Computational Materials Science in the High-Throughput Era

with AiiDA and the Materials Cloud

Leopold Talirz, Aliaksandr V. Yakutovich, Daniele Ongari

MAX DRIVING THE EXASCALE TRANSITION

MARVEL

EPFL ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Today's schedule

9:00-10:30	Introductory lecture
10:30-11:00	Coffee break
11:00-12:00	Getting everybody set up Group A: Room X | Group B: Room Y
12:00-13:00	Lunch break
13:00-17:00	Tutorial & exercises Group A: Room X | Group B: Room Z

Outline

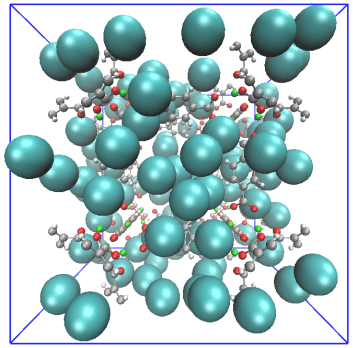
AiiDA

❶ Motivation, Architecture

MATERIALS CLOUD

❷

AiiDAlab



❸ Topic of today's tutorial

Outline

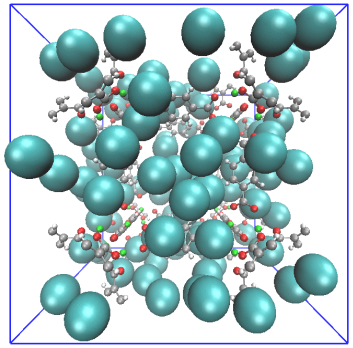
AiiDA

❶ Motivation, Architecture

MATERIALS CLOUD

❷

AiiDAlab



❸ Topic of today's tutorial

Motivation

Computational Materials Science Challenges

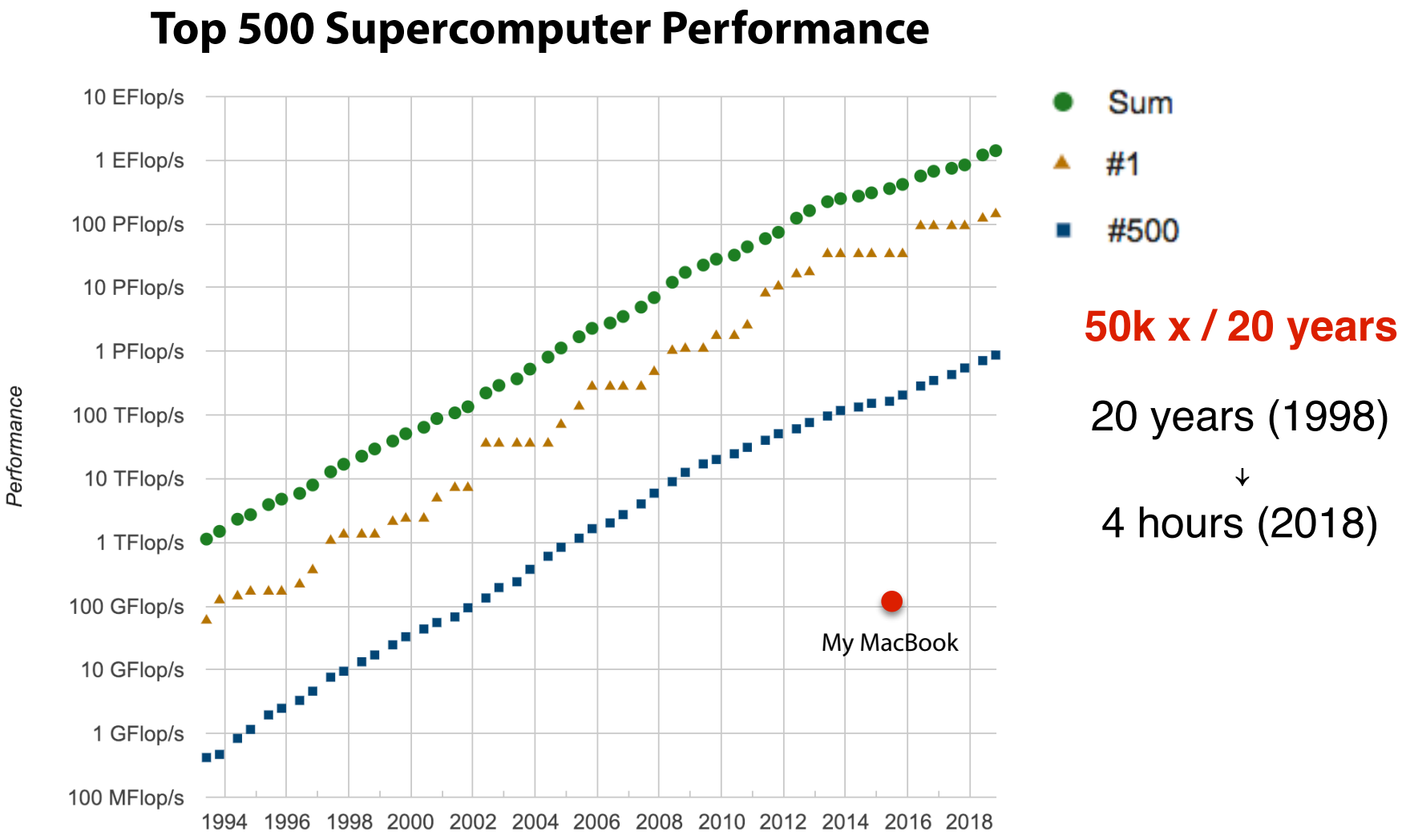
High-Throughput

Reproducibility

Open Science

Knowledge Transfer

Challenge 1 – High Throughput



www.top500.org/statistics/perfdevel

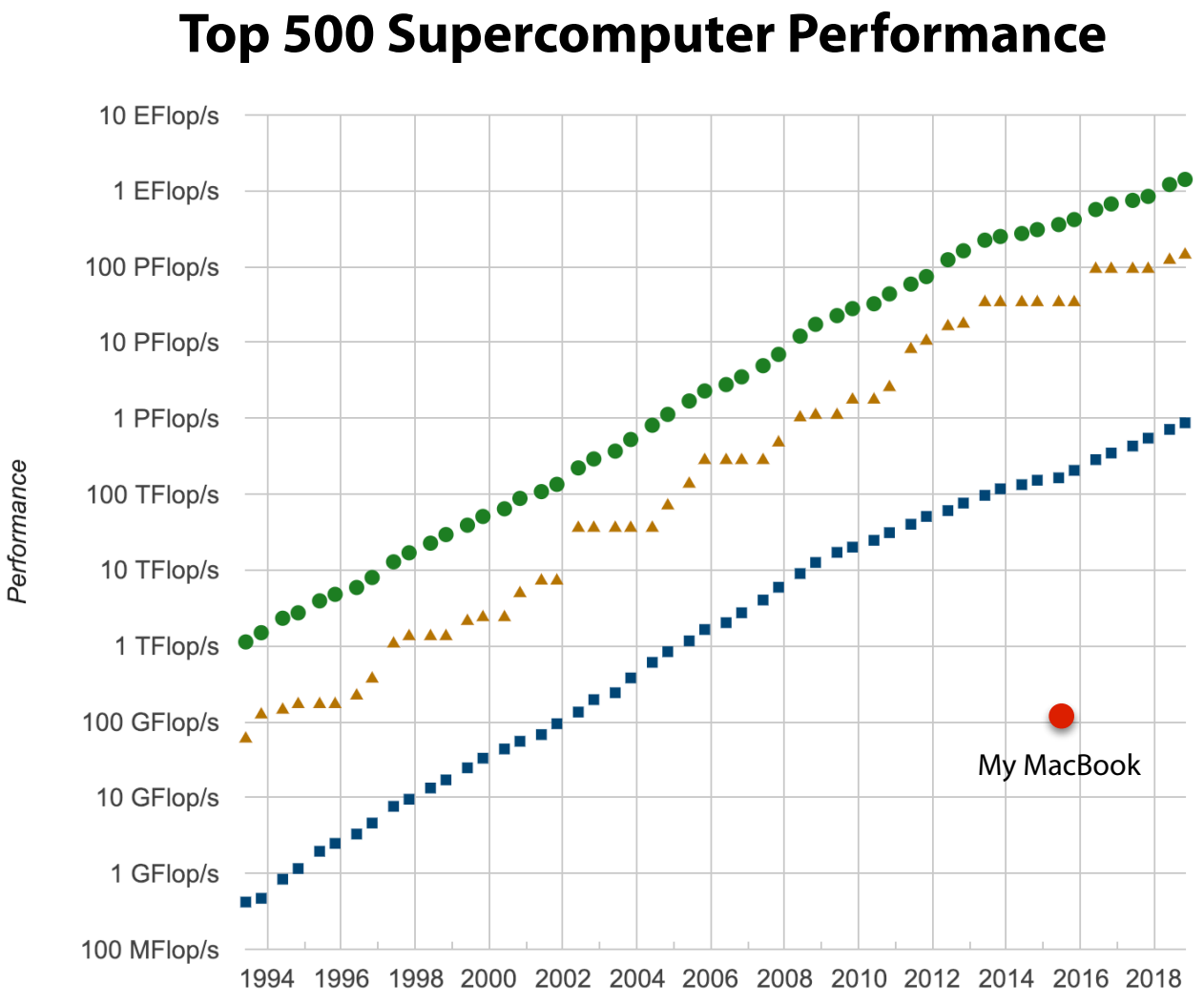
50k x / 20 years

20 years (1998)

↓

4 hours (2018)

Challenge 1 – High Throughput



www.top500.org/statistics/perfdevel

50k x / 20 years

20 years (1998)

↓

4 hours (2018)

OR

1 material (1998)

↓

50k materials (2018)

Motivation

Computational Materials Science Challenges

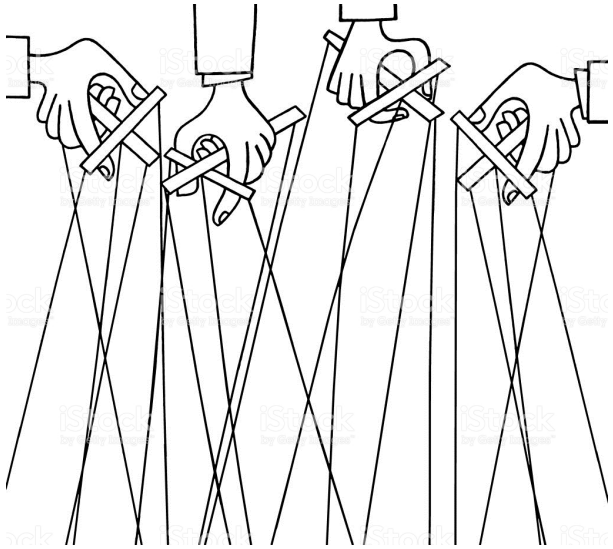
High-Throughput

Reproducibility

Open Science

Knowledge Transfer

- Organize large numbers of calculations
- Deal with corner cases (theory, code, infrastructure)
- Many strings to pull



Source: istockphoto.com

Motivation

Computational Materials Science Challenges

High-Throughput

Reproducibility

Open Science

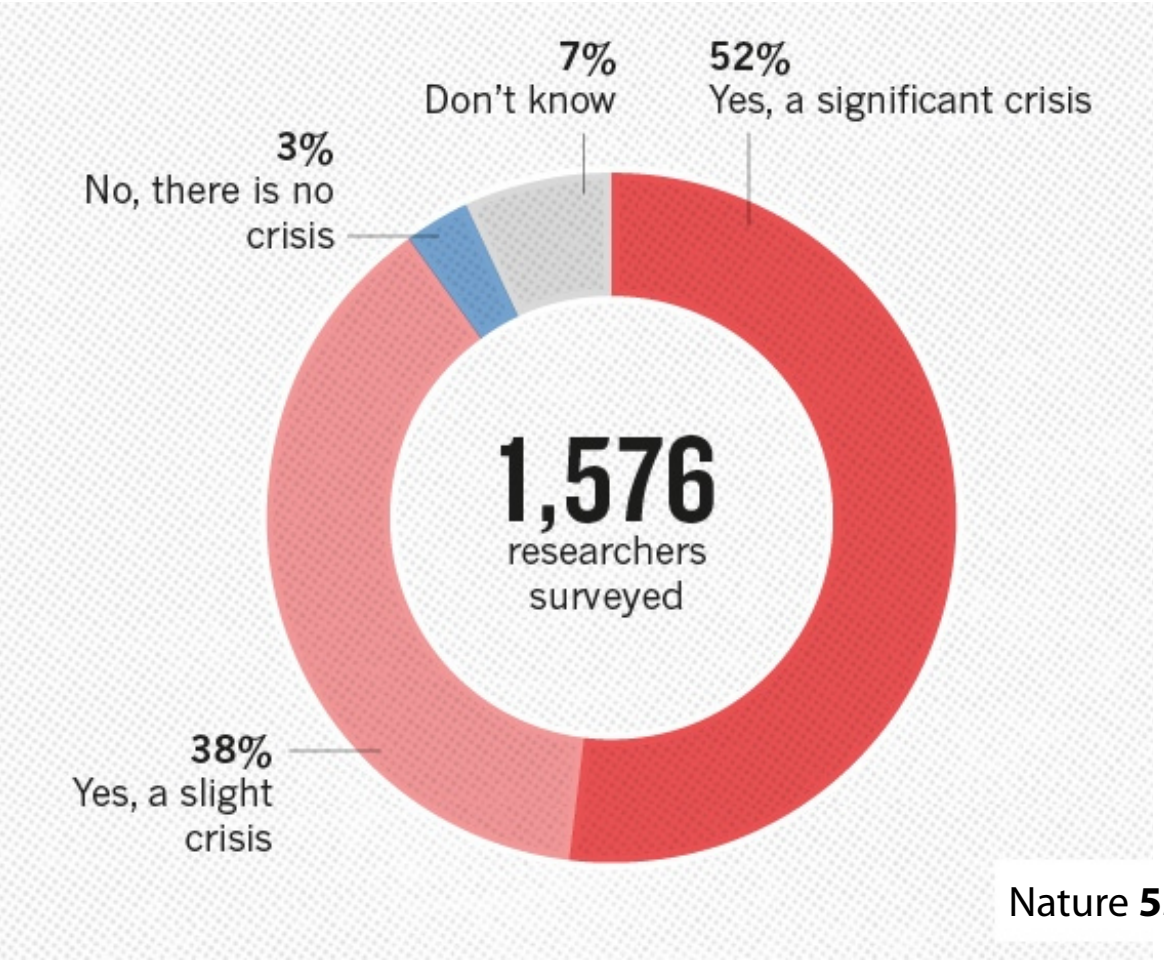
Knowledge Transfer

- Keep track of what you calculate
- Keep track of how you did it
- Within a research group:
  Can Alice reproduce what Bob computed 1 year ago?

Source: academiccoachingandwriting.org

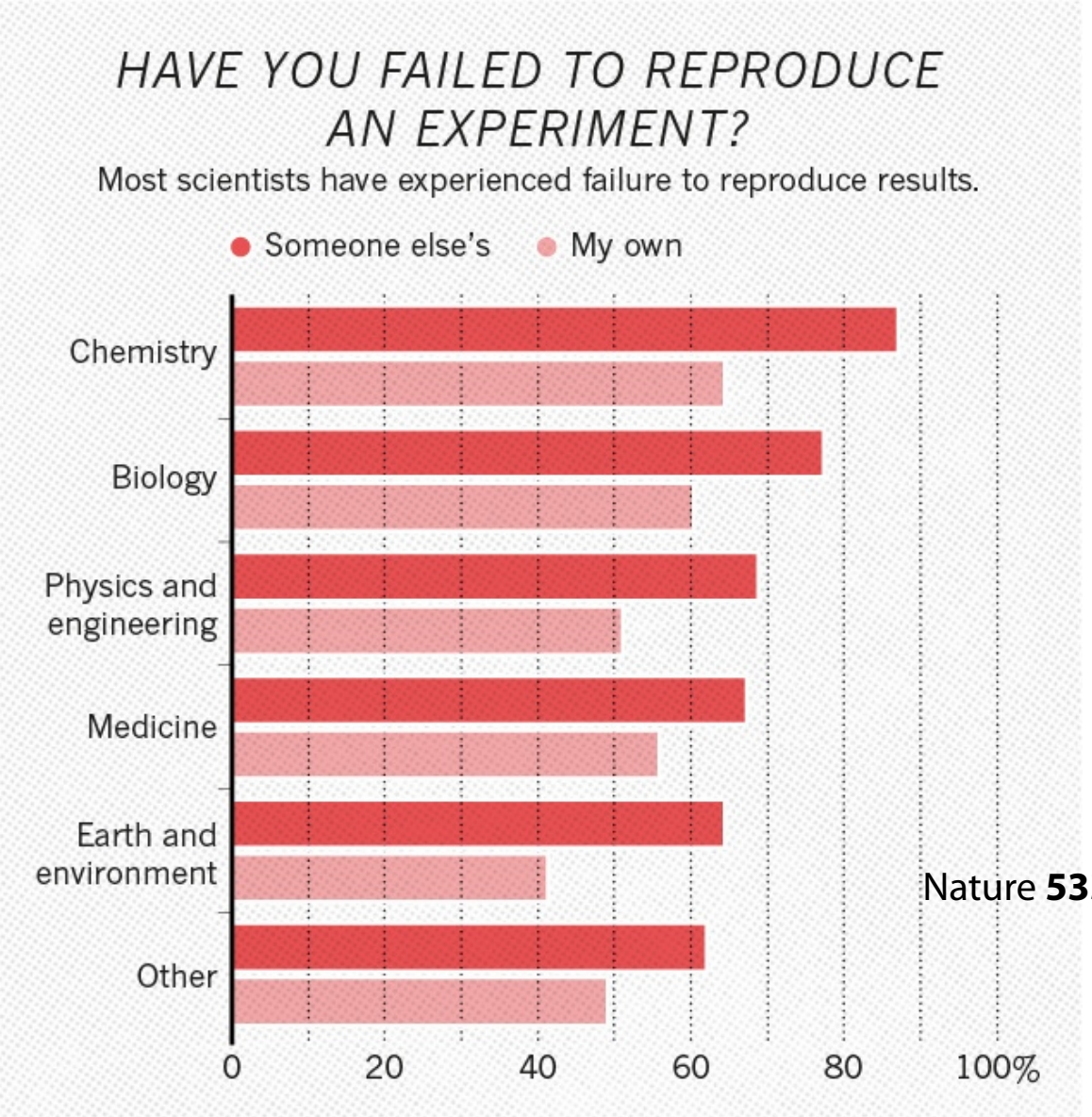
Challenge 2 – Reproducibility

IS THERE A REPRODUCIBILITY CRISIS?



