Automating materials science workflows with pymatgen, FireWorks, and atomate

Anubhav Jain
Energy Technologies Area
Lawrence Berkeley National Laboratory
Berkeley, CA

ASE/Fireworks workshop 2021

Slides (already) posted to hackingmaterials.lbl.gov

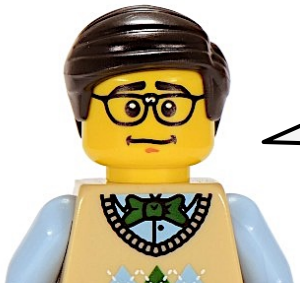
“Civilization advances by extending the number of important operations which we can perform without thinking about them.”

- Alfred North Whitehead

Outline

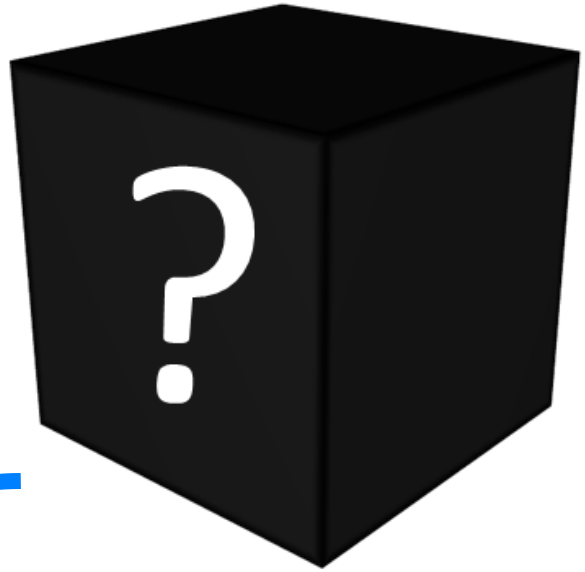
1. Overview of vision and goals
2. Case studies of usage
3. Current implementation
4. Future developments
5. Getting started

A "black-box" view of performing a calculation



What is the GGA-PBE elastic tensor of GaAs?

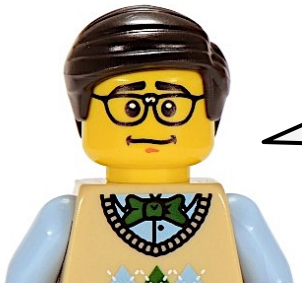
researcher



"something"

Results!

Unfortunately, the inside of the "black box" is usually tedious and "low-level"



researcher

What is the GGA-PBE elastic tensor of GaAs?

Input file flags
SLURM format
how to fix ZPOTRF?

- ❑ set up the structure coordinates
- ❑ write input files, double-check all the flags
- ❑ copy to supercomputer
- ❑ submit job to queue
- ❑ deal with supercomputer headaches
- ❑ monitor job
- ❑ fix error jobs, resubmit to queue, wait again
- ❑ repeat process for subsequent calculations in workflow
- ❑ parse output files to obtain results
- ❑ copy and organize results, e.g., into Excel

lots of tedious, low-level work...

Results!

All the low-level steps can lead to errors!

Let's take a look at two alternate universes:

1

you

have coffee

copy files from previous simulation

edit 5 lines

run simulation, analyze data

2

you

forget coffee

copy files from previous simulation

edit 4 lines but forget LHFCALC=F

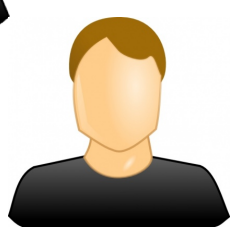
run simulation, looks fine at first, in a month you discover it was wrong

It is too easy to end up in the wrong timeline!

Today, it is difficult to learn and apply several computational procedures due to steep learning curve

Because of the multiple low-level steps and subtleties that all need to be done correctly to apply computational methods, there is often a single group "expert" for each technique

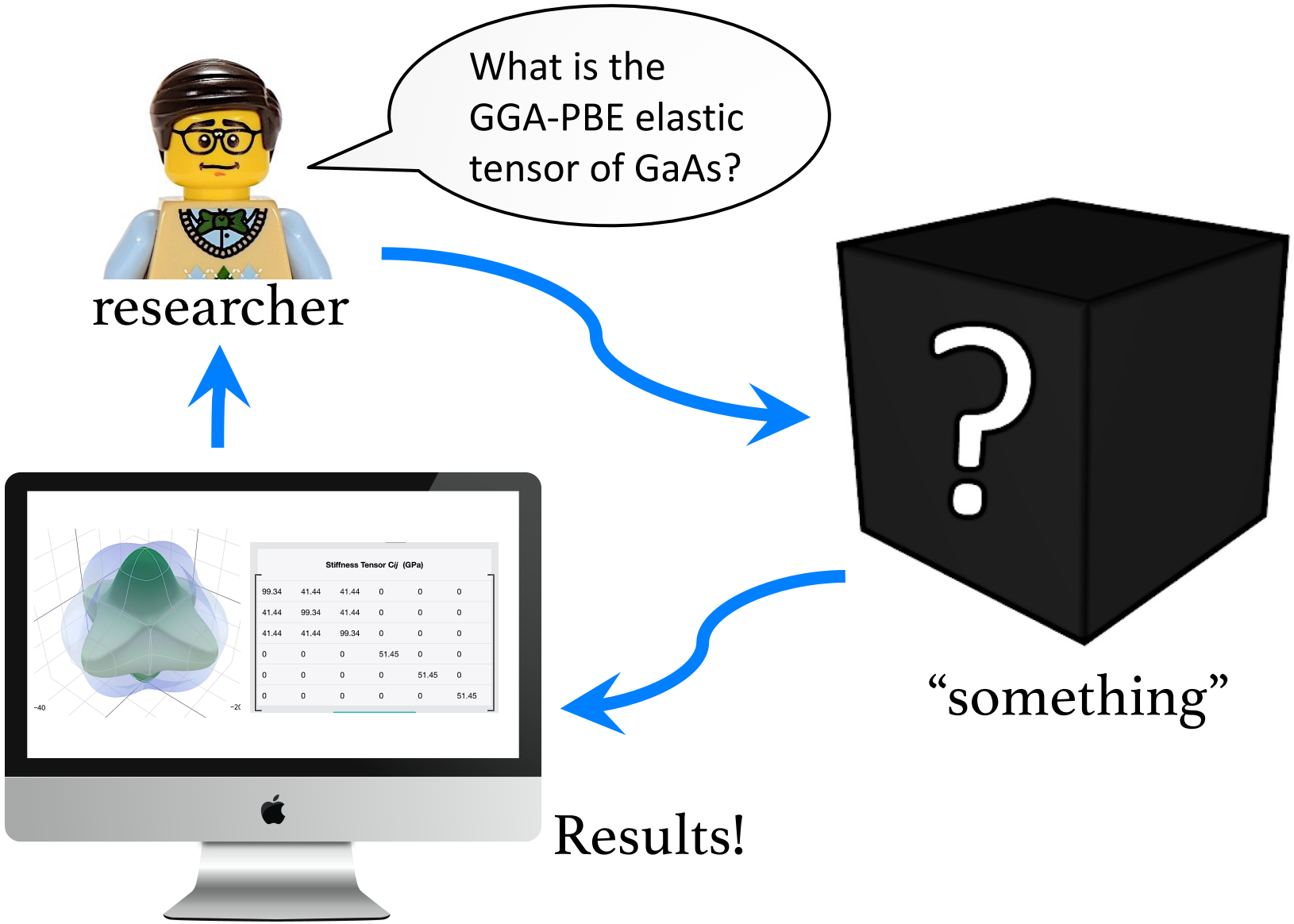
"Alice knows how to do charged defect calculations."



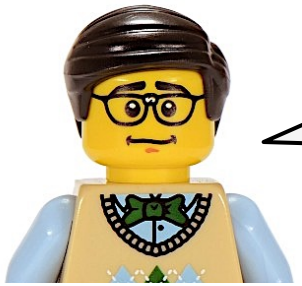
"Bob is the one who can properly converge GW runs."

"Olga has all the scripts for phonon calculations."

What would be a better way?

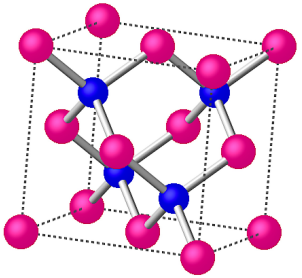


What would be a better way?



researcher

What is the GGA-PBE elastic tensor of GaAs?



Workflows to run

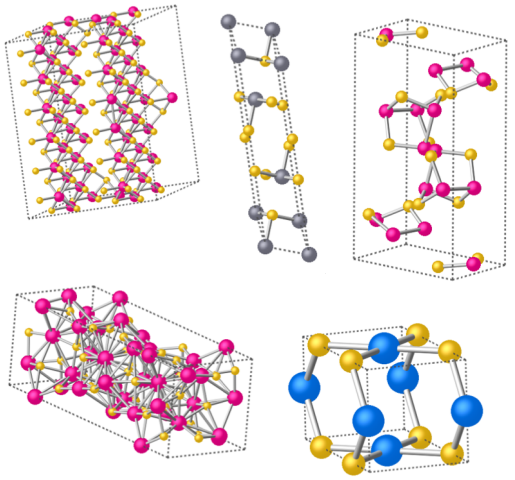
- ❑ band structure
- ❑ surface energies
- ✓ elastic tensor
- ❑ Raman spectrum
- ❑ QH thermal expansion
- ❑ spin-orbit coupling

Results!

Ideally the method should scale to millions of calculations

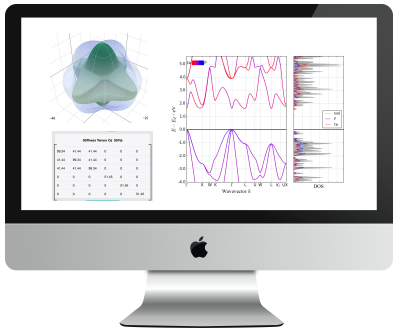
researcher

Start with all binary oxides, replace O->S, run several different properties



Workflows to run

- ✓ band structure
- ✓ surface energies
- ✓ elastic tensor
- ❑ Raman spectrum
- ❑ QH thermal expansion
- ❑ spin-orbit coupling

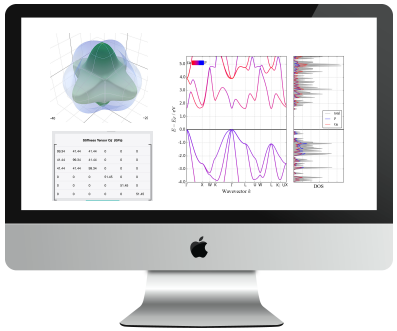


Results!

