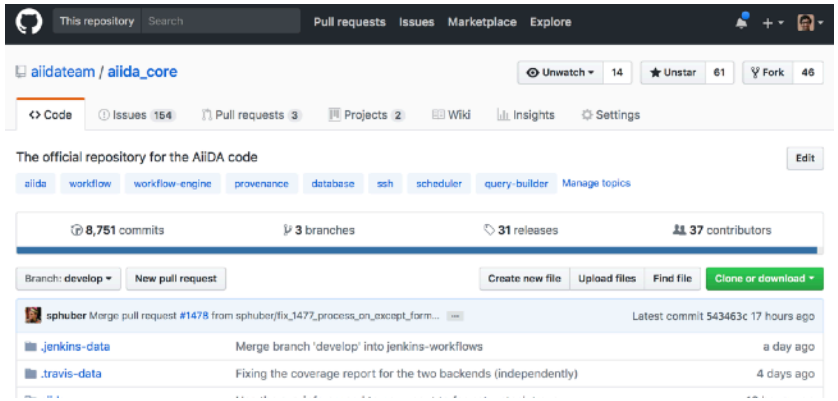
We can and **must** be fully reproducible

CHALLENGE #2: make open-science **easier**

Nature **533**, 452–454 (2016)

AiiDA development timeline

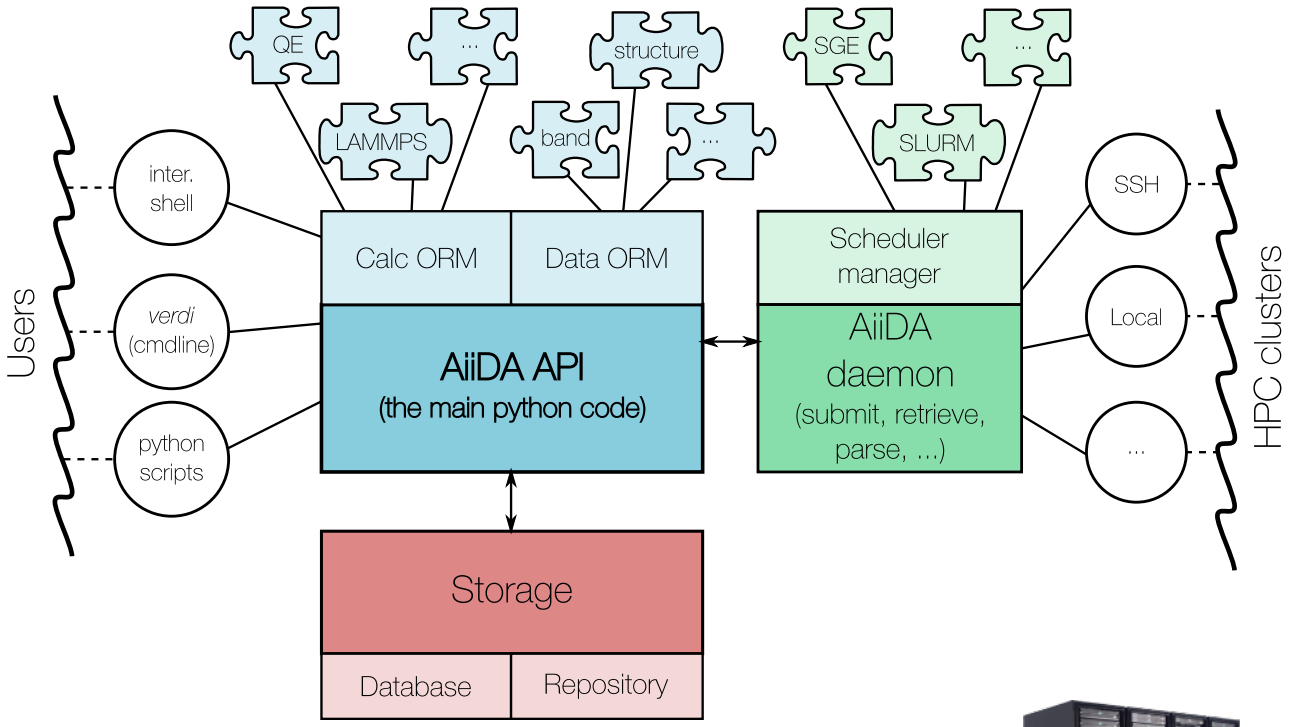




G. Pizzi et al., Comp. Mat. Sci. 111, 218-230 (2016)

http://www.aiida.net

AiiDA



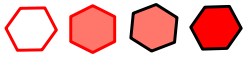
AiiDA

Main features

- Python (2.7 & 3.6) infrastructure
- SQL database backend, access via a Python ORM
- Local connection to clusters, or via ssh using a python API
- Interface to various job schedulers (SGE, Torque, LSF, PBS Pro, SLURM, ...)
- Event-based daemon with remote management and workflow execution manager
- REST APIs using Flask to expose one own's data
- Plugin management system and extended code support
- Easy sharing of the results with other users in the community

AiiDA submission

```
code = Code.get('pw-6.3@daint-mr25')
builder = code.new_builder()

builder.metadata.options = {
    ‘max_wallclock_seconds’: 600,
    ‘max_wallclock_seconds’: {"num_machines": 2}}

Structure = DataFactory('structure')
structure = Structure(ase = read('TiO2.cif'))

Dict = DataFactory('dict')
parameters = Dict(dict={
    'CONTROL': {
        'calculation': 'scf',
        'restart_mode': 'from_scratch'},
    'SYSTEM': {‘ecutwfc': 40.}})

Kpoints = DataFactory('array.kpoints')
kpoints = Kpoints(kpoints_mesh = [4,4,4])

builder.structure = structure
builder.parameters = parameters
builder.kpoints = kpoints
builder.pseudos = get_pseudos_from_family(structure,
’SSSP_efficiency_v1.0')

aiida.engine.submit(builder)
```



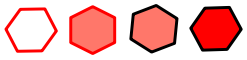
AiiDA submission

```
code = Code.get('pw-6.3@daint-mr25')
builder = code.new_builder()

builder.metadata.options = {
    'max_wallclock_seconds': 600,
    'max_wallclock_seconds': {"num_machines": 2}}

Structure = DataFactory('structure')
structure = Structure(ase = read('TiO2.cif'))

Dict = DataFactory('dict')
parameters = Dict(dict={
    'CONTROL': {
        'calculation': 'scf',
        'restart_mode': 'from_scratch'},
    'SYSTEM': {'ecutwfc': 40.}})

Kpoints = DataFactory('array.kpoints')
kpoints = Kpoints(kpoints_mesh = [4,4,4])

builder.structure = structure
builder.parameters = parameters
builder.kpoints = kpoints
builder.pseudos = get_pseudos_from_family(structure,
'SSSP_efficiency_v1.0')

aiida.engine.submit(builder)
```