Nature **533**, 452–454 (2016)

Challenge 2 – Reproducibility



Nature **533**, 452–454 (2016)

Challenge 2 – Reproducibility

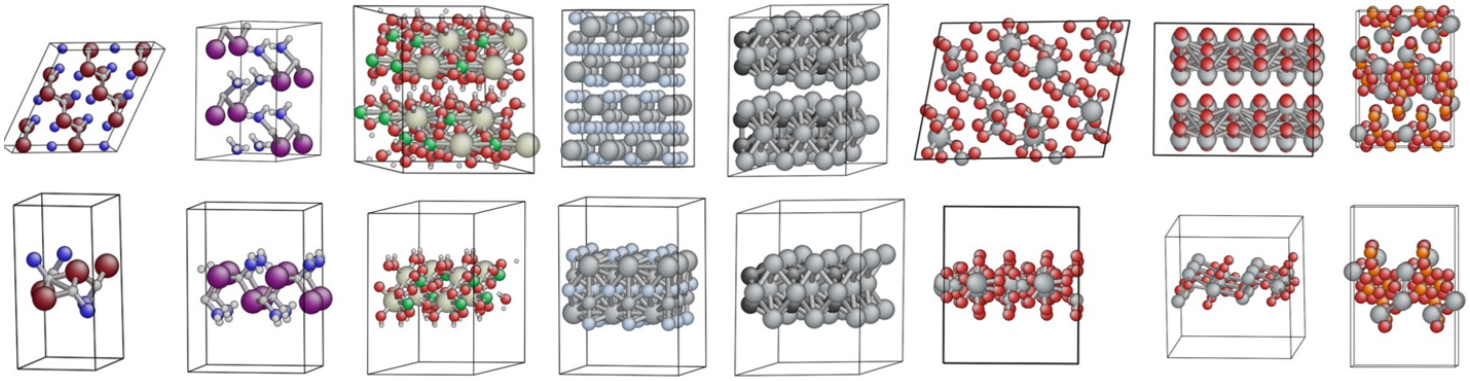
No excuses in computational science

We can and **must** be fully reproducible

Nature **533**, 452–454 (2016)

High-throughput Example

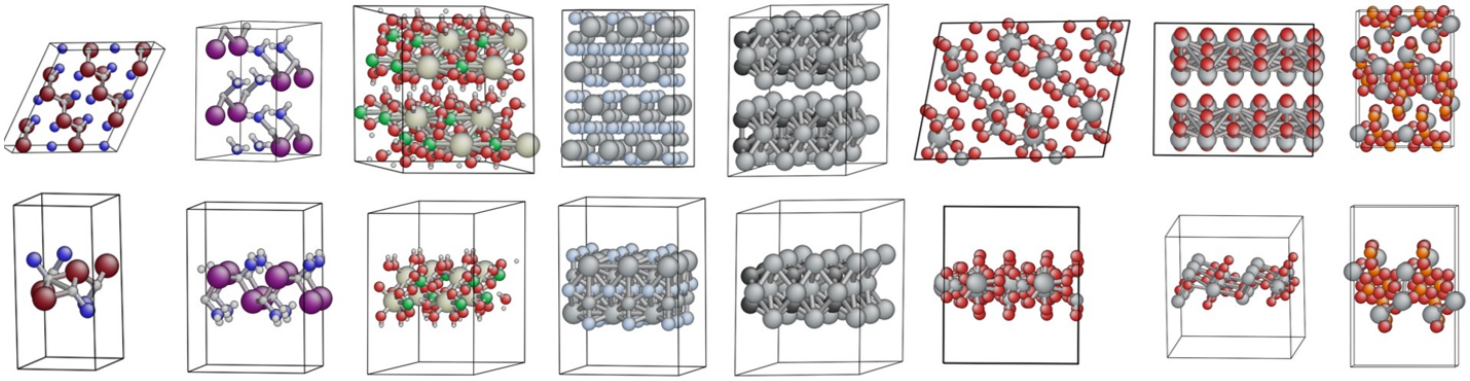
Discovering new two-dimensional materials



N. Mounet et al. Nat Nanotech **13**, *246-52* (2018). doi: 10.1038/s41565-017-0035-5

High-throughput Example

Discovering new two-dimensional materials



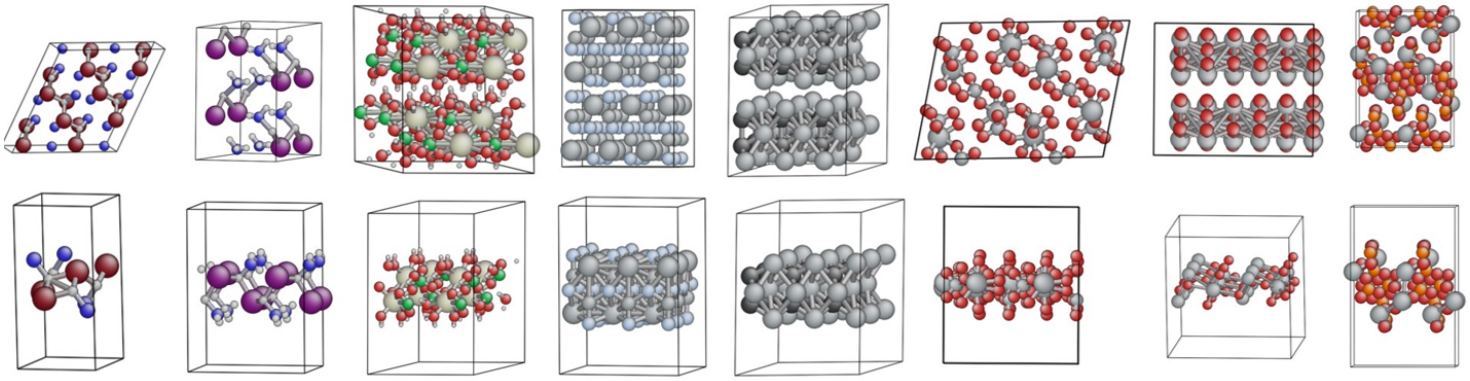
STARTING FROM ICSD/COD DATABASE:

- **108 423** unique 3D structures
- **5619** layered structures
- >**100 000** DFT calculations
- >**30 000** material properties
- >**1·10^{9}** attributes

N. Mounet et al. Nat Nanotech **13**, *246-52* (2018). doi: 10.1038/s41565-017-0035-5

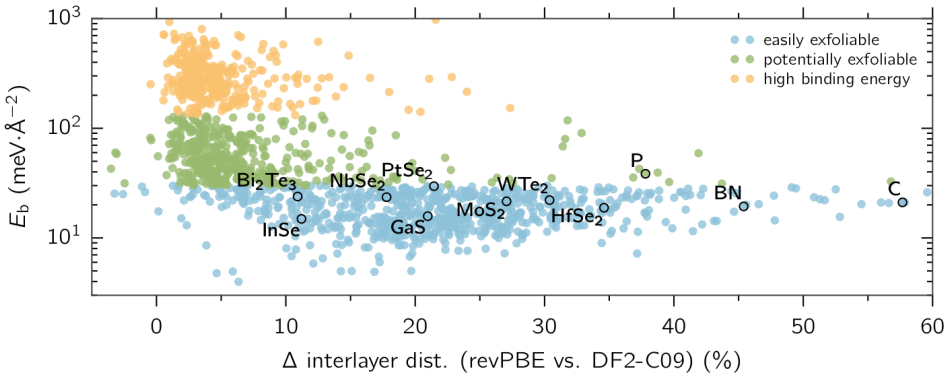
High-throughput Example

Discovering new two-dimensional materials



Starting from ICSD/COD database:

- **108 423** unique 3D structures
- **5619** layered structures
- >**100 000** DFT calculations
- >**30 000** material properties
- >$\mathbf{1 \cdot 10^9}$ attributes

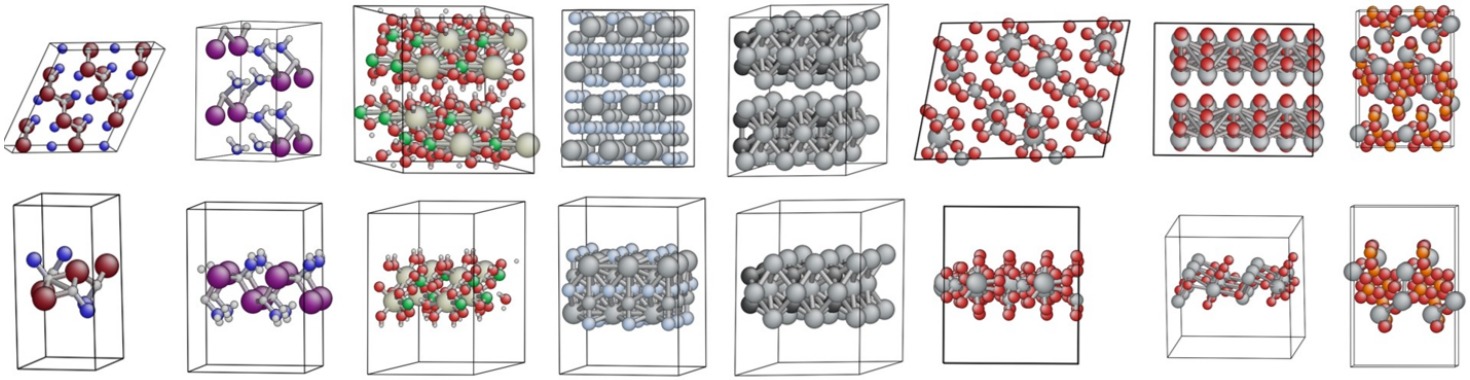


Data needs to be condensed in a few plots

N. Mounet et al. Nat Nanotech **13**, *246-52* (2018). doi: 10.1038/s41565-017-0035-5

High-throughput Example

Discovering new two-dimensional materials



Starting from ICSD/COD database:

- **108 423** unique 3D structures
- **5619** layered structures
- >**100 000** DFT calculations
- >**30 000** material properties
- >**1·10^9** attributes

III. METHODS

A. Reproducibility and provenance

It is an often repeated tenet that results of scientific research must be reproducible. This objective is, however, challenging, especially in high-throughput research, due to the large number of simulations involved and the complex sequence of logical steps needed in the study. To ensure reproducibility we use AiiDA[30] as a materials' informatics infrastructure to implement the ADES model of automation, data, environment, and sharing, as dis-

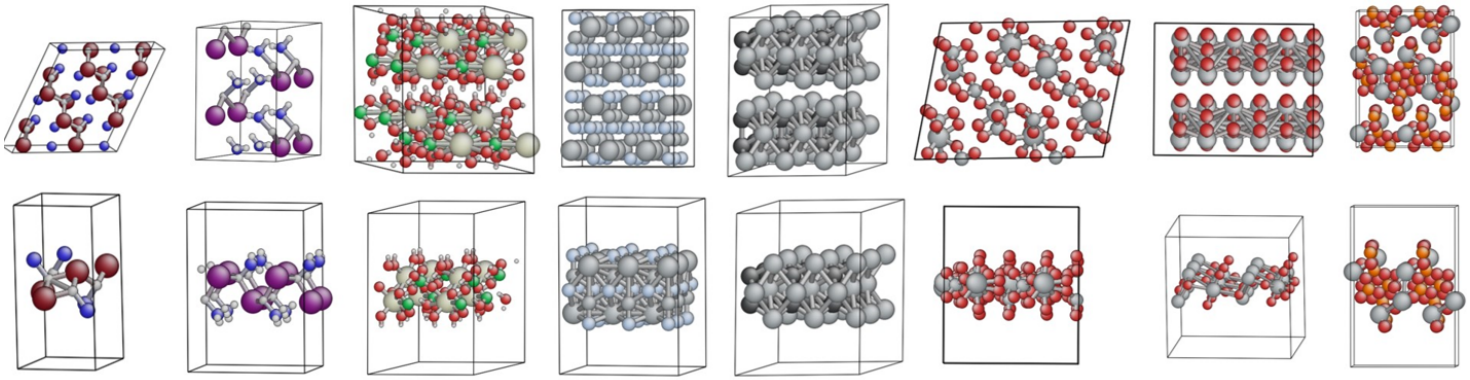
As already mentioned in the main text, in this step we consider only structures coming from experimental measurements, while the source databases are partially populated by purely theoretical structures. This is done using the flags set on the database entries by their curators. We also implemented some heuristics to detect clearly wrong CIF files. For example, we discard structures where the chemical formula provided in the file is inconsistent with the elements in the unit cell. Regardless of all these efforts, it is possible that some incomplete or incorrect structures are still not filtered out from the original databases.

Methods: Impossible to describe every detail

N. Mounet et al. Nat Nanotech **13**, *246-52* (2018). doi: 10.1038/s41565-017-0035-5

High-throughput Example

Discovering new two-dimensional materials



STARTING FROM ICSD/COD DATABASE:

- **108 423** unique 3D structures
- **5619** layered structures
- >**100 000** DFT calculations
- >**30 000** material properties
- >$1 \cdot 10^9$ attributes

III. METHODS

A. Reproducibility and provenance

It is an often repeated tenet that results of scientific research must be reproducible. This objective is, however, challenging, especially in high-throughput research, due to the large number of simulations involved and the complex sequence of logical steps needed in the study. To ensure reproducibility we use AiiDA[30] as a materials' informatics infrastructure to implement the ADES model of automation, data, environment, and sharing, as dis-

As already mentioned in the main text, in this step we consider only structures coming from experimental measurements, while the source databases are partially populated by purely theoretical structures. This is done using the flags set on the database entries by their curators. We also implemented some heuristics to detect clearly wrong CIF files. For example, we discard structures where the chemical formula provided in the file is inconsistent with the elements in the unit cell. Regardless of all these efforts, it is possible that some incomplete or incorrect structures are still not filtered out from the original databases.

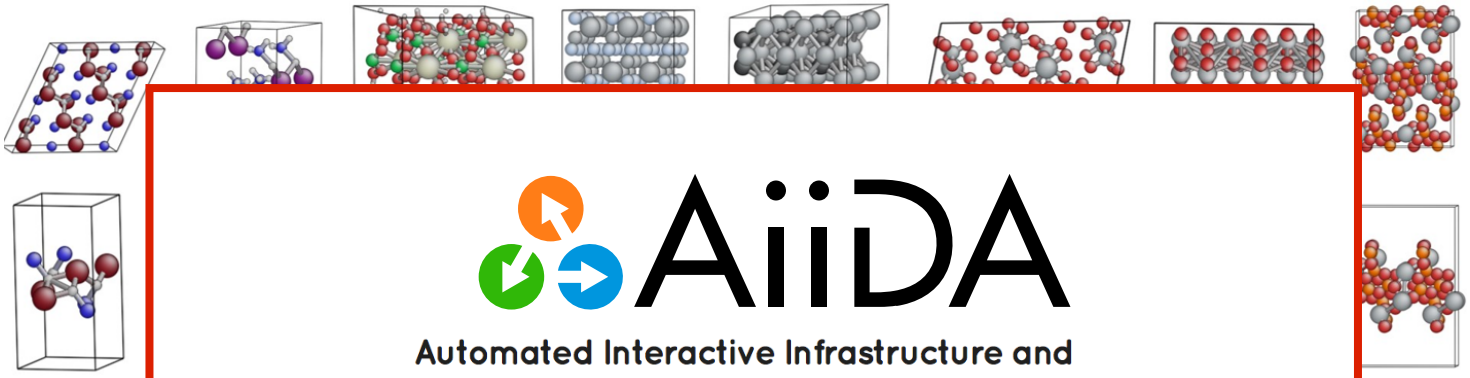
Methods: Impossible to describe every detail

For **authors**, reproducing all data is challenging.
For **peers**, reproducing all data is almost impossible.

N. Mounet et al. Nat Nanotech **13**, *246-52* (2018). doi: 10.1038/s41565-017-0035-5

High-throughput Example

DISCOVERING NEW TWO-DIMENSIONAL MATERIALS



AiiDA

Automated Interactive Infrastructure and Database for Computational Science

- Computational science platform
- for high-throughput calculations
- with automatic data provenance

STARTING FROM

- **108 423** unic
- **5619** layered
- >**100 000** DF
- >**30 000** mat
- >$1 \cdot 10^9$ attributes

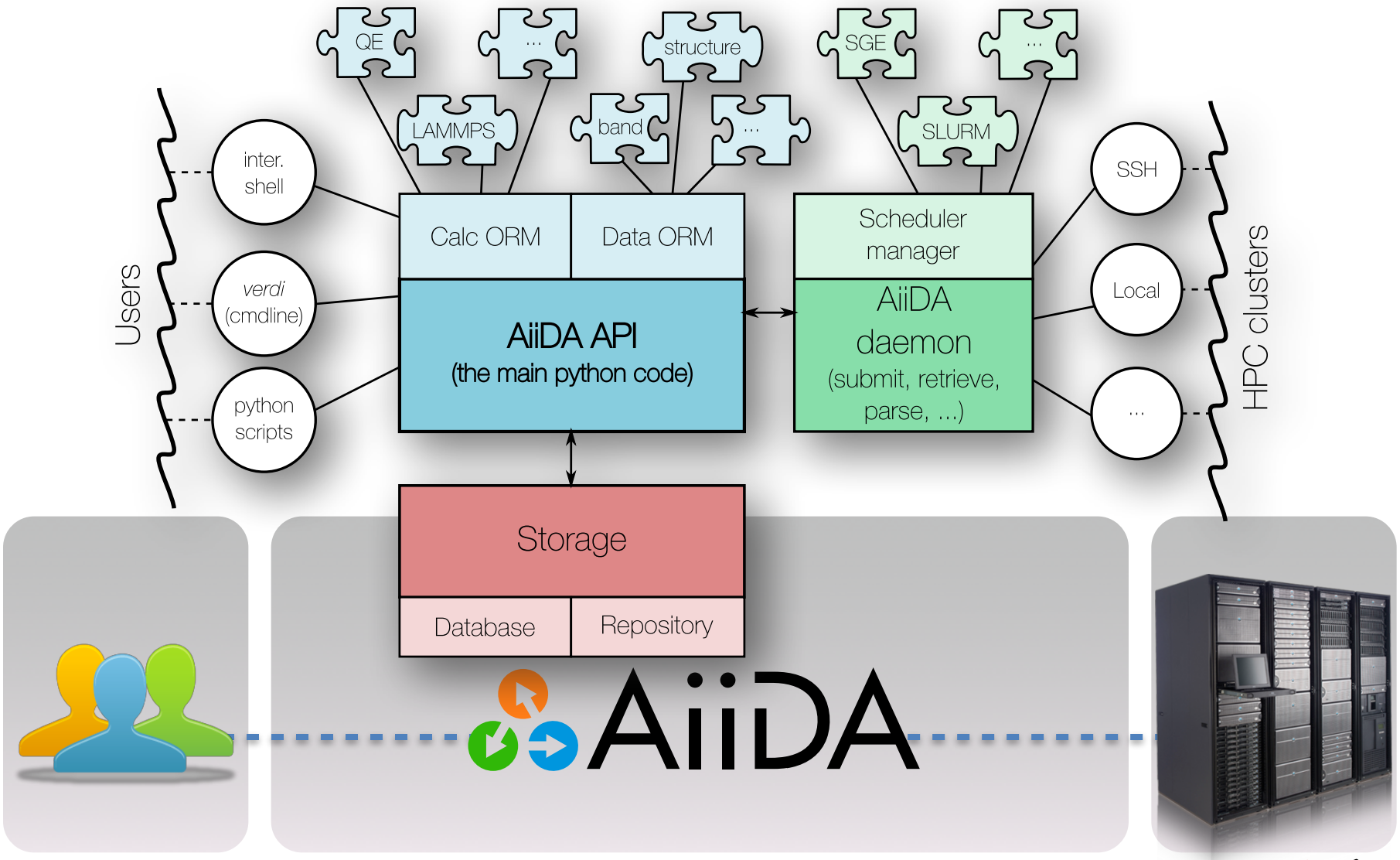
ed in the main text, in this step
ctures coming from experimental
he source databases are partially
eoretical structures. This is done
the database entries by their cu-
mented some heuristics to detect
s. For example, we discard struc-
cal formula provided in the file is
lements in the unit cell. Regard-
it is possible that some incomplete
are still not filtered out from the