Atomate's vision is to make it easy, automatic, and flexible to generate data with existing simulation packages

Run many different properties of many different materials!

researcher



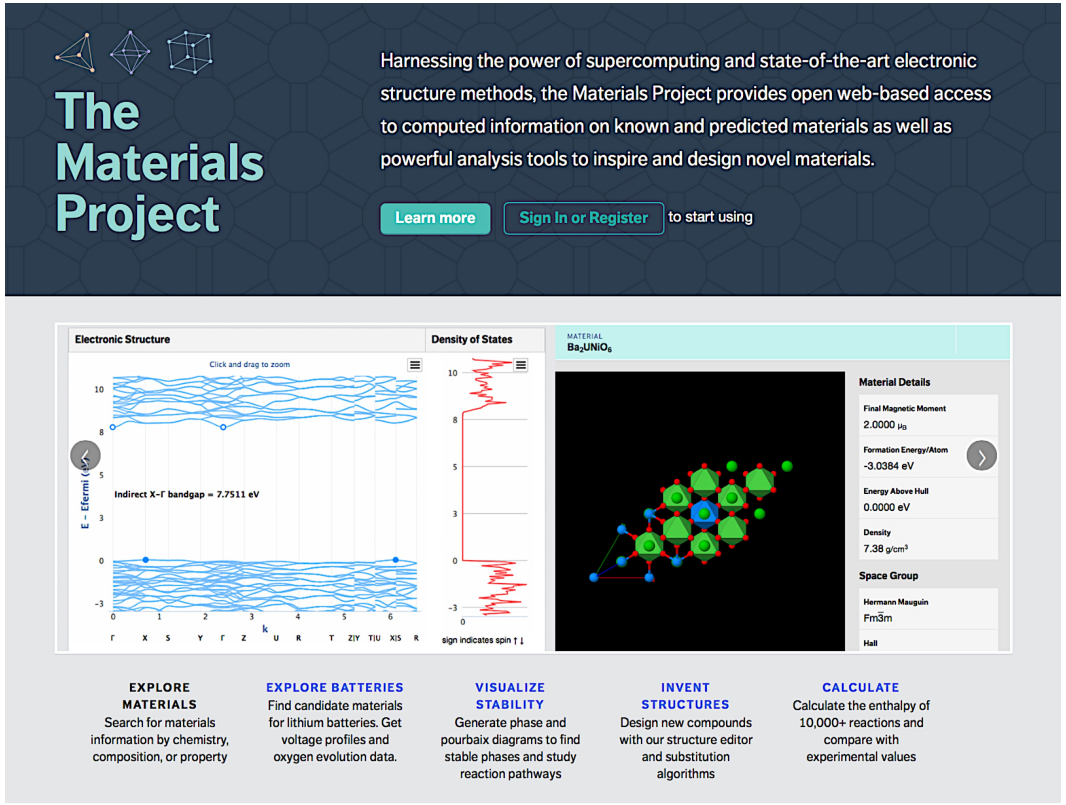
Results!

Outline

1. Overview of vision and goals
2. Case studies of usage
3. Current implementation
4. Future developments
5. Getting started

Example: The Materials Project database

The Materials Project (http://www.materialsproject.org)



Jain*, Ong*, Hautier, Chen, Richards, Dacek, Cholia, Gunter, Skinner, Ceder, and Persson, APL Mater., 2013, 1, 011002. *equal contributions

Free

>200,000 registered users around the world

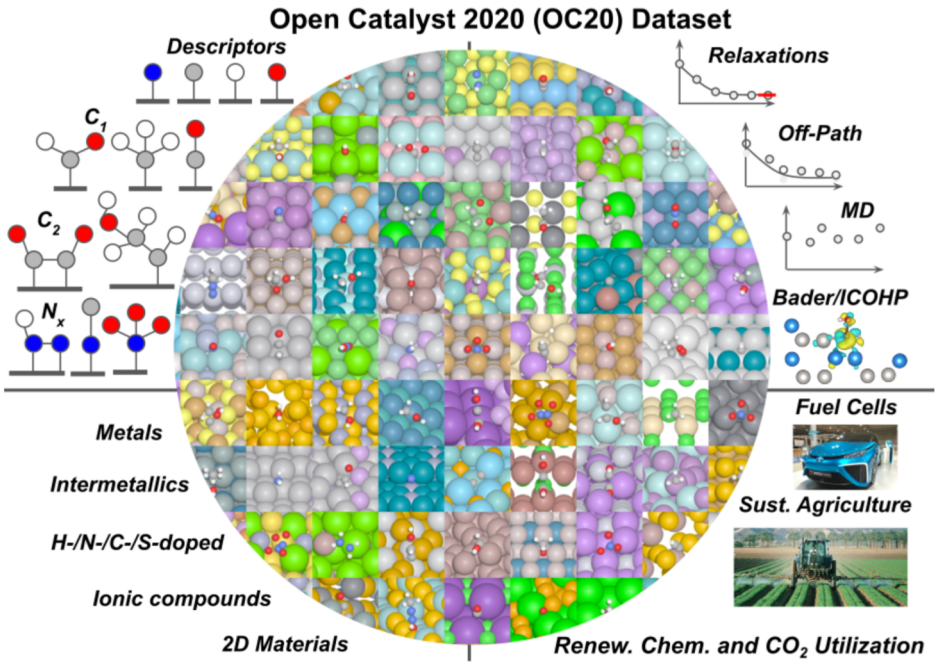
>150,000 compounds calculated multiple properties / compound

Data includes:

- thermodynamic props.
- electronic band structure
- aqueous stability (E-pH)
- elasticity tensors
- piezoelectric tensors
- electrolyte molecules
- much more

~1 billion CPU-hours invested

Example: Open Catalyst Project



>1.2 million DFT adsorbate relaxations

>250 million single point DFT calculations

GASpy

Automating surface chemistry or catalysis calculations can be complicated. Our workflows are built on top of fireworks, pymatgen, luigi, ase, and other helpful toolkits. With this system we perform ~100-200 DFT calculations per day across various chemistries and materials. A lot of work is put into the active learning system to find and schedule interesting calculations. You can read more about this system

- Dynamic Workflows for Routine Materials Discovery in Surface Science
- Active learning across intermetallics to guide discovery of electrocatalysts for CO2 reduction and H2 evolution

Example: many open data sets generated with high-throughput computing

SCIENTIFIC DATA

OPEN

DATA DESCRIPTOR

2DMatPedia, an open computational database of two-dimensional materials from top-down and bottom-up approaches

Received: 19 February 2019
Accepted: 7 May 2019
Published online: 12 June 2019

Jun Zhou[1], Lei Shen[2], Miguel Dias Costa[3], Kristin A. Persson[4,5], Shyue Ping Ong[6], Patrick Huck[5], Yunhao Lu[7], Xiaoyang Ma[1], Yiming Chen[6], Hanmei Tang[6] & Yuan Ping Feng[1,3]

SCIENTIFIC DATA

Check for updates

OPEN

DATA DESCRIPTOR

An automatically curated first-principles database of ferroelectrics

Tess E. Smidt[1,2], Stephanie A. Mack[1,2,3], Sebastian E. Reyes-Lillo[1,2,4], Anubhav Jain[3] & Jeffrey B. Neaton[1,2,5]✉

SCIENTIFIC DATA

OPEN

SUBJECT CATEGORIES

» Computational methods

» Surfaces, interfaces and thin films

» Density functional

Data Descriptor: Surface energies of elemental crystals

Richard Tran[1], Zihan Xu[1], Balachandran Radhakrishnan[1], Donald Winston[2], Wenhao Sun[3], Kristin A. Persson[2,4] & Shyue Ping Ong[1]

SCIENTIFIC DATA

OPEN

ATA DESCRIPTOR

High-throughput computation and evaluation of raman spectra

Qiaohao Liang[1], Shyam Dwaraknath[2] & Kristin A. Persson[1,2]

SCIENTIFIC DATA

OPEN

Data Descriptor: High-throughput computational X-ray absorption spectroscopy

Kiran Mathew[1,*], Chen Zheng[2,*], Donald Winston[3], Chi Chen[2], Alan Dozier[4], John J. Rehr[5], Shyue Ping Ong[2] & Kristin A. Persson[1]

Received: 11 December 2017

scientific data

Check for updates

OPEN

DATA DESCRIPTOR

Database of ab initio L-edge X-ray absorption near edge structure

Yiming Chen[1], Chi Chen[1], Chen Zheng[1], Shyam Dwaraknath[2], Matthew K. Horton[2], Jordi Cabana[3], John Rehr[4], John Vinson[5], Alan Dozier[6], Joshua J. Kas[4], Kristin A. Persson[7,8] & Shyue Ping Ong[1]✉

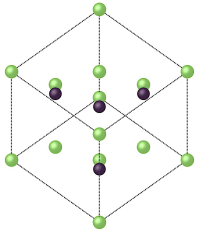
Outline

1. Overview of vision and goals
2. Case studies of usage
3. Current implementation
4. Future developments
5. Getting started

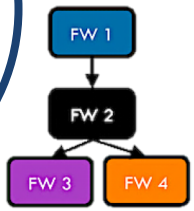
Steps for theory & simulation

material generation

(crystal, surface/interface, molecule, etc...)



simulation procedure



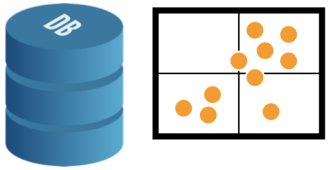
simulation parameters and sequence (workflow)

calculation & execution

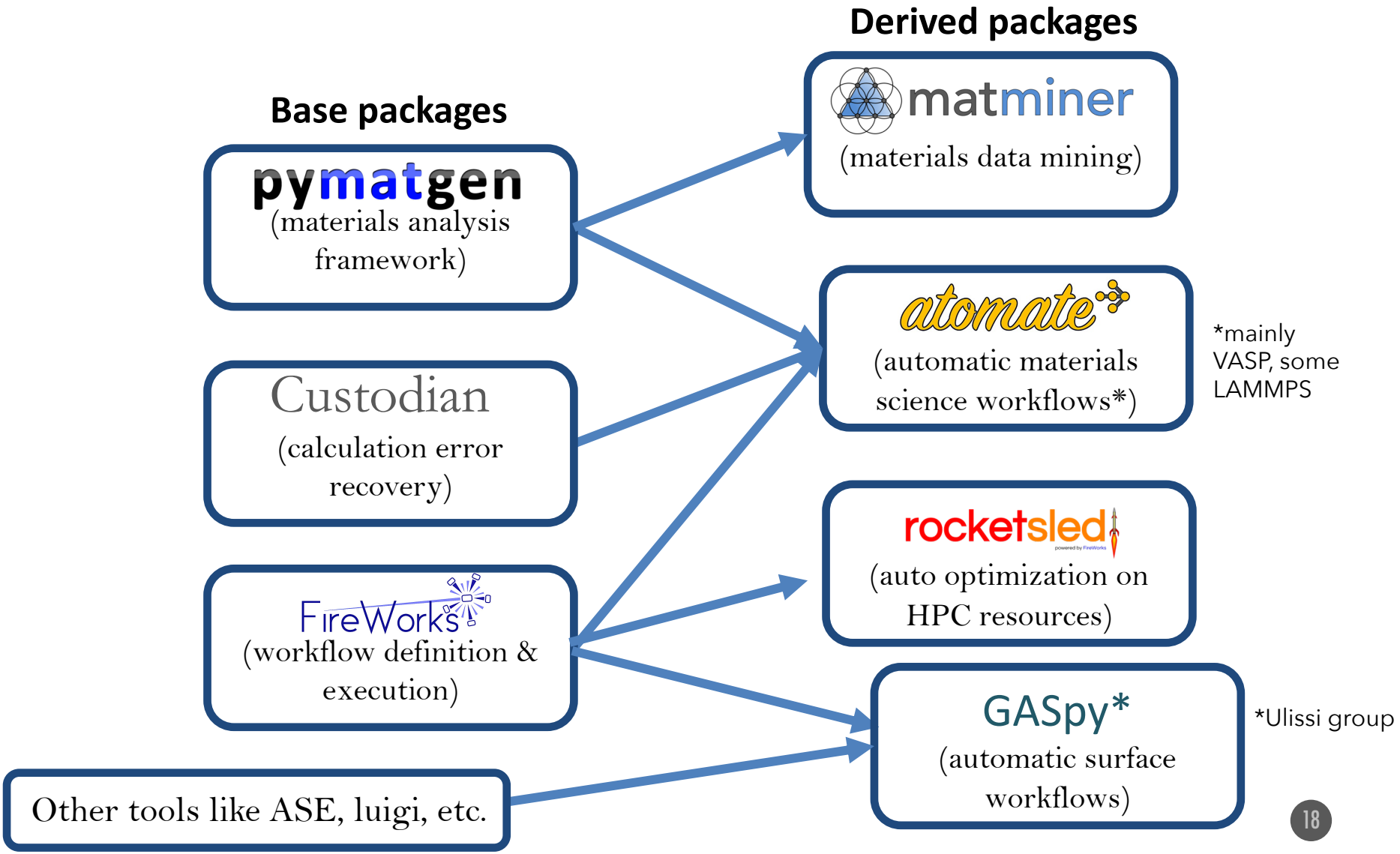
job management, execution, error recovery