Switch computers in one line
supports different schedulers, version of codes, …

Define (only) necessary inputs
Interface designed by plugin

Inputs stored in the DB, and handing over to the daemon

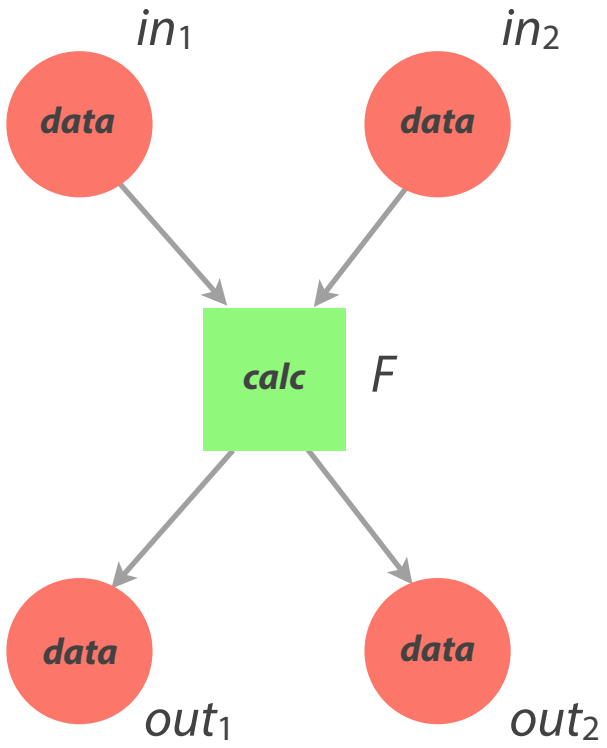


Storage and provenance

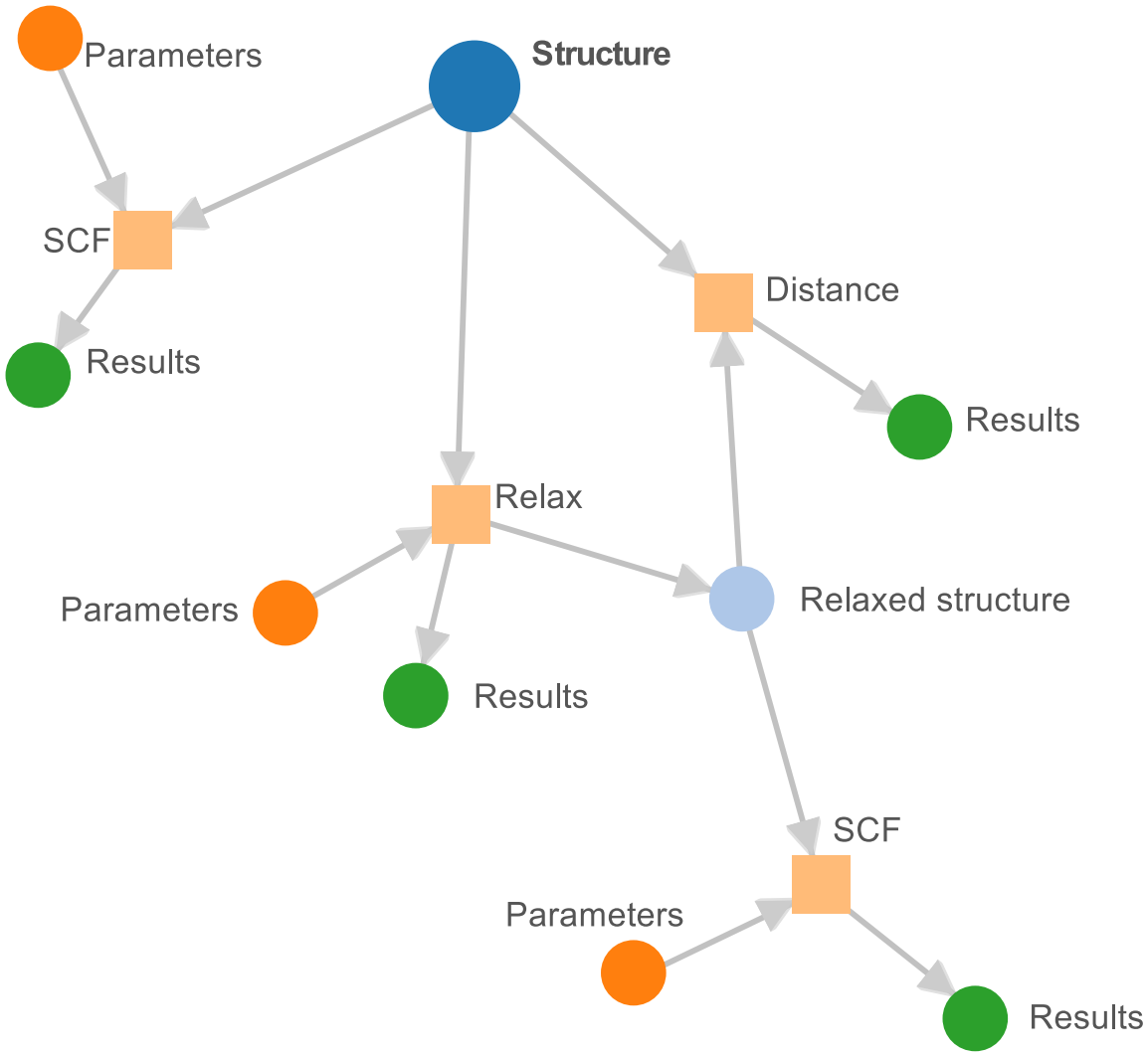
- *Calculated properties*: result of complex, connected calculations
- How do we store simulations **preserving the connected structure?**
- Inspiration from the *open provenance model*
- **Any calculation: a function,** converting inputs to outputs:

$$\mathbf{out_1, out_2 = F(in_1, in_2)}$$

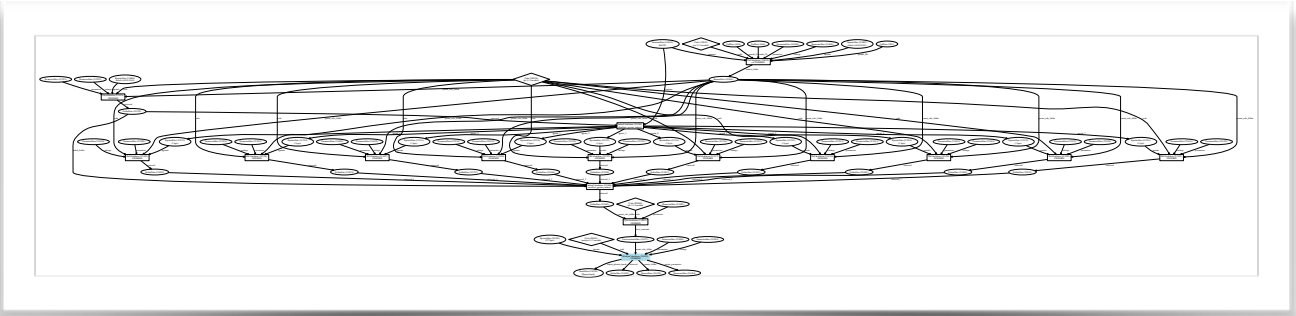
- **Each object is a node in a graph,** connected by directional labeled links
- Output nodes can be used as inputs



Data provenance: Directed Acyclic Graphs

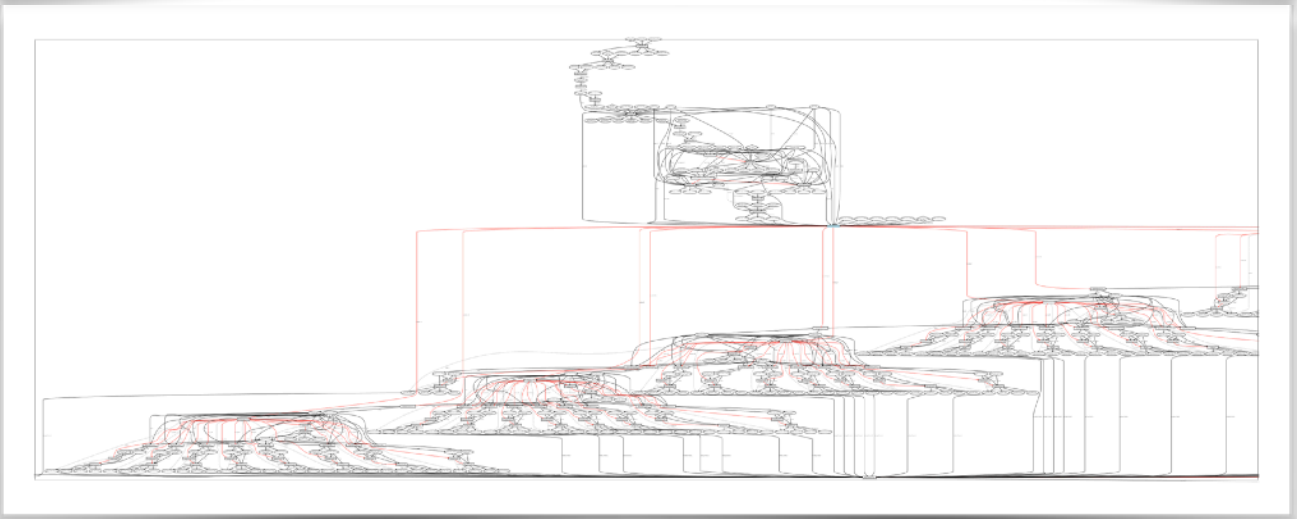


“Simple” graphs of workflows for a single material



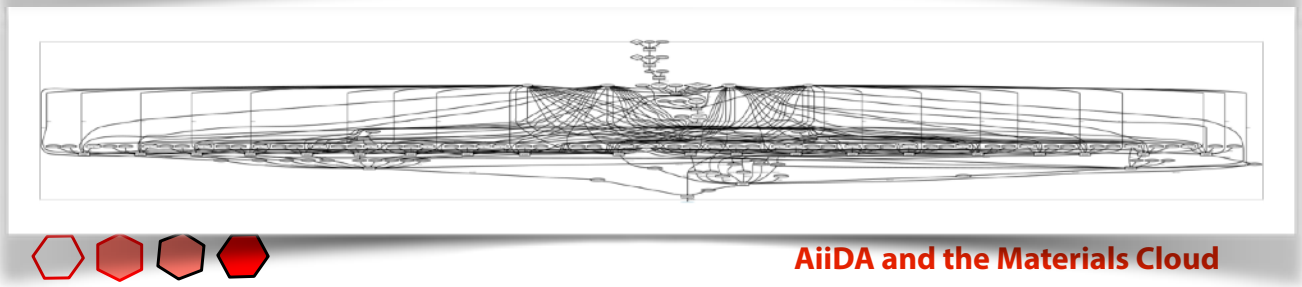
Phonon dispersion

(atom oscillations around equilibrium positions: thermal transport, electronic mobility, …)