Methods: Impossible to describe every detail

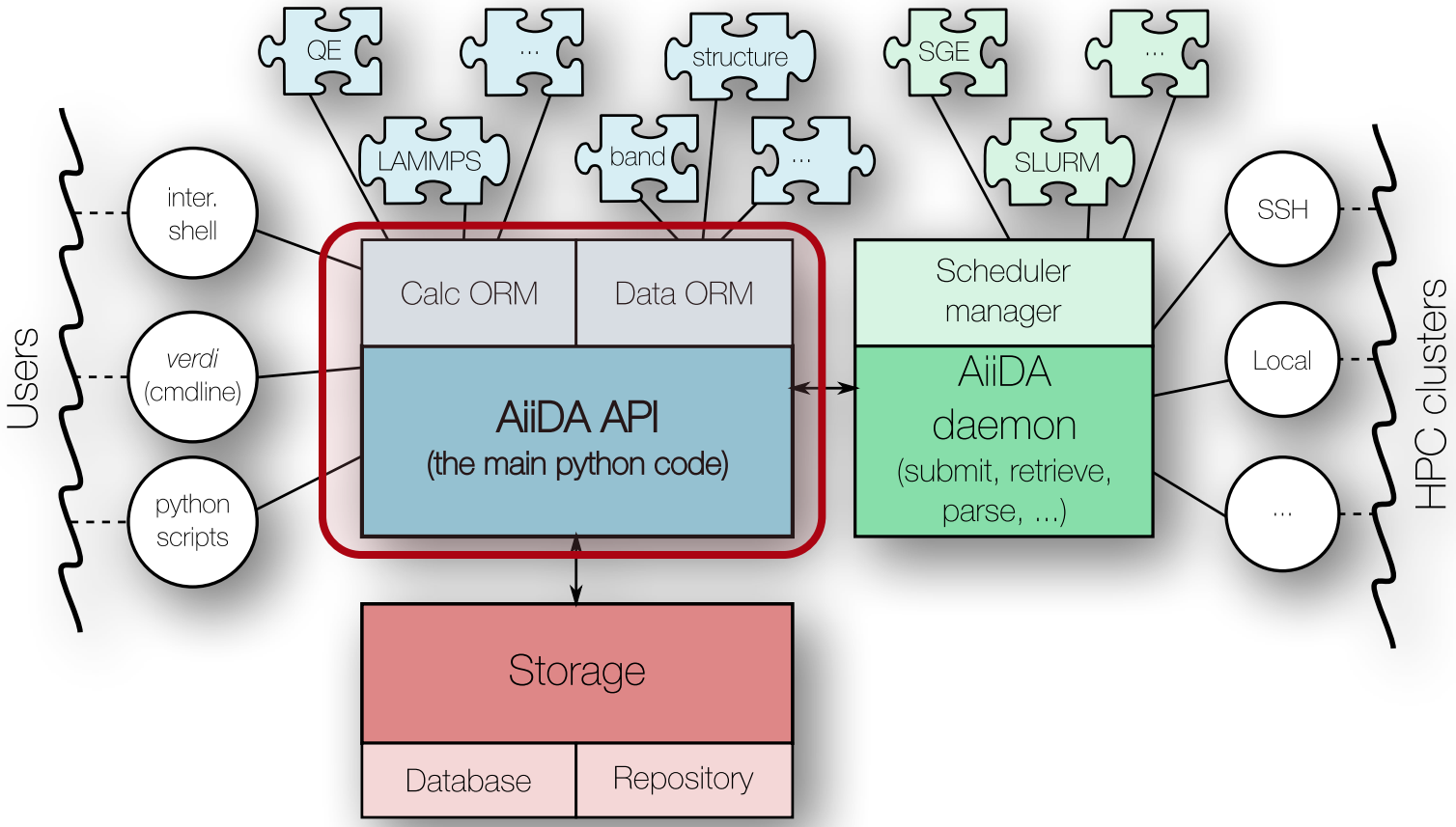
For **authors**, reproducing all data is challenging.
For **peers**, reproducing all data is almost impossible.

N. Mounet et al. Nat Nanotech **13**, *246-52* (2018). doi: 10.1038/s41565-017-0035-5

AiiDA architecture

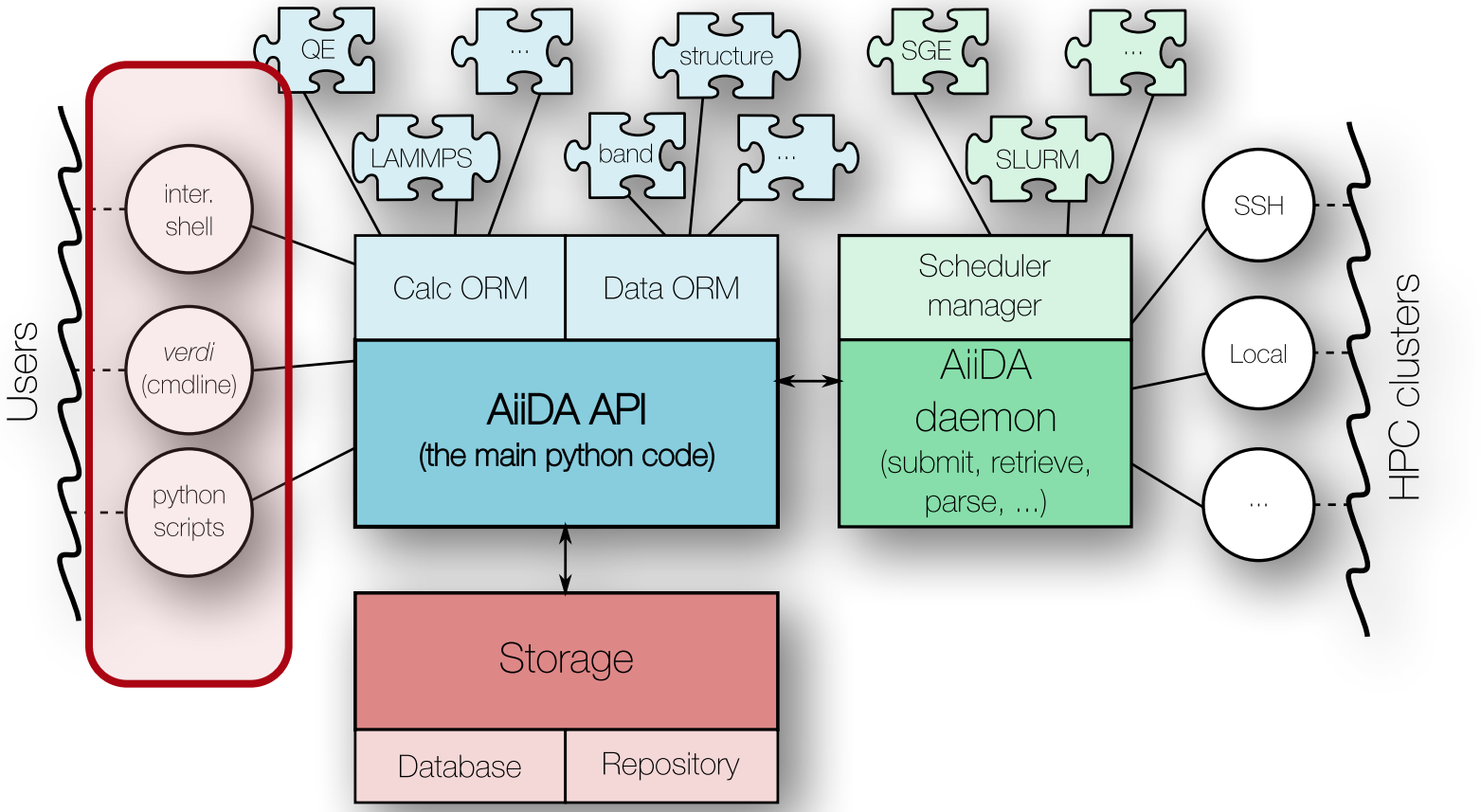


AiiDA architecture



1. The core: **AiiDA python API**

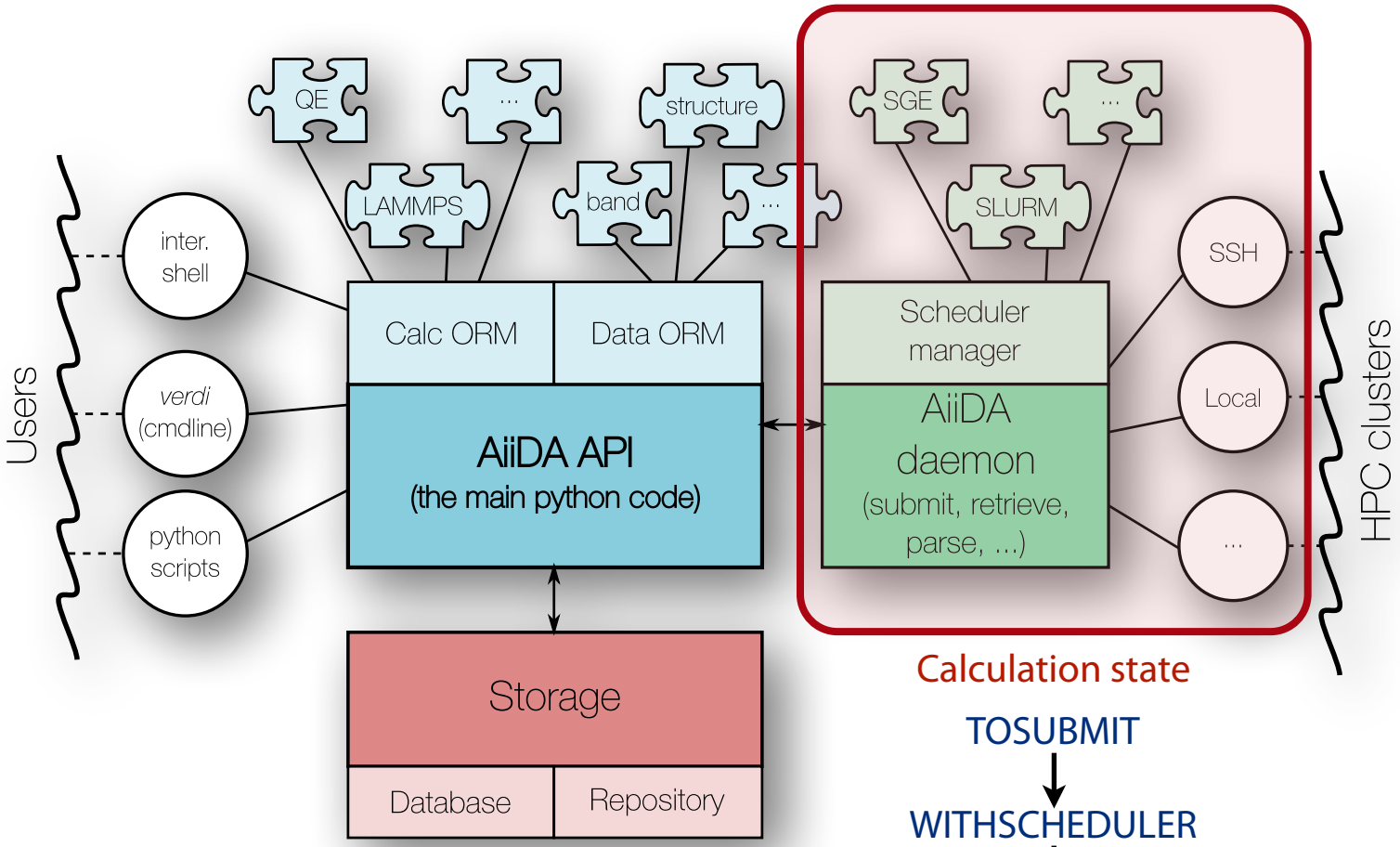
AiiDA architecture

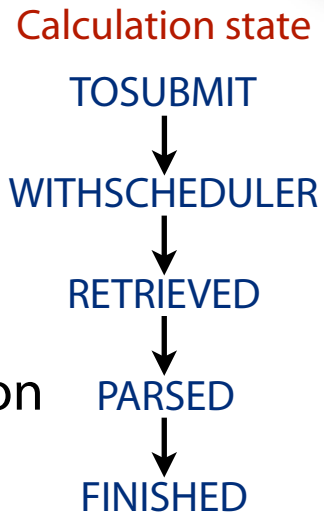


2. **User interface:**

python scripts, `verdi` command line tool, `verdi shell`

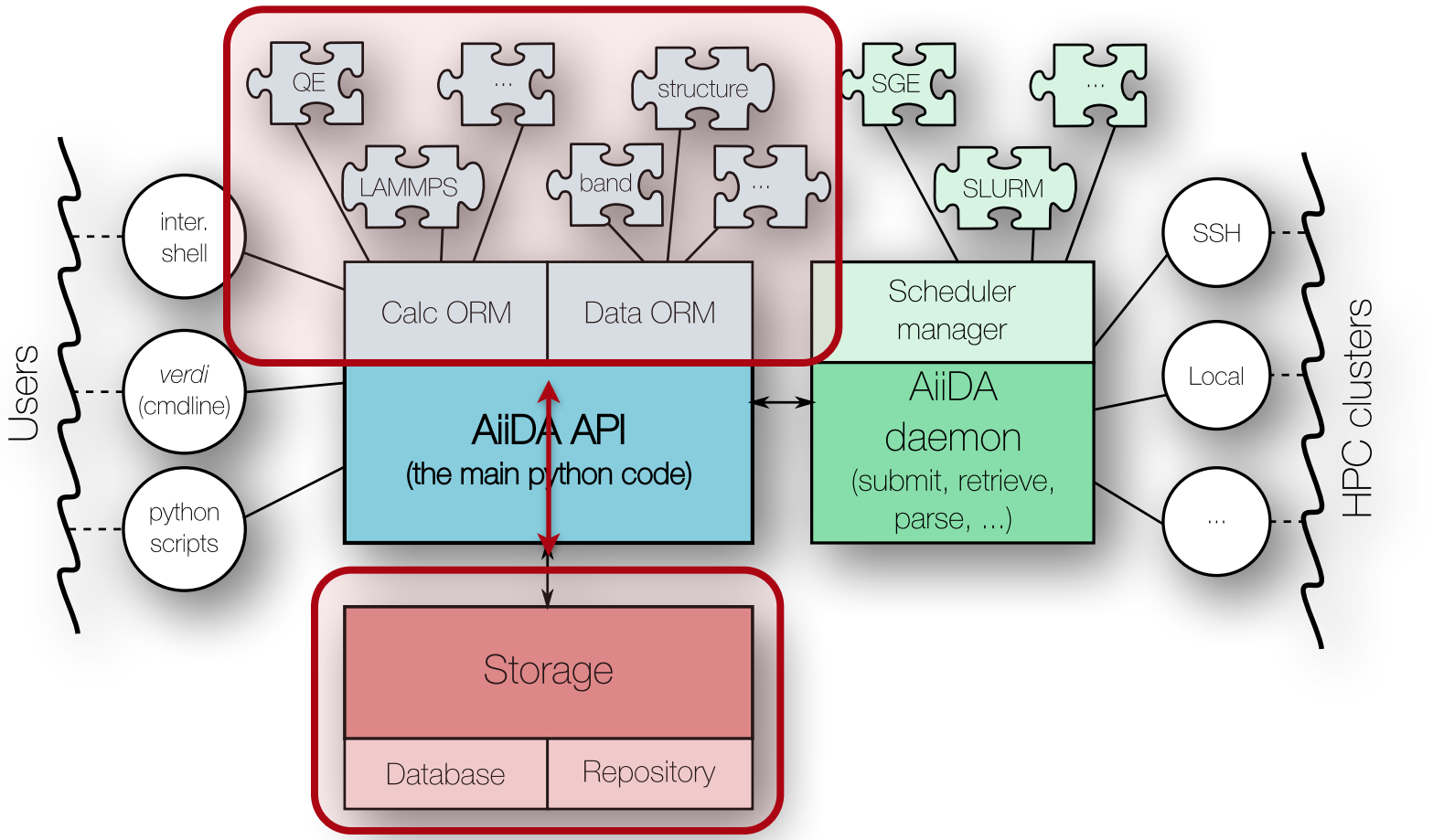
AiiDA architecture





3. AiiDA **daemon:** manage interaction with remote computers without user intervention

AiiDA architecture



4. **AiiDA Object-Relational Mapper** (ORM):
stores data, codes and calculations in local database

AiiDA: Calculation example

```
code = Code.get_from_string('pw-6.3@daint-mr25')
calc = code.new_calc()

calc.set_max_wallclock_seconds(600)
calc.set_resources({"num_machines": 2})

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

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

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

calc.use_structure(structure)
calc.use_parameters(parameters)
calc.use_kpoints(kpoints)
calc.use_pseudos_from_family('SSSP_efficiency_v1.0')
calc.store_all()

calc.submit()
```

AiiDA: Calculation example

```
code = Code.get_from_string('pw-6.3@daint-mr25')
calc = code.new_calc()

calc.set_max_wallclock_seconds(600)
calc.set_resources({"num_machines": 2})

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

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

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

calc.use_structure(structure)
calc.use_parameters(parameters)
calc.use_kpoints(kpoints)
calc.use_pseudos_from_family('SSSP_efficiency_v1.0')
calc.store_all()

calc.submit()
```

Switch computers in one line

supports different schedulers, version of codes, …

AiiDA: Calculation example

```
code = Code.get_from_string('pw-6.3@daint-mr25')
calc = code.new_calc()

calc.set_max_wallclock_seconds(600)
calc.set_resources({"num_machines": 2})

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

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

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

calc.use_structure(structure)
calc.use_parameters(parameters)
calc.use_kpoints(kpoints)
calc.use_pseudos_from_family('SSSP_efficiency_v1.0')
calc.store_all()

calc.submit()
```

Switch computers in one line

supports different schedulers, version of codes, …

Define (only) necessary inputs

Interface designed by plugin

AiiDA: Calculation example

```
code = Code.get_from_string('pw-6.3@daint-mr25')
calc = code.new_calc()

calc.set_max_wallclock_seconds(600)
calc.set_resources({"num_machines": 2})

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

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

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

calc.use_structure(structure)
calc.use_parameters(parameters)
calc.use_kpoints(kpoints)
calc.use_pseudos_from_family('SSSP_efficiency_v1.0')
calc.store_all()

calc.submit()
```

Switch computers in one line

supports different schedulers, version of codes, …

Define (only) necessary inputs

Interface designed by plugin

Inputs stored in the DB

AiiDA: Calculation example

```
code = Code.get_from_string('pw-6.3@daint-mr25')
calc = code.new_calc()

calc.set_max_wallclock_seconds(600)
calc.set_resources({"num_machines": 2})

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

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

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

calc.use_structure(structure)
calc.use_parameters(parameters)
calc.use_kpoints(kpoints)
calc.use_pseudos_from_family('SSSP_efficiency_v1.0')
calc.store_all()

calc.submit()
```