data analysis

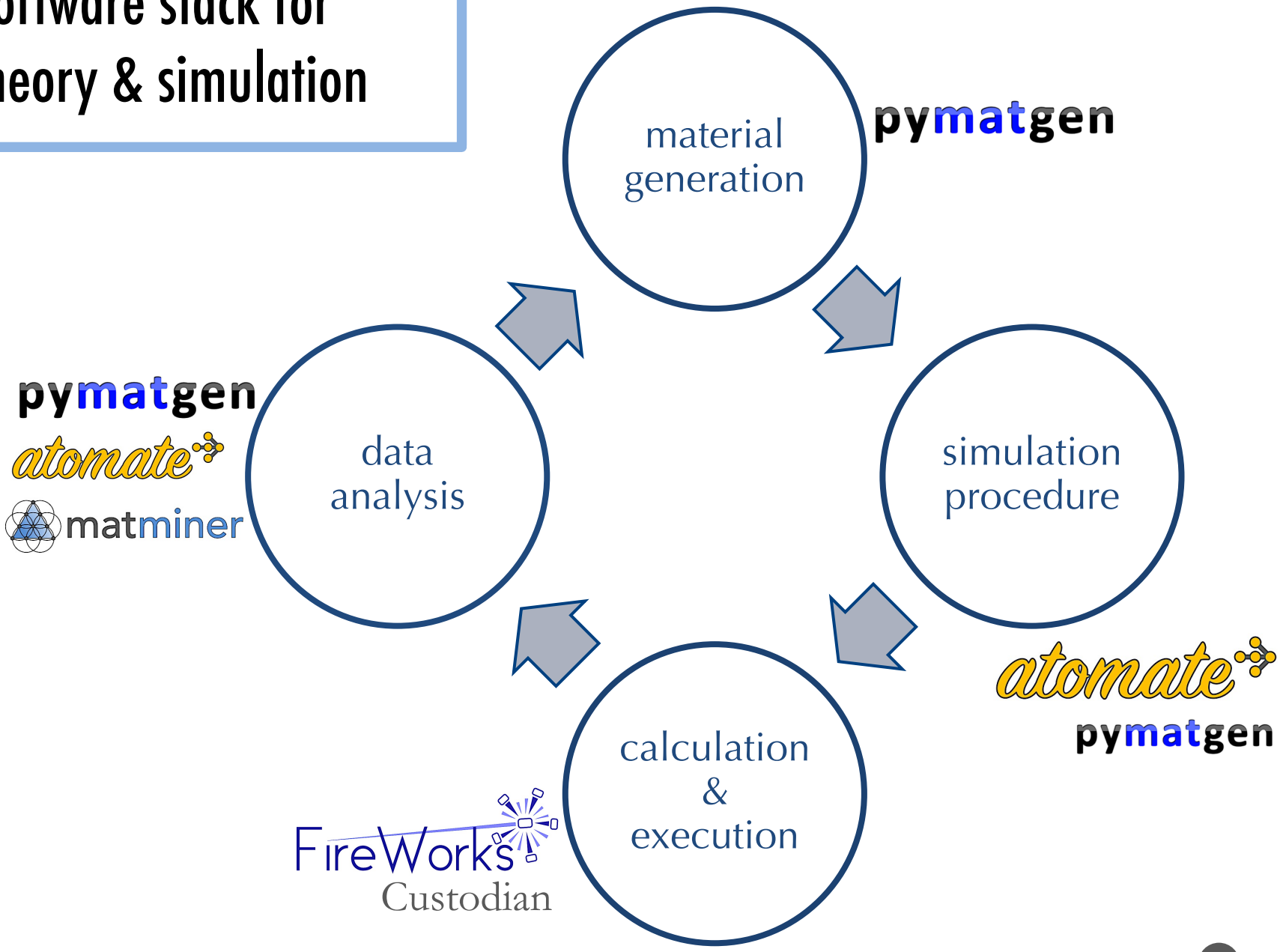
databases, plotting, analysis, insight



Software toolkits and frameworks for theory / calculation



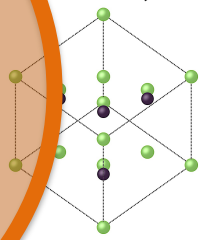
Software stack for theory & simulation



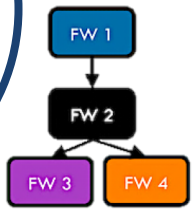
Steps for theory & simulation

material generation

(crystal, surface/interface, molecule, etc...)



simulation procedure



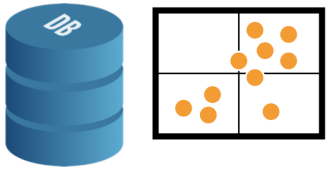
simulation parameters and sequence (workflow)

calculation & execution

job management, execution, error recovery

data analysis

databases, plotting, analysis, insight

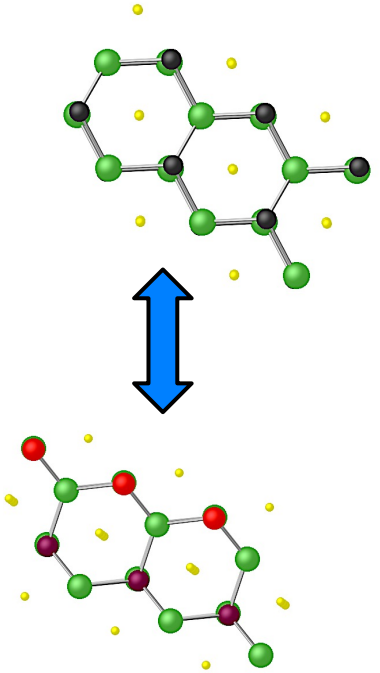


Step 1: materials generation

material generation

pymatgen

- Pymatgen can help generate models for:
  - crystal structures
  - molecules
  - systems (surfaces, interfaces, etc.)
- Tools include:
  - order-disorder (shown at right) and SQS
  - interstitial finding
  - surface / slab generation
  - structure matching and analysis
  - get structures from Materials Project
- Won't cover materials generation in this presentation!

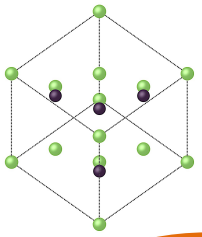


Example: Order-disorder
resolve partial or mixed occupancies into a fully ordered crystal structure (e.g., mixed oxide-fluoride site into separate oxygen/fluorine)

Steps for theory & simulation

material generation

(crystal, surface/interface, molecule, etc...)

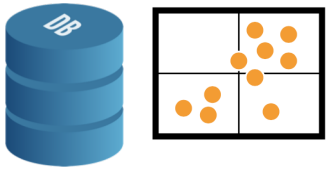


simulation procedure

simulation parameters and sequence (workflow)

data analysis

databases, plotting, analysis, insight



calculation & execution

job management, execution, error recovery

Atomate contains a library of simulation procedures

b-initio VASP ackage ienna imulation

Q-CHEM™ A QUANTUM LEAP INTO THE FUTURE OF CHEMISTRY

FEFF

VASP-based

- band structure
- spin-orbit coupling
- hybrid functional calcs
- elastic tensor
- piezoelectric tensor
- Raman spectra
- NEB
- GIBBS method
- QH thermal expansion
- AIMD
- ferroelectric
- surface adsorption
- work functions
- NMR spectra
- Bader charges
- Magnetic orderings
- SCAN functionals

Other

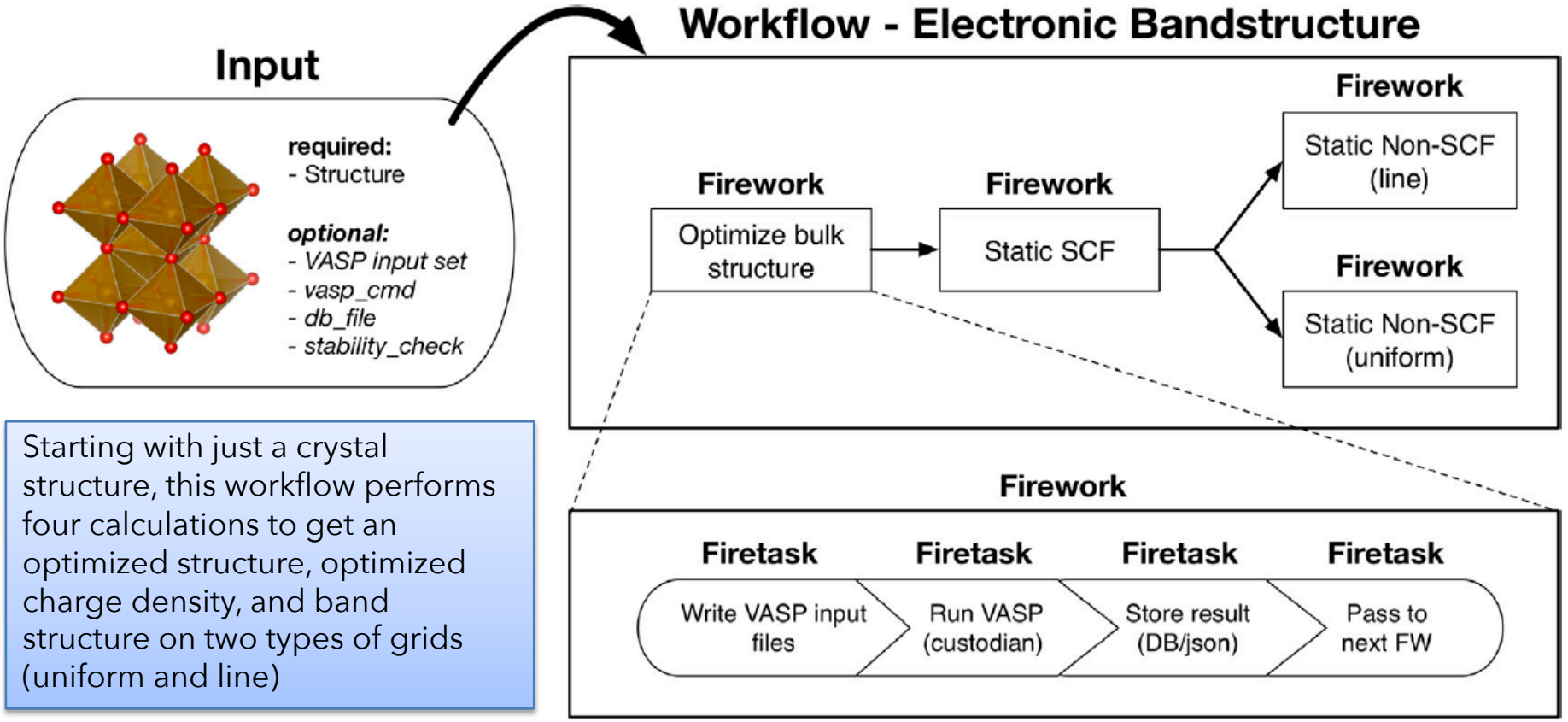
- BoltzTraP
- FEFF method
- Q-Chem

In progress

- AMSET e-transport
- HiPhive for phonons

Mathew, K. et al Atomate: A high-level interface to generate, execute, and analyze computational materials science workflows, Comput. Mater. Sci. 139 (2017) 140–152.