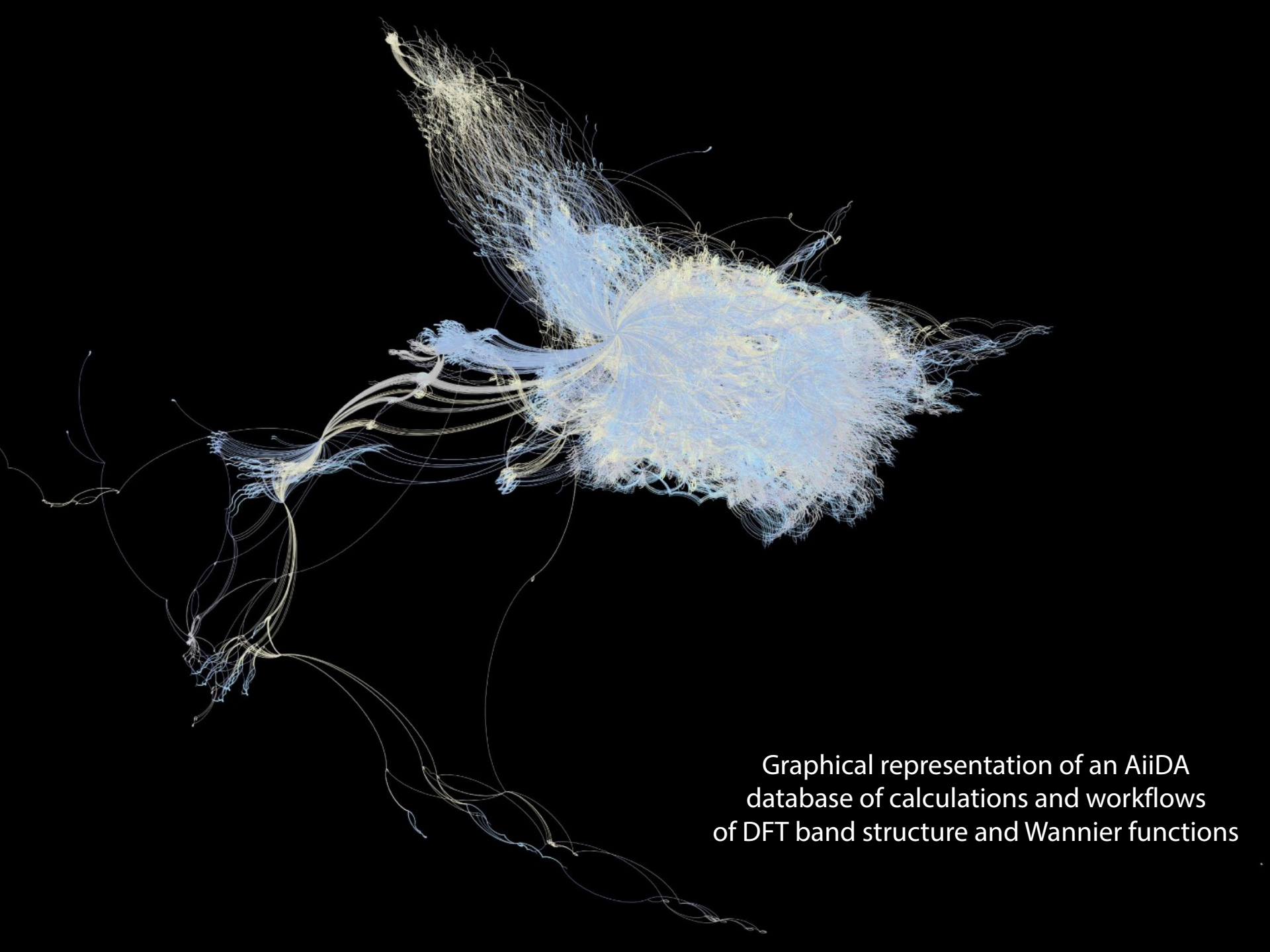
Molecular dynamics of Lithium in a solid electrolyte

(Discover novel, safe and efficient electrolytes for Li-batteries)



Elastic constants

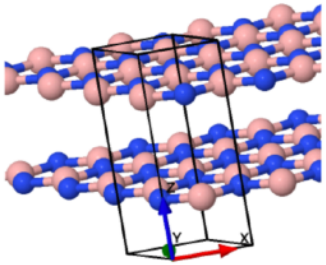
(response of materials to stresses and deformations)

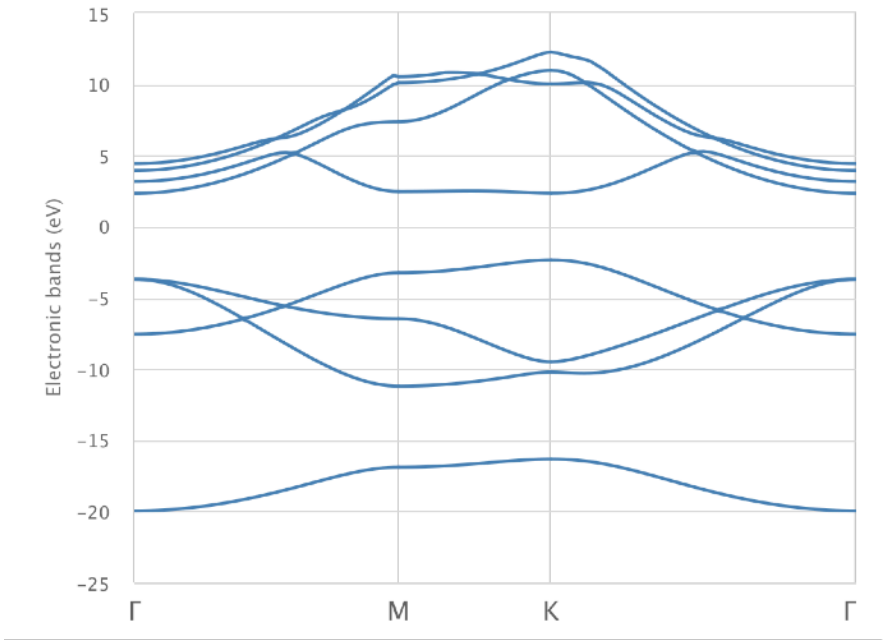


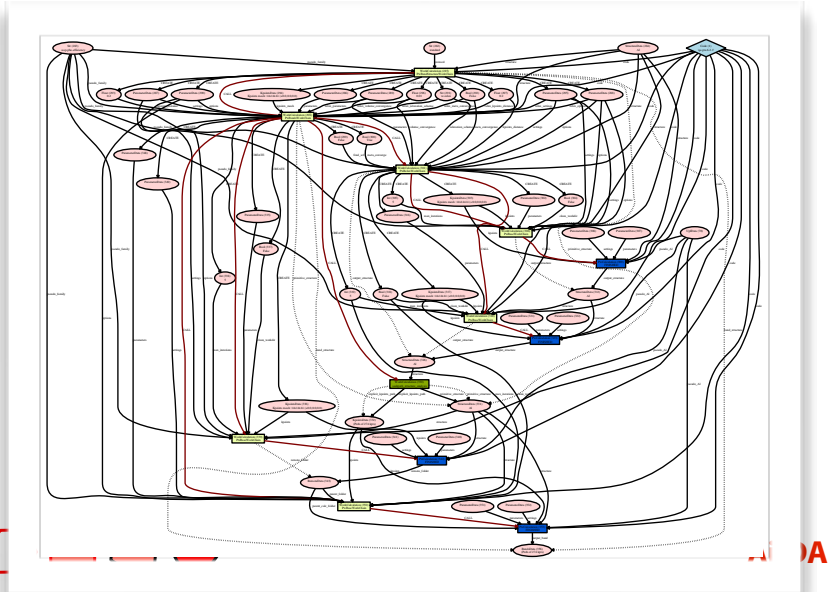
Graphical representation of an AiiDA
database of calculations and workflows
of DFT band structure and Wannier functions

