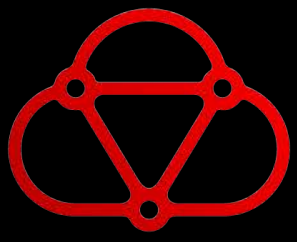
AiiDA AND MATERIALS CLOUD UPDATES

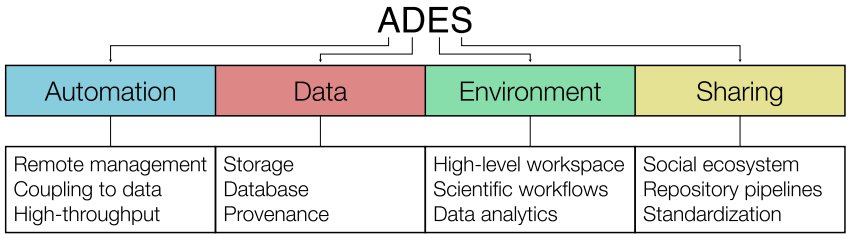
NICOLA MARZARI, EPFL



AiiDA: OBJECTIVES

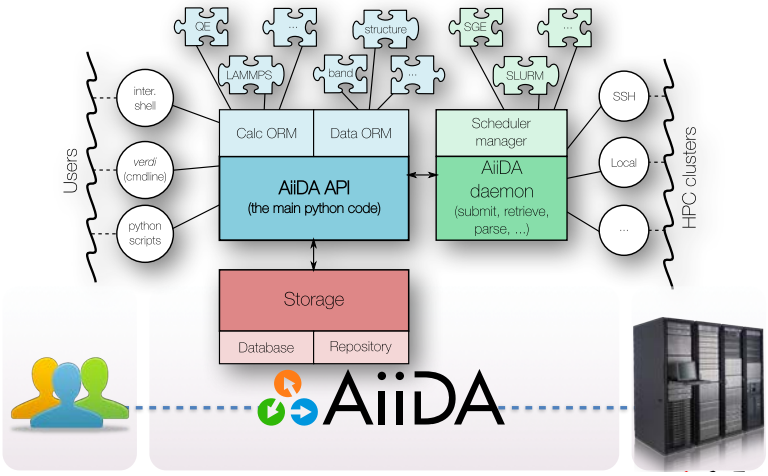
- **Automation**: run thousands of calculations daily
- **Provenance**: all children and all parent data are recorded
- **Reproducibility**: go back to a simulation years later, and redo it with new parameters or codes
- **Extensible/agnostic** to models, codes and formats
- **Workflows**: dynamical, robust, complex "turnkey solutions" that calculate desired properties on demand
- **Sharing**: provide the distributed environment to disseminate workflows and data and to provide services

ADES MODEL FOR COMPUTATIONAL SCIENCE



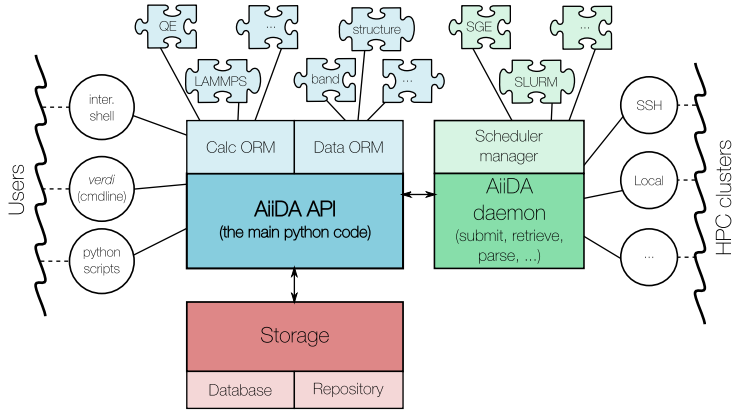
G. Pizzi et al., Comp. Mat. Sci 111, 218-230 (2016)

Automation in AiiDA



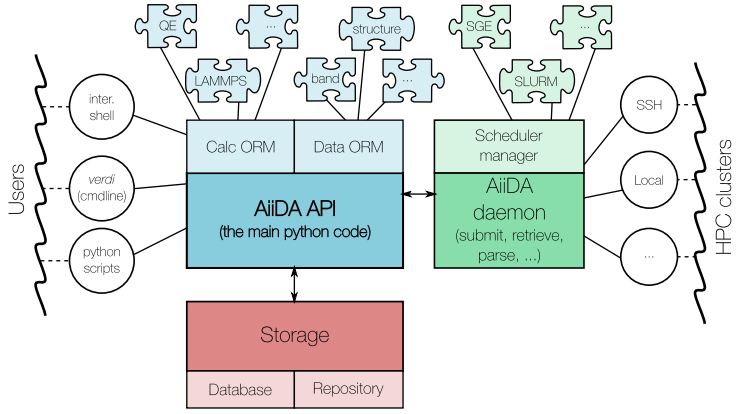
1. The core of the code is the **AiiDA API** (Application Programming Interface), a set of Python classes that exposes the users to the key objects: **Calculations, Codes, and Data.**