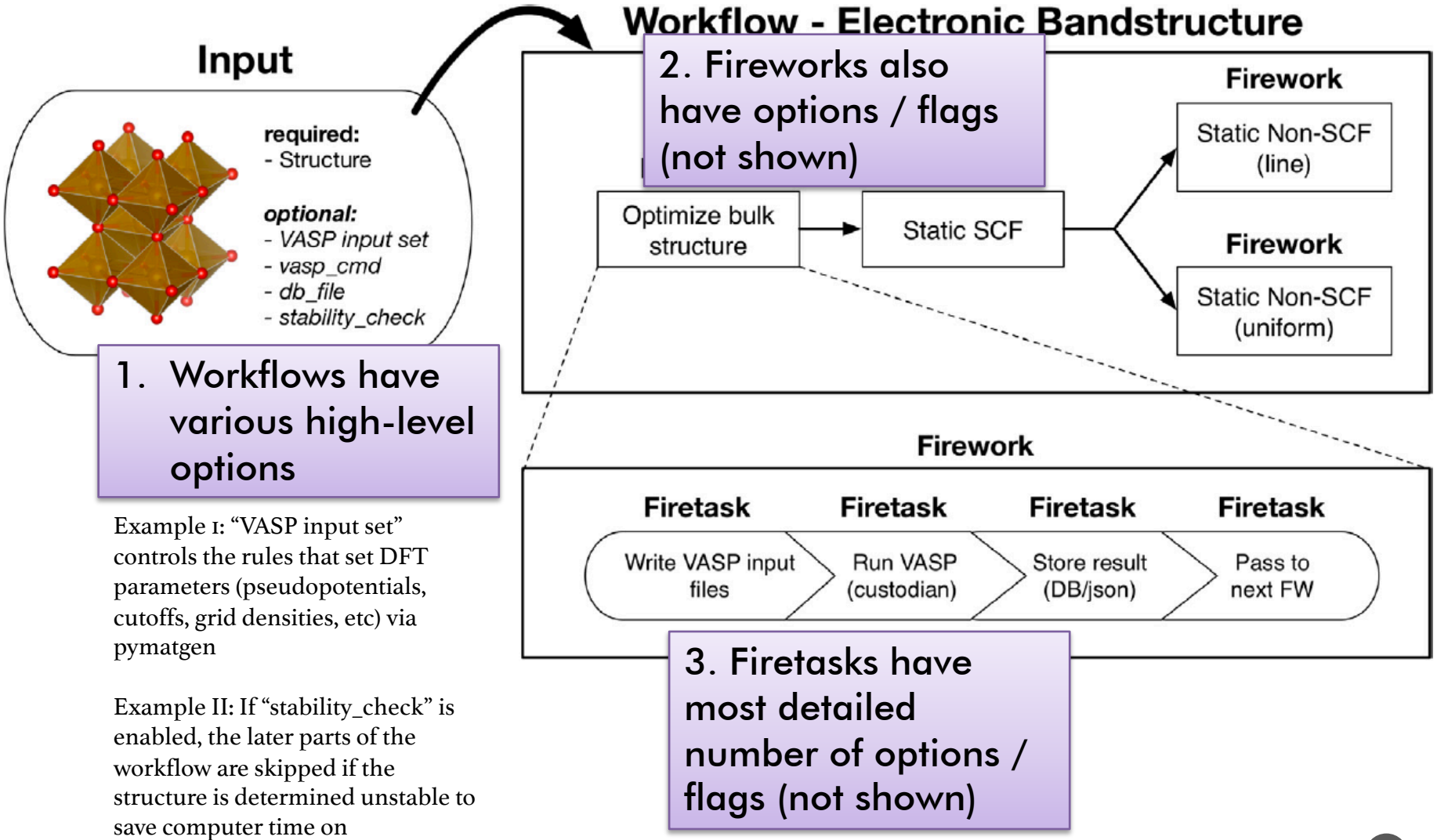
Each simulation procedure in atomate is composed of multiple levels of detail / abstraction



Starting with just a crystal structure, this workflow performs four calculations to get an optimized structure, optimized charge density, and band structure on two types of grids (uniform and line)

Workflow – complete set of calculations to get a materials property
Firework – one step in the Workflow (typically one DFT calculation)
Firetask – one step in a Firework

Workflow parameters can be customized at multiple levels of detail



1. Workflows have various high-level options

Example I: "VASP input set" controls the rules that set DFT parameters (pseudopotentials, cutoffs, grid densities, etc) via pymatgen

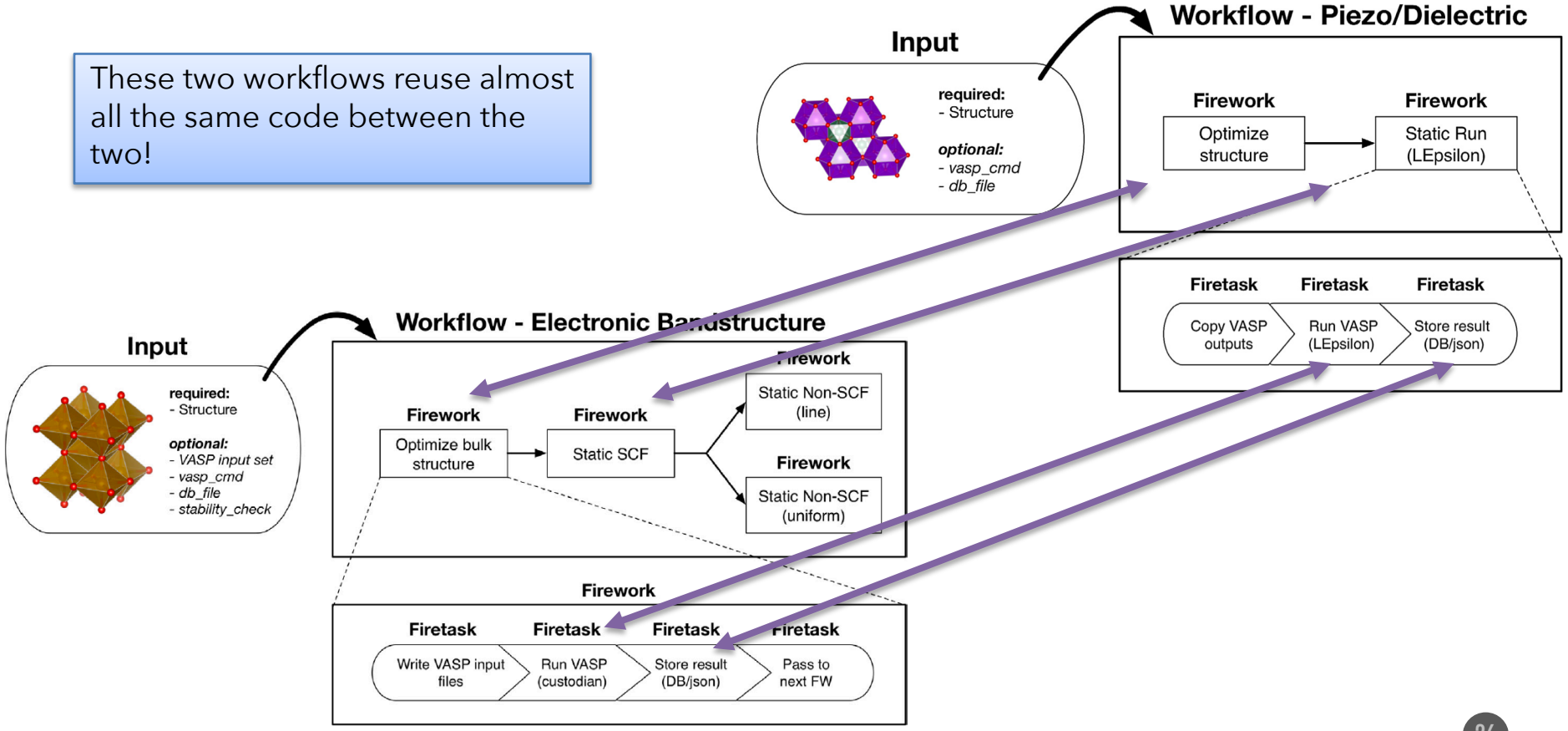
Example II: If "stability_check" is enabled, the later parts of the workflow are skipped if the structure is determined unstable to save computer time on uninteresting structures

3. Firetasks have most detailed number of options / flags (not shown)

You can build workflows from scratch or reuse components to assemble workflows

Multiple workflows are built with the same components stacked together in different ways

These two workflows reuse almost all the same code between the two!



atomate allows you to leverage the prior efforts and knowledge of many researchers

atomate +  → 

All past and present knowledge, from everyone in the group, everyone previously in the group, and our collaborators, about how to run calculations


K. Mathew


J. Montoya


S. Dwaraknath


A. Faghaninia


B. Bocklund


T. Smidt


M. Aykol


H. Tang


I.H. Chu


M. Horton


J. Dagdalen


B. Wood


Z.K. Liu


J. Neaton


S.P. Ong

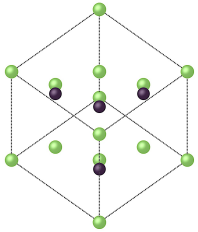

K. Persson


A. Jain

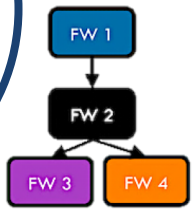
Steps for theory & simulation

material generation

(crystal, surface/interface, molecule, etc...)