Switch computers in one line

supports different schedulers, version of codes, …

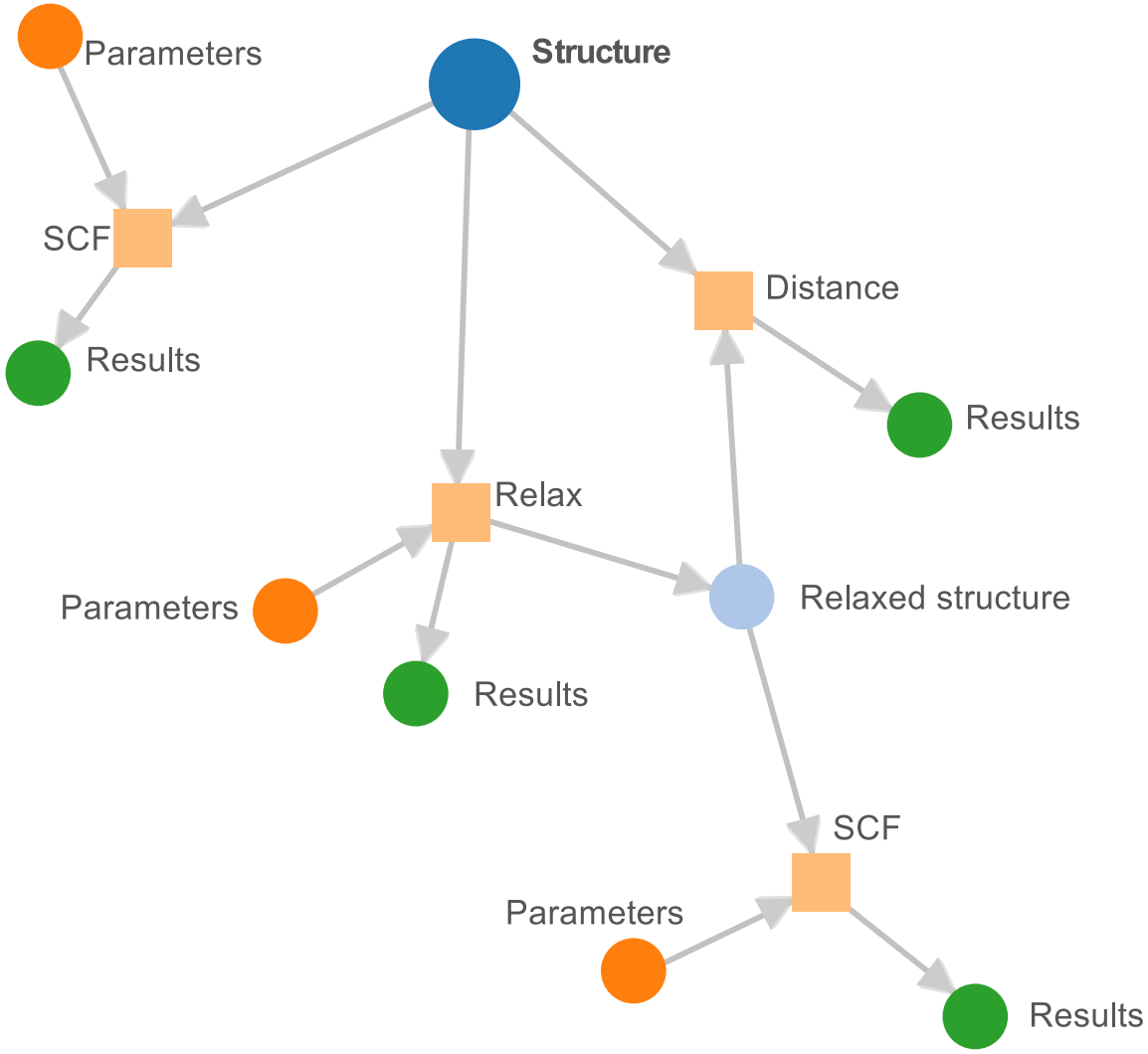
Define (only) necessary inputs

Interface designed by plugin

Inputs stored in the DB

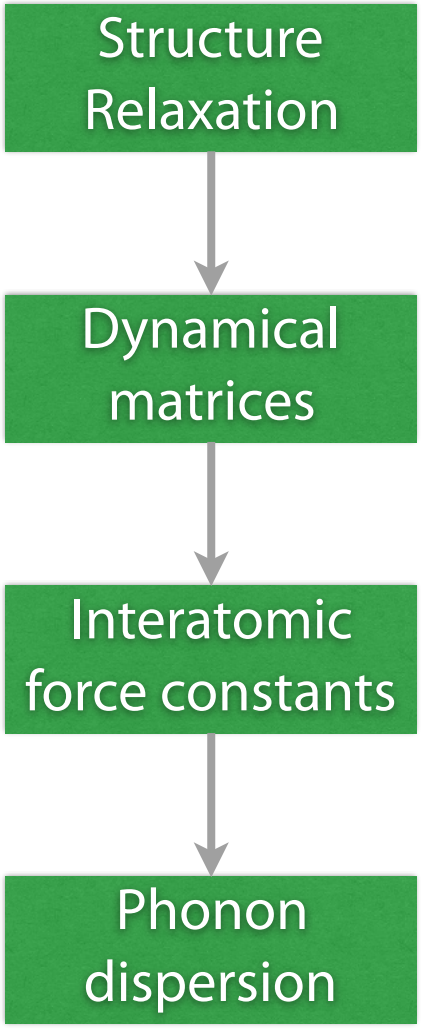
Handing over to the daemon

Data provenance: Directed Acyclic Graphs



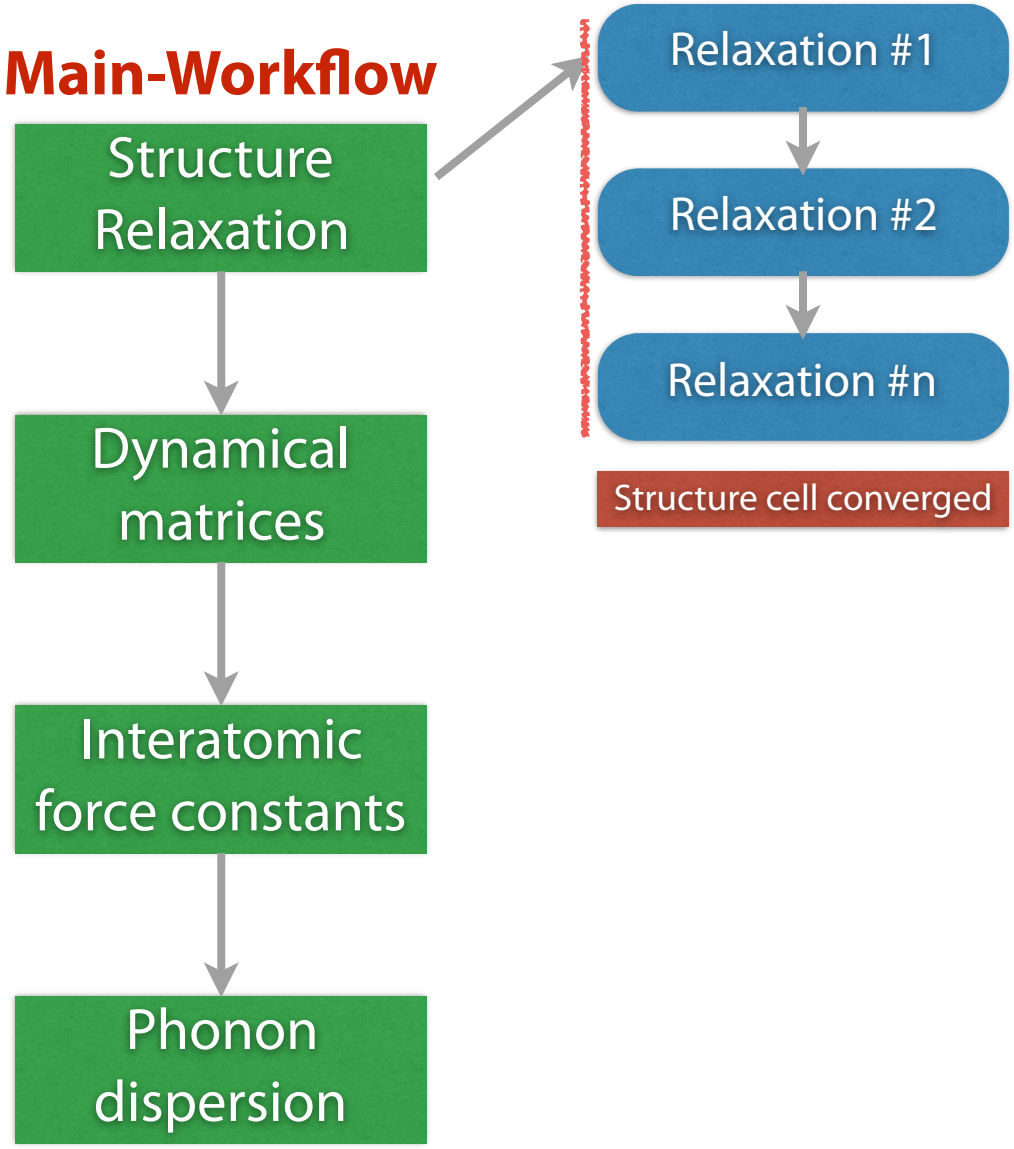
From calculations to workflows: **phonon dispersion**

Main-Workflow



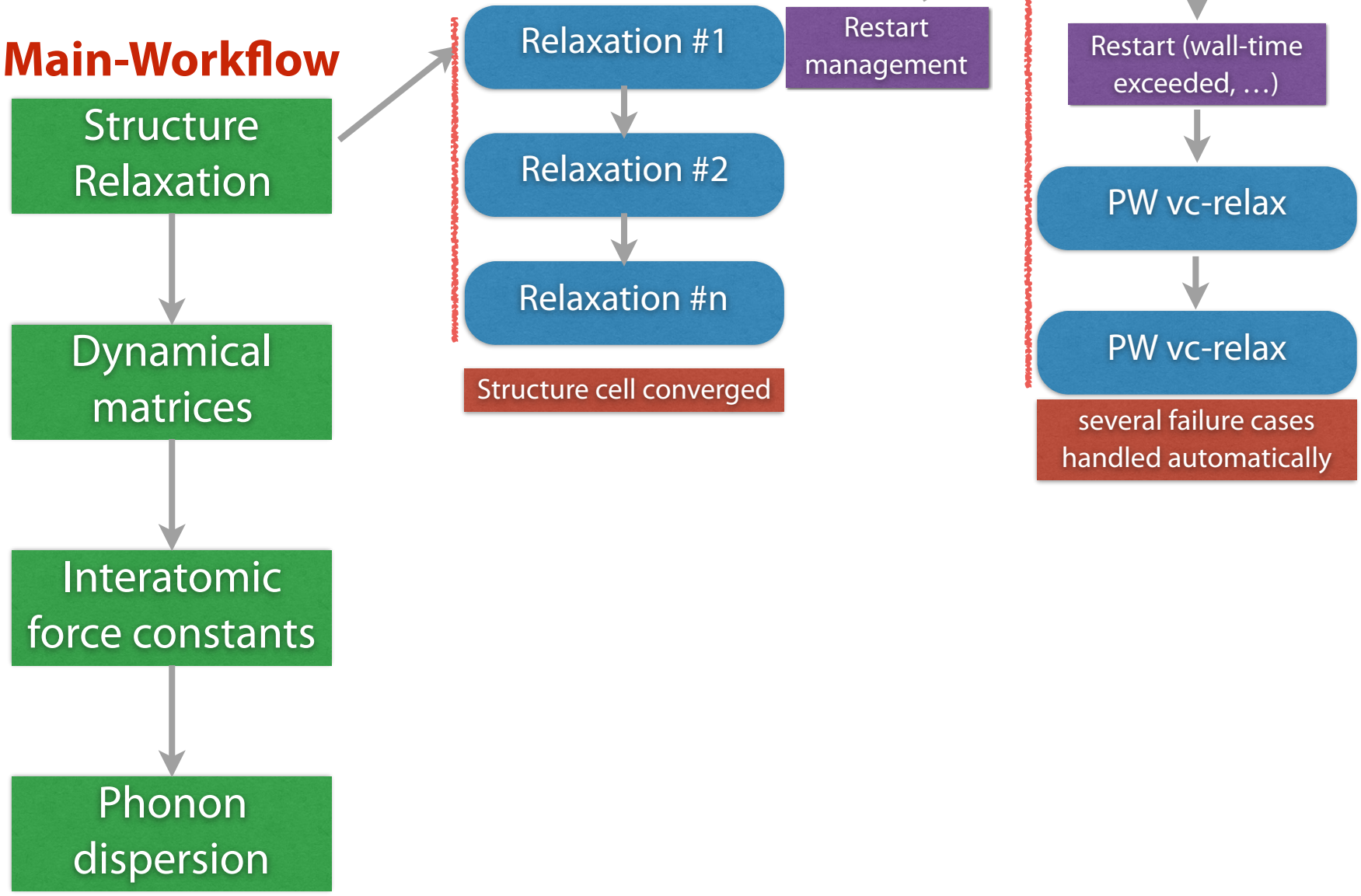
N. Mounet et al.

From calculations to workflows: **phonon dispersion**



N. Mounet et al.

From calculations to workflows: **phonon dispersion**



N. Mounet et al.

From calculations to workflows: **phonon dispersion**

N. Mounet et al.

Workflows = Encapsulation

```
params = {'input': {'kpoints_density': 0.2,
                    'convergence': 'tight'},
          'structure': structure,
          'pseudo_family': pseudo_family,
          'machinename': 'mycluster',
          'pw_input':{'volume_conv_threshold': 5e-2},
          'pw_parameters': {
             'SYSTEM': {'ecutwfc': 30.},
             'ELECTRONS': {'conv_thr': 1.e-10}}
          'ph_input':{
             'distance_kpoints_in_dispersion': 0.005,
             'diagonalization': 'cg'}
           }
future = submit(PhBandsWorkflow, **params)
```

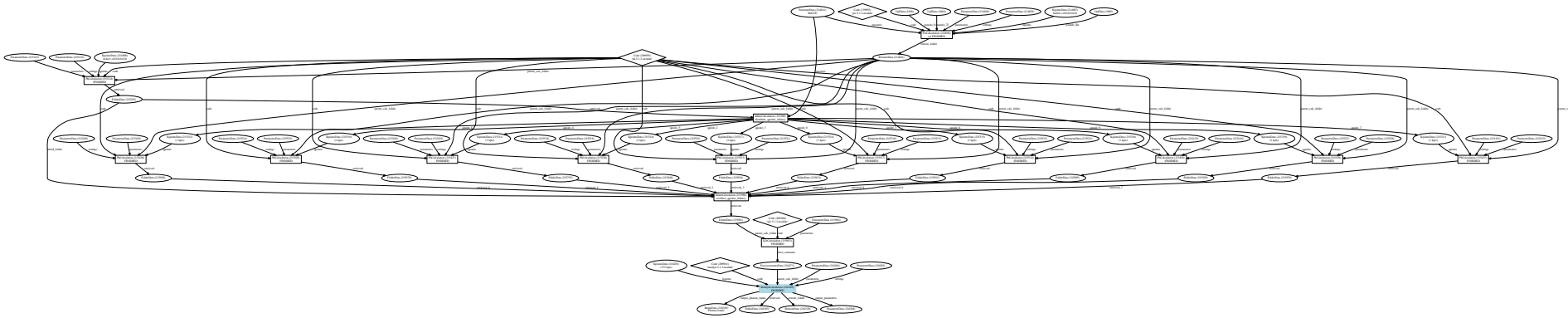
From minimal inputs ...

Workflows = Encapsulation

```
params = {'input': {'kpoints_density': 0.2,
                    'convergence': 'tight'},
          'structure': structure,
          'pseudo_family': pseudo_family,
          'machinename': 'mycluster',
          'pw_input':{'volume_conv_threshold': 5e-2},
          'pw_parameters': {
             'SYSTEM': {'ecutwfc': 30.},
             'ELECTRONS': {'conv_thr': 1.e-10}}
          'ph_input':{
             'distance_kpoints_in_dispersion': 0.005,
             'diagonalization': 'cg'}
           }
future = submit(PhBandsWorkflow, **params)
```

From minimal inputs ...

... to complex workflows



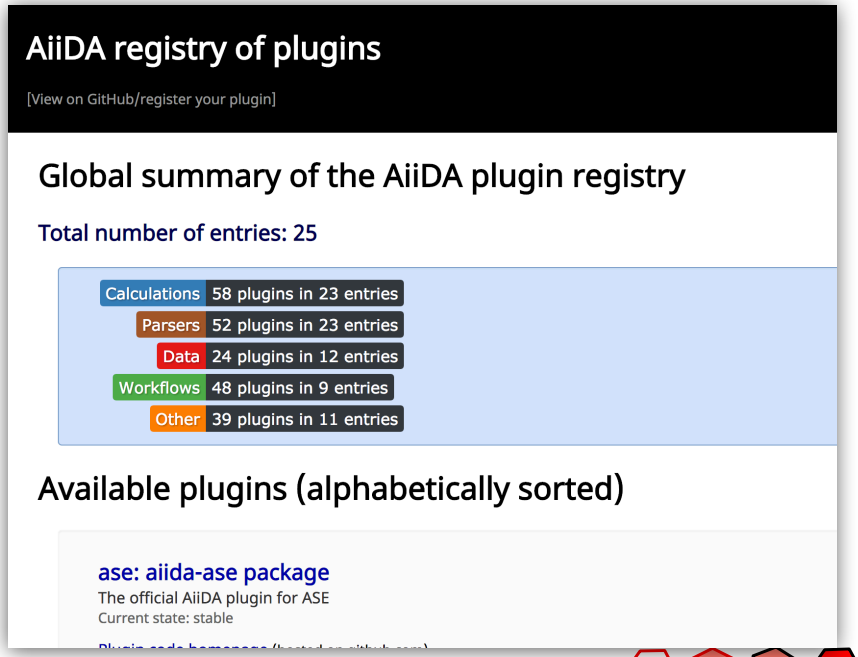
AiiDA – Development

- **Free and open source (MIT License)**
- github.com/aiidateam/aiida_core
- 36 releases, latest stable version: v0.12.2
- 40 contributors since 2012

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

AiiDA plugins

ase, castep, codtools, cp2k, crystal17, fleur, gollum, kkr, lammps, nwchem, phonopy, quantumespresso, raspa, siesta, vasp, wannier90, yambo, zeo++, and more ... (+plugin template)



❷

MATERIALS
CLOUD

AiiDAlab

Motivation

Computational Materials Science Challenges

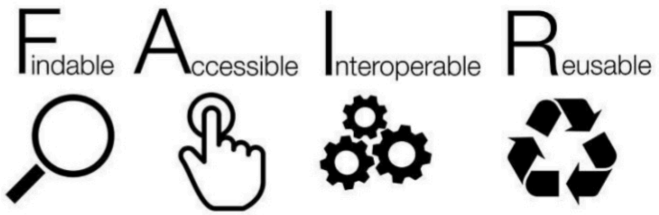
High-Throughput

Reproducibility

Open Science

Knowledge Transfer

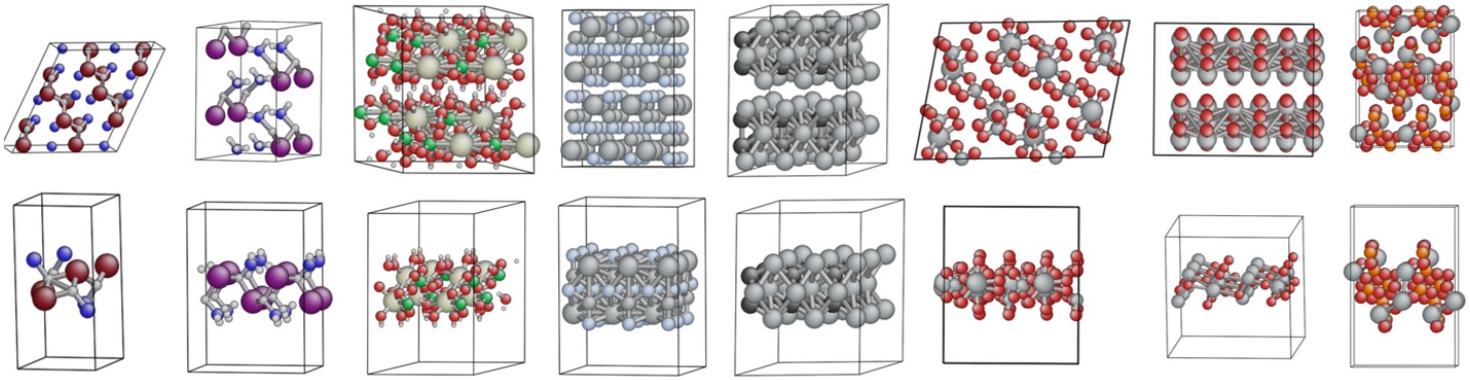
- Open Science
  - ~~Supporting Information~~
  - ~~Just upload everything~~
  - FAIR data
- Making data FAIR is hard, can we make it easier?