Automation in AiiDA



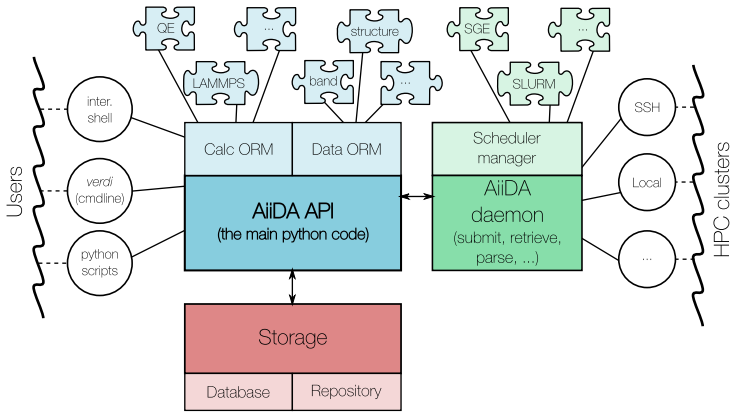
2. The **AiiDA Object-Relational Mapper** (ORM) maps AiiDA objects into Python Classes, so that the objects can be created/modified/queried via an agnostic high-level interface. Any interaction with Storage occurs transparently via Python calls.

Automation in AiiDA



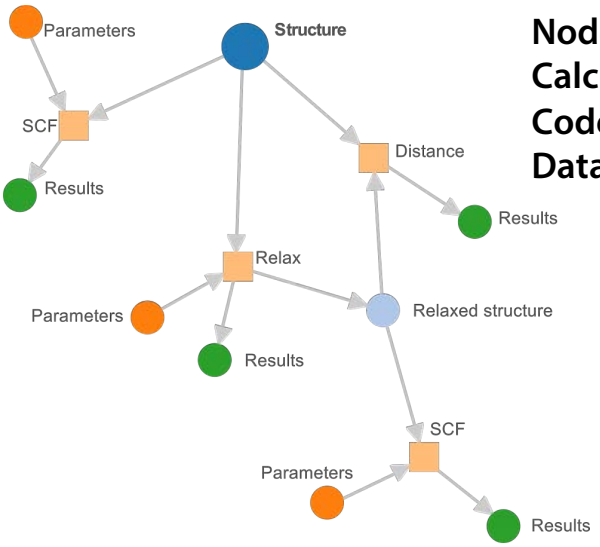
3. A daemon manages calculation states (submission, retrieval, parsing…) without user intervention (uses Python celery+supervisor modules), through remote transports and Slurm/PBS Pro/SGE/Torque plugins.

Automation in AiiDA



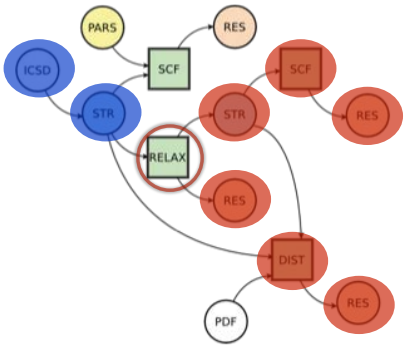
4. User interactions occurs via the command line tool **`Verdi`**, the interactive shell or via Python scripts