simulation procedure



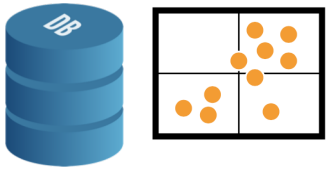
simulation parameters and sequence (workflow)

calculation & execution

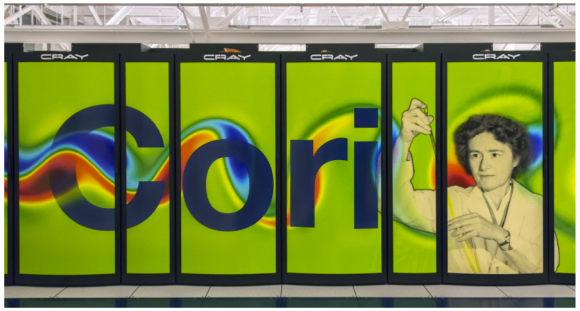
job management, execution, error recovery

data analysis

databases, plotting, analysis, insight



FireWorks allows you to write your workflow once and execute (almost) anywhere



- Execute workflows locally or at a supercomputing center
- Queue systems supported
  - PBS
  - SGE
  - SLURM
  - IBM LoadLeveler
  - NEWT (a REST-based API at NERSC)
  - Cobalt (Argonne LCF)
- Cloud based services (user-generated)
  - https://github.com/CovertLab/borealis

Dashboard with status of all jobs

FireWorks

Workflow Dashboard

Newest Workflows

B4 C1 READY ID: 1573745

- B4_C1--Controller_add_Electronic_Structure_v2
- B4_C1--VASP_db_insertion
- B4_C1--GGA_optimize_structure_(2x)
- B4_C1--Add_to_SNL_database

Ba2 Fe1 Nb1 O6 READY ID: 1573739

- Ba2_Fe1_Nb1_O6--Controller_add_Electronic_Structure_v2
- Ba2_Fe1_Nb1_O6--VASP_db_insertion
- Ba2_Fe1_Nb1_O6--GGAU_optimize_structure_(2x)
- Ba2_Fe1_Nb1_O6--VASP_db_insertion
- Ba2_Fe1_Nb1_O6--GGA_optimize_structure_(2x)
- Ba2_Fe1_Nb1_O6--Add_to_SNL_database

Ba2 Fe1 Nb1 O6 READY ID: 1573733

- Ba2_Fe1_Nb1_O6--Controller_add_Electronic_Structure_v2
- Ba2_Fe1_Nb1_O6--VASP_db_insertion
- Ba2_Fe1_Nb1_O6--GGAU_optimize_structure_(2x)
- Ba2_Fe1_Nb1_O6--VASP_db_insertion
- Ba2_Fe1_Nb1_O6--GGA_optimize_structure_(2x)
- Ba2_Fe1_Nb1_O6--Add_to_SNL_database

Current Database Status

	Fireworks	Workflows
RUNNING	994	4,728
ARCHIVED	109,576	22,992
WAITING	167,134	0
FIZZLED	76,949	44,682
READY	24,199	18,197
RESERVED	799	0
COMPLETED	1,187,041	111,402
DEFUSED	6,588	3,656
TOTAL	1,573,280	205,657

Summary Reports

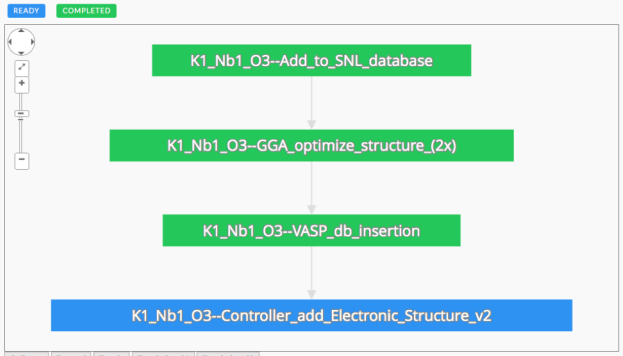
Get a report of all jobs for the past:

- 30 minutes
- 24 hours
- 7 days
- 30 days
- 6 months
- 24 months
- 10 years

For more reporting options, use the "lpad report --help" command line tool.

FireWorks

Workflow 1951337



Collapse | Expand | Toggle | Toggle level1 | Toggle level2


```
{
    created_on: "2017-07-28T17:00:13.517000",
  + launch_dirs: { … },
  + links: { … },
  + metadata: { … },
    name: "K1 Nb1 O3",
  + parent_links: { … },
    state: "RUNNING",
  + states: { … },
    updated_on: "2017-08-01T08:01:19.128000"
}
```

Job provenance and automatic metadata storage

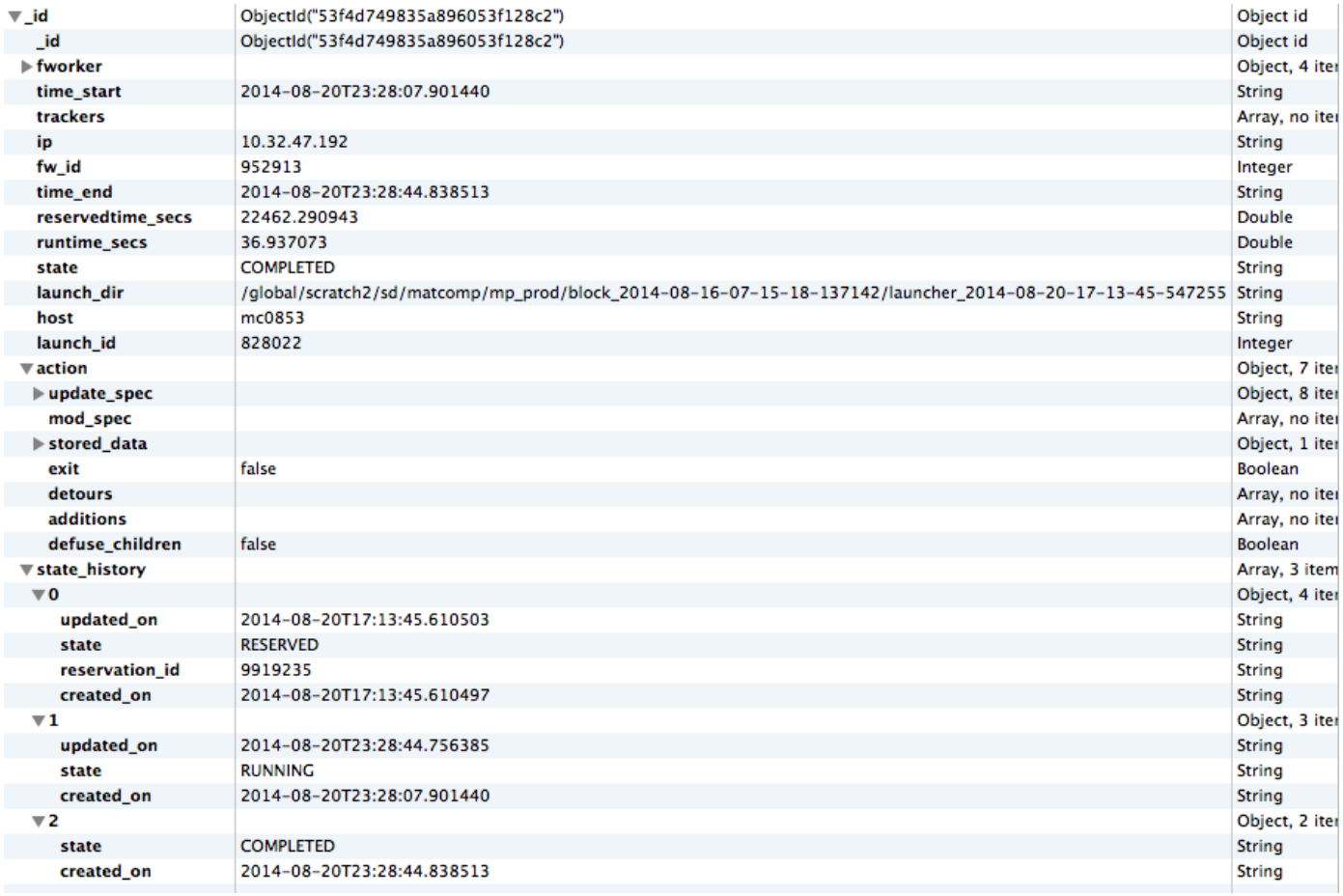
what machine
what time
what directory

what was the output

when was it queued

when did it start running

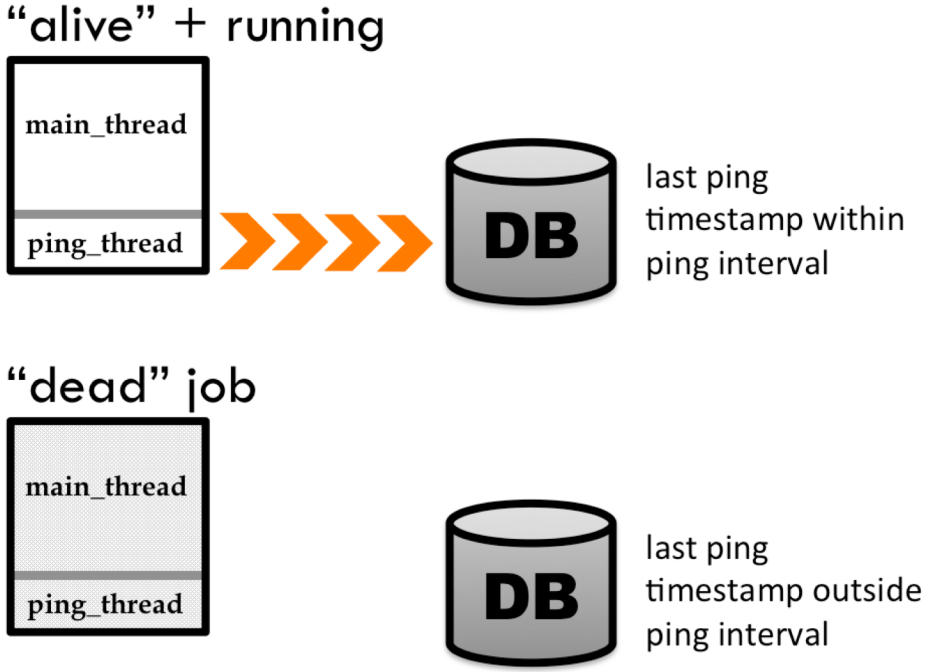
when was it completed



▼ _id	ObjectId("53f4d749835a896053f128c2")	Object id
_id	ObjectId("53f4d749835a896053f128c2")	Object id
▶ fworker		Object, 4 ite
time_start	2014-08-20T23:28:07.901440	String
trackers		Array, no ite
ip	10.32.47.192	String
fw_id	952913	Integer
time_end	2014-08-20T23:28:44.838513	String
reservedtime_secs	22462.290943	Double
runtime_secs	36.937073	Double
state	COMPLETED	String
launch_dir	/global/scratch2/sd/matcomp/mp_prod/block_2014-08-16-07-15-18-137142/launcher_2014-08-20-17-13-45-547255	String
host	mc0853	String
launch_id	828022	Integer
▼ action		Object, 7 ite
▶ update_spec		Object, 8 ite
mod_spec		Array, no ite
▶ stored_data		Object, 1 ite
exit	false	Boolean
detours		Array, no ite
additions		Array, no ite
defuse_children	false	Boolean
▼ state_history		Array, 3 item
▼ 0		Object, 4 ite
updated_on	2014-08-20T17:13:45.610503	String
state	RESERVED	String
reservation_id	9919235	String
created_on	2014-08-20T17:13:45.610497	String
▼ 1		Object, 3 ite
updated_on	2014-08-20T23:28:44.756385	String
state	RUNNING	String
created_on	2014-08-20T23:28:07.901440	String
▼ 2		Object, 2 ite
state	COMPLETED	String
created_on	2014-08-20T23:28:44.838513	String