Source: Prof. Michel Dumontier

High-throughput Example

Discovering new two-dimensional materials



STARTING FROM ICSD/COD DATABASE:

- **108 423** unique 3D structures
- **5619** layered structures
- >**100 000** DFT calculations
- >**30 000** material properties
- >**1·10^9** attributes

III. METHODS

A. Reproducibility and provenance

It is an often repeated tenet that results of scientific research must be reproducible. This objective is, however, challenging, especially in high-throughput research, due to the large number of simulations involved and the complex sequence of logical steps needed in the study. To ensure reproducibility we use AiiDA[30] as a materials' informatics infrastructure to implement the ADES model of automation, data, environment, and sharing, as dis-

As already mentioned in the main text, in this step we consider only structures coming from experimental measurements, while the source databases are partially populated by purely theoretical structures. This is done using the flags set on the database entries by their curators. We also implemented some heuristics to detect clearly wrong CIF files. For example, we discard structures where the chemical formula provided in the file is inconsistent with the elements in the unit cell. Regardless of all these efforts, it is possible that some incomplete or incorrect structures are still not filtered out from the original databases.

Methods: Impossible to describe every detail

For **authors**, reproducing all data is challenging.
For **peers**, reproducing all data is almost impossible.

N. Mounet et al. Nat Nanotech **13**, *246-52* (2018). doi: 10.1038/s41565-017-0035-5

FAIR sharing example

ARTICLES
https://doi.org/10.1038/s41565-017-0035-5
nature nanotechnology

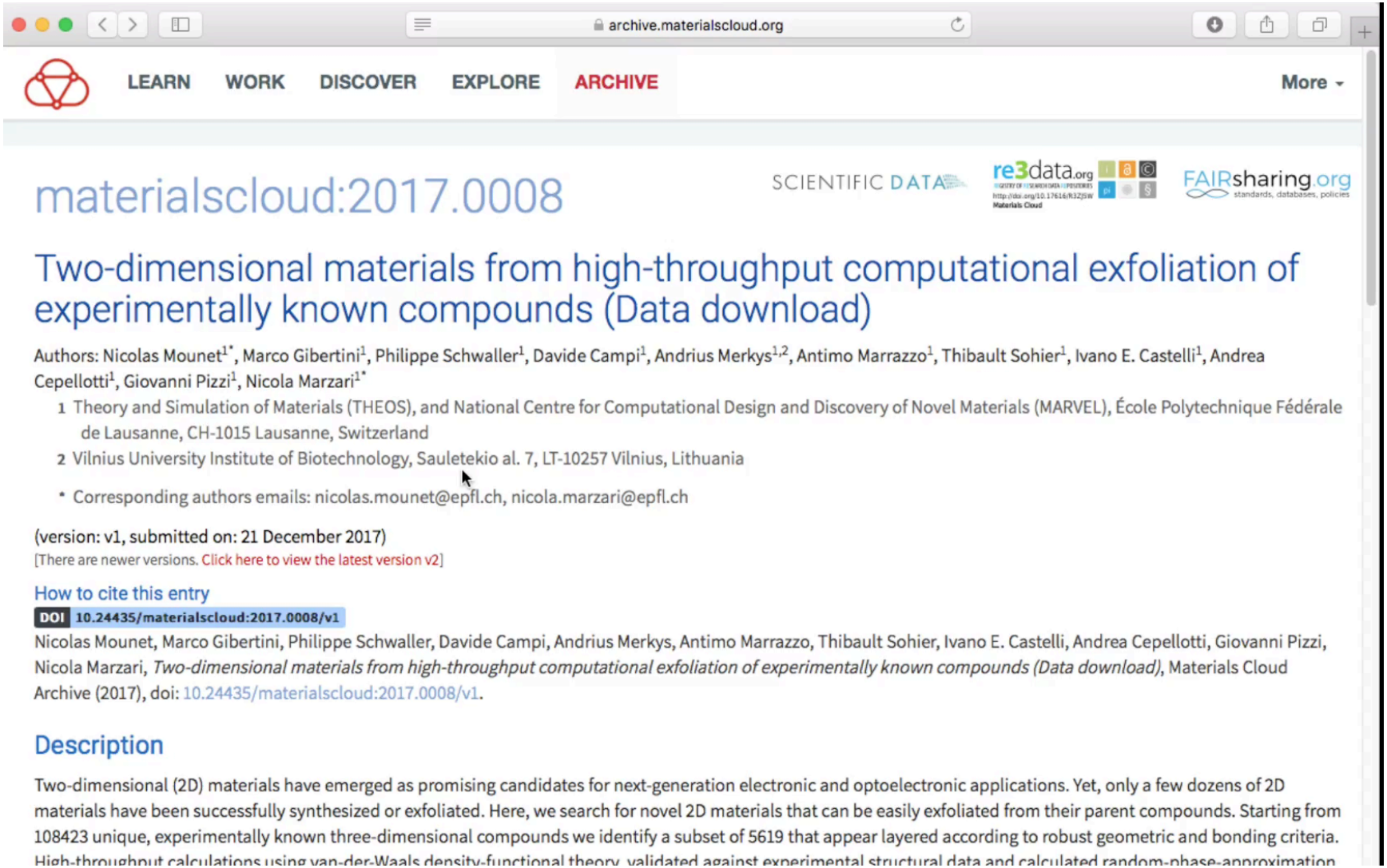
Two-dimensional materials from high-throughput computational exfoliation of experimentally known compounds

order. All results can be fully reproduced thanks to the deployment of the AiiDA infrastructure that tracks the provenance of each data entry. To date, this is the largest available database of 2D compounds and it is available online on the Materials Cloud platform[51].

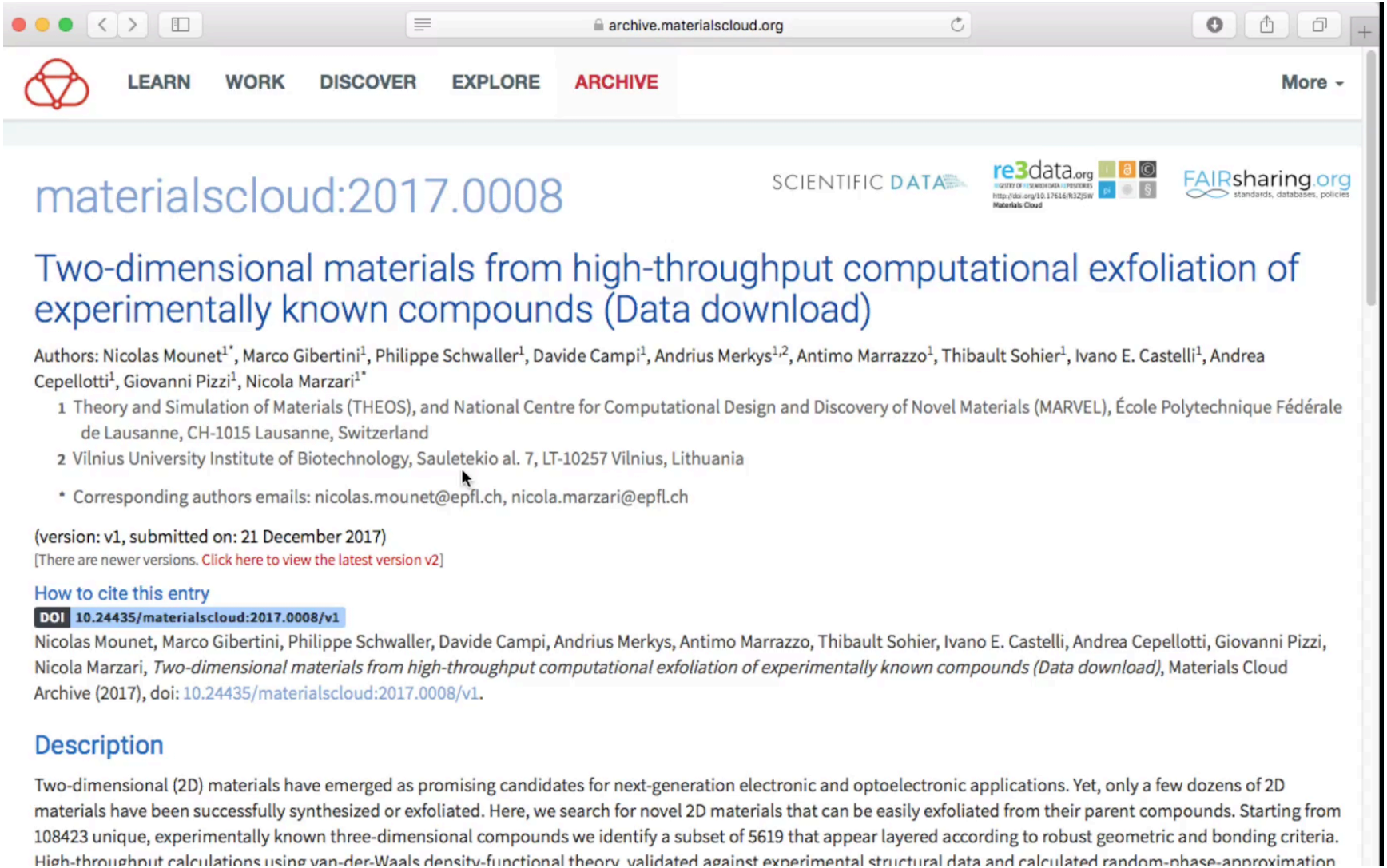
51. Mounet, N. et al. Two-dimensional materials from high-throughput computational exfoliation of experimentally known compounds (data download). *Materials Cloud Archive* (2017); https://doi.org/10.24435/materialscloud:2017.0008/v1