Turn-key workflows in AiiDA

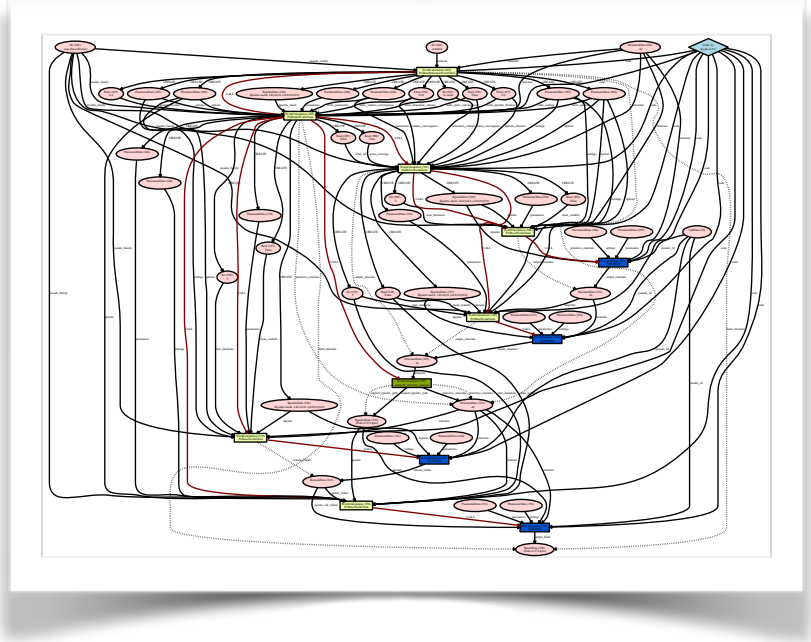
- Given a material, we often need to compute advanced quantities
- These are often non-trivial and result from a complex workflow

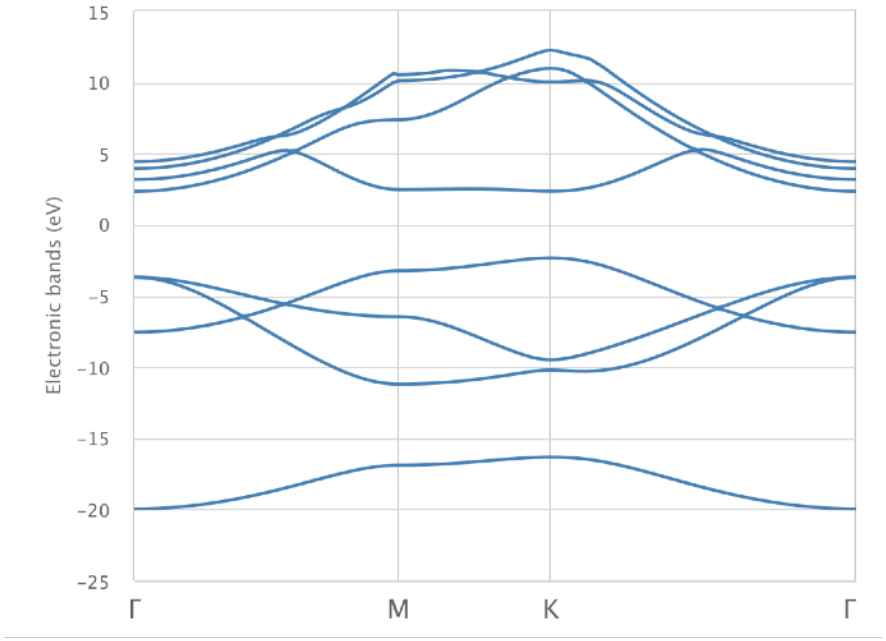






Turn-key workflows in AiiDA



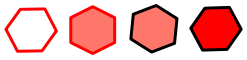


- The AiiDA provenance graph allows to know how the structure was computed and to **reproduce that single specific calculation:** *log of "what happened in the past"*
- We need also an **easy way to re-run the same calculation again** with different parameters or for a different material: **turn-key workflows**

Turn-key workflows in AiiDA

```
class PwBandsWorkChain(WorkChain):
    @classmethod
    def define(cls, spec):

        spec.input('code',
                   valid_type=Code)
        spec.input('structure',
                   valid_type=StructureData)
        spec.input('pseudo_family',
                   valid_type=Str)

        spec.outline(
            cls.setup,
            cls.validate_inputs,
            if_(cls.should_do_relax)(
                cls.run_relax,
            ),
            cls.run_seekpath,
            cls.run_scf,
            cls.run_bands,
            cls.results,
        )
```

- **"Operating system" for all calculations**
- Automatic provenance tracking in the DB
- Control provenance granularity
  store level of detail relevant to the workflows
- Progress checkpointing
  restart from arbitrary step, retry on failure, allows to shut down daemon and continue later
- Easy debugging, self-documenting



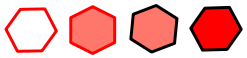
Turn-key workflows in AiiDA

```
class PwBandsWorkChain(WorkChain):
    @classmethod
    def define(cls, spec):

        spec.input('code',
                   valid_type=Code)
        spec.input('structure',
                   valid_type=StructureData)
        spec.input('pseudo_family',
                   valid_type=Str)

        spec.outline(
            cls.setup,
            cls.validate_inputs,
            if_(cls.should_do_relax)(
                cls.run_relax,
            ),
            cls.run_seekpath,
            cls.run_scf,
            cls.run_bands,
            cls.result
        )
```

- **"Operating system" for all calculations**
- Automatic provenance tracking in the DB
- Control provenance granularity
  store level of detail relevant to the workflows
- Progress checkpointing
  restart from arbitrary step, retry on failure, allows to shut down daemon and continue later
- Easy debugging, self-documenting

- Turn-key solution:

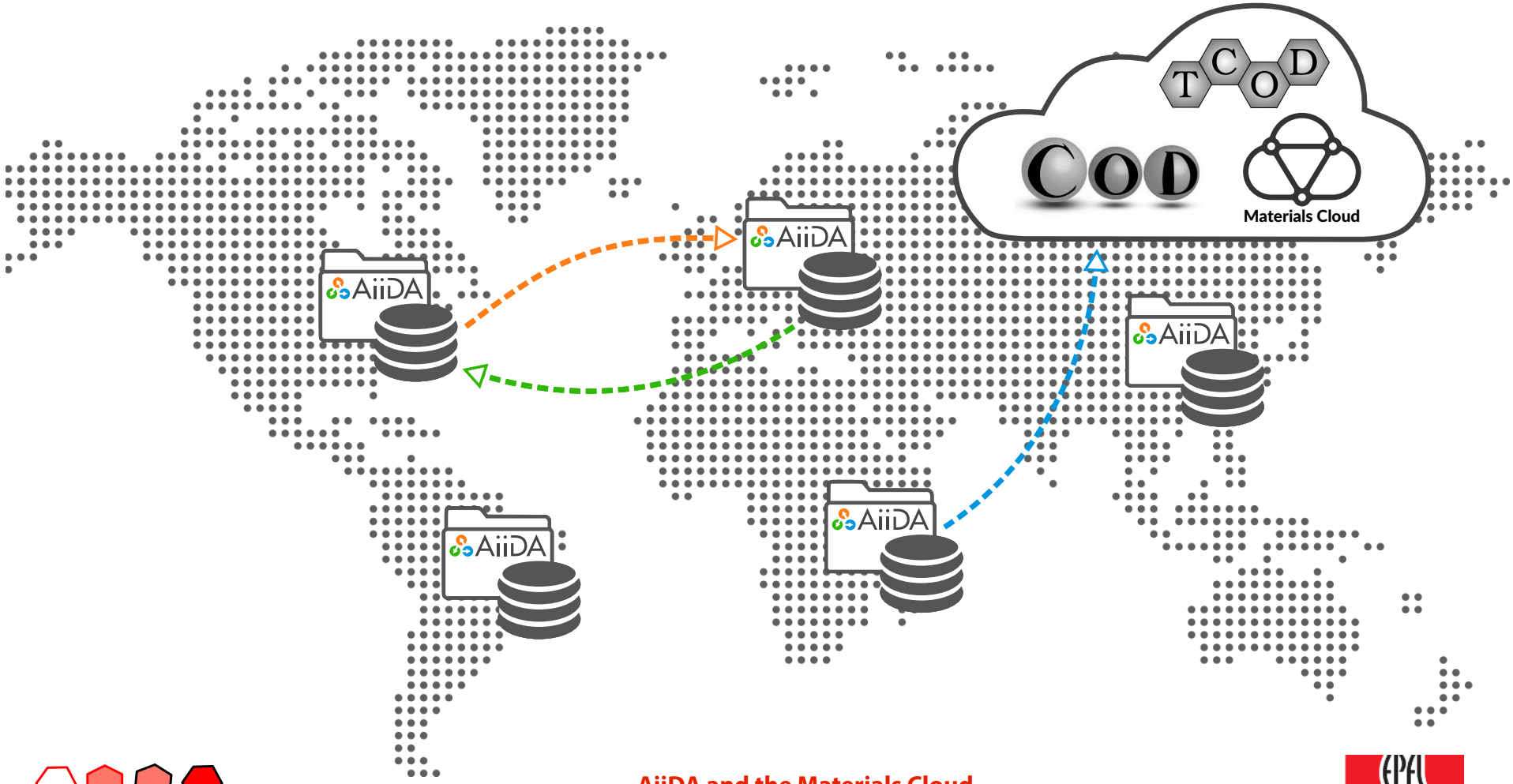
```
PwBandStructureWorkChain.run(
    code=Code.get_from_string(
        'qe-pw-6.2.1@localhost'),
    structure=StructureData(
        ase=ase.build.bulk('Al')),
     pseudo_family=Str('sssp-pbe-efficiency'))
```