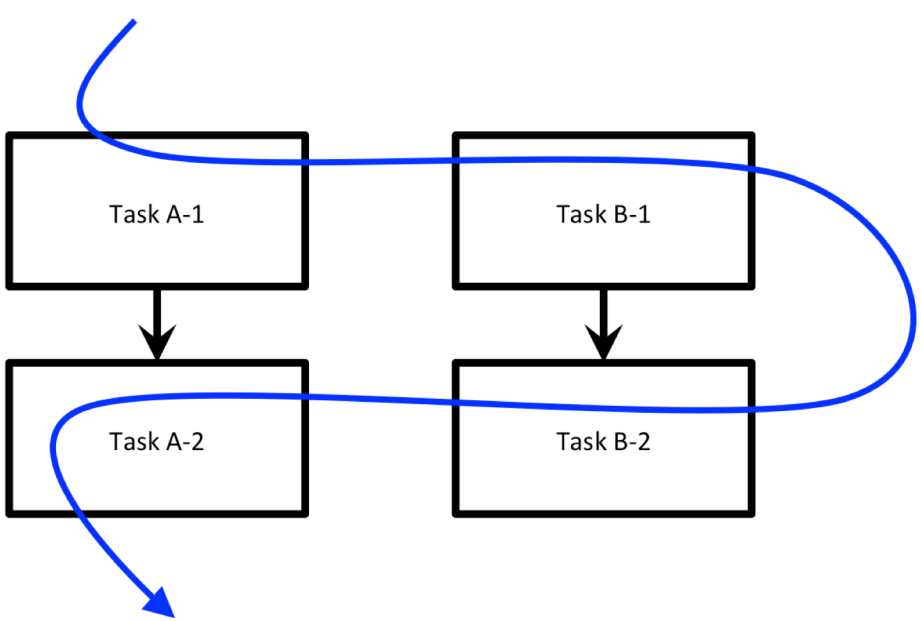
Detect and rerun failures

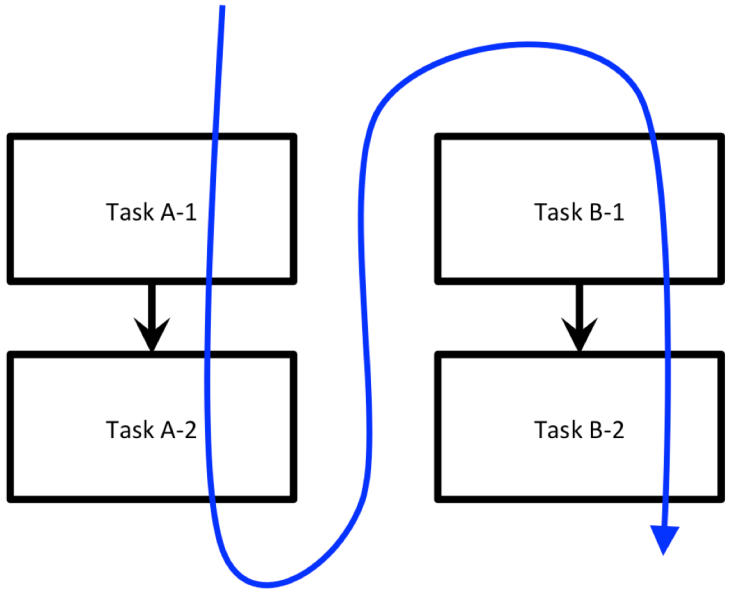
- All kinds of failures can be detected and rerun
  - Soft failures (job quits with error code)
  - hard failures (computing center goes down)



Customize job priorities

- Within workflow, or between workflows
- Completely flexible and can be modified / updated whenever you want





Track output files remotely

- Can bring up the last few lines of each of your output files – and can be combined with queries / filters

Now seems like a good time to bring up the last few lines of the OUTCAR of all failed jobs…

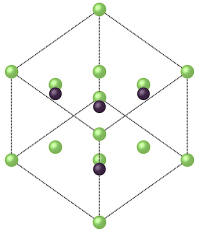
FireWorks workflow software: the main ideas

- FireWorks is used to execute Workflow objects at supercomputing centers
- It is very well-suited to high-throughput applications and has been used to execute millions of jobs and is well tested
- There are many features and advantages to using FireWorks as your workflow manager, which has made it one of the most popular scientific workflow software in use today

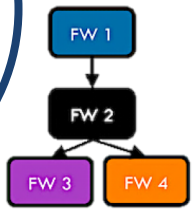
Steps for theory & simulation

material generation

(crystal, surface/interface, molecule, etc...)



simulation procedure



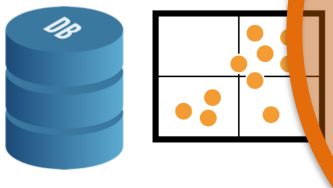
simulation parameters and sequence (workflow)

calculation & execution

job management, execution, error recovery

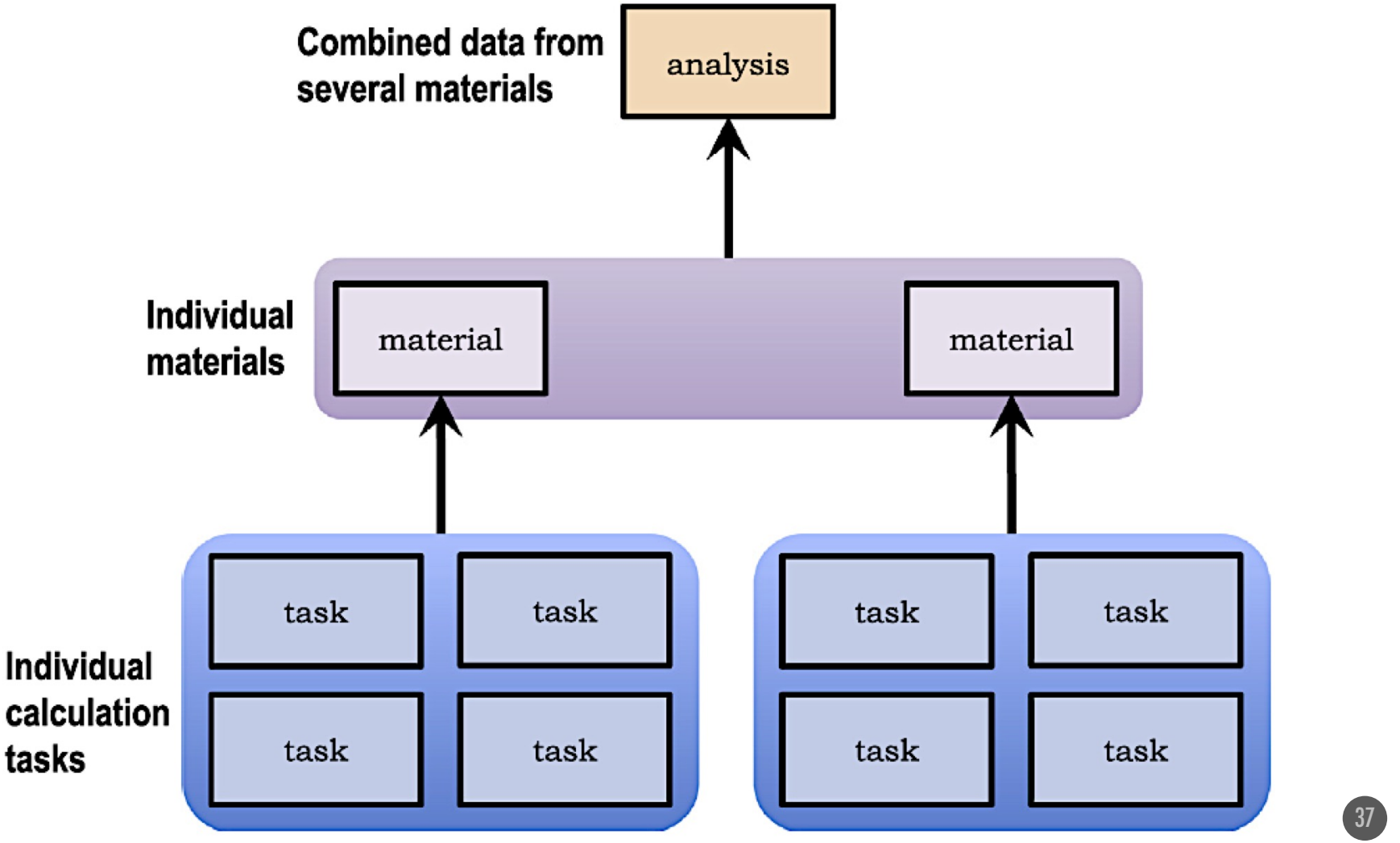
data analysis

databases, plotting analysis, insight



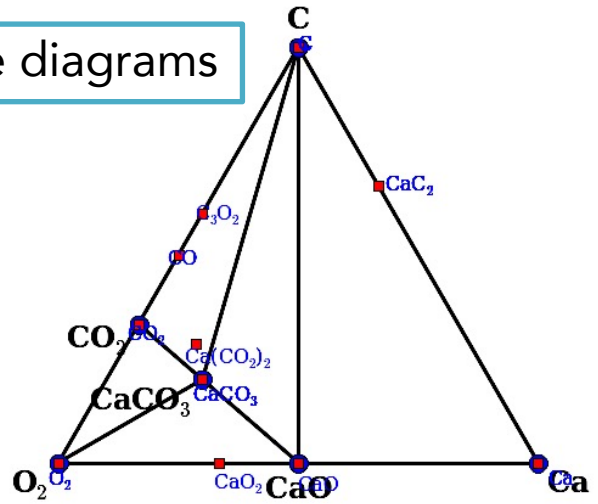
Atomate – builders framework

"Builders" start with base collections in a database and create higher-level collections that summarize information or add metadata

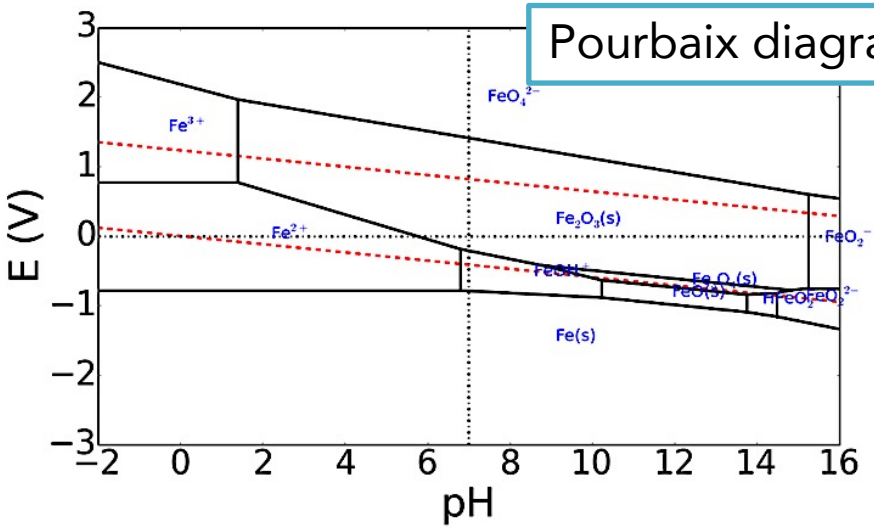


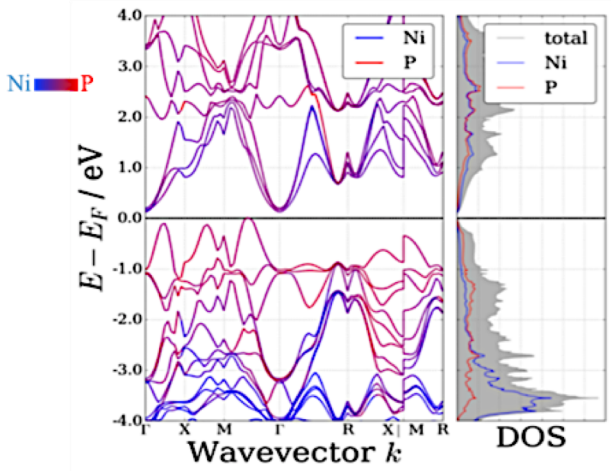
Examples of analyses

phase diagrams

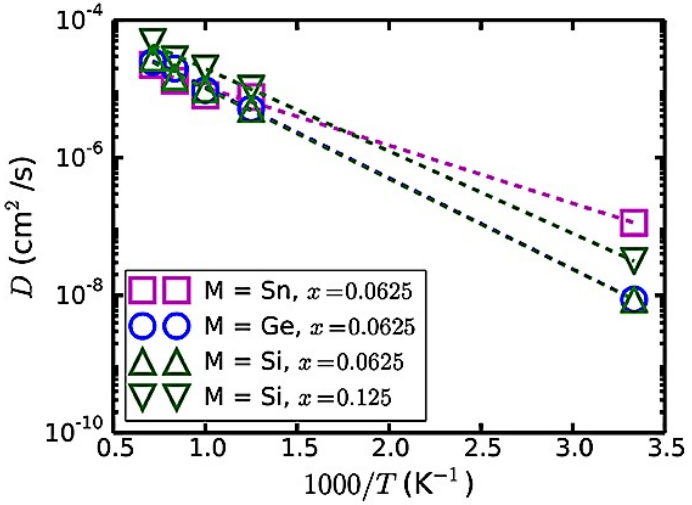


Pourbaix diagrams



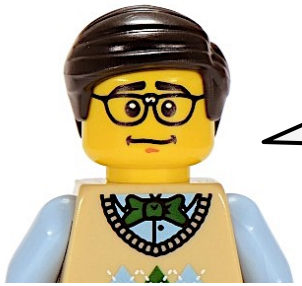


band structure analysis



diffusivity from MD

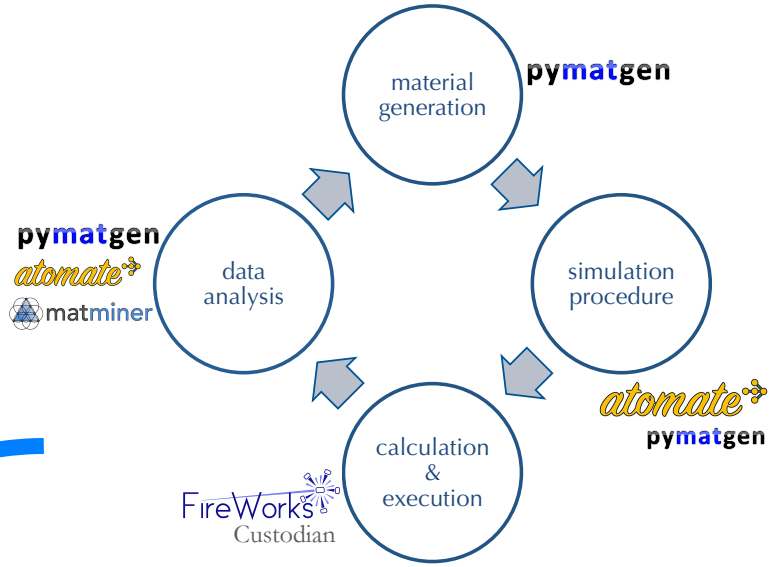
Doing simulations programmatically can become second nature – and very fast / automatic / scalable



What is the GGA-PBE elastic tensor of GaAs?

researcher

Results!



software infrastructure

Outline

1. Vision of atomate
2. History of atomate
3. Current implementation
4. Future developments
5. Getting started

Atomate v2 – a "jobflow" layer to FireWorks

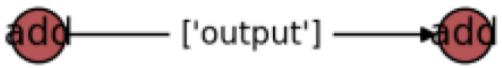
- After using atomate for ~4-5 years, we realize there are still things that are awkward to do. For example:
  - Data flow between workflows requires manual thinking to manage
  - Reusing workflow components with slight modifications often requires code duplication
- We decided to add a library on top of FireWorks called "jobflow" that helps make workflows more programmable
  - https://github.com/materialsproject/jobflow
- We suggest:
  - Try using jobflow when building on top of FireWorks (no need to use pymatgen or any other specific codes)
  - If wanting to use atomate, be on the lookout for atomate2 which will use jobflow
  - We are also working with the ABINIT team (G.M. Rignanese) to incorporate these new ideas into ABINIT workflows

Jobflow idea #1: More explicit data dependencies

```python
from jobflow import job, Flow

@job
def add(a, b):
    return a + b

add_first = add(1, 5)
add_second = add(add_first.output, 5)

flow = Flow([add_first, add_second])
flow.draw_graph().show()
```

Can refer to output of a job before the data is ever created. Makes chaining of workflows and Fireworks much easier

Automatically infer workflow diagram if desired (based on data dependencies)



Jobflow idea #2: Maker classes to enhance code reusability

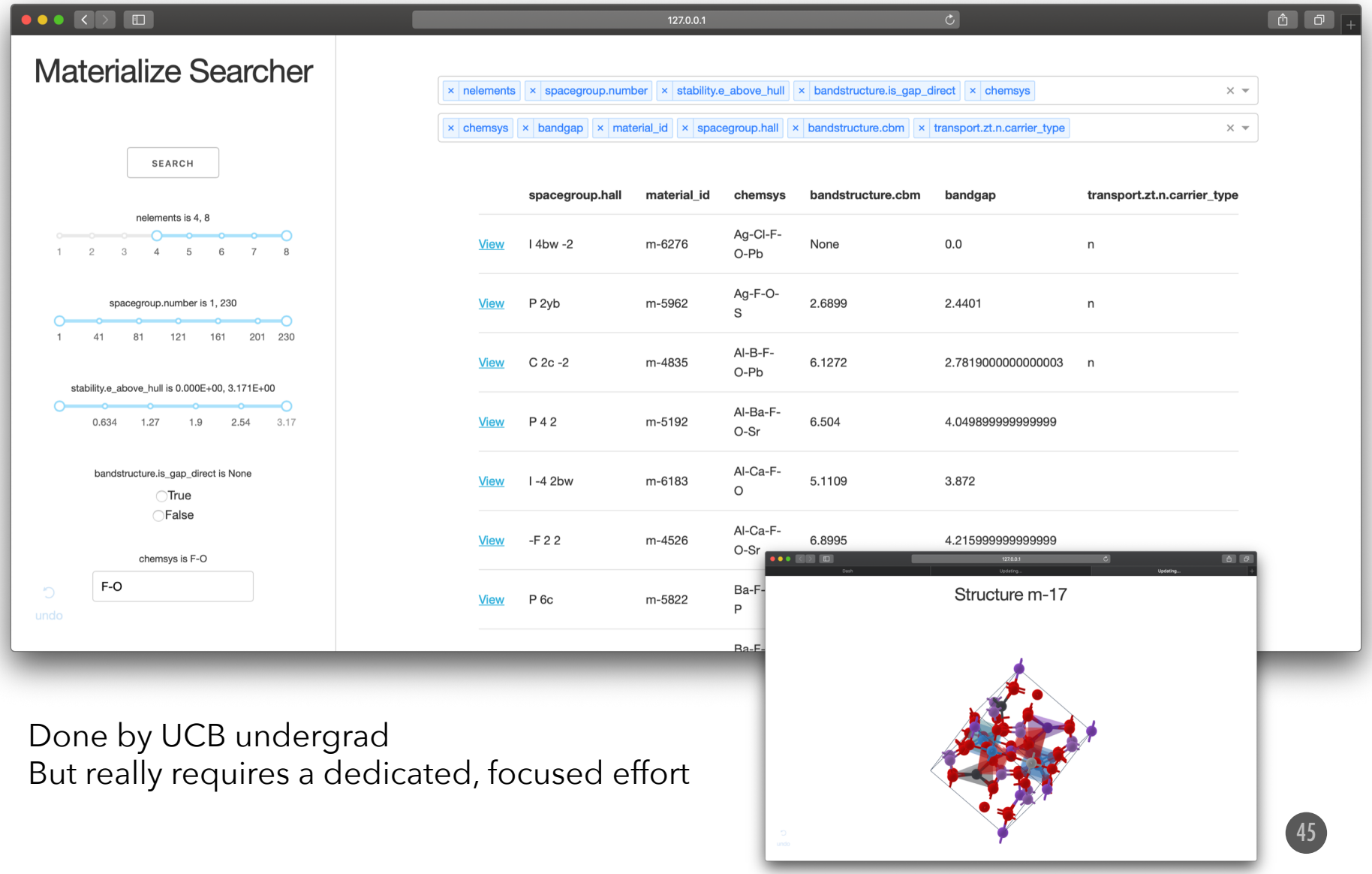
```python
@dataclass
class ElasticMaker(Maker):
    name: str = "elastic"
    relax_maker: Maker = VaspRelaxMaker()
    static_maker: Maker = VaspStaticMaker()

    def make(self, structure):
        relax = self.relax_maker.make(structure)
        perturbations = generate_perturbations(
            relax.output.structure, self.static_maker
        )
        elastic = fit_elastic(perturbations.output)
        return Activity([relax, perturbations, elastic], elastic.output)
```

E.g., this could be replaced with `AbinitRelaxMaker`

Atomate GUI

- Most of “atomate” (and all the various software tools) are largely programmatic, not interactive / GUI-driven
- The goal of atomate was always to work up to a visual interface
  - Stage 1: Exploring results using a web interface rather than by database queries
  - Stage 2: Submitting pre-defined workflow templates using a web interface
  - Stage 3: Designing and submitting custom workflows using a web interface
- Currently working on stage 1

Atomate GUI prototype



Done by UCB undergrad
But really requires a dedicated, focused effort

Outline

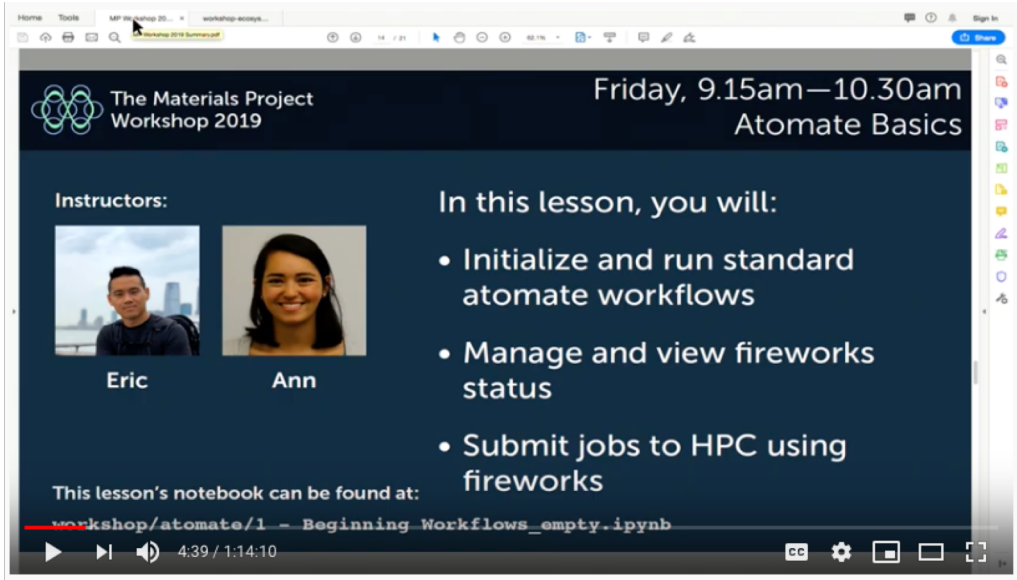
1. Vision of atomate
2. History of atomate
3. Current implementation
4. Future developments
5. Getting started

What resources are available to learn?