https://archive.materialscloud.org/2017.0008/v1

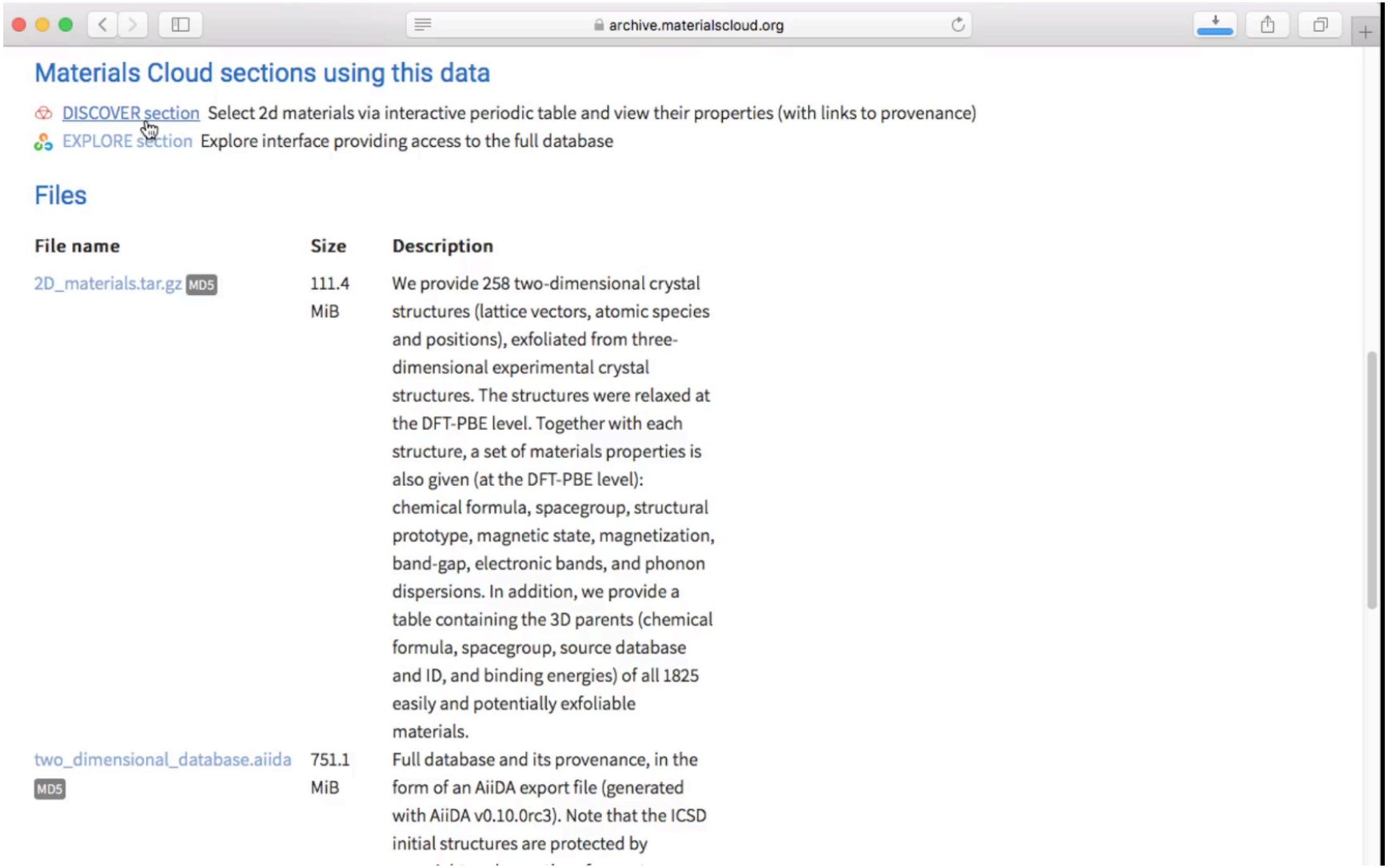
FAIR sharing example



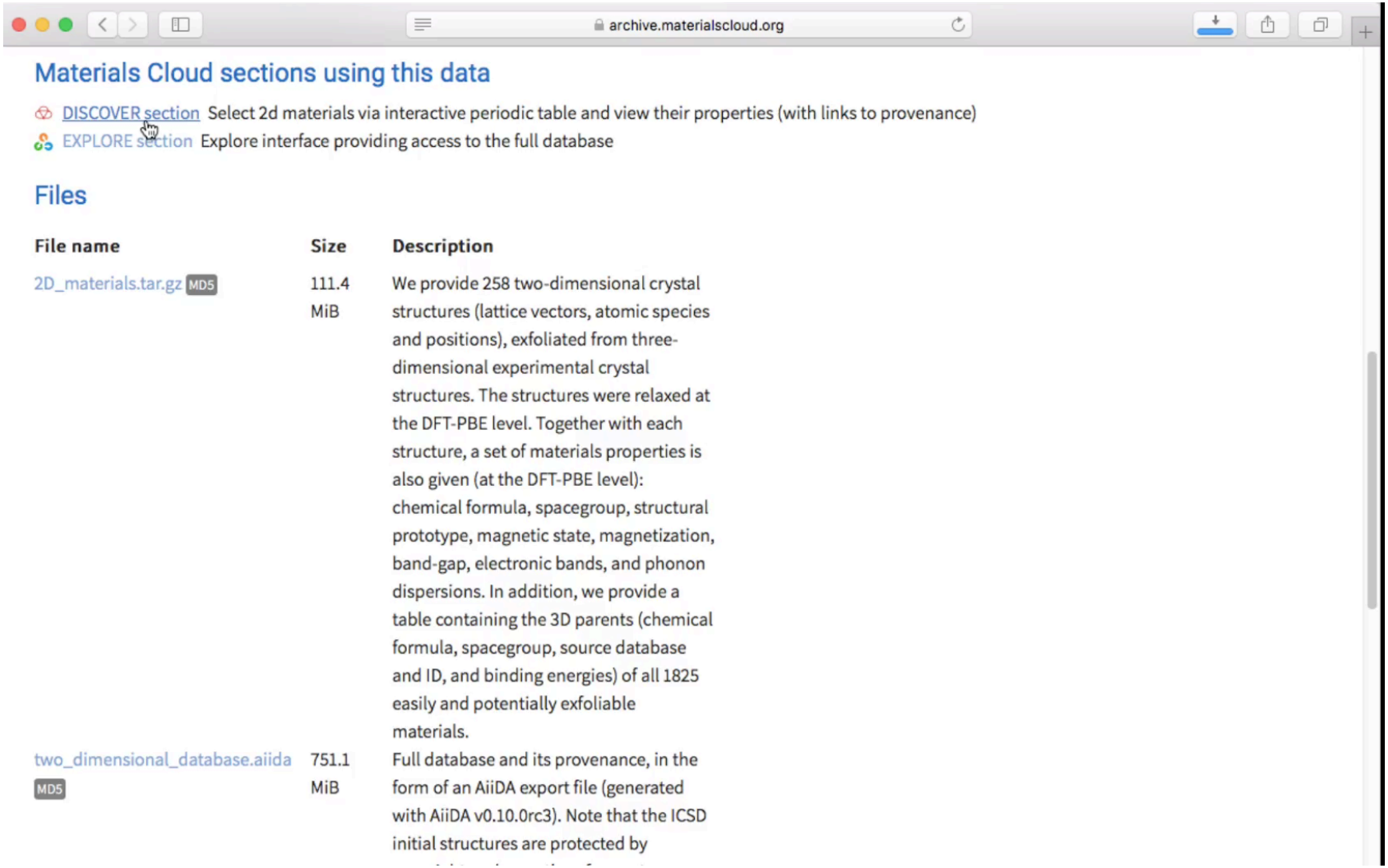
FAIR sharing example



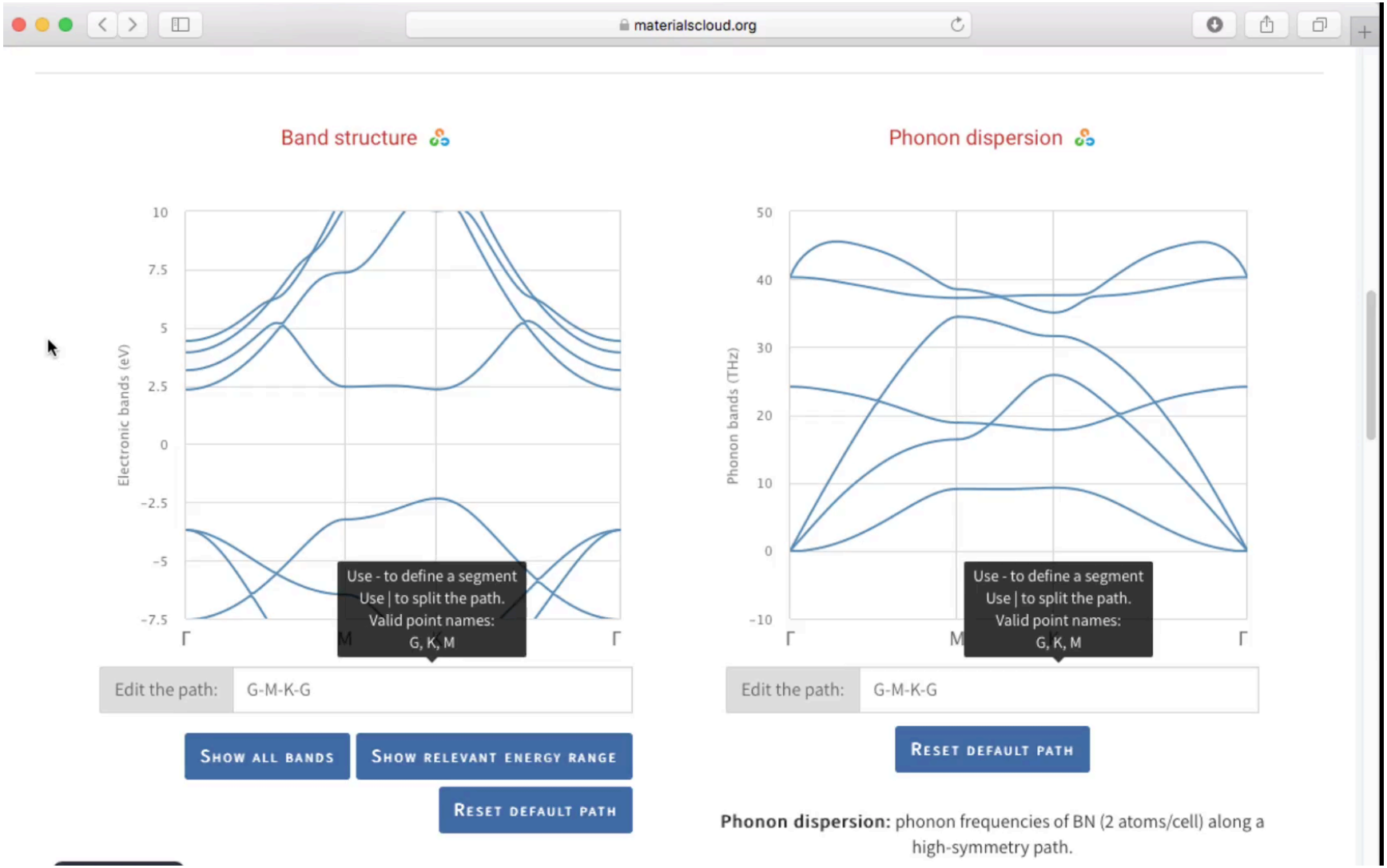
FAIR sharing example



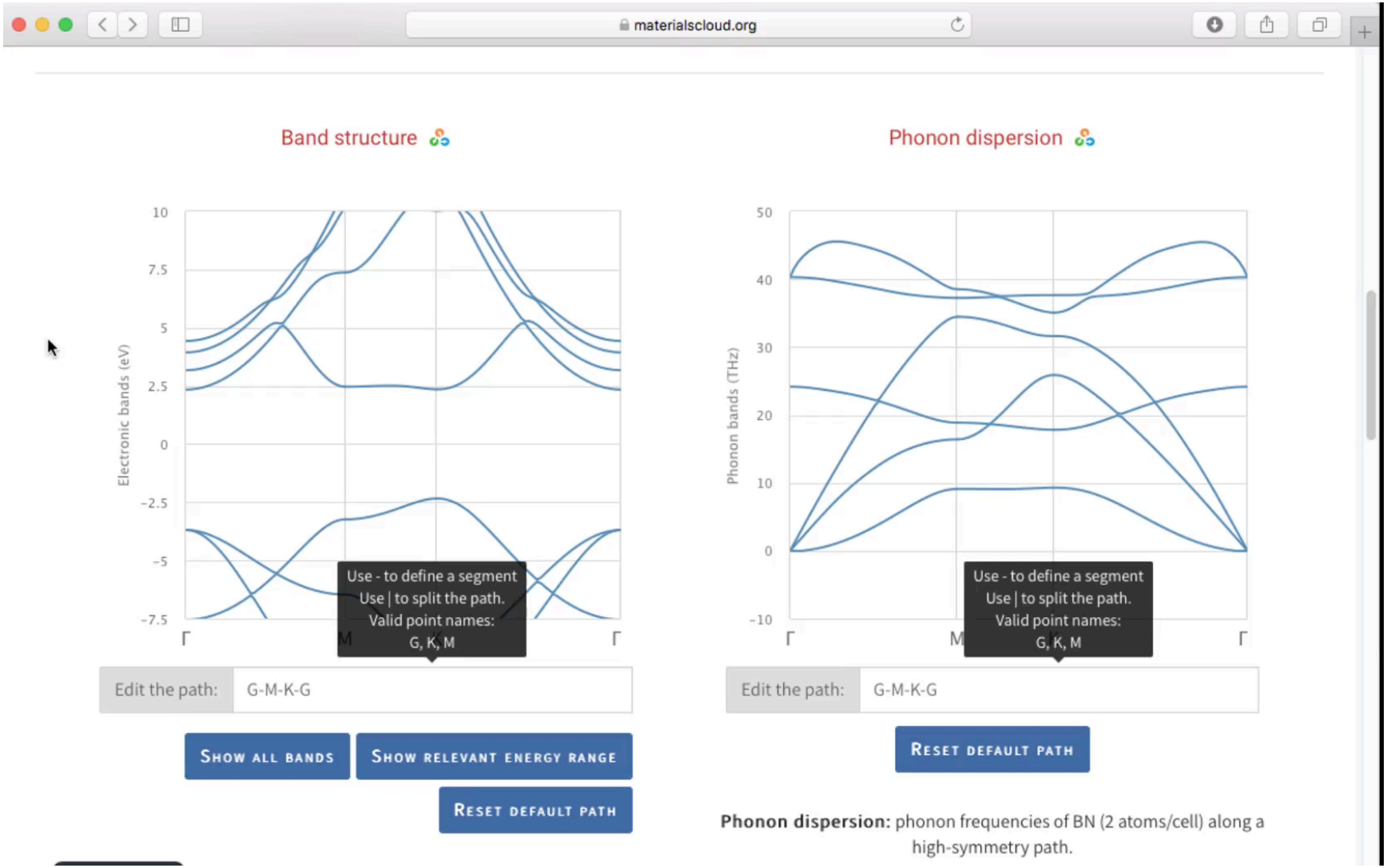
FAIR sharing example



FAIR sharing example

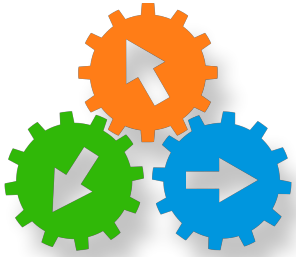


FAIR sharing example



❷

MATERIALS
CLOUD



AiiDAlab

Motivation

Computational Materials Science Challenges

High-Throughput

Reproducibility

Open Science

Knowledge Transfer

- How to transfer your insights & expertise, e.g.
  - reporting data sets to an experimental group
  - providing a workflow for your code to a collaborator/ company
  - ...

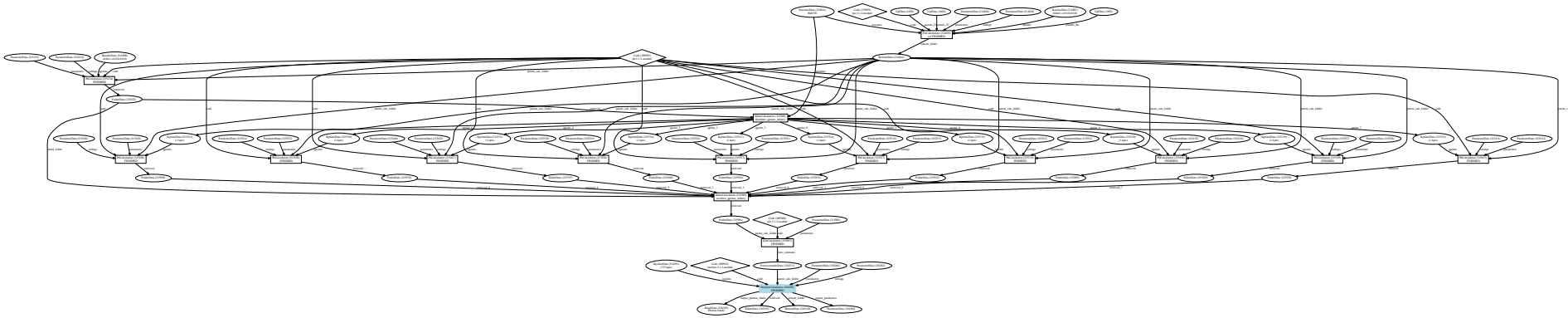
Source: quote.ucsd.edu

Workflows = Encapsulation

```
params = {'input': {'kpoints_density': 0.2,
                    'convergence': 'tight'},
          'structure': structure,
          'pseudo_family': pseudo_family,
          'machinename': 'mycluster',
          'pw_input':{'volume_conv_threshold': 5e-2},
          'pw_parameters': {
             'SYSTEM': {'ecutwfc': 30.},
             'ELECTRONS': {'conv_thr': 1.e-10}}
          'ph_input':{
             'distance_kpoints_in_dispersion': 0.005,
             'diagonalization': 'cg'}
           }
future = submit(PhBandsWorkflow, **params)
```

From minimal inputs ...

... to complex workflows



User base

User	Skills	Goals	Solution

User base

User	Skills	Goals	Solution
Computational Scientist	Knows Unix, bash, python	• run high-throughput calculations • write complex workflows • develop AiiDA plugins	**AiiDA** on the laptop

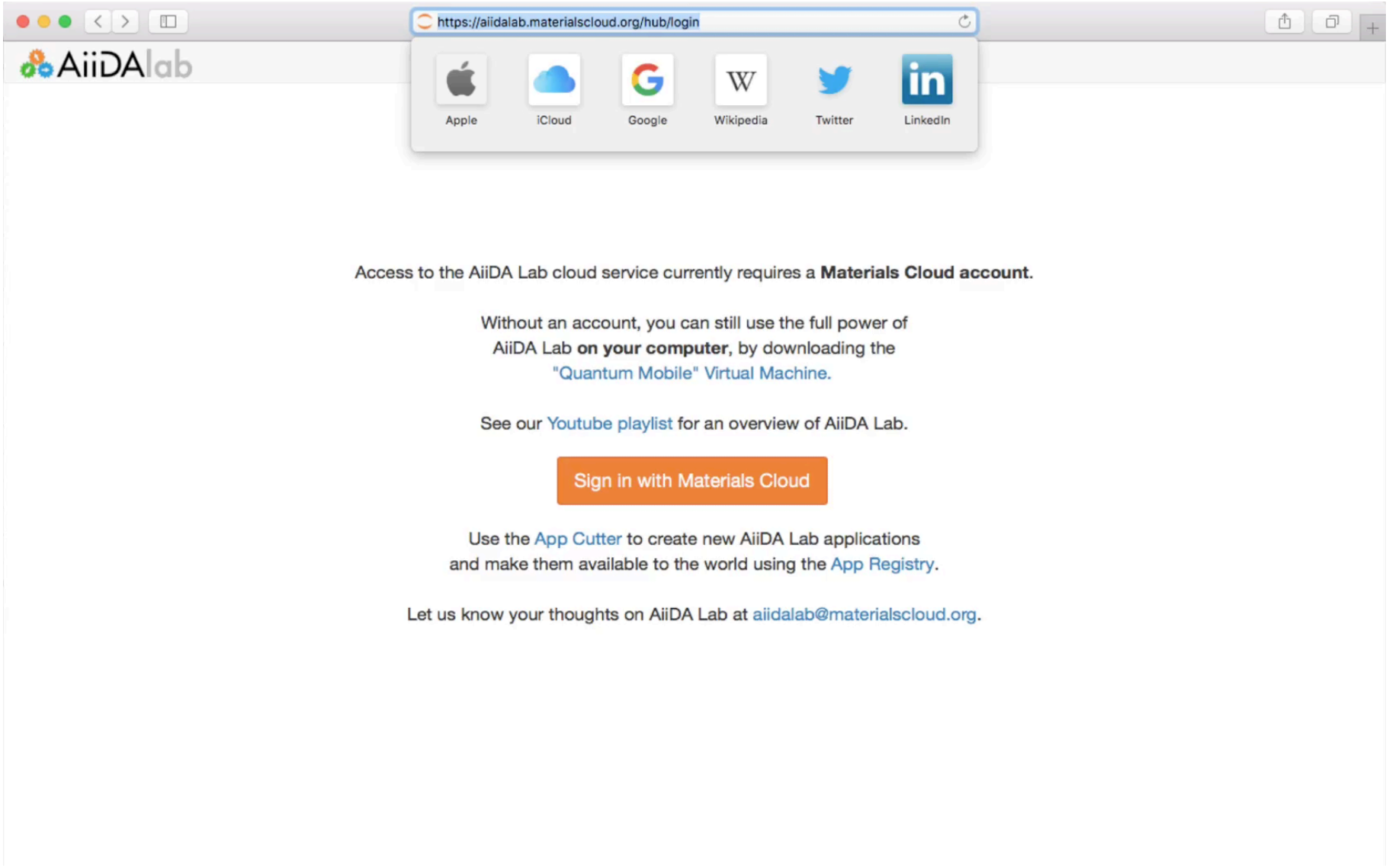
User base

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

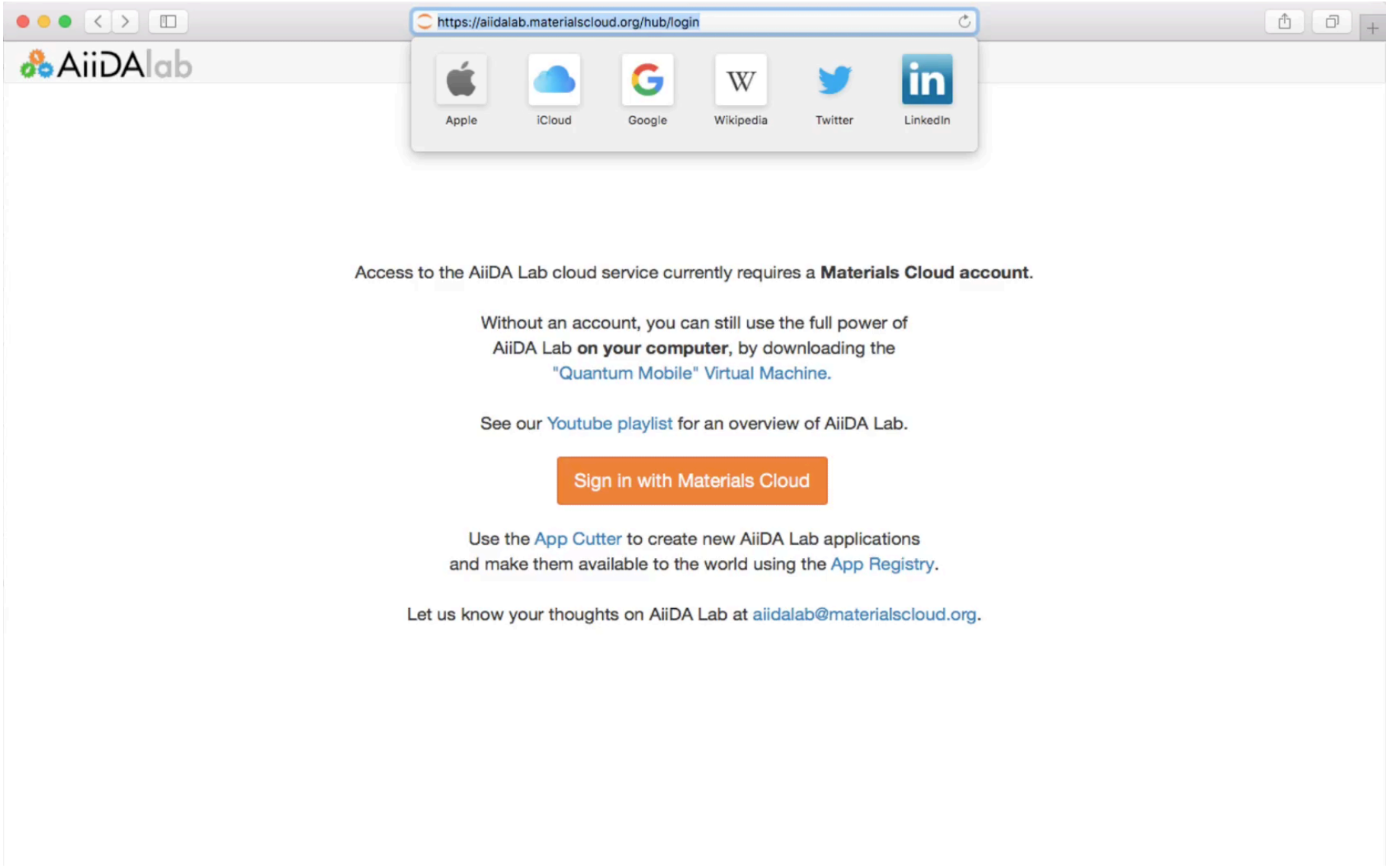
User base

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

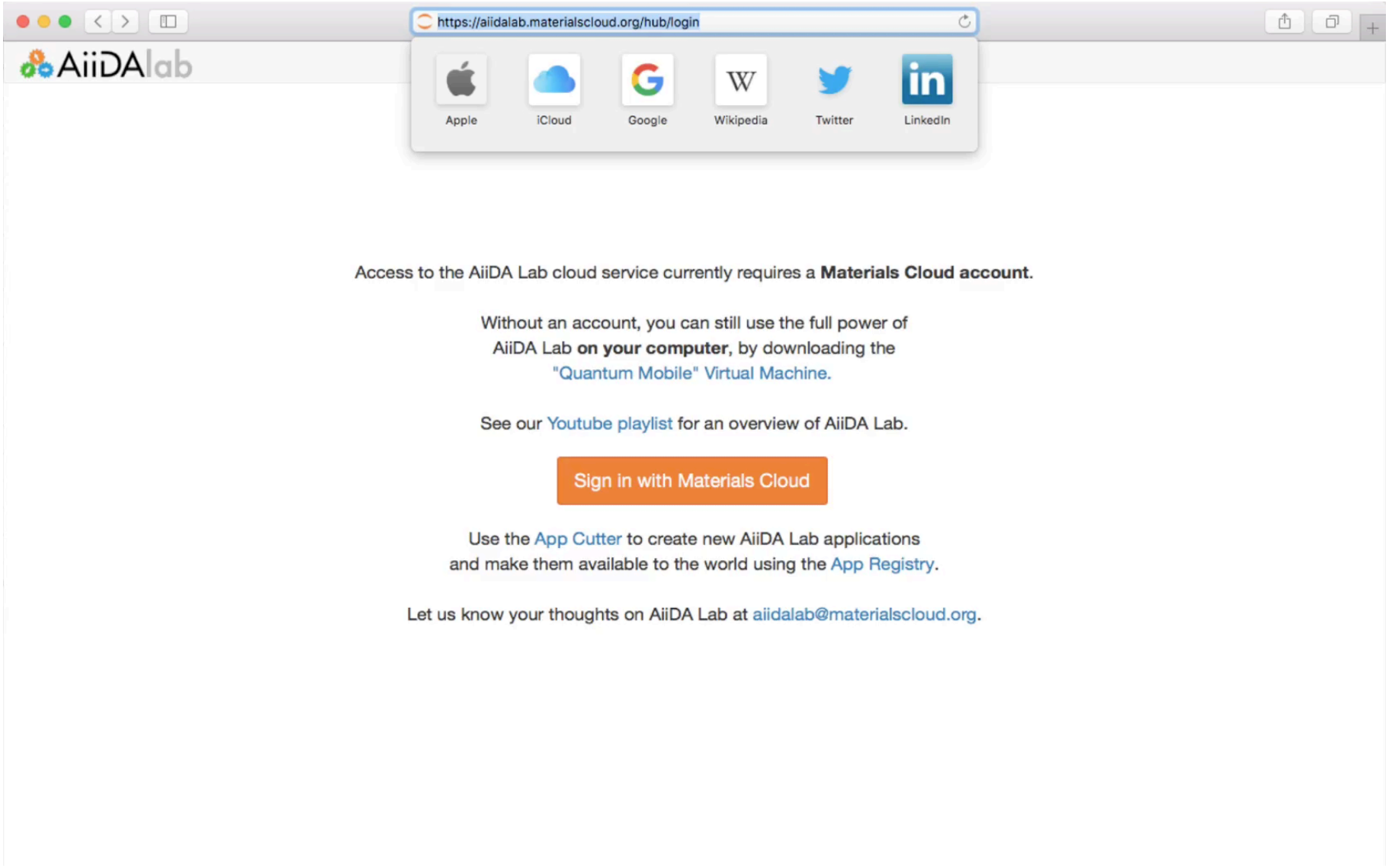
AiiDA Lab demo - login



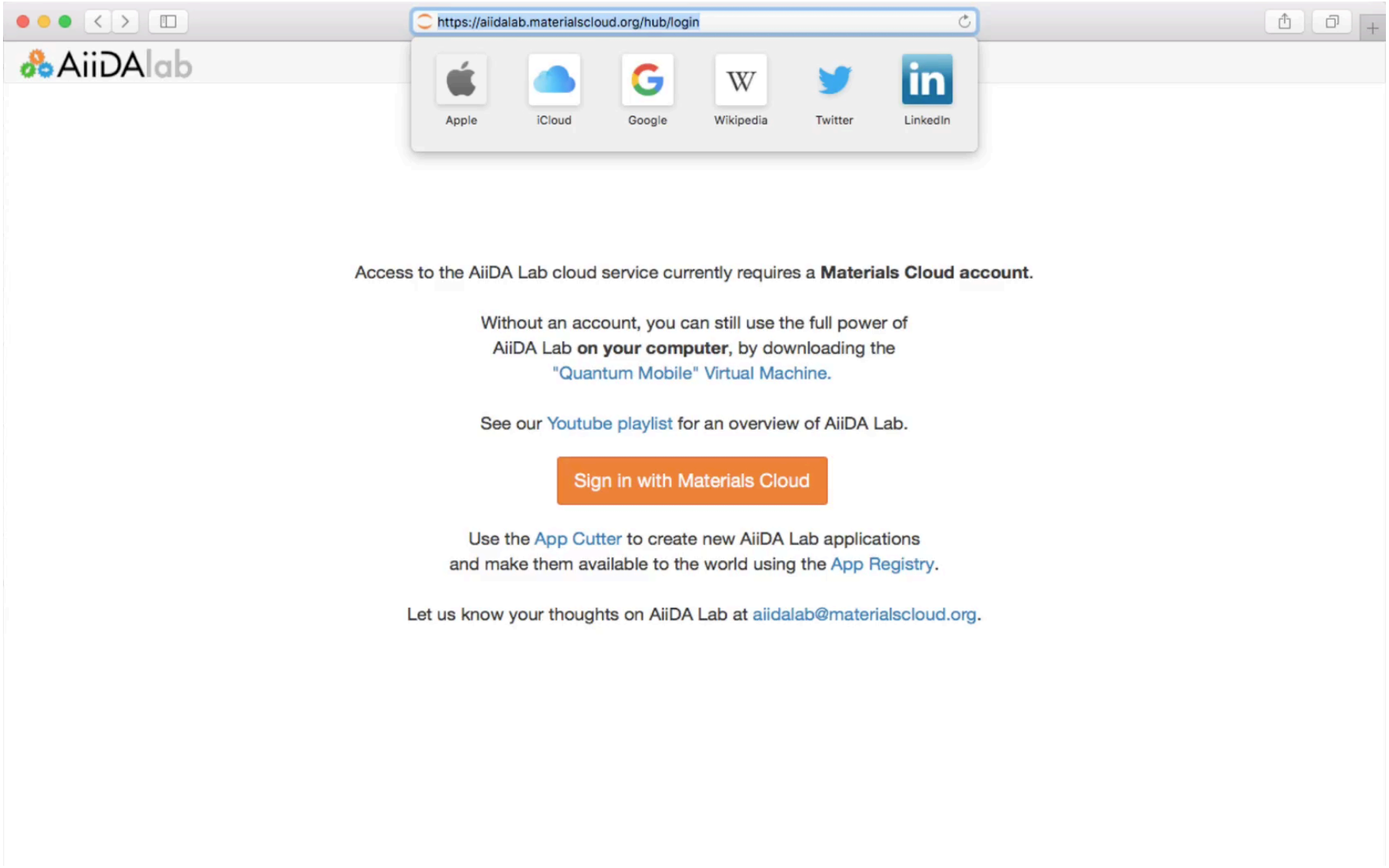
AiiDA Lab demo - login



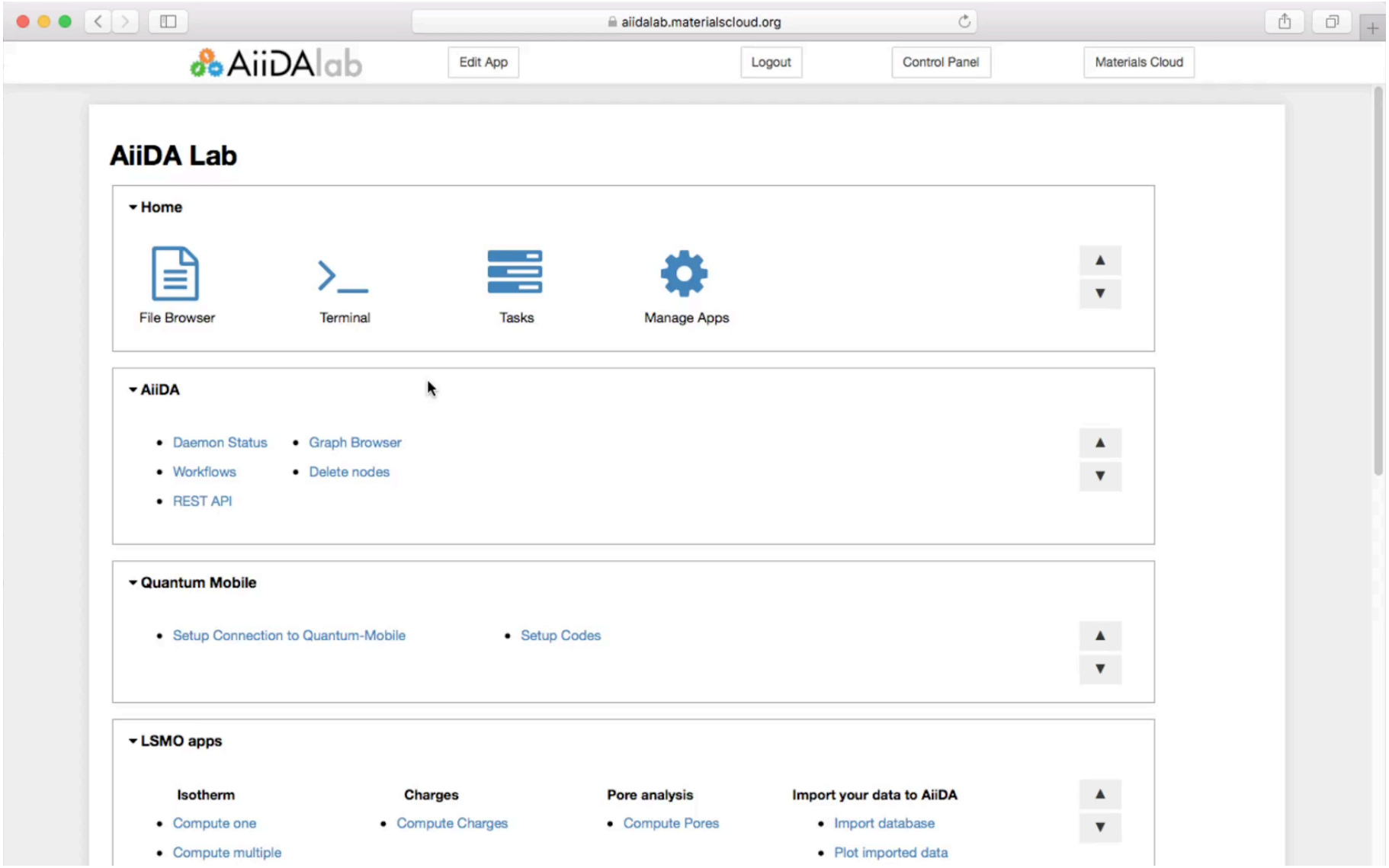
AiiDA Lab demo - login



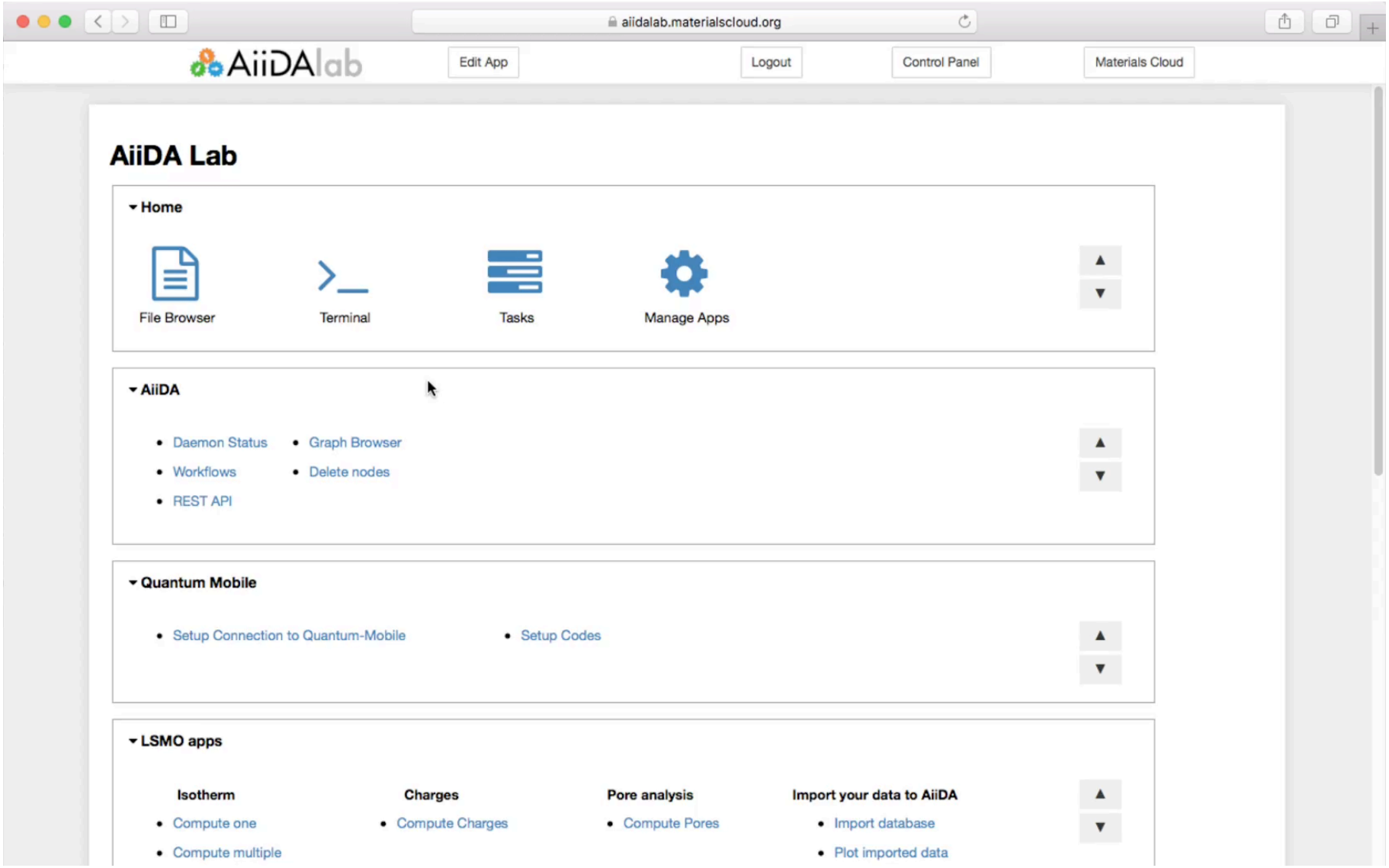
AiiDA Lab demo - login



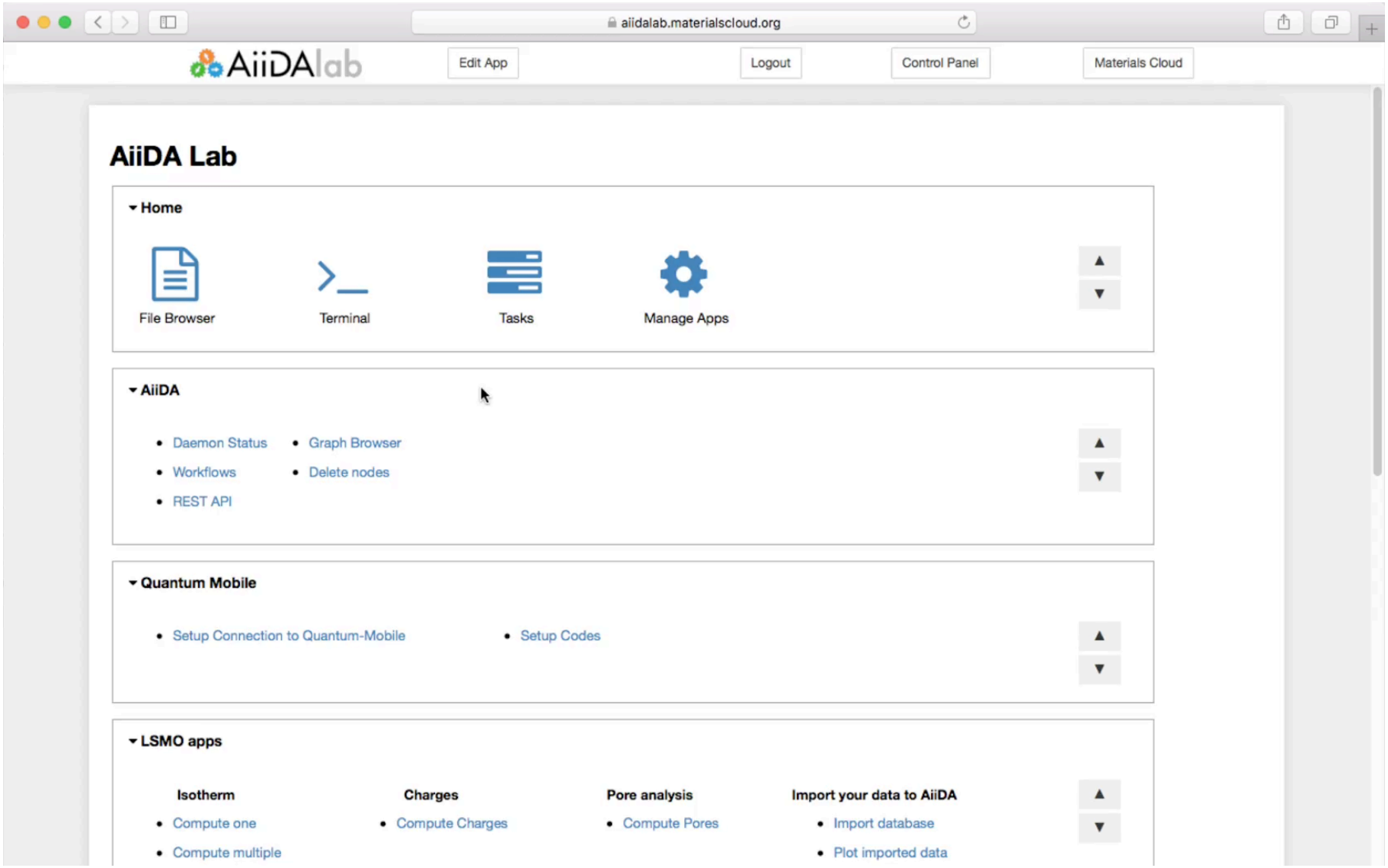
AiiDA Lab demo - appmode



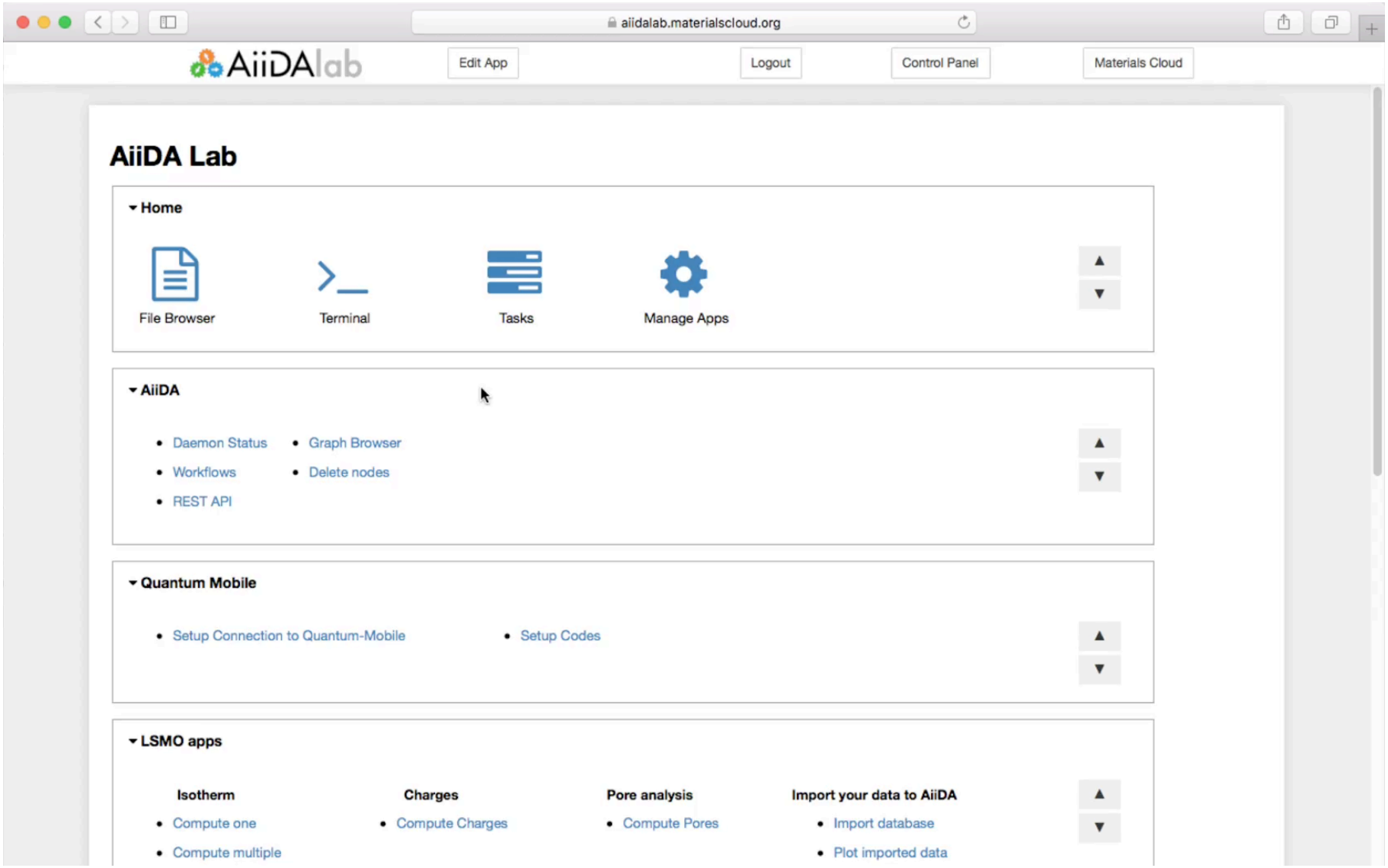
AiiDA Lab demo - appmode



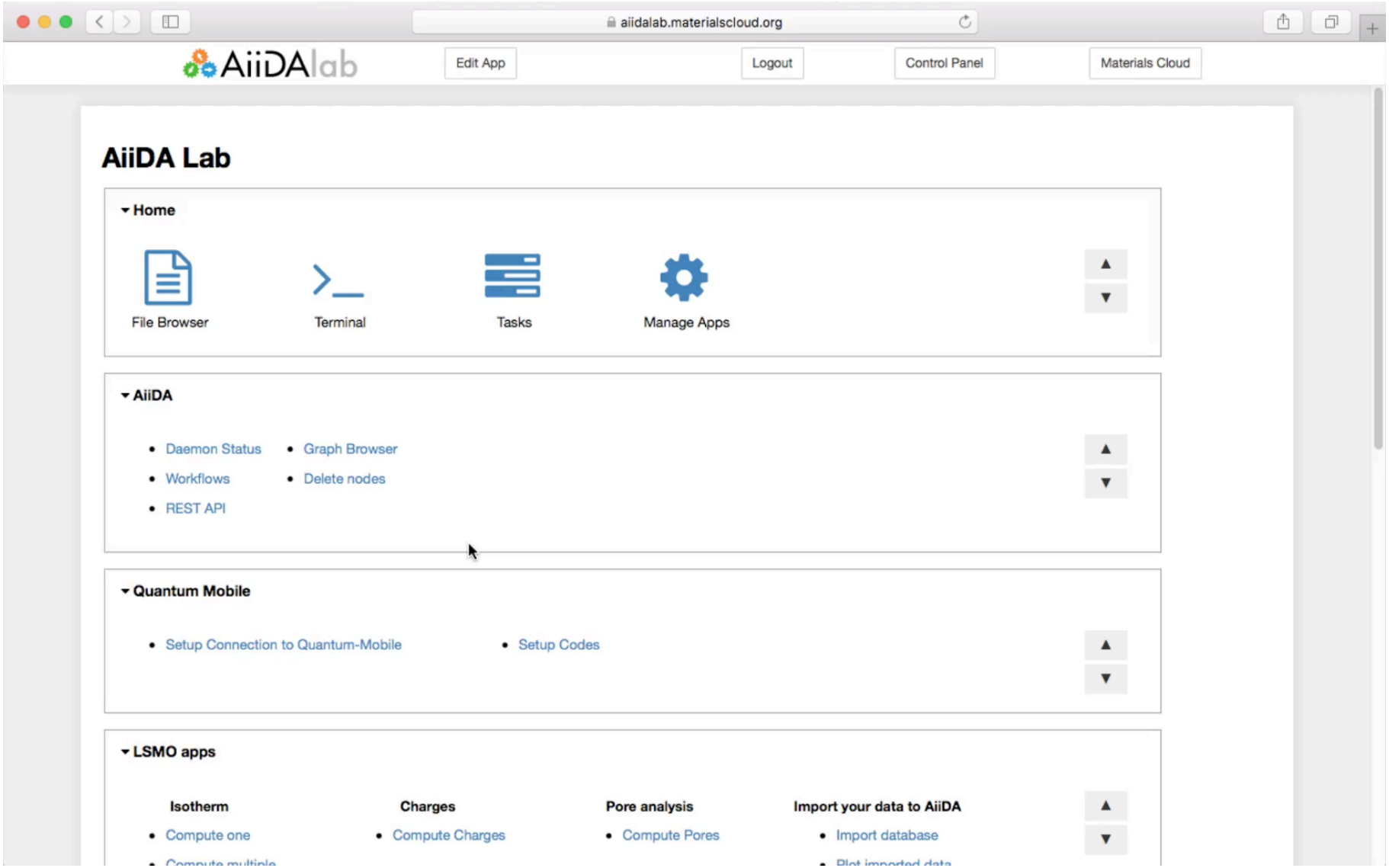
AiiDA Lab demo - terminal



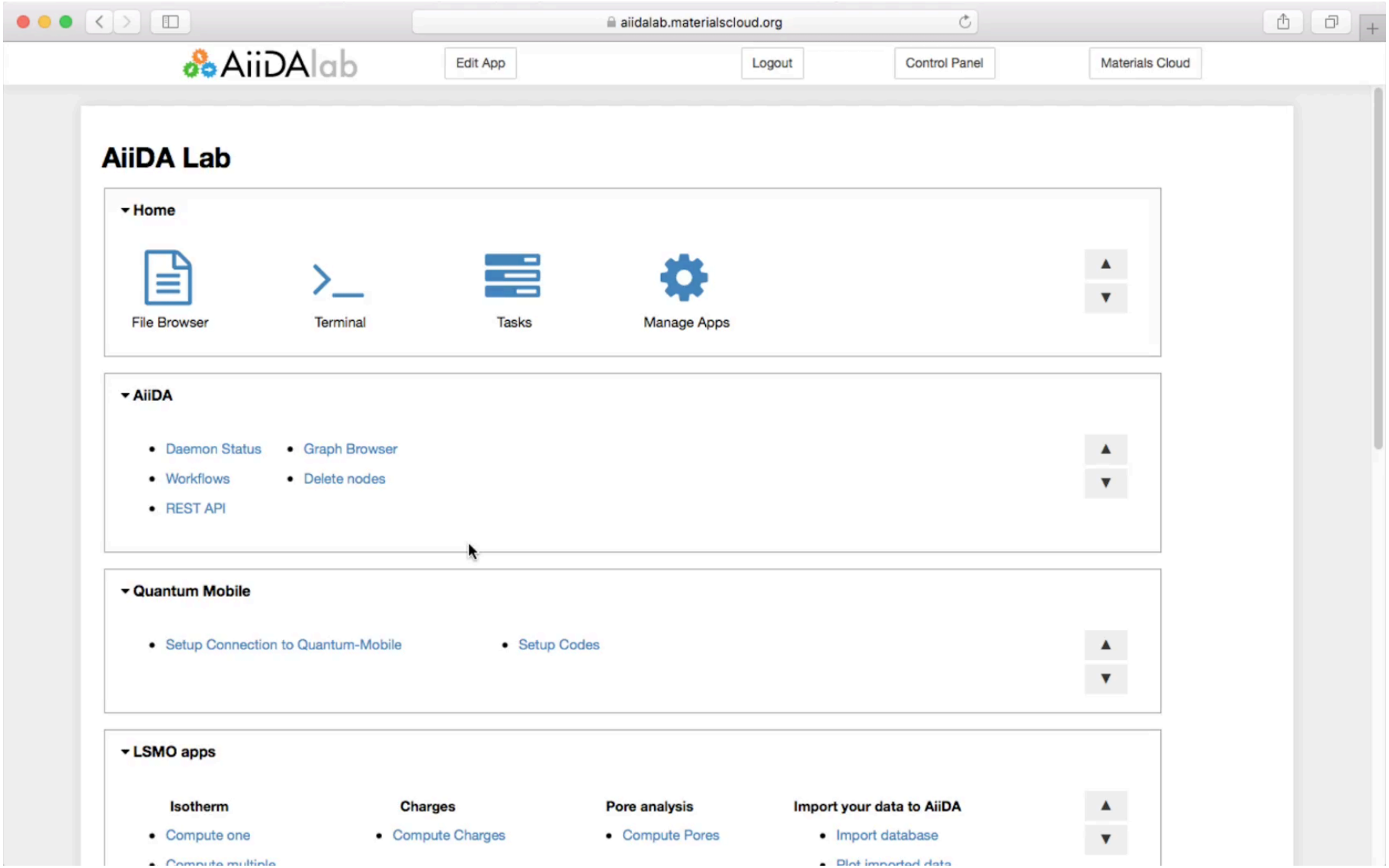
AiiDA Lab demo - terminal



AiiDA Lab demo - phonon submit



AiiDA Lab demo - phonon submit



AiiDA Lab demo - phonon results

aiidalab.materialscloud.org

.start-15 | .phonon turnkey solution-14

AiiDAlab | Edit App | Logout | Control Panel | Materials Cloud

```
{'automatic_parallelization': {'max_num_machines': 2,
  'max_wall_time_seconds': 86400,
  'target_time_seconds': 3600},
 'cutoff_factor': 1.0,
 'deg': 0.02,
 'do_bands': True,
 'do_mag': False,
 'do_phonons': True,
 'kpts_dist': 0.2,
 'matdyn_codename': 'matdyn-6.1@daint-s761',
 'mixing_beta': 0.7,
 'ph_codename': 'ph-6.1@daint-s746',
 'psfam': 'SSSP_eff_PBE',
 'pw_codename': 'pw-6.1@daint-s746',
 'q2r_codename': 'q2r-6.1@daint-s761',
 'relaxation_scheme': 'vc-relax',
 'smear': 'cold'}
```

structure

```
<StructureData: uuid: abcab877-1aee-4c8a-85a2-6500a507b578 (unstored)>
```

Cell

```
[[0.0, 1.78, 1.78], [1.78, 0.0, 1.78], [1.78, 1.78, 0.0]]
```

Sites

```
[<Site: kind name 'C' @ 0.0,0.0,0.0>, <Site: kind name 'C' @ 0.89,0.89,0.89>]
```

▶ Run workflow with validated input

The workflow was run with **workflow ID = 114**. Please note down this number for future reference.

Monitor the status and check results of workflow 114.

AiiDA Lab demo - phonon results

aiidalab.materialscloud.org

.start-15 | .phonon turnkey solution-14

AiiDAlab | Edit App | Logout | Control Panel | Materials Cloud

```
{'automatic_parallelization': {'max_num_machines': 2,
  'max_wall_time_seconds': 86400,
  'target_time_seconds': 3600},
 'cutoff_factor': 1.0,
 'deg': 0.02,
 'do_bands': True,
 'do_mag': False,
 'do_phonons': True,
 'kpts_dist': 0.2,
 'matdyn_codename': 'matdyn-6.1@daint-s761',
 'mixing_beta': 0.7,
 'ph_codename': 'ph-6.1@daint-s746',
 'psfam': 'SSSP_eff_PBE',
 'pw_codename': 'pw-6.1@daint-s746',
 'q2r_codename': 'q2r-6.1@daint-s761',
 'relaxation_scheme': 'vc-relax',
 'smear': 'cold'}
```

structure

```
<StructureData: uuid: abcab877-1aee-4c8a-85a2-6500a507b578 (unstored)>
```

Cell

```
[[0.0, 1.78, 1.78], [1.78, 0.0, 1.78], [1.78, 1.78, 0.0]]
```

Sites

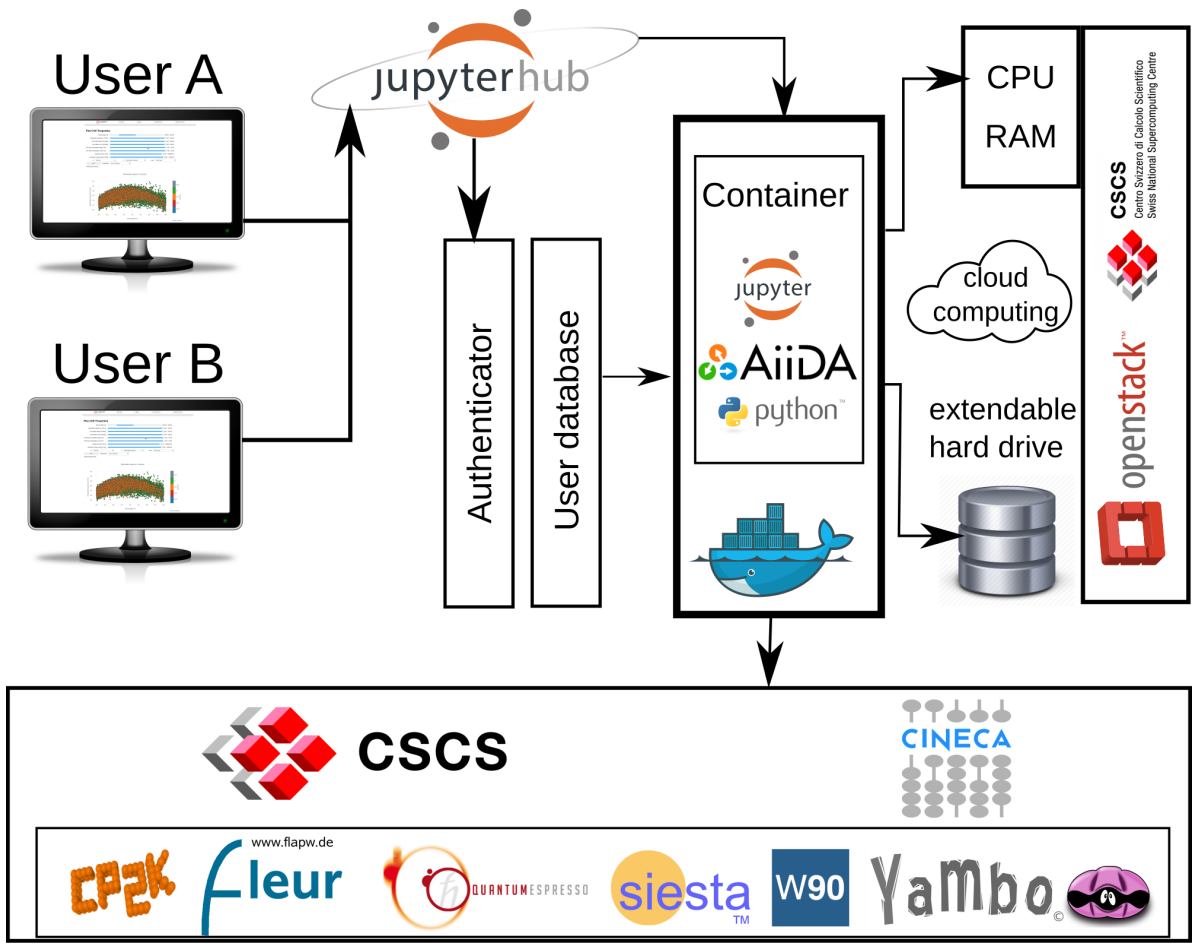
```
[<Site: kind name 'C' @ 0.0,0.0,0.0>, <Site: kind name 'C' @ 0.89,0.89,0.89>]