Sharing in AiiDA: data and graphs

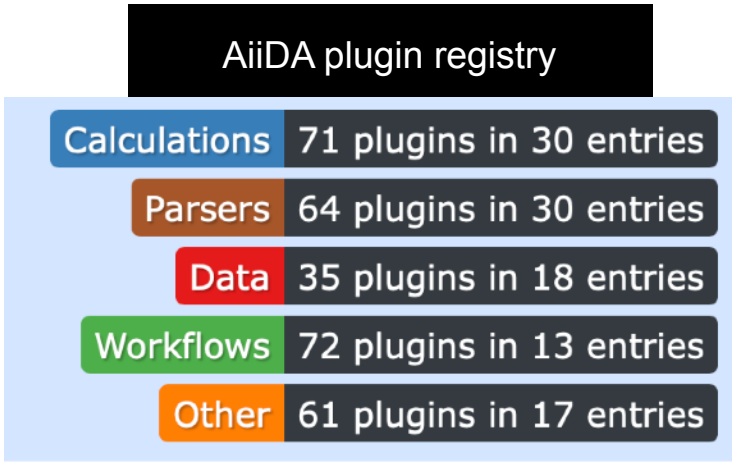
- *Private* AiiDA instances
- UUIDs to uniquely identify nodes

- Data can be shared to other AiiDA repositories **or to online repositories**



Sharing in AiiDA: codes, plugins and workflows

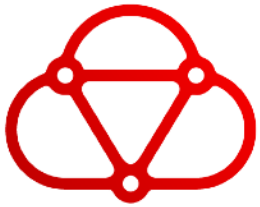
Calculation | **Data** | **Parsers** | **Transport and scheduler** | **Workflows** | **Importers & exporters**



- Plugins are collected in the AiiDA plugin registry
- Over **70 different code executables currently supported from over 30 different codes,** with almost 50 workflows
- Many are **community-contributed**

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

OPEN SCIENCE PLATFORM:

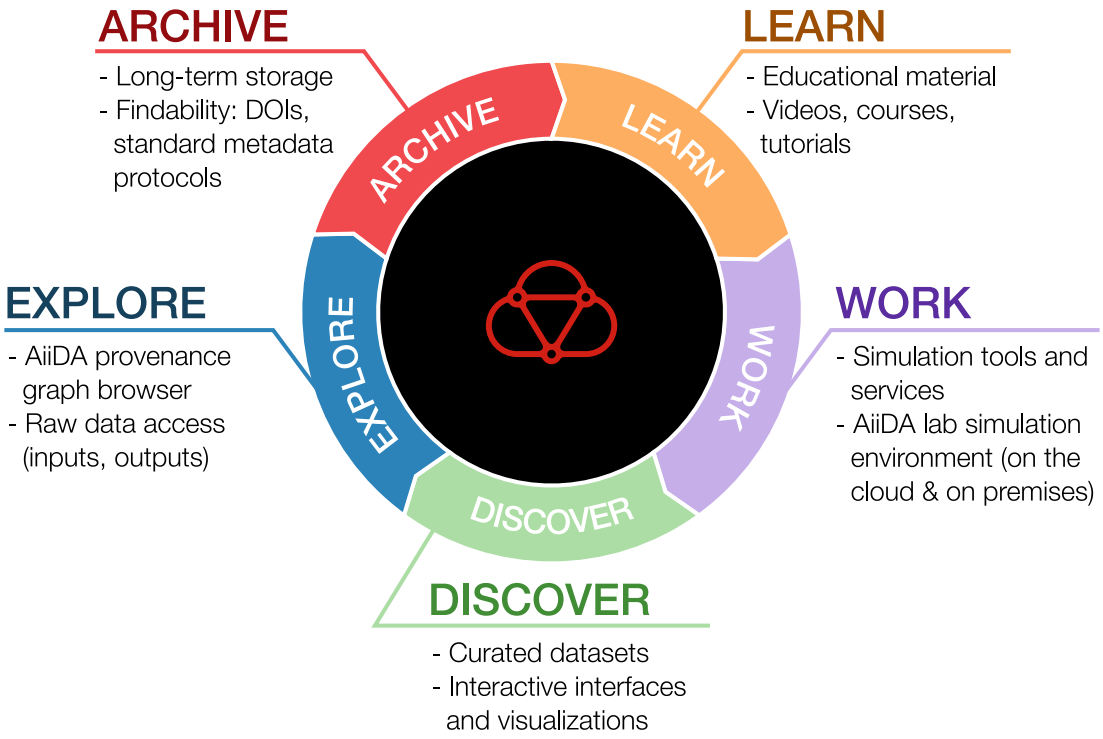


MATERIALSCLOUD

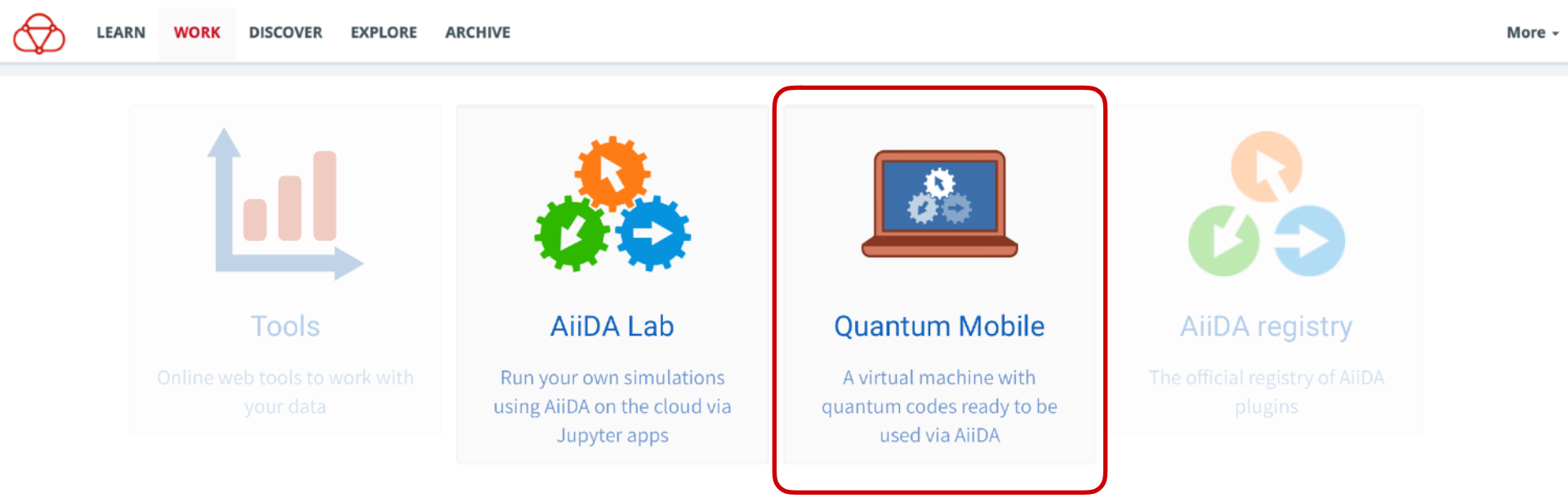
https://www.materialscloud.org

Materials Cloud

- **AiiDA** is the 'engine', like **Git -** used in production *since 2015*
- **Materials Cloud** is the dissemination platform (like **GitHub**) **and more** (cloud computing and data generation platform) - online since *Dec 2017*