- **Papers:** good general overview / vision but of no practical help
- **Online MP Workshop videos and tutorials:** good way to get started if you have at least some knowledge, but usually not very deep. A “zero to one” situation
- **Code documentation:** Most comprehensive way to learn, although some experience probably necessary
- **Help channels:** Great if you already started and run into problems

Online MP workshop and videos

- Resources on http://workshop.materialsproject.org
- Videos on MP channel of Youtube:
  - https://www.youtube.com/channel/UC6pqY-__Nu-mKkv0LMP8FJQ



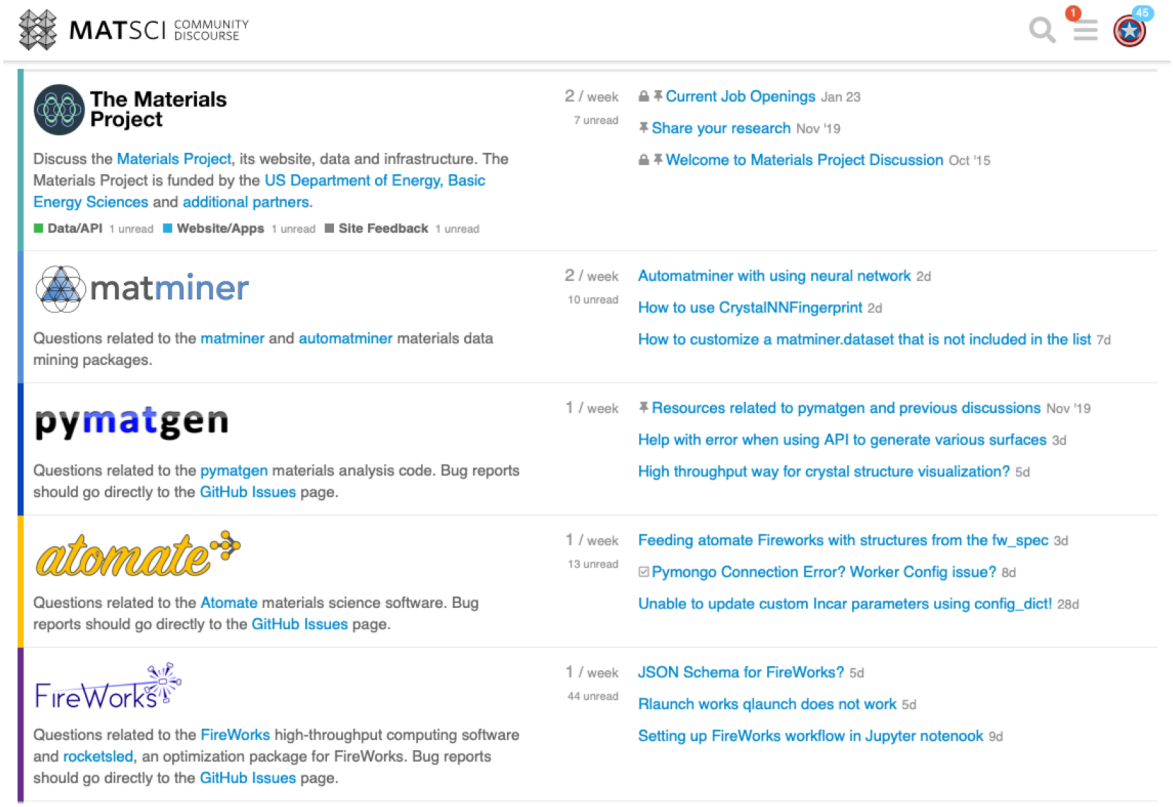
2019 MP Workshop - Atomate basics - Building a workflow

Online documentation

- Online documentation is the most comprehensive writeup
  - www.pymatgen.org
  - https://materialsproject.github.io/fireworks/
  - https://materialsproject.github.io/custodian/
  - https://hackingmaterials.github.io/atomate/
- The online documentation includes installation, examples, tutorials, and descriptions of how to use the code
- If you want to do “everything”, suggest starting with atomate and going from there

Help lists

- A help forum for each code is at https://discuss.matsci.org



Note: matsci.org now includes help forums for ~25 different codes!

Including LAMMPS, ASE, NOMAD, and many more

Questions? Comments?

A big thank you to all the contributors of the various codes!

- Pymatgen: 182 contributors
  - https://github.com/materialsproject/pymatgen/graphs/contributors
- Fireworks: 46 contributors
  - https://github.com/materialsproject/fireworks/graphs/contributors
- Atomate: 36 contributors
  - https://github.com/hackingmaterials/atomate/graphs/contributors

Funding through the U.S. Department of Energy, Basic Energy Sciences, Materials Science Division

Slides (already) posted to hackingmaterials.lbl.gov

Appendix slides

Network security issues

LAUNCHPAD
(MongoDB)

FIREWORKER
(computing resource)

LaunchPad and FireWorker within the same network firewall

→ *Works great*

LAUNCHPAD
(MongoDB)

FIREWORKER
(computing resource)

LaunchPad and FireWorker separated by firewall, BUT login node of FireWorker is open to MongoDB connection

→ *Works great if you have a MOM node type structure*

→ *Otherwise "offline" mode is a non-ideal but viable option*

LAUNCHPAD
(MongoDB)

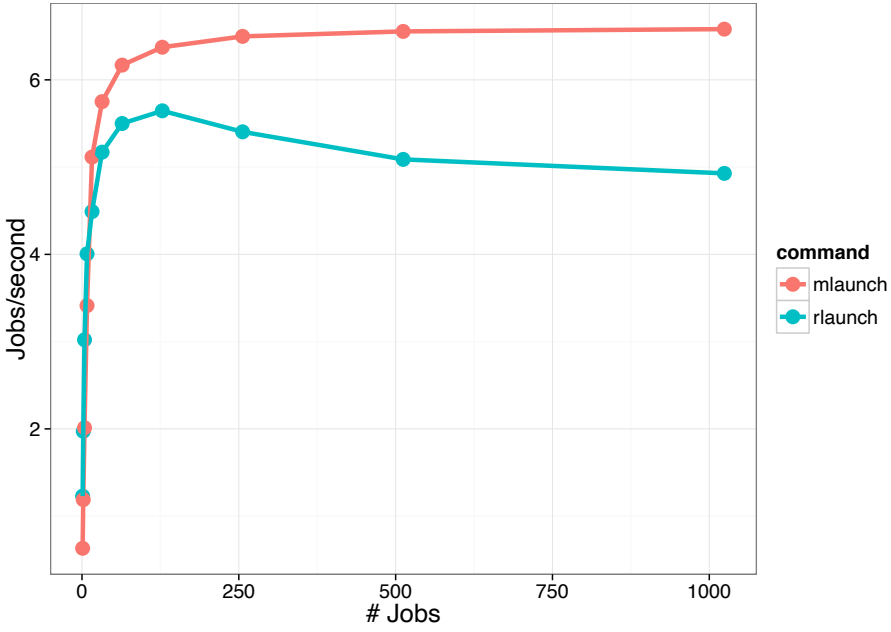
FIREWORKER
(computing resource)

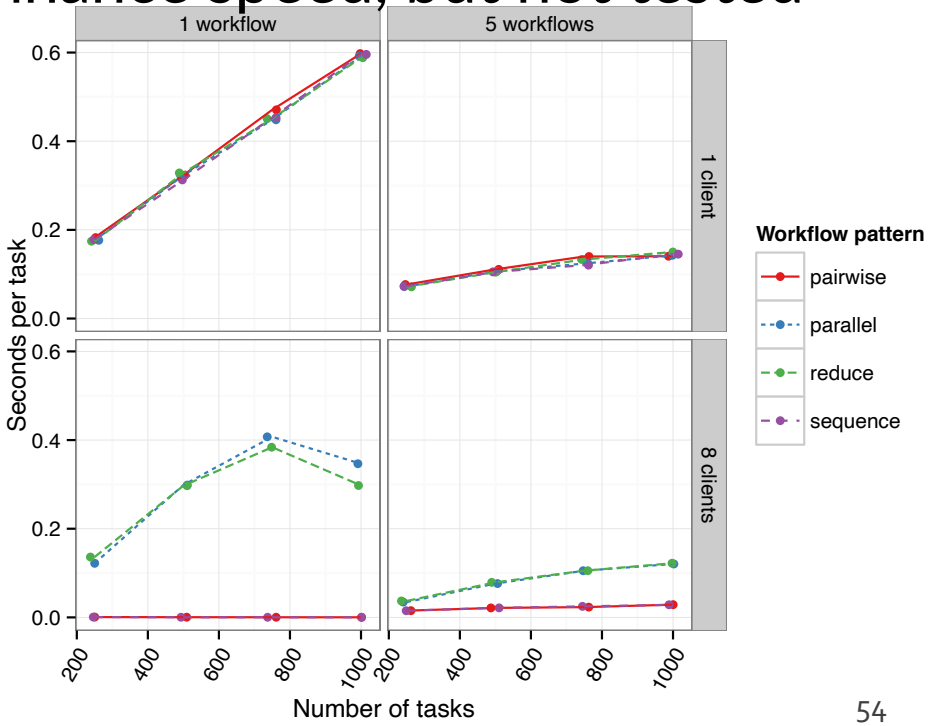
LaunchPad and FireWorker separated by firewall, no communication allowed

→ *Doesn't work!*

Performance issues

- Tests indicate the FireWorks can handle a throughput of about 6-7 jobs finishing per second
- Overhead is 0.1-1 sec per task
- Recently changes might enhance speed, but not tested





Job packing issues

- Computing center issues
  - Almost all computing centers limit the number of “mpirun”-style commands that can be executed within a single job
  - Typically, this sets a limit to the degree of job packing that can be achieved
  - Currently, no good solution; may need to work on “hacking” the MPI communicator. e.g., “wraprun” is one effort at Oak Ridge.