DIRECTED ACYCLIC GRAPHS



Nodes:
Calculations
Codes
Data

Saving the DAGs: Nodes and Links



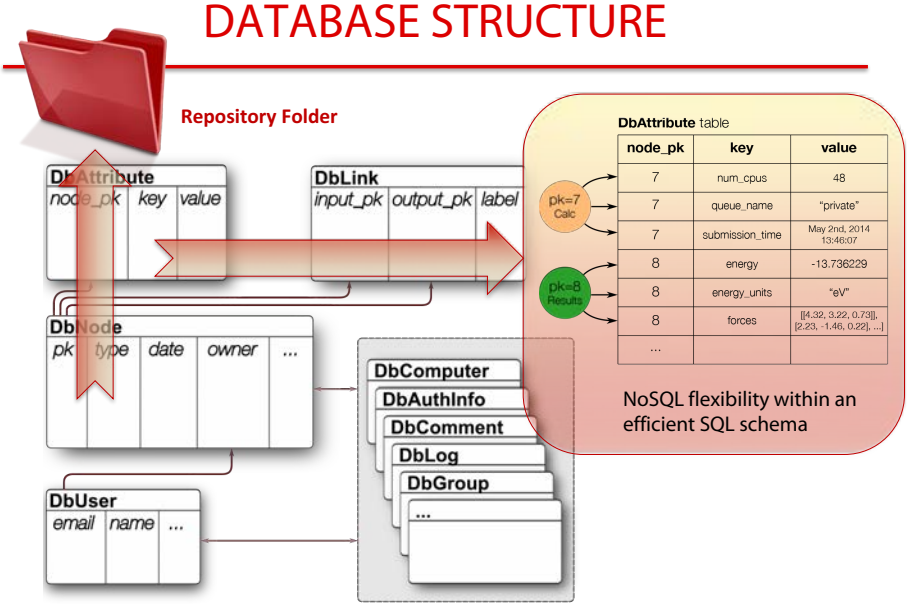
Nodes and links: a graph structure

- *Each node*: row in a SQL table + folder for files
- Links also stored in a SQL table ⇒ *provenance*

Transitive closure

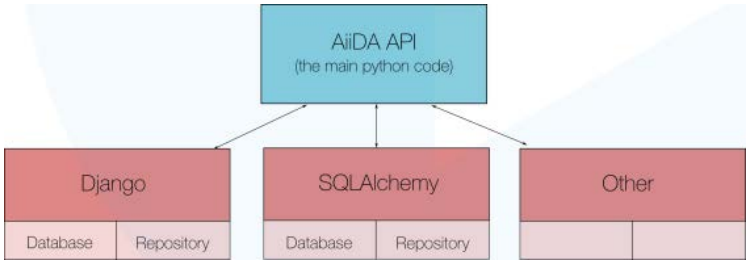
- Allows queries that *traverse the graph*
- Either by automatically updated table (via db triggers) or recursive common table expressions

DATABASE STRUCTURE



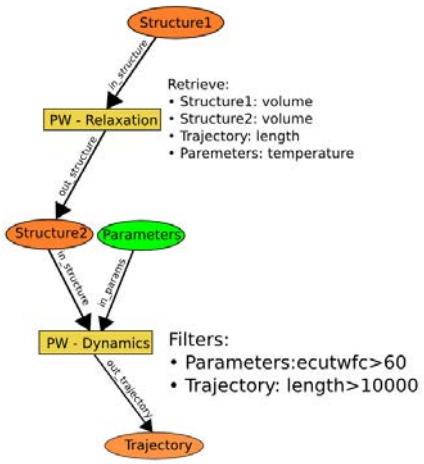
DbNode: entry for each node. **DbLink:** all links. Everything else in **DbAttribute** (+DbExtra for later).

MULTIPLE STORAGE-BACKEND SUPPORT



- AiiDA API decoupled from object-relational mapper
- **Two ORM implemented (Django and SQL Alchemy)**
- Flexible backend choice based on needs
- Easy incorporation of graph databases like Neo4J and Titan

TOOLS FOR QUERYING

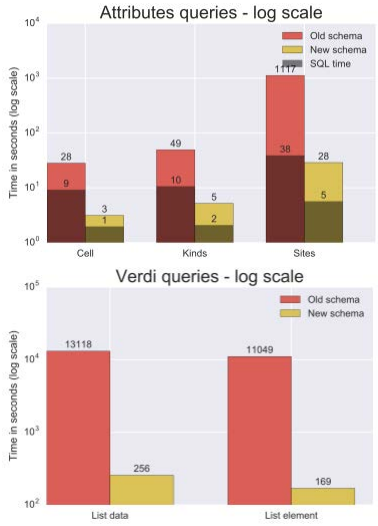


Support for complex graph queries

- Any combination of filtering and projections on AiiDA graph nodes
- Full graph traversal
- No SQL required from the user