```

▶ Run workflow with validated input

The workflow was run with **workflow ID = 114**. Please note down this number for future reference.

Monitor the status and check results of workflow 114.

AiiDA Lab Architecture



Technologies:

- JupyterHub (multi-user server for Jupyter notebooks)
- Jupyter notebooks (interactive python, + appmode)
- Docker (isolated environment for every user)
- Openstack (software to manage computing clouds)

Conclusions

- Workflows & Daemon help with automation
- AiiDA graph = map of the data jungle
- Your calculations are ready for Open Science

- FAIR sharing of data
- Free: 5 GB/record
  (50 GB for AiiDA graphs)

re3data.org
REGISTRY OF RESEARCH DATA REPOSITORIES
http://doi.org/10.17616/R3ZJ5W
Materials Cloud

FAIRsharing.org
standards, databases, policies

SCIENTIFIC DATA

- write simple apps in python
- Run where you like:
  - on your machine (your machine, your setup)
  - on Quantum Mobile (your machine, our setup)
  - on AiiDA Lab (our machine, our setup)

Acknowledgements

Supported by

MARVEL NCCR

www.nccr-marvel.ch

MaX Centre

www.max-centre.eu

SNSF

http://www.snf.ch

EPFL

www.epfl.ch

CSCS

www.cscs.ch

PRACE

www.prace-ri.eu

Partners - Research

NFFA

www.nffa.eu

EMMC

www.emmc.info

MarketPlace project

the-marketplace-project.eu

ERC

erc.europa.eu

Developers

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)

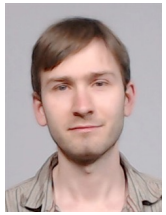

Elsa Passaro (EPFL)


Giovanni Pizzi (EPFL)


Thomas Schulthess (ETHZ,CSCS)


Berend Smit (EPFL)


Leopold Talirz (EPFL)


Martin Uhrin (EPFL)


Joost VandeVondele (ETHZ,CSCS)


Casper Welzel (EPFL)


Aliaksandr Yakutovich (EPFL)


Spyros Zoupanos (EPFL)

Contributors for the 23+ 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, Dominik Gresch, Rico Häuselmann, Eric Hontz, Christoph Koch, Espen Flage-Larsen, Andrius Merkys, Nicolas Mounet, Tiziano Müller, Riccardo Sabatini, Ole Schütt, Phillippe Schwaller

The CSCS support teams

Contacts and info

Website: aiida.net

Docs: aiida-core.readthedocs.io

Git repo: github.com/aiidateam/aiida_core/

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

 facebook.com/aiidateam  @aiidateam

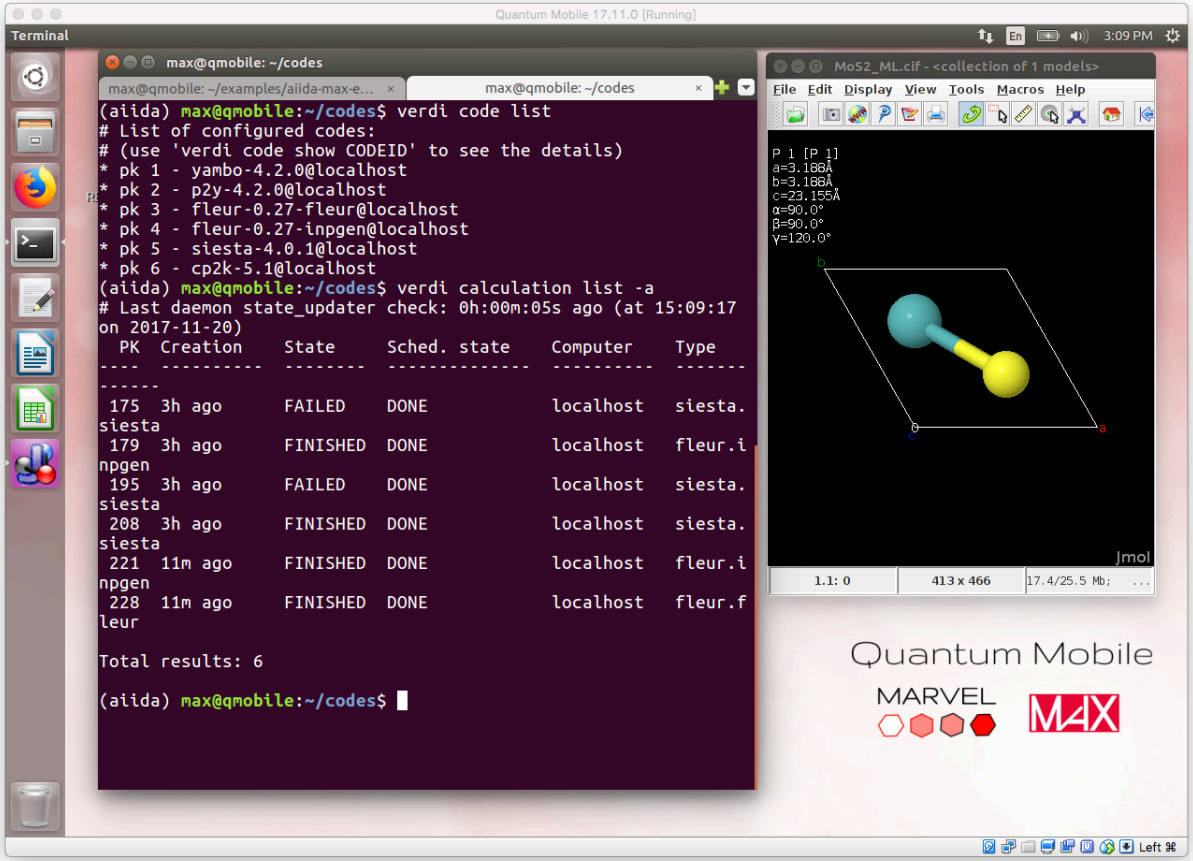
Materials Cloud: materialscloud.org

- **Archive**: archive.materialscloud.org
- **AiiDA Lab**: aiidalab.materialscloud.org
- **Quantum Mobile:** materialscloud.org/work/quantum-mobile

3 open positions for research software engineers/materials scientists

nccr-marvel.ch

Quantum Mobile Virtual Machine



- Ubuntu 16.04 LTS
- **AiiDA, AiiDA Lab**
- **QE, Siesta, fleur, yambo, cp2k, ... + AiiDA plugins**
- Visualization tools (xcrysden, …)
- used in tutorials & lectures at EPFL, ETH, ...
- **Modular setup: roll your own**

Runs on Linux, MacOS and Windows hosts using VirtualBox

Download: materialscloud.org/work/quantum-mobile