Data generation: Materials Cloud **Work**



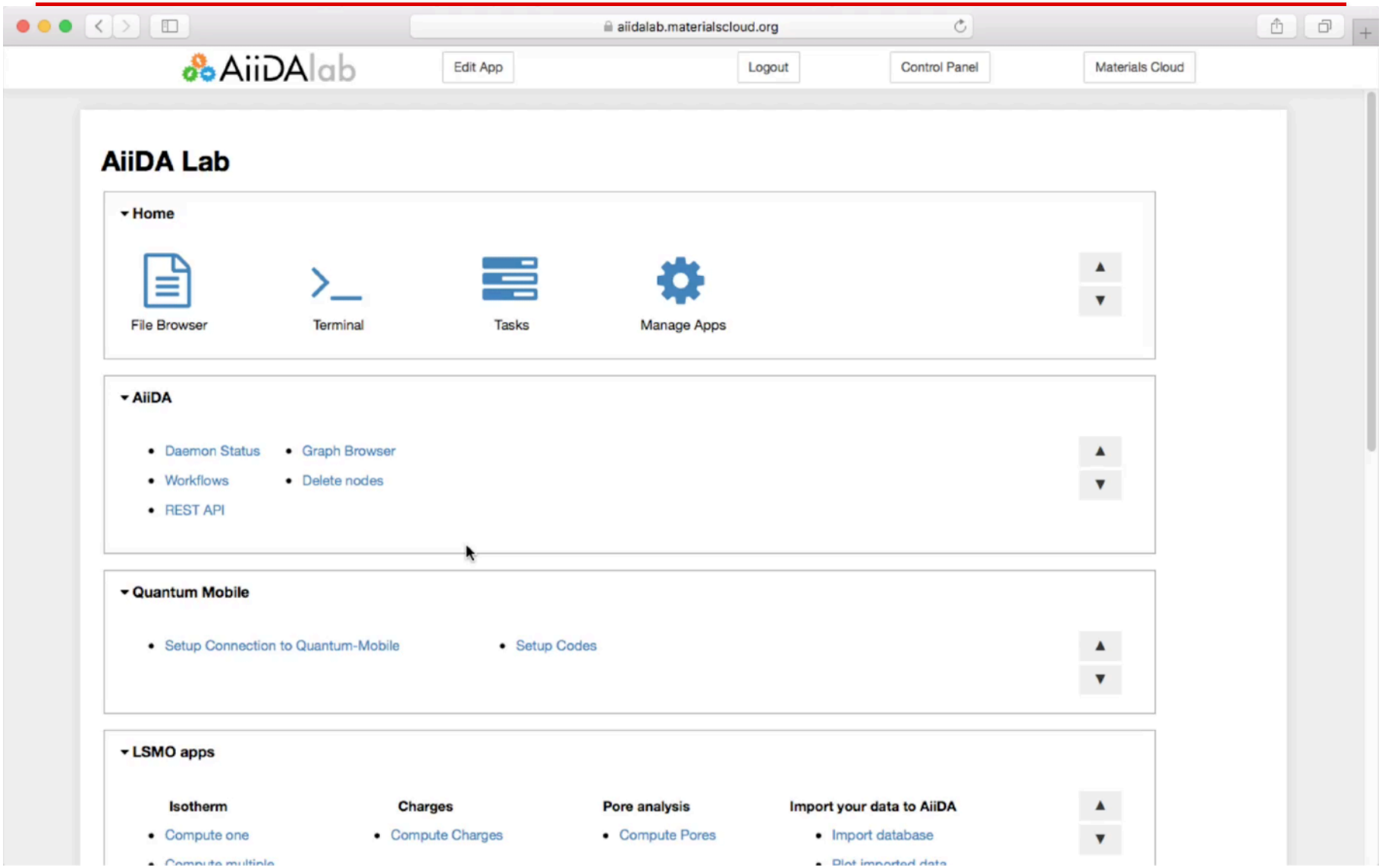
AiiDA Lab

- Comes with a preconfigured AiiDA setup, **ideal interface for turn-key workflows**
- Custom **AppMode** extension to make notebooks look&feel like real web apps knowing only python
- Using JupyterHub + DockerSpawner

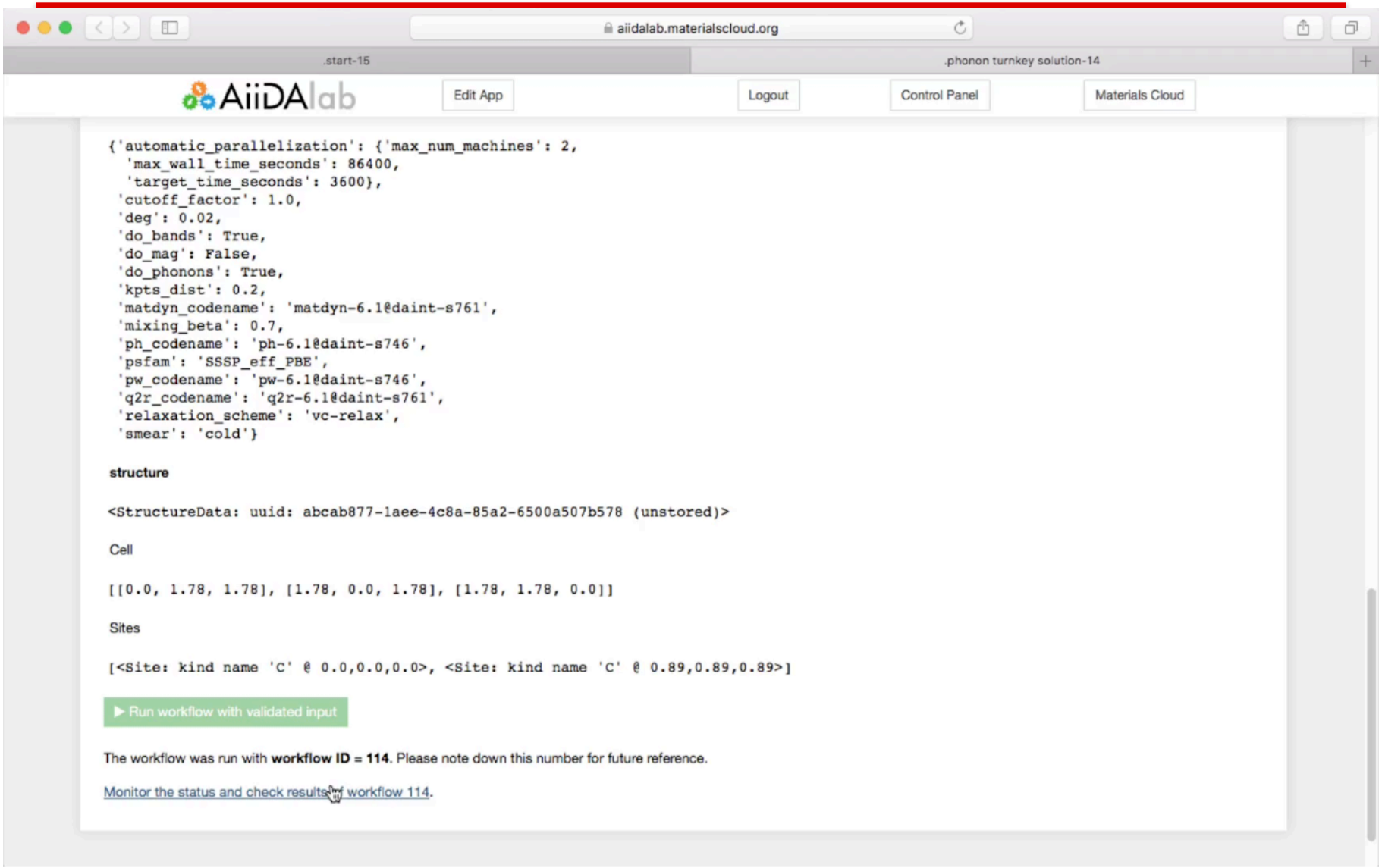
Quantum Mobile

- **Downloadable VM** with preinstalled **AiiDA and codes** like QE, Yambo, Fleur, Siesta, CP2K, …
- Includes **same AiiDA Lab apps environment** as on Materials Cloud
- Ideal for **education**