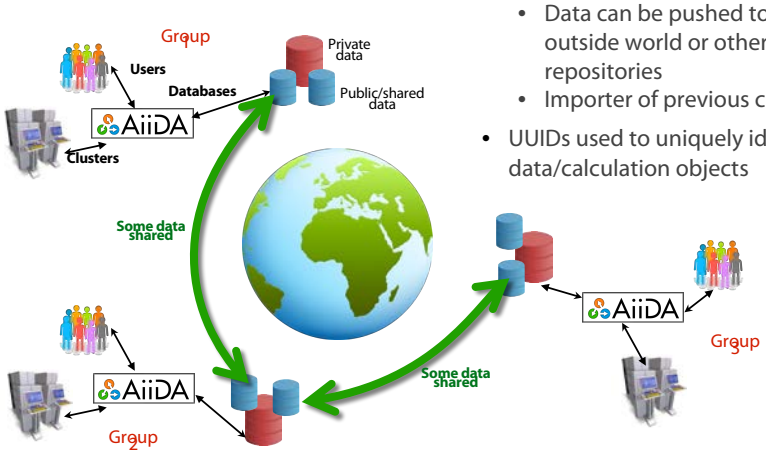
```
query = aiida.orm.QueryBuilder()
query.append(
        StructureData, tag='Structure1',
        project='attributes.volume')
query.append(
        PwCalculation, tag='Relaxation')
query.append(
        StructureData, tag='Structure2',
        project='attributes.volume')
query.append(
        PwCalculation, tag='Dynamics')
query.append(
          TrajectoryData, tag='Trajectory',
          project='attributes.length',
          filters={'attributes.length':{'>':10000}})
query.append(
          ParameterData, tag='Parameters',
          input_of='Dynamics',
          project='attributes.IONS.ion_temperature',
          filters={'attributes.SYSTEM.ecutwfc':{'>':60.}})
      )
# Retrieving all results in the database:
results=query.all()
```

NEW ORM (65x performance increase)



- AiiDA inherently supports JSON even with non-JSON enabled databases.
- New ORM SQLAlchemy has been added to store and query JSON in a JSON-enabled database.
- PostgreSQL supports JSON since version 9.3. Version 9.4 supports JSONB & indexing, version 9.5 inline JSON updates.
- Improved performance on queries and command line operations related to JSON encoded information.
- Experiments performed using PostgreSQL 9.4 on Dual Xeon E5606 (2.13GHz, 2x4 cores, 48GB RAM)
- **Database size: 948.233 nodes, 4.986.089 links.**

Sharing in AiiDA



- *Sharing model in AiiDA*
  - Data can be pushed to the outside world or other repositories
  - Importer of previous calculations
- UUIDs used to uniquely identify all data/calculation objects

Available commands - Sender

- Status
  - Changes since last commit
- Commit
  - A new commit is created with changes
- Push
  - A new sub-commit is created per destination

Commit visibility

- Commit only locally visible until push
- Push also notifies destinations for the update

Available commands - Receiver

- Fetch
  - Receives/fetches the sub-commits related to specific repository
- Merge
  - Merges the sub-commits to the main repository

Orchestration

- For every repository that there is a subscription, checks if there is an update
- SHA of the last sub-commit and last major commit are used to resolve status
- Sub-commits contain commit hash and node ids
- Nodes are retrieved independently

The REST API interface (+ OPTIMADE API)

https://aiidaserver/api/v1/dbnode/?type=calc&state=FINISHED&user__email=giovanni.pizzi@epfl.ch

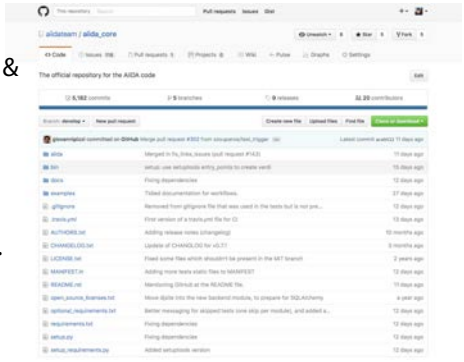
- *Features*:
  - *Full access* to the data
  - *Creation/editing* of settings
  - **Full query capability**
  - Programmatically access, submit and analyze data
  - JSON serialization (easy to use from Java, JavaScript, Python, ...)


```
{
  "meta":{
    "total_count":2
  },
  "objects":[
    {
      "resource_uri":"/aiida_test/api/v1/dbnode/24/",
      "label":"Test QE pw.x",
      "description":"Test calculation with the pw.x code",
      "dbcomputer":"daint",
      "plugin_string":"quantumespresso.pw",
      "class_name":"PwCalculation",
      "public":false,
      "state":"FINISHED",
      "user":"giovanni.pizzi@epfl.ch",
      "uuid":"3babfa26-faac-4bad-9851-cebd0608109f"
    },{
      "resource_uri":"/aiida_test/api/v1/dbnode/217/",
      "label":"Test QE pw.x with AiiDA",
      "description":"Test pw.x calculation with alat 4.079",
      "dbcomputer":"titan",
      "plugin_string":"quantumespresso.pw",
      "class_name":"PwCalculation",
      "public":false,
      "state":"FINISHED",
      "user":"giovanni.pizzi@epfl.ch",
      "uuid":"4fd4aea9-0581-408f-9fa1-599aaab3e334"
    }
  ]
}
```


GitHub & Travis CI

Advantages

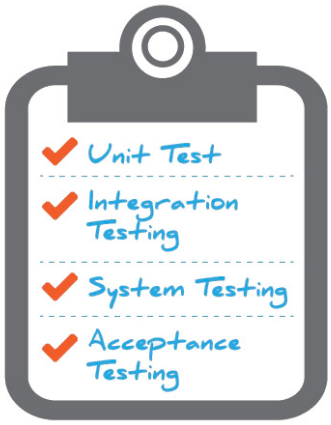
- Easy configuration of GitHub & Travis
- Very good integration of GitHub with Travis
- Good management of pull requests, issues, commits etc.
- Various and interesting statistics



Disadvantages

- Restriction to have your repository to GitHub
- Private repositories are charged

Testing levels



415 for django, 420 for SQLAlchemy, at each commit/pull request (in reality there are ~430-450 test functions, most of them running twice, once for each backend)

Unit tests

- Verify the functionality of a part of code (function, method, class)
- Isolated from the remaining system

Integration tests

- Verify the functionality of different components of the code

System tests

- End-to-end tests for a full system (security, performance, …)

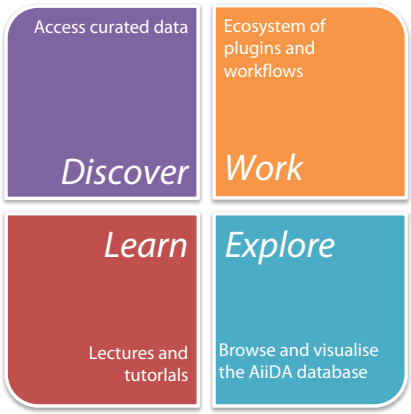
Acceptance tests

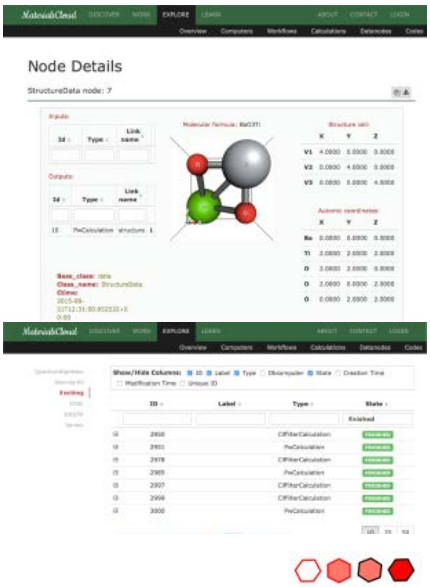
- Operational readiness tests / pre-release tests.

MATERIALS CLOUD

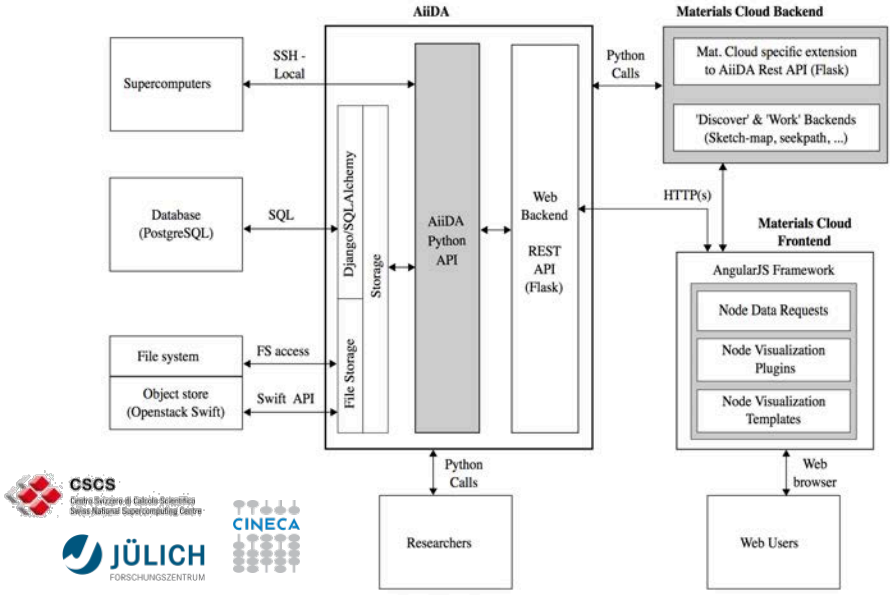
MATERIALS CLOUD

Portal grouped into four areas





MATERIALS CLOUD INFRASTRUCTURE



Ver. 2: Material calculations on the cloud

Computer centers are moving from HPC only to service providers

On-going work between AiiDA team and CSCS

We identified services to enable automatic computations "on the cloud" (i.e. on federated supercomputer centres)

- **Database** (to store and query information)
  - PostgreSQL 9.5 supporting data intensive queries, JSON and multiple users
- **Object store** (to store large files)
  - Apache Swift: Efficient storage and retrieval of large objects
- **Web backends** (hosting of web services)
  - Apache: Discovery, exploration of existing materials, calculations, workflows & launch of new ones
- **AAI** (authentication and authorization infrastructure)
  - In progress: Keystone, Shibboleth, identity management and authentication for federated access

Ver. 2: Material calculations on the cloud

Currently: CSCS provides proof of concepts of each service, and AiiDA+MaterialsCloud use them:

- Web frontend:
  - materialscloud.org portal, will go online in April
- Object store & databases
  - MaterialsCloud uses the DB from CSCS, and can store files as objects in Apache Swift
  - DB and Object Store will authenticate users via the common AAI
- AAI
  - Via AAI, services above can be federated with other computer centres
  - Result: data shared; HPC computation, querying and web access distributed; higher availability of services and resources

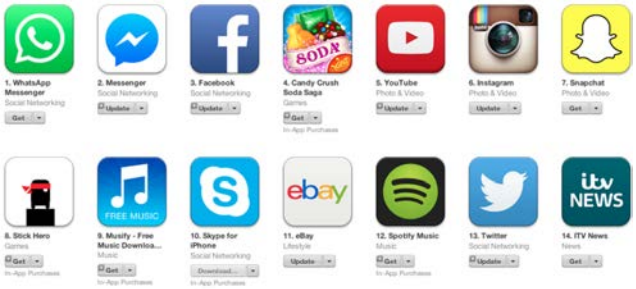
If supercomputer centres provide these federated services to users:

1. AiiDA can be installed on supercomputers and used with zero effort to run workflows, manage provenance
2. *Beneficial also beyond AiiDA*: easier to federate files (via object store); easier to run big-data analysis (thanks to DB)

APP STORE MODEL

App-store (@Apple) model for Plugins & Workflows that can be freely contributed or sold by users or companies. E.g.

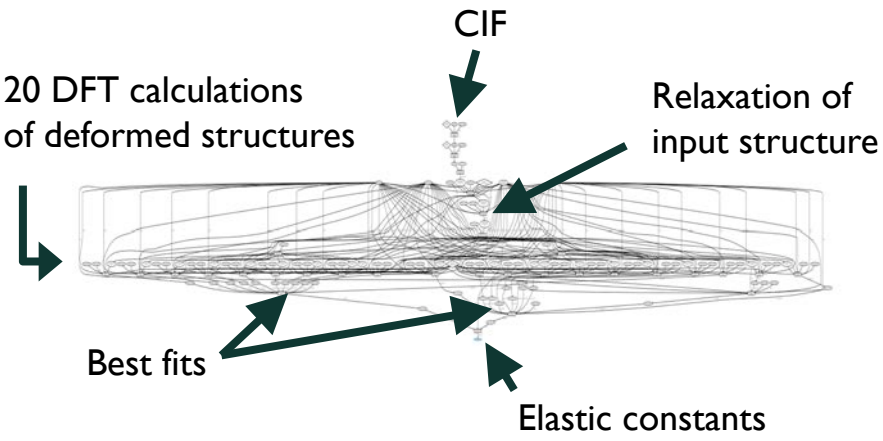
- Workflows: look for the workflow that computes the desired property.
- Calculations: look for the plugin supporting your favourite software (Quantum Espresso, Vasp, Gpaw, Yambo, …)
- Experimental database: load structures and data from COD, ICSD, …
- Computers: install a new cluster from the web
- ……



http://www.aiida.net/plugins/

Code	Contact person(s)	Link to repository	Status
Quantum ESPRESSO: PW, CP	The AiiDA Team Davide Campi (EPFL)	Included in aiida_core	01 - stable
NWChem	The AiiDA Team Andrius Merkys (Vilnius U.)	Included in aiida_core	01 - stable
cod-tools	The AiiDA Team Andrius Merkys (Vilnius U.)	Included in aiida_core	01 - stable
Quantum ESPRESSO: PH, DOS, PP, projwfc, NEB, Environ, Q2R, Matdyn	The AiiDA Team Davide Campi (EPFL)	Included in aiida_epfl	01 - stable
Wannier90	The AiiDA Team	Included in aiida_epfl	01 - stable
ASE / GPAW	The AiiDA Team Andrea Cepellotti (EPFL)	Included in aiida_epfl	01 - stable
FLEUR	Jens Bröder (PGI-1)	Available on BitBucket	01 - stable
CP2K	Aliaksandr Yakutovich (EMPA)	Available on BitBucket	02 - working: testing needed/ features missing
PhonoPY	Abel Carreras (Kyoto University)	Available upon request	02 - working: included in the next aiida_epfl release
SIESTA	Alberto Garcia (ICMAB-CSIC, Spain) Victor Garcia-Suarez (U. Oviedo, Spain)	Available upon request on BitBucket	02 - working: beta version
Z2PACK	Antimo Marrazzo (EPFL)		03 - under development
Yambo	Antimo Marrazzo (EPFL) Michael Atambo (U. Modena e Reggio Emilia)	Available upon request on BitBucket	02 - working: beta version
Exciting	Anton Kozhevnikov (CSCS)	Available on BitBucket (forked from aiida_core)	03 - under development
Sirius	Anton Kozhevnikov (CSCS)	Available on BitBucket (forked from aiida_core)	03 - under development
VASP	Mario Žic (Trinity College Dublin) Rico Häuselmann (ETHZ)	available upon request	03 - under development
i-PI	Venkat Kapil (EPFL)		04 - under design, collaboration welcome

WORKFLOWS

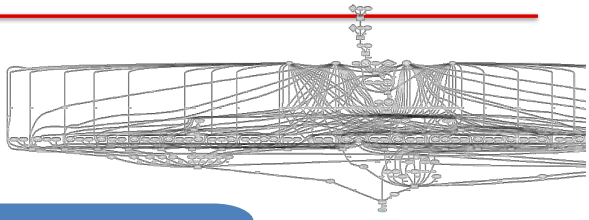


Workflows features

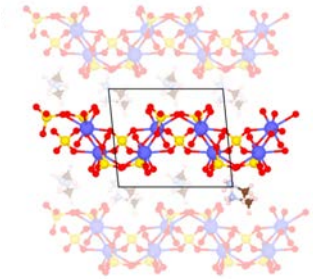
- Automatic provenance tracking, stored in DB using simple python functions
  inputs, outputs, function calls stored by adding simple decorator to existing functions
- Serial and parallel execution support
  can launch long running tasks on separate threads and wait for result when needed
- Control provenance granularity
  store level of detail relevant to the workflows
- Seamless mixing of local and remote jobs
- Progress checkpointing
  restart from arbitrary step, retry on failure
- Easy debugging
  execute workflows in IDE and observe/change states of variables as it runs
- Background execution
  daemon execution allows machine to be shutdown and continue from last point, essential for running long remote jobs

FROM HALF A MILLION TO…?

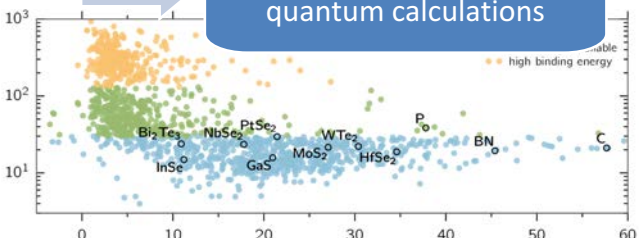
Database(s) pipeline:
500,000 candidates



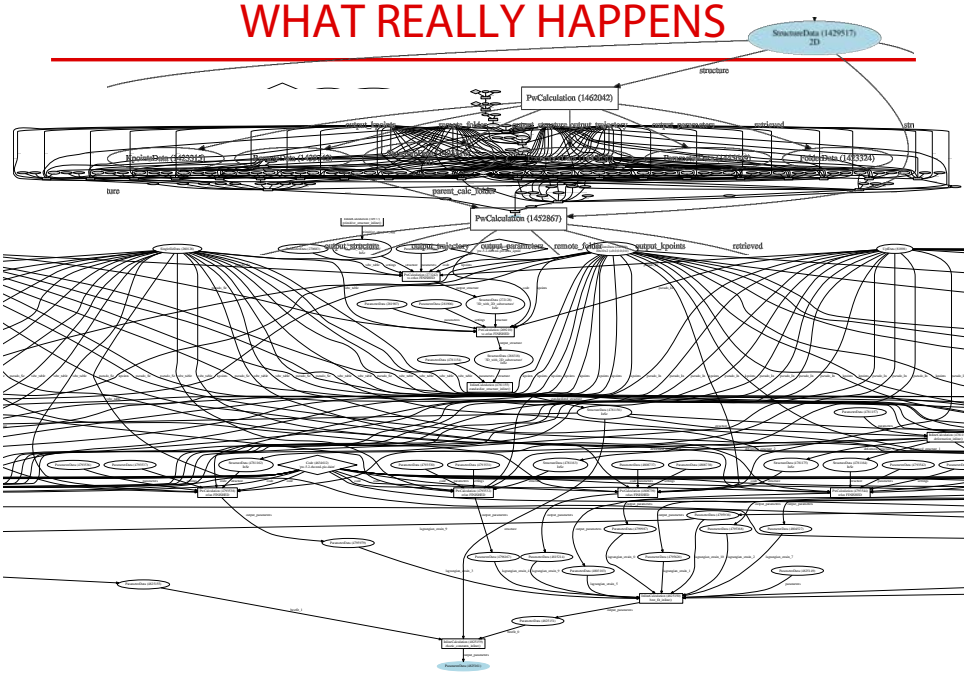
Layered? 6,400 candidates



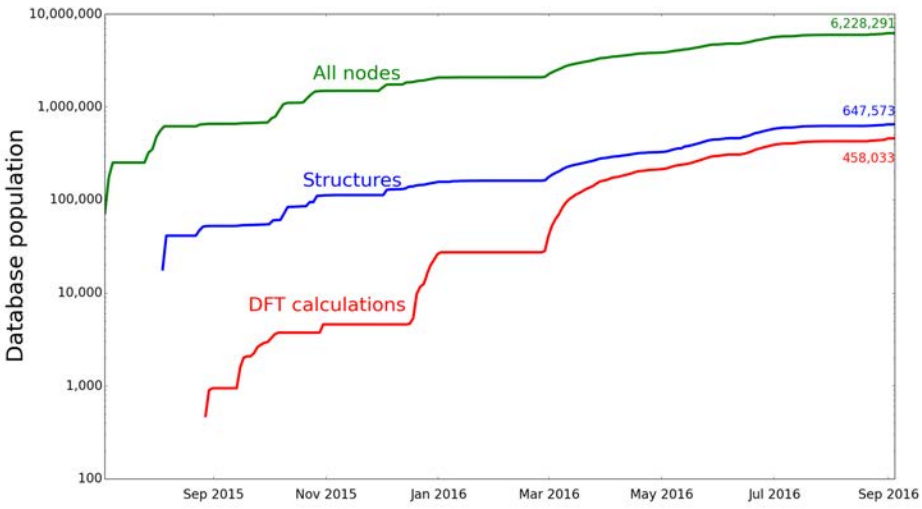
Exfoliable? 450,000
quantum calculations



WHAT REALLY HAPPENS

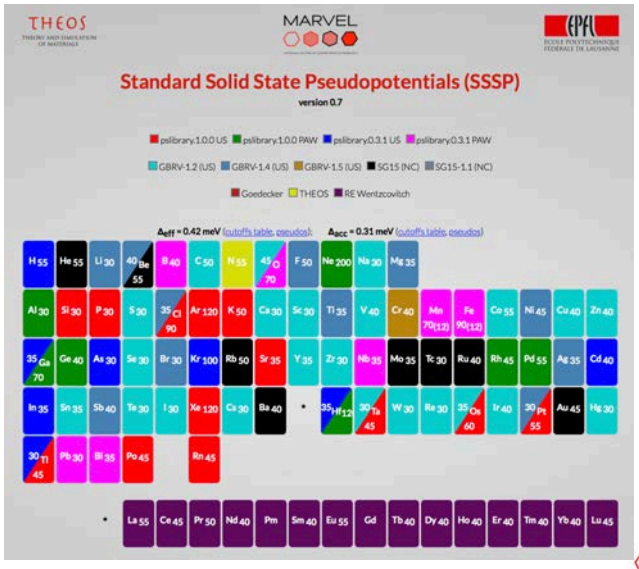


2-DIMENSIONAL DATABASE

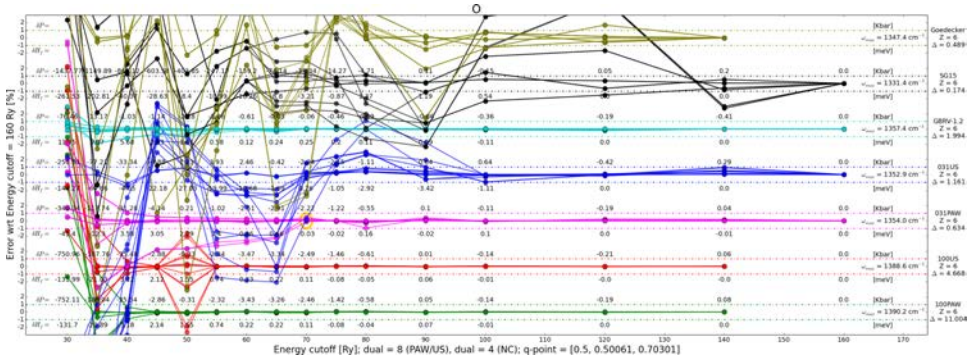


MATERIALS CLOUD TOOLS

http://materialscloud.org/sssp



http://materialscloud.org/sssp



Cutoff/dual convergence of phonon frequencies, stