The ADES model and the AiiDA infrastructure for Computational Materials Science

Giovanni Pizzi
Theory and Simulation of Materials, EPFL, Switzerland

THEOS
THEORY AND SIMULATION OF MATERIALS

MARVEL

EPFL
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

MAX DRIVING THE EXASCALE TRANSITION

MARVEL, MaX, etc..

http://nccr-marvel.ch

- **MARVEL - National Centre on Computational Design and Discovery of Novel Materials (Switzerland)**
  - 2014 to 2026 (3 phases of 4 years)
  - 39 PIs
  - **Hardware platform** (@CSCS) + **Software** platform and dissemination:
    - **AiiDA: materials informatics platform**
    - **Materials Cloud: dissemination** of tools, curated properties, data, and **workflows**
    - Domain-specific libraries

MARVEL, MaX, etc..

MARVEL

http://nccr-marvel.ch

- **MARVEL - National Centre on Computational Design and Discovery of Novel Materials (Switzerland)**
  - 2014 to 2026 (3 phases of 4 years)
  - 39 PIs
  - **Hardware platform** (@CSCS) + **Software** platform and dissemination:
    - **AiiDA: materials informatics platform**
    - **Materials Cloud: dissemination** of tools, curated properties, data, and **workflows**
    - Domain-specific libraries

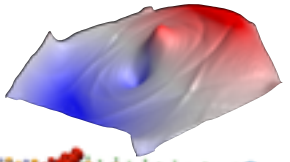
MAX DRIVING THE EXASCALE TRANSITION

http://max-center.eu

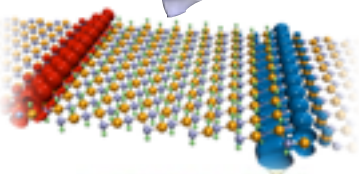
- **MaX - Materials Design at the Exascale (Europe)**
  - one of the 3 EU H2020 e-Infra Centres of Excellence dedicated to materials
  - 2015 to 2018 - EU thinking at renewals/consolidation
  - 10 groups, 5 Supercomputing Centres, 5 codes: i-PI, Quantum ESPRESSO, SIESTA, FLEUR, YAMBO
  - Exascale through HPC and HTC, via the creation of **workflows and turn-key solutions for the computation of materials properties**

Some current applications and workflows

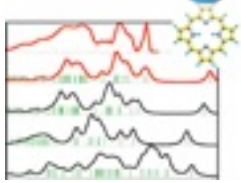
Phonon-phonon scattering in 2D — *Phonon hydrodynamics in 2D materials*



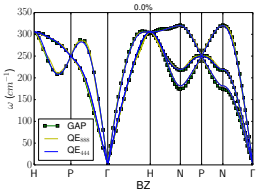
1D metallic wires at interfaces — *Engineering polar discontinuities in 2D*



Functional development — *Development of a Koopmans' compliant functional*

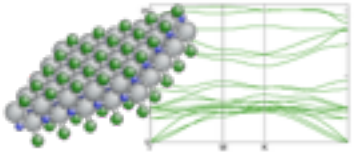


Neural Network potentials — *Generating databases for neural network potentials*

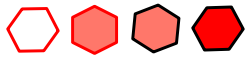


Pseudopotential database — *Creation of a Standard Solid State Pseudopotentials library*

Thermodynamical properties of 2D materials — *Discovering 2D materials and creating a database of their thermodynamical properties*



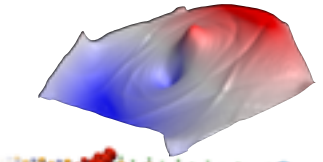
Paraelectric-ferroelectric transition in perovskites, ...



Some current applications and workflows

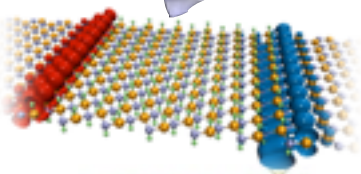
Phonon-phonon scattering in 2D

Phonon hydrodynamics in 2D materials



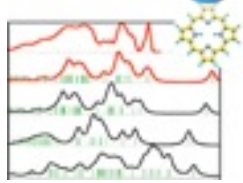
1D metallic wires at interfaces

Engineering polar discontinuities in 2D



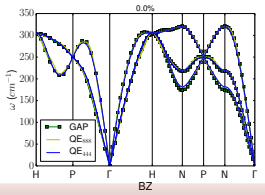
Functional development

Development of a Koopmans' compliant functional



Neural Network potentials

Generating databases for neural network potentials

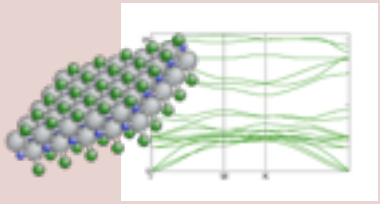


Pseudopotential database

Creation of a Standard Solid State Pseudopotentials library

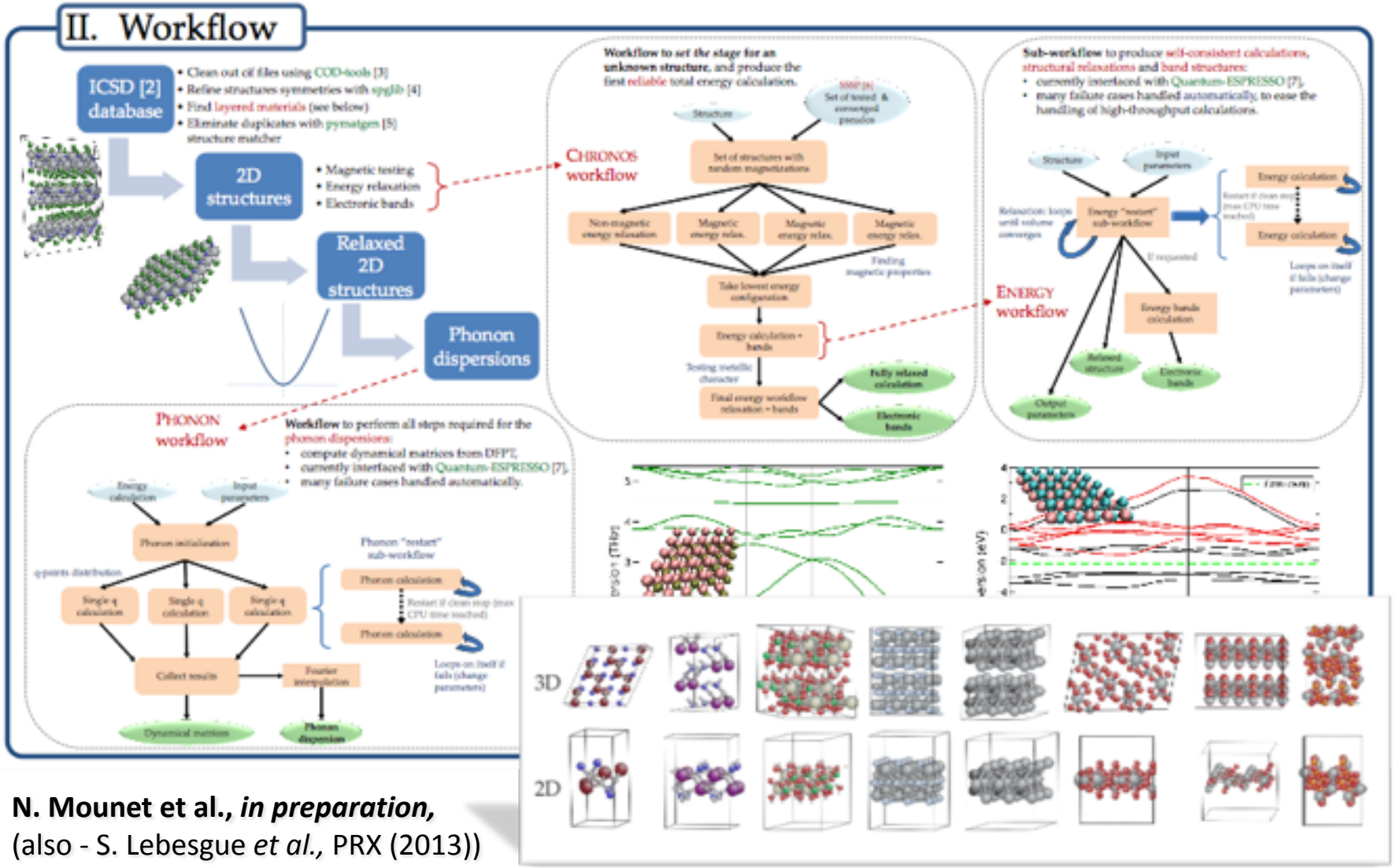
Thermodynamical properties of 2D materials

Discovering 2D materials and creating a database of their thermodynamical properties



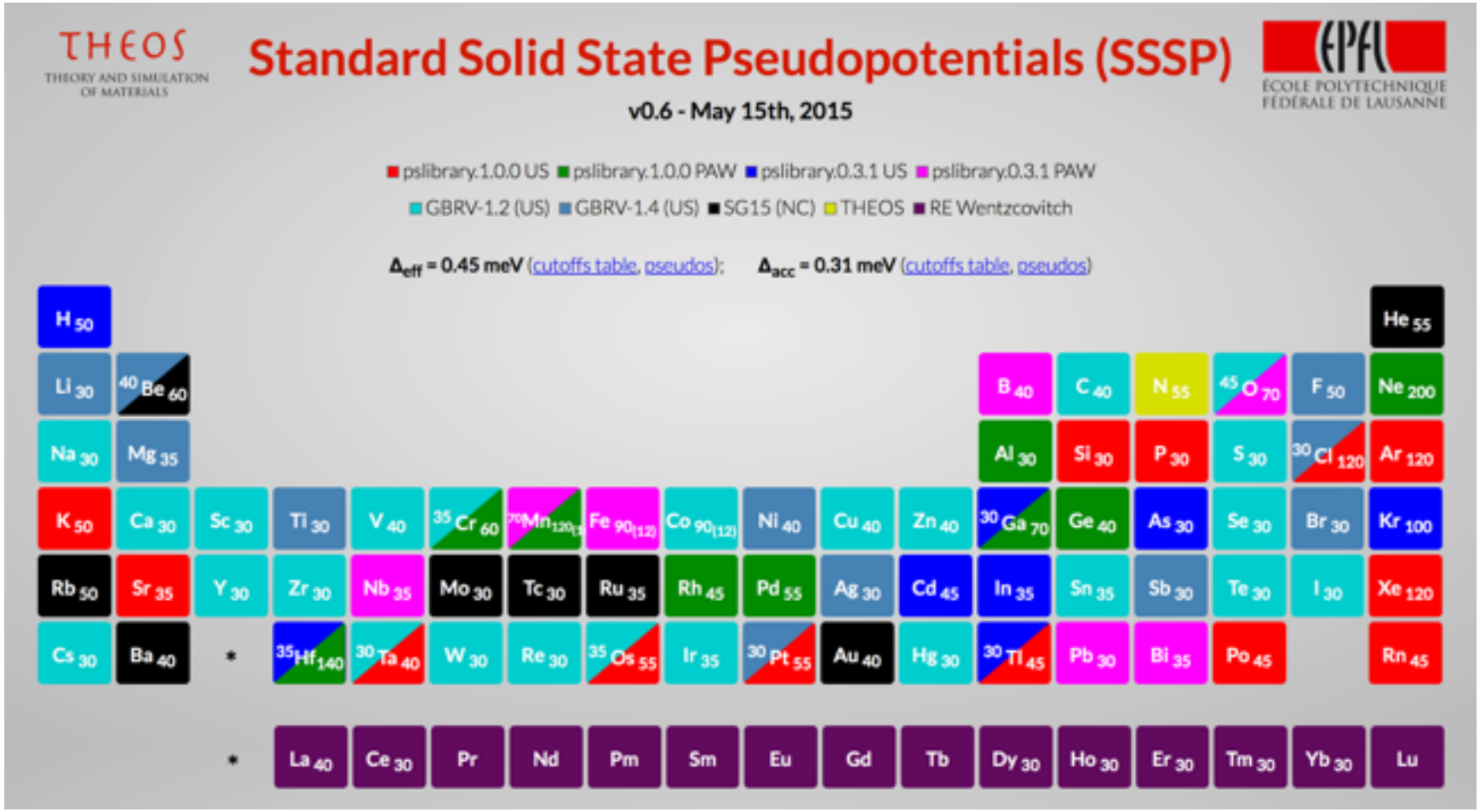
Paraelectric-ferroelectric transition in perovskites, ...

Example 1: discovery of novel 2D materials



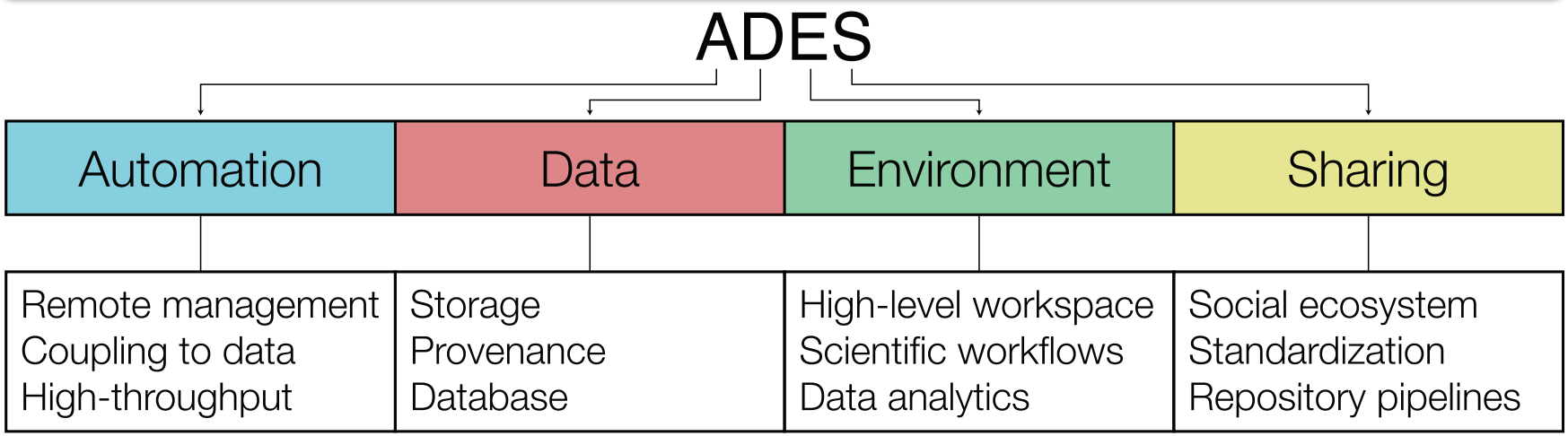
N. Mounet et al., *in preparation,*
(also - S. Lebesgue *et al.,* PRX (2013))

Example 2: Standard Solid State Pseudopotentials



I. Castelli, N. Mounet, and N. Marzari, *in preparation* (2015)

Developing novel models for computational science

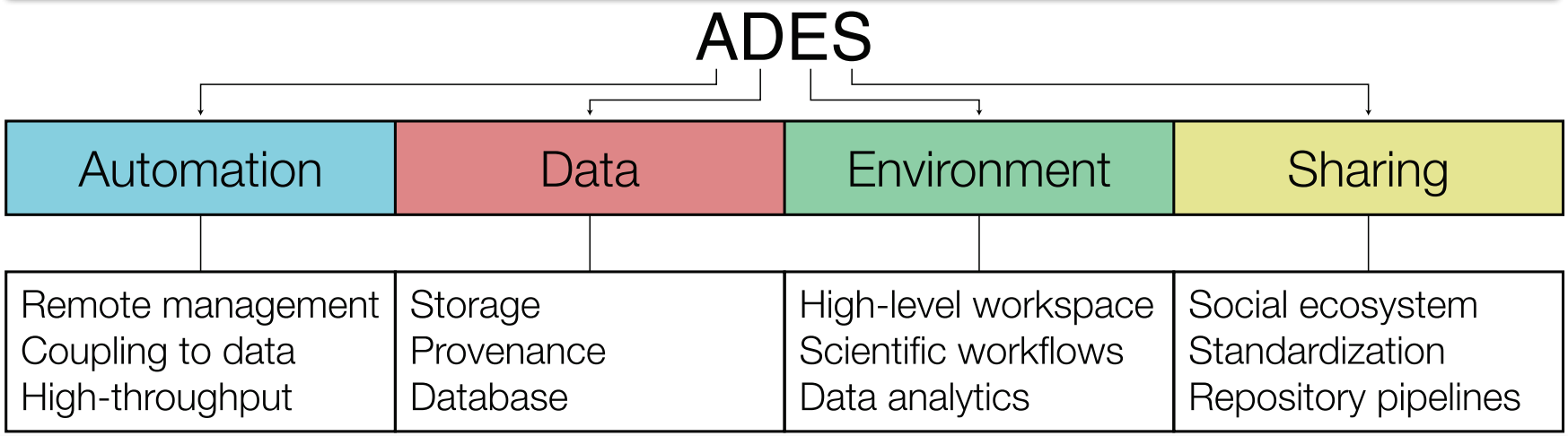


In collaboration with Robert Bosch RTC, Cambridge (MA)
open-source BSD-like license

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

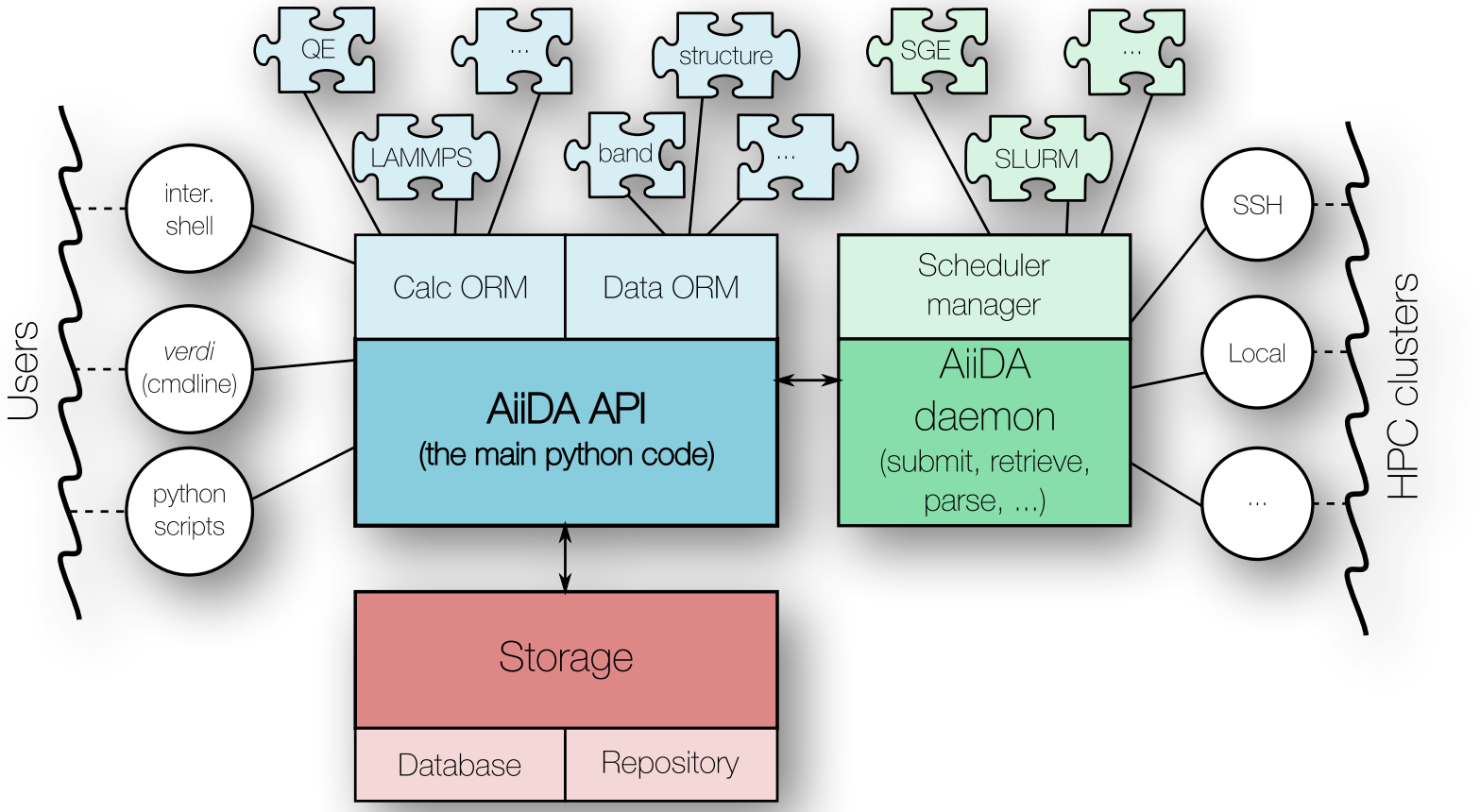
www.aiida.net

Materials informatics objectives



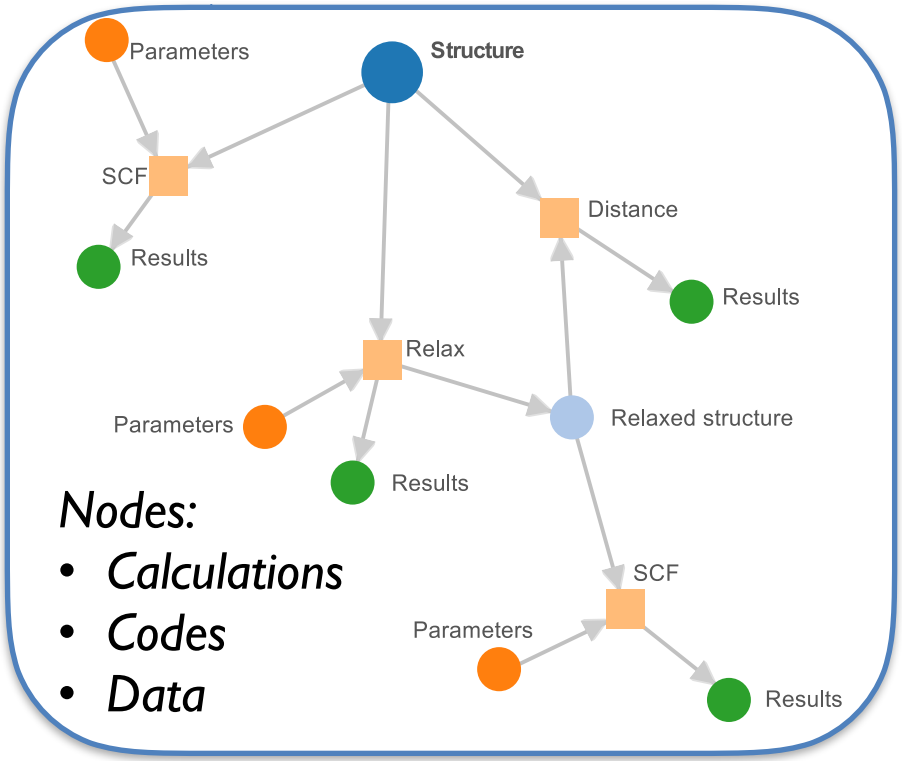
- **Automation** – thousands of calculations daily
- **Provenance** – we need to know how data were produced, and what they were used for
- **Reproducibility** – we might go back to a simulation years later, and redo it with new parameters/tools
- **Data and metadata –** key are "(generalized) structure" and "properties"
- **Workflows** –these are the "turn-key solutions" that generate calculated properties
- **Sharing** - platforms to disseminate workflows, data, codes

AiiDA structure



The core of the code is the **AiiDA API** (Application Programming Interface), a set of Python classes that exposes the users to the key objects: **Calculations, Codes, and Data.**

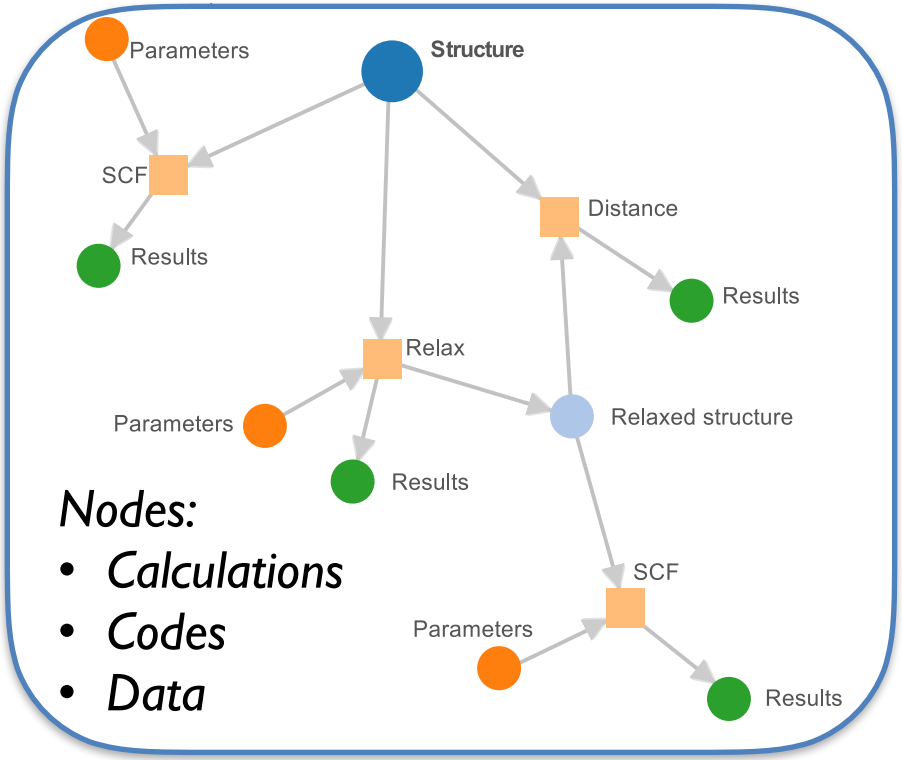
Storing the *provenance*: Directed Acyclic Graphs



Nodes:

- *Calculations*
- *Codes*
- *Data*

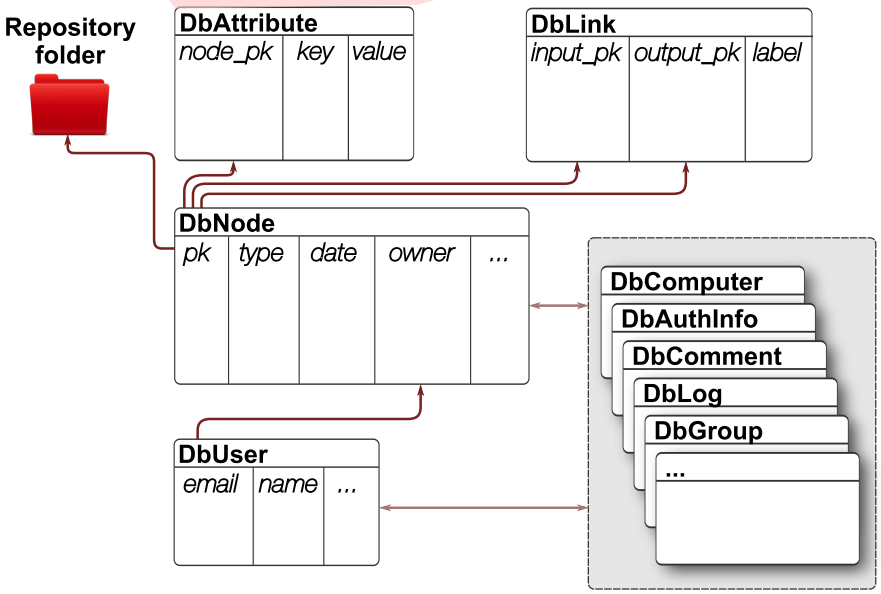
Storing the *provenance*: Directed Acyclic Graphs



Nodes:

- *Calculations*
- *Codes*
- *Data*

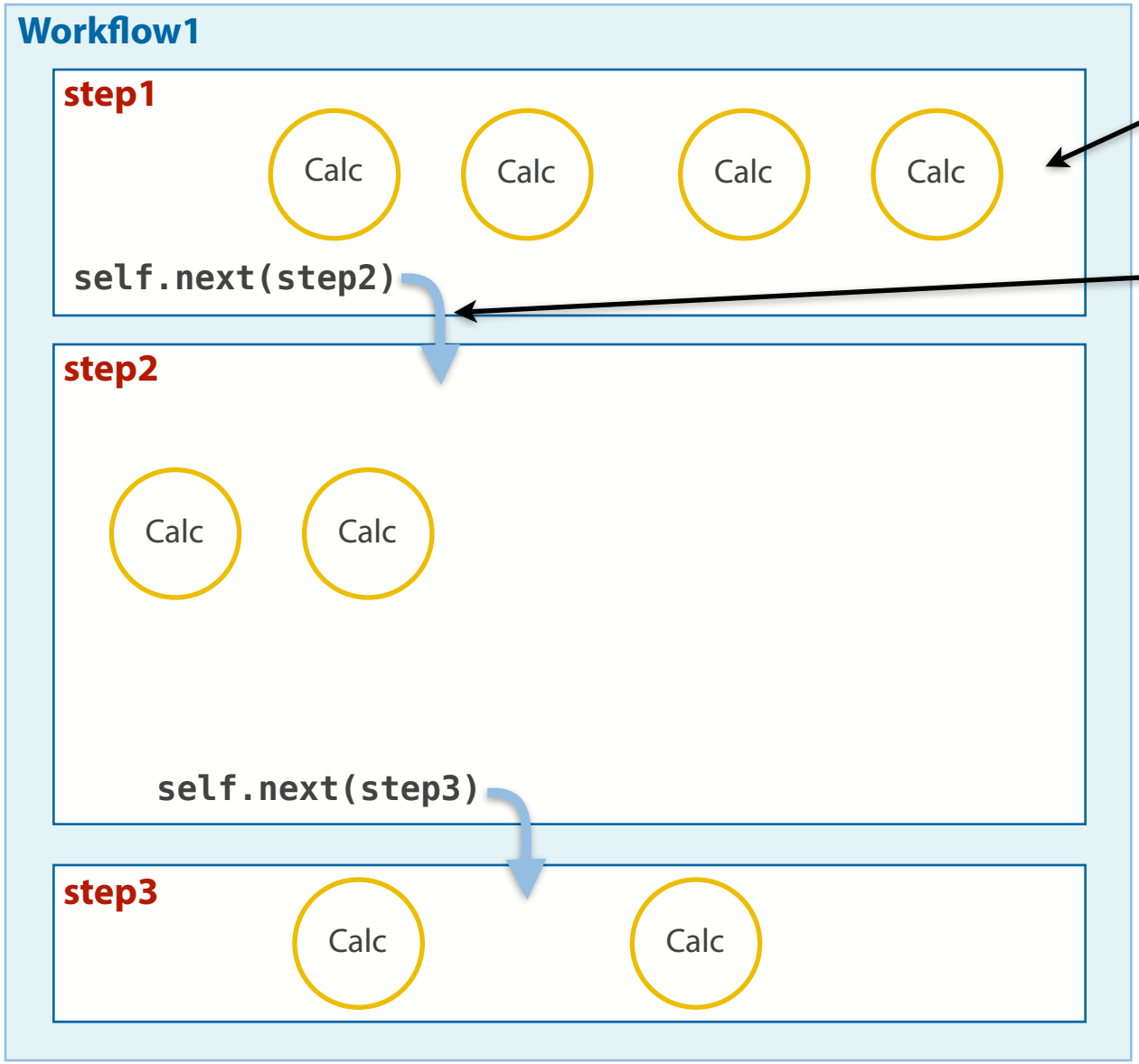
	node_pk	key	value
pk=7 Calc	7	num_cpus	48
	7	queue_name	“private”
	7	submission_time	May 2nd, 2014 13:46:07
pk=8 Results	8	energy	-13.736229
	8	energy_units	“eV”
	8	forces	[[4.32, 3.22, 0.73]], [2.23, -1.46, 0.22], ...]
	...		



With **10millions+ nodes**, we need appropriate techniques and database backends to store and query the results!

In AiiDA: migration to SQLAlchemy + JSONB fields, graph-oriented DBs (Neo4j, TitanDB)

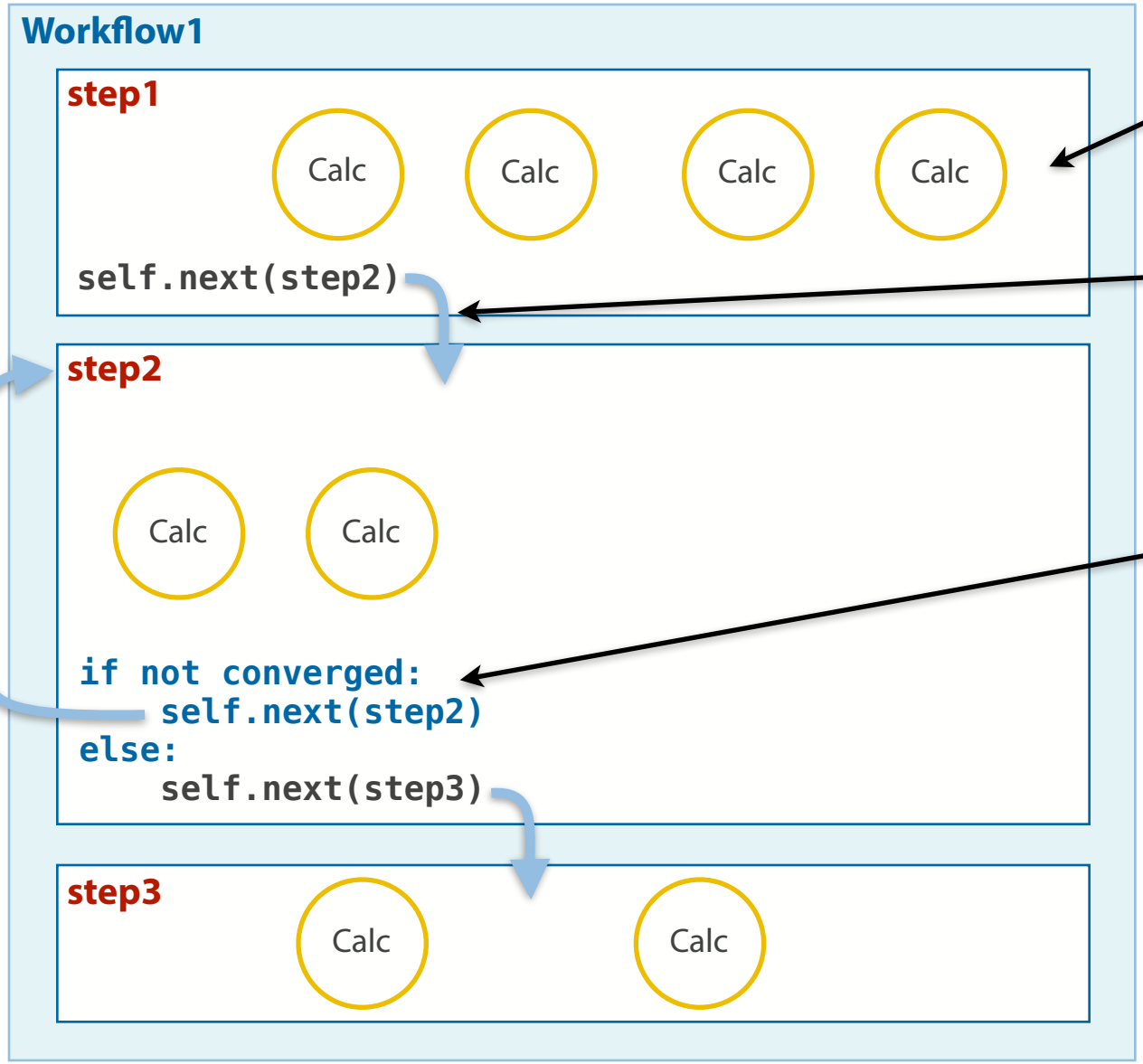
Environment in AiiDA: Workflows



Calculations in the same step: run in **parallel**

Different steps: run in **serial**; relationships via a **.next()** method

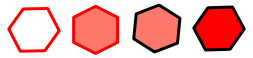
Environment in AiiDA: Workflows



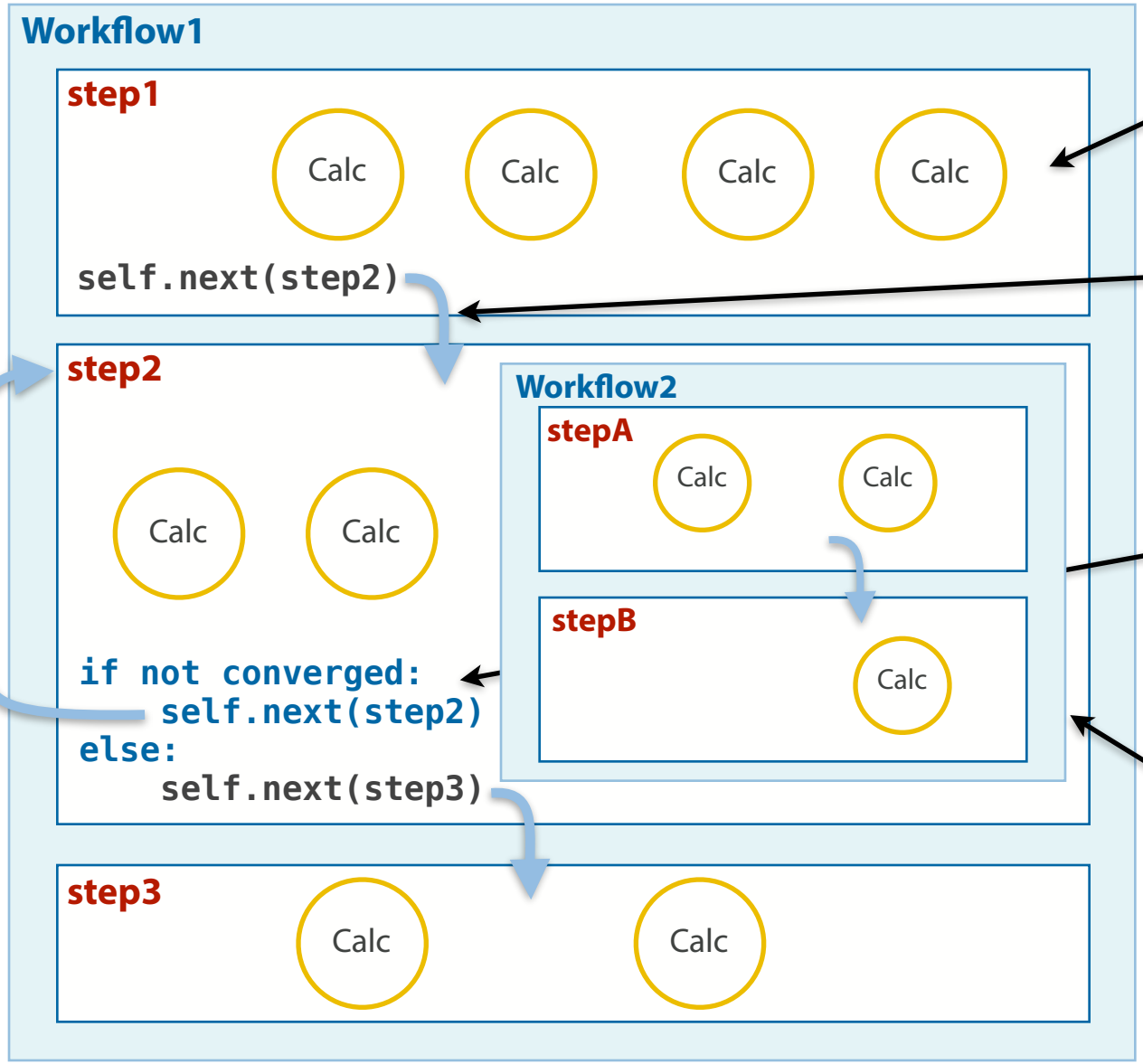
Calculations in the same step: run in **parallel**

Different steps: run in **serial**; relationships via a **.next()** method

Python logic (`if`, `for`, ...) possible: *conditional loops, convergence, ...*



Environment in AiiDA: Workflows



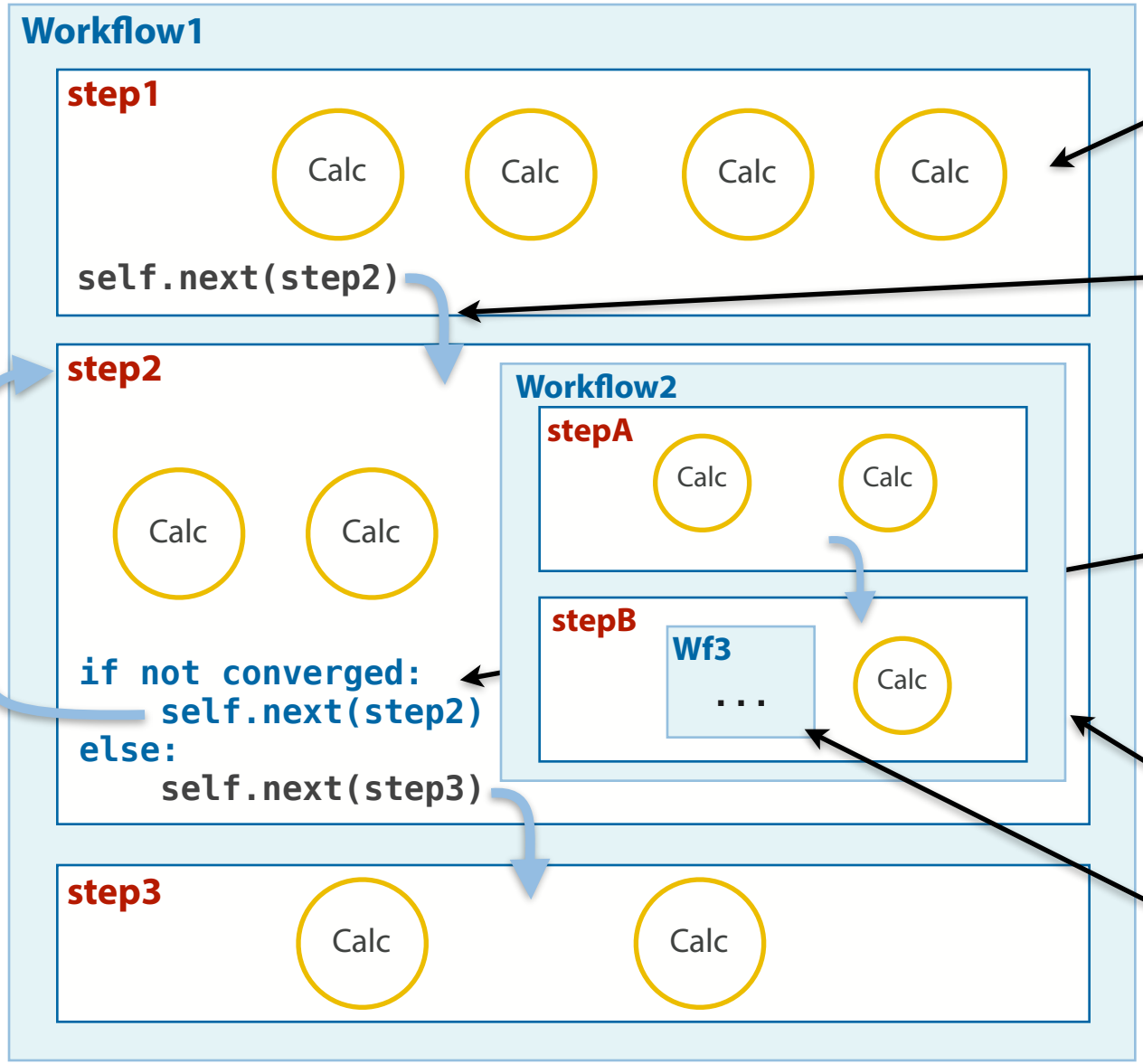
Calculations in the same step: run in **parallel**

Different steps: run in **serial**; relationships via a **.next()** method

Python logic (`if`, `for`, ...) possible: *conditional loops, convergence, ...*

Within a step: also subworkflows

Environment in AiiDA: Workflows



Calculations in the same step: run in **parallel**

Different steps: run in **serial**; relationships via a **.next()** method

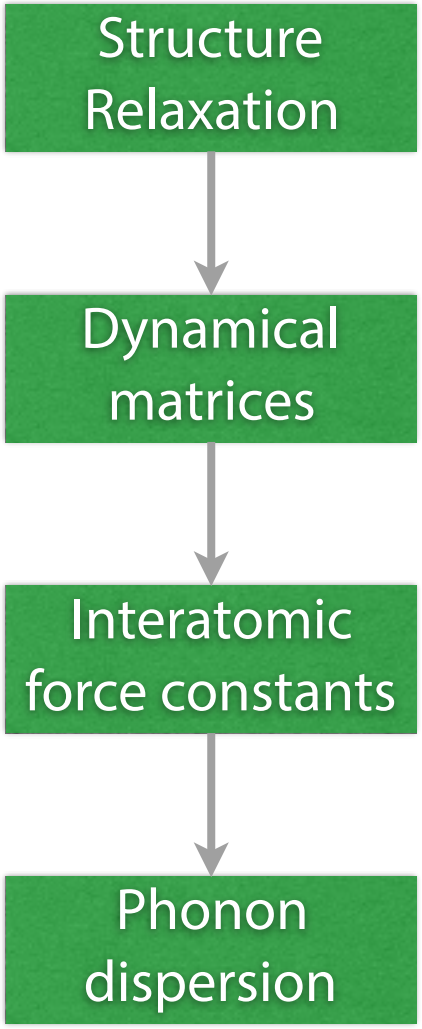
Python logic (`if`, `for`, ...) possible: *conditional loops, convergence, ...*

Within a step: also subworkflows

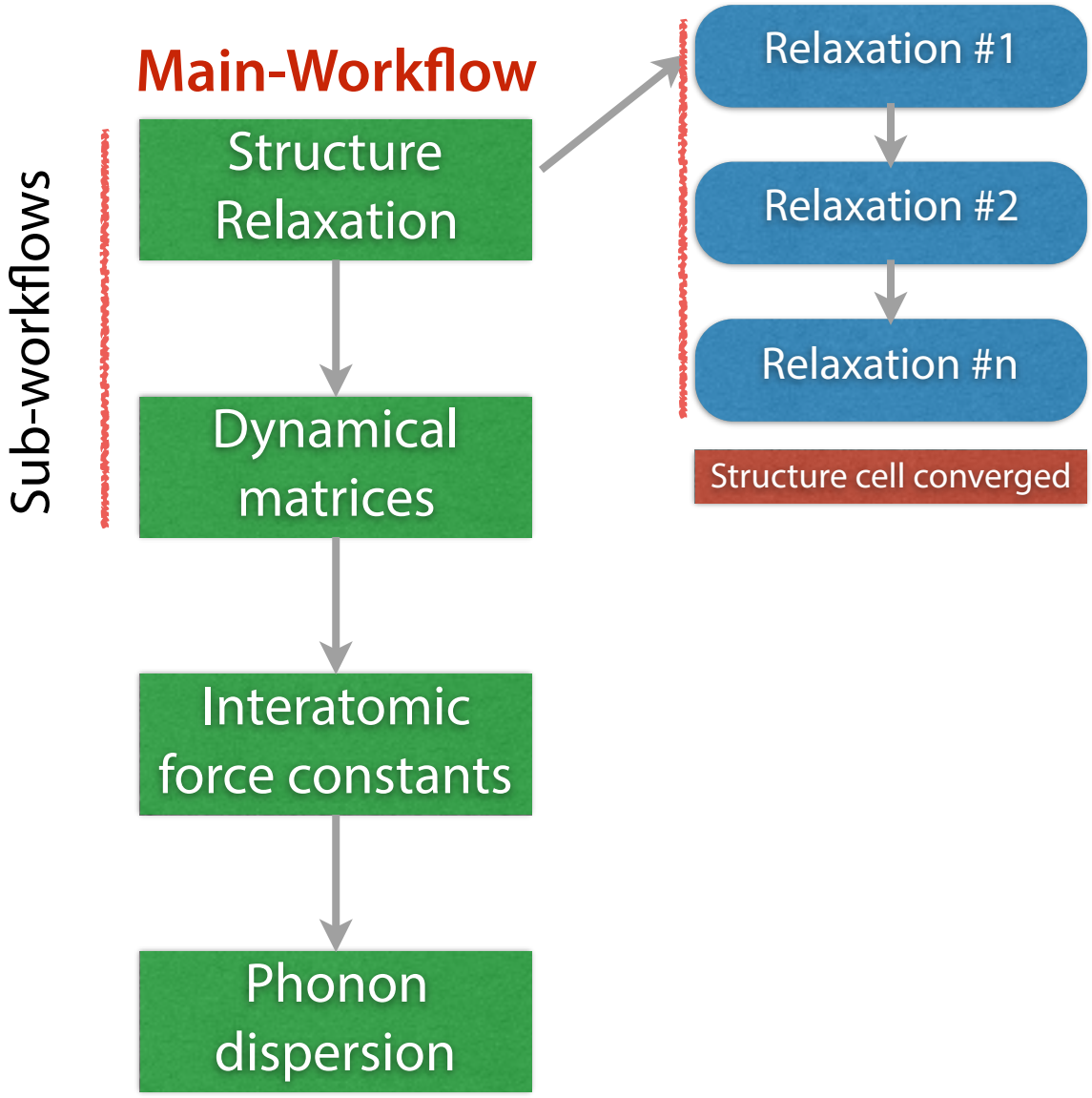
Any nesting level

A real-life workflow example: phonon dispersions

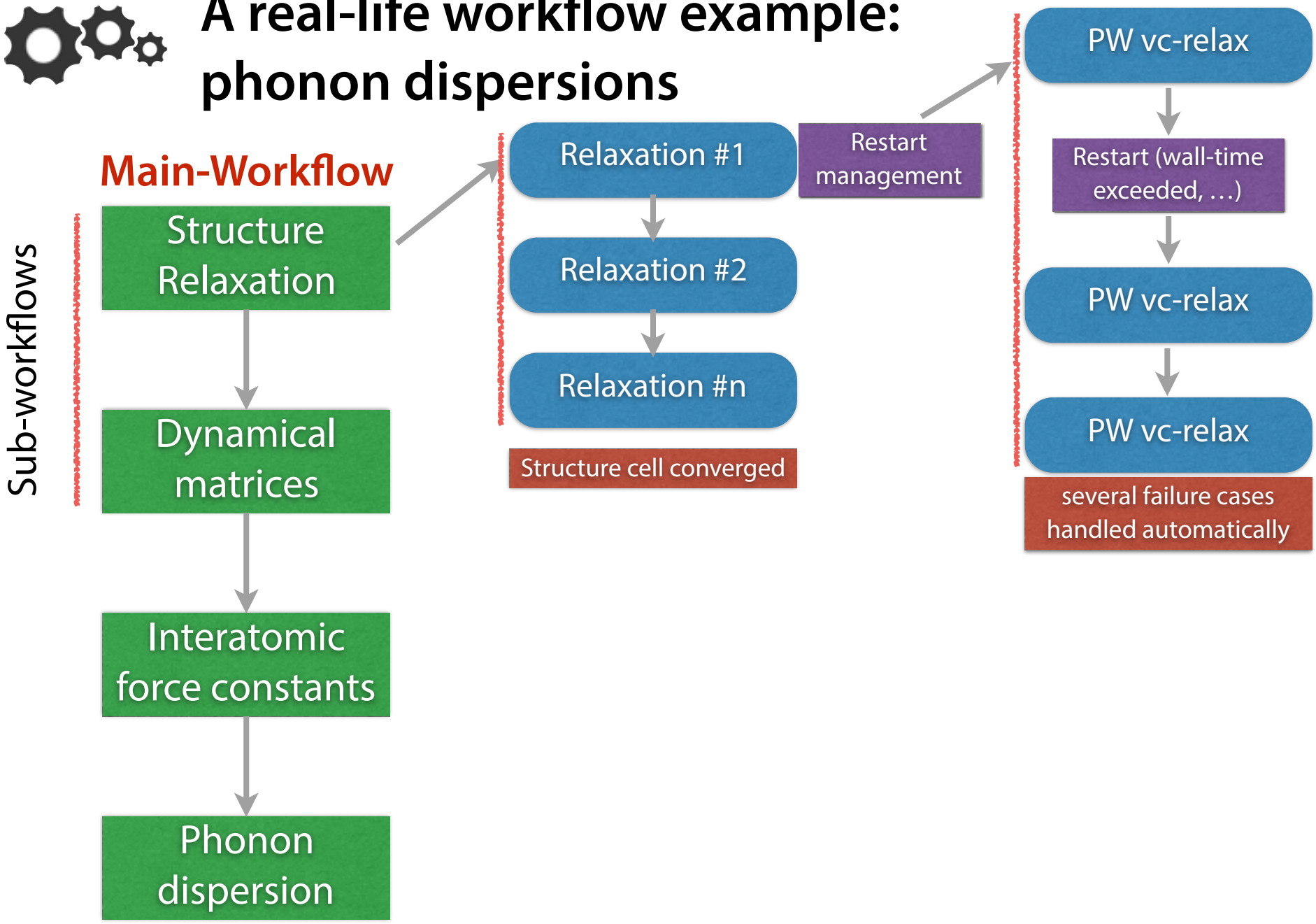
Main-Workflow



A real-life workflow example: phonon dispersions



A real-life workflow example: phonon dispersions



A real-life workflow example: phonon dispersions

Sub-workflows

Main-Workflow

- Structure Relaxation
- Dynamical matrices
- Interatomic force constants
- Phonon dispersion

Structure Relaxation:

- Relaxation #1
- Relaxation #2
- Relaxation #n

Structure cell converged

Restart management:

- PW vc-relax
- Restart (wall-time exceeded, …)
- PW vc-relax
- PW vc-relax

several failure cases handled automatically

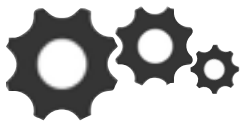
Dynamical matrices:

- Initialize PH
- PH on q-grid
- Collect phonons

Parallelization:

- PH on $\mathbf{q}_1$
- PH on $\mathbf{q}_2$
- PH on $\mathbf{q}_n$

A real-life workflow example: phonon dispersions



Main-Workflow

Sub-workflows

Single calculations

Structure Relaxation

Dynamical matrices

Interatomic force constants

Phonon dispersion

Relaxation #1

Relaxation #2

Relaxation #n

Structure cell converged

Restart management

PW vc-relax

Restart (wall-time exceeded, …)

PW vc-relax

PW vc-relax

several failure cases handled automatically

Initialize PH

PH on q-grid

Collect phonons

Parallelization

PH on $\mathbf{q}_1$

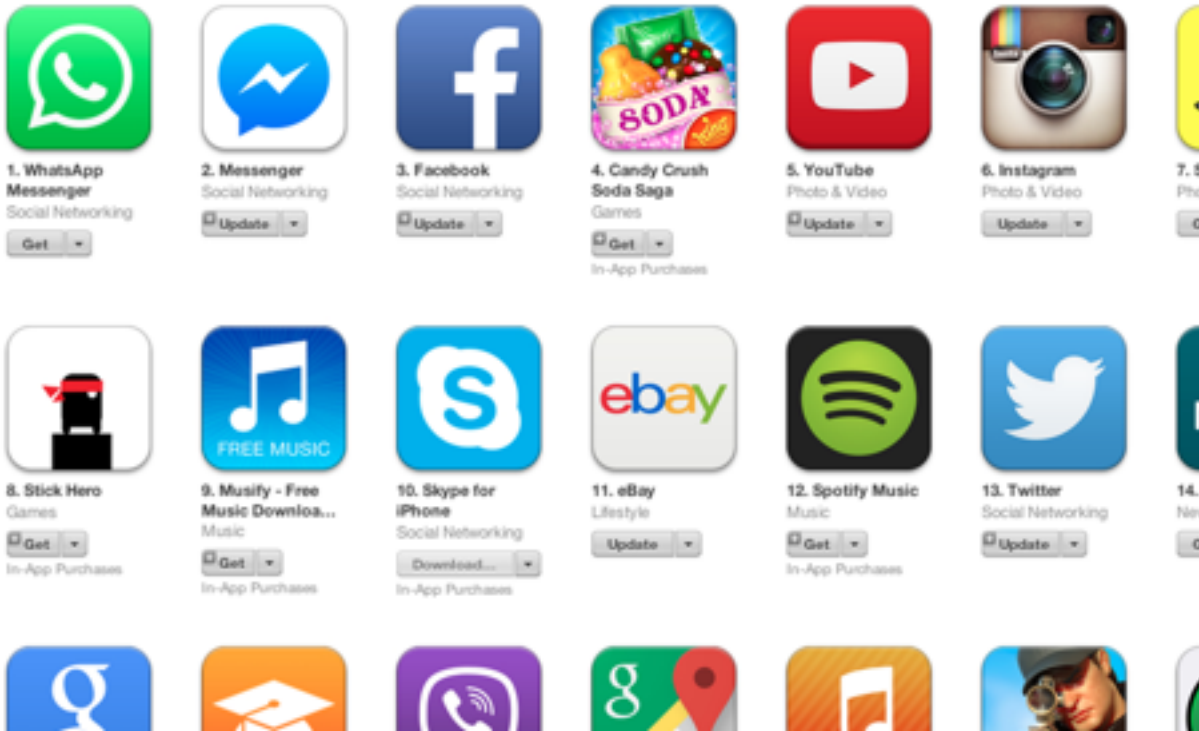
PH on $\mathbf{q}_2$

PH on $\mathbf{q}_n$

Outlook: App store model

App-store (@Apple) model for Plugins & Workflows, e.g.

- **Computers**: automatically setup a new cluster or supercomputer
- **DB importers**: load structures and data from COD, ICSD, …
- **Calculations**: find plugin to support your favorite software (Quantum ESPRESSO, VASP, GPAW, Yambo, …)
- **Turn-key solutions**: workflows to compute a desired property, with dependencies (see *pip install*)
- …



Acknowledgements and level of effort

The AiiDA team

Giovanni Pizzi (EPFL)

Andrea Cepellotti (EPFL)

Riccardo Sabatini (EPFL)

Nicola Marzari (EPFL)

Boris Kozinsky (BOSCH)

Nicolas Mounet (EPFL)

Andrius Merkys (Vilnius)

Martin Uhrin (EPFL)

Spyros Zoupanos (EPFL)

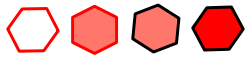
Snehal Waychal (EPFL)

Leonid Kahle (EPFL)

Plugin contributors — **Quantum ESPRESSO NEB**: Marco Gibertini (EPFL); **Quantum ESPRESSO DOS, PDOS; Wannier90**: Daniel Marchand (EPFL); **CP2K**: Aliaksandr Yakutovich (EMPA), Uli Schauer (ETHZ), Tiziano Müller, Andreas Glosse, Patrick Seewald (UZH); **FLEUR**: Jens Broeder, Gregor Michalicek, Daniel Wortmann (Jülich); **Exciting**: Anton Kozhevnikov (CSCS); **YAMBO**: Andrea Ferretti, Giovanni Borghi, Daniele Varsano (CNR-NANO), Gianluca Prandini (EPFL); **SIESTA**: Victor Garcia Suarez (Uni Oviedo); **i-PI**: Venkat Kali (EPFL); **VASP**: Mario Zic (Trinity College Dublin).
Contributors — Prof. Christoph Koch, Jocelyn Boullier (EPFL); Valentin Bersier, Philippe Schwaller (THEOS EPFL); Marco Dorigo (ICAMS - Bochum); Eric Hontz (MIT & Bosch RTC)
Early beta testers — Giovanni Borghi, Ivano Castelli, Marco Gibertini (THEOS EPFL); Prateek Mehta (Bosch RTC)

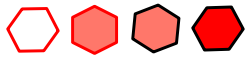
Our workflow requirements (implemented in AiiDA)

- *Easy to write* a new plugin for scientists
  (possibly in Python with access to libraries as numpy, spglib, …)



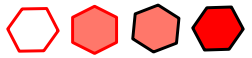
Our workflow requirements (implemented in AiiDA)

- *Easy to write* a new plugin for scientists
  (possibly in Python with access to libraries as numpy, spglib, …)
- *Easy to* build on existing plugins and *extend functionality*



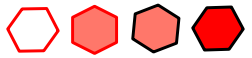
Our workflow requirements (implemented in AiiDA)

- *Easy to write* a new plugin for scientists
  (possibly in Python with access to libraries as numpy, spglib, …)
- *Easy to* build on existing plugins and *extend functionality*
- Integrated *management of remote calculations* on supercomputers



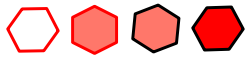
Our workflow requirements (implemented in AiiDA)

- *Easy to write* a new plugin for scientists
  (possibly in Python with access to libraries as numpy, spglib, …)
- *Easy to* build on existing plugins and *extend functionality*
- Integrated *management of remote calculations* on supercomputers
- On-the-fly *storage of full provenance* in the graph database



Our workflow requirements (implemented in AiiDA)

- *Easy to write* a new plugin for scientists
  (possibly in Python with access to libraries as numpy, spglib, …)
- *Easy to* build on existing plugins and *extend functionality*
- Integrated *management of remote calculations* on supercomputers
- On-the-fly *storage of full provenance* in the graph database
- *"Dynamic" workflows*: graph not decided at the start, but depends on intermediate results
  (e.g. convergence workflows, …)



Our workflow requirements (implemented in AiiDA)

- *Easy to write* a new plugin for scientists
  (possibly in Python with access to libraries as numpy, spglib, …)
- *Easy to* build on existing plugins and *extend functionality*
- Integrated *management of remote calculations* on supercomputers
- On-the-fly *storage of full provenance* in the graph database
- *"Dynamic" workflows*: graph not decided at the start, but depends on intermediate results
  (e.g. convergence workflows, …)

Current efforts:

- Implement plugins and workflows for various codes and materials science applications
- Improve the workflow interface to scale up and to make them easier to develop and debug
- Allowing reuse of results of existing calculations in the DB (if calculations give the same results given the same inputs)