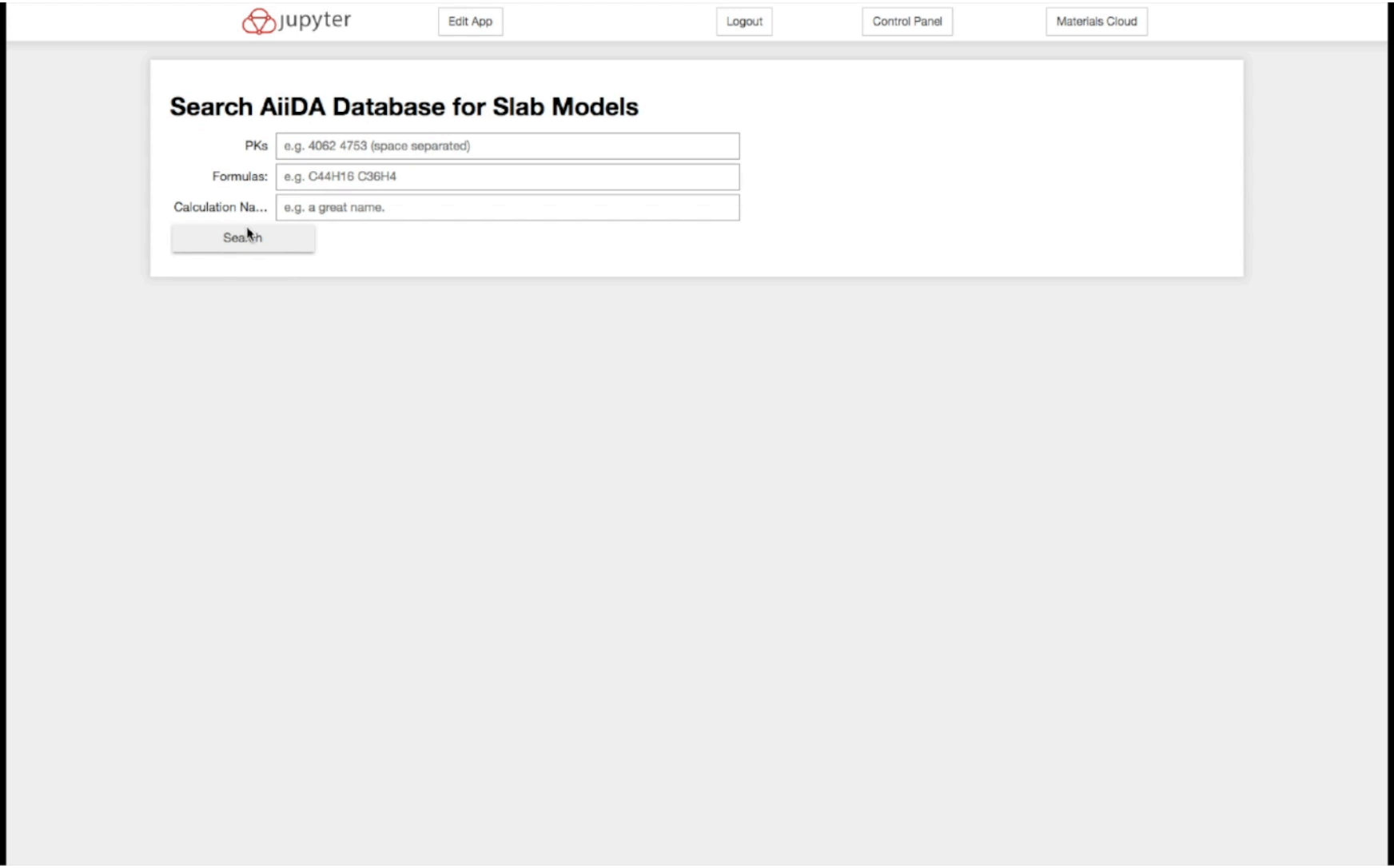
AiiDA Lab: submitting a turn-key solution (phonons)

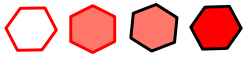


AiiDA Lab: checking the workflow results (phonons)



AiiDA Lab: checking the workflow results (STM)





github.com/cpignedoli/mc-empa-surfaces

ECOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

A solution tailored for every user

User	Skills	Goals	Solution
Computational Scientist	*Knows* Unix, bash, python	• run high-throughput calculations • write complex workflows • develop AiiDA plugins	**AiiDA** on the laptop
Experimental Scientist	*Doesn't know* Unix, bash, python	• run pre-defined workflows • analyze results	**AiiDA Lab** in the cloud
Student (tutorial/ lecture)	*some familiarity with* Unix, bash, python	• learn how to use AiiDA • learn how to use ab-initio codes • take materials home	**Quantum Mobile** on the laptop

Open data sharing: **Archive, Discover, Explore**

materialscloud:2017.0008

SCIENTIFIC DATA re3data.org FAIRsharing.org

Two-dimensional materials from high-throughput computational exfoliation of experimentally known compounds

Authors: Nicolas Mounet[1*], Marco Gibertini[1], Philippe Schwaller[1], Davide Campi[1], Andrius Merkys[1,2], Antimo Marrazzo[1], Thibault Sohier[1], Ivano E. Castelli[1], Andrea Cepellotti[1], Giovanni Pizzi[1], Nicola Marzari[1*]

1 Theory and Simulation of Materials (THEOS), and National Centre for Computational Design and Discovery of Novel Materials (MARVEL), École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland

2 Vilnius University Institute of Biotechnology, Sauletekio al. 7, LT-10257 Vilnius, Lithuania

* Corresponding authors emails: nicolas.mounet@epfl.ch, nicola.marzari@epfl.ch

DOI 10.24435/materialscloud:2017.0008/v2 (version v2, submitted on 21 March 2018)

How to cite this entry

Nicolas Mounet, Marco Gibertini, Philippe Schwaller, Davide Campi, Andrius Merkys, Antimo Marrazzo, Thibault Sohier, Ivano E. Castelli, Andrea Cepellotti, Giovanni Pizzi, Nicola Marzari, *Two-dimensional materials from high-throughput computational exfoliation of experimentally known compounds*, Materials Cloud Archive (2018), doi: 10.24435/materialscloud:2017.0008/v2.

Description

Two-dimensional (2D) materials have emerged as promising candidates for next-generation electronic and optoelectronic applications. Yet, only a few dozens of 2D materials have been successfully synthesized or exfoliated. Here, we search for novel 2D materials that can be easily exfoliated from their parent compounds. Starting from 108423 unique, experimentally known three-dimensional compounds we identify a subset of 5619 that appear layered according to robust geometric and bonding criteria. High-throughput calculations using van-der-Waals density-functional theory, validated against experimental structural data and calculated random-phase-approximation binding energies, allow to identify 1825 compounds that are either easily or potentially exfoliable. In particular, the subset of 1036 easily exfoliable cases provides novel structural prototypes and simple ternary compounds as well as a large portfolio of materials to search from for optimal properties. For a subset of 258 compounds we explore vibrational, electronic, magnetic, and topological properties, identifying 56 ferromagnetic and antiferromagnetic systems, including half-metals and half-semiconductors. This archive entry contains the database of 2D materials (structural parameters, band structures, binding energies, etc.) together with the provenance of all data and calculations as stored by AiiDA.

Materials Cloud sections using this data

Select 2d materials via interactive periodic table and view their properties (with links to provenance)

Explore interface providing access to the full database

DOIs assigned

Direct links to Discover & Explore

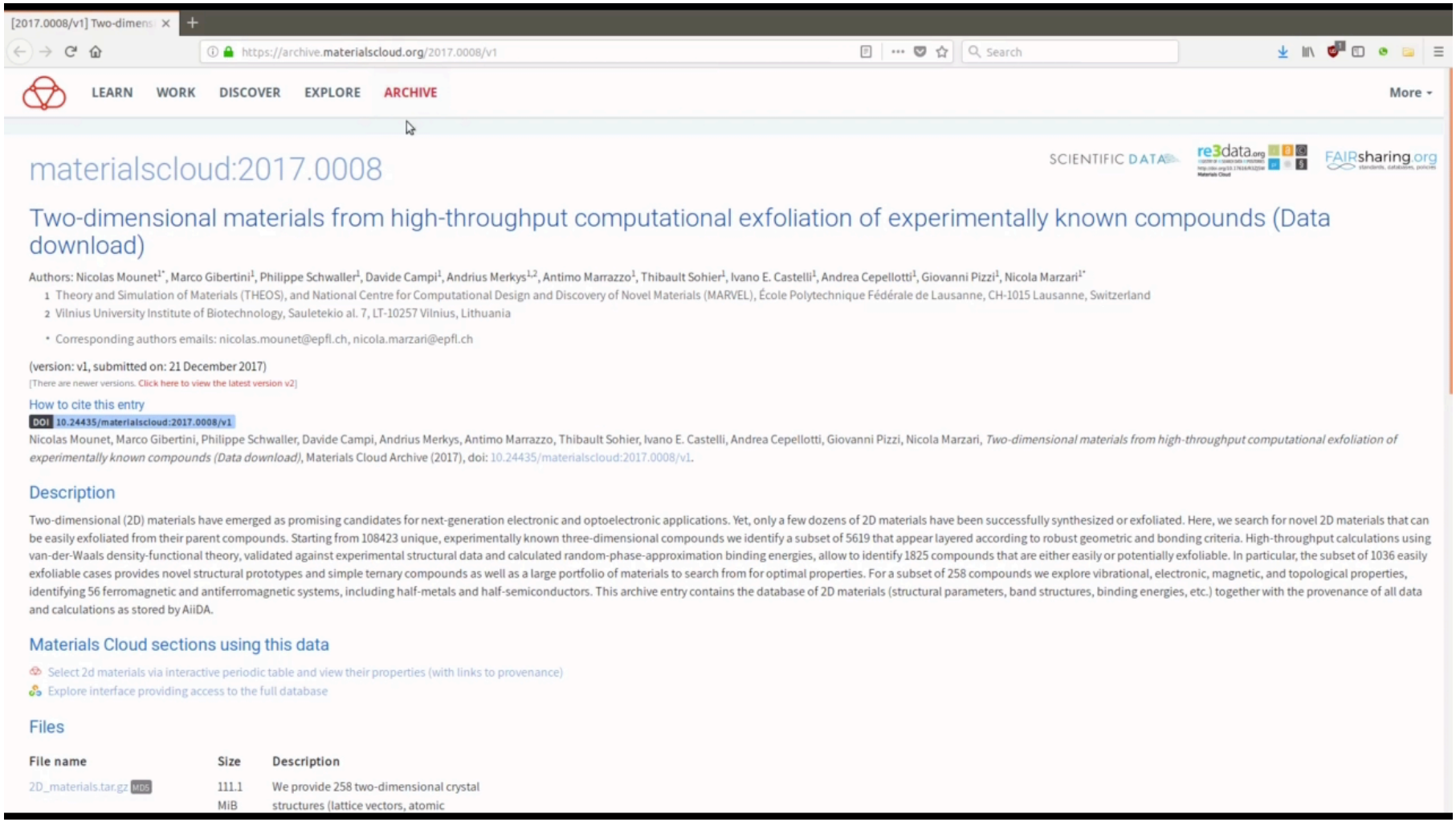
FAIRsharing.org
re3data.org

+

Recommended data repository by Nature's journal *Scientific Data*

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Open data sharing: **Archive, Discover, Explore**



DATA MANAGEMENT PLANS AND FAIR PRINCIPLES

- Combination of **AiiDA + Materials Cloud** (Discover, Explore, Archive): **FAIR-compliant sharing**
  - **Findable**: DOIs with standardized metadata
  - **Accessible**: web interface to browse data, calculations and provenance, curated data in Discover section
  - **Interoperable**: data linked via the AiiDA directed graph; data structures reusable between different codes
  - **Reusable**: downloadable data, encourage open (CC) licences, reproduce in the AiiDA Lab thanks to *full provenance*

- We provide DMP templates for researchers using Materials Cloud (Swiss SNSF, EU H2020, and more in the future)

Below, we provide templates for data management plans using the Materials Cloud Archive (with and without AiiDA).

Funding Body	DMP template (using AiiDA)	DMP template (no AiiDA)
SNF	.docx .odt .pdf	.docx .odt .pdf

Acknowledgements and funding

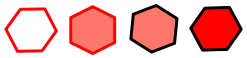
SNSF NCCR "MARVEL"
Discovery of new materials via simulations and dissemination of curated data

H2020 Centre of Excellence "MaX"
Scaling towards exascale machines and high-throughput efficiency

Swissuniversities P-5 "Materials Cloud"
Scaling the web platform, extending to more disciplines

EPFL Open Science Fund "OSSCAR"
Creating a hub of computational resources geared also towards education and teaching

Moreover: *H2020 Marketplace* (providing data and simulation services in a EU Markeplace platform also for industry); *H2020 Intersect* (develop AiiDA workflows to compute transport properties of materials)



Acknowledgements

The Materials Cloud And AiiDA teams


Marco Borelli (EPFL)


Sebastiaan P. Huber (EPFL)


Boris Kozinsky (BOSCH)


Snehal P. Kumbhar (EPFL)


Leonid Kahle (EPFL)


Nicola Marzari (EPFL)

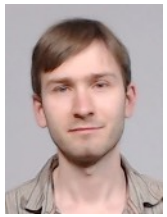

Elsa Passaro (EPFL)

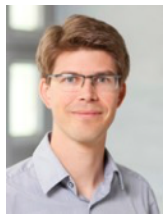

Giovanni Pizzi (EPFL)


Thomas Schulthess (ETHZ,CSCS)


Berend Smit (EPFL)


Leopold Talirz (EPFL)


Martin Uhrin (EPFL)


Joost VandeVondele (ETHZ,CSCS)


Aliaksandr Yakutovich (EPFL)


Spyros Zoupanos (EPFL)

Contributors for the 20+ plugins: Quantum ESPRESSO, Wannier90, CP2K, FLEUR, YAMBO, SIESTA, VASP, …

Contributors to aiida_core and former AiiDA team members — Valentin Bersier, Jocelyn Boullier, Jens Broeder, Andrea Cepellotti, Fernando Gargiulo, Christoph Koch, Dominik Gresch, Rico Haüselmann, Eric Hontz, Andrius Merkys, Nicolas Mounet, Tiziano Müller, Ole Schütt, Riccardo Sabatini, Phillippe Schwaller

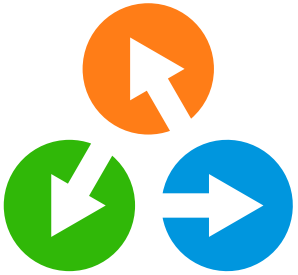
The CSCS support teams

Summary

- **AiiDA**: the reproducibility and automation engine
  - Define turn-key workflows, automate them on supercomputers
  - Keep track of the provenance in the form of a graph
  - Share data, plugins and workflows
- **Materials Cloud**: the dissemination portal
  - **AiiDA Lab**: data generation platform, both on the cloud or locally (Quantum Mobile VM)
  - *Archive*: Findable long-term storage + *Discover* (curated data) + *Explore* (raw AiiDA data): FAIR sharing
  - Supports researchers in data management and DMPs

Open Science: not only *open data, codes and workflows*, but also ***straightforward access to them***

Contacts



Website: http://www.aiida.net

Docs: http://aiida-core.readthedocs.io

Git repo: https://github.com/aiidateam/aiida_core/

Plugin registry: http://aiidateam.github.io/aiida-registry

https://www.facebook.com/aiidateam

@aiidateam

Materials Cloud: http://www.materialscloud.org

- **AiiDA Lab**: http://aiidalab.materialscloud.org
- **Archive**: http://archive.materialscloud.org

Quantum Mobile: http://www.materialscloud.org/work/quantum-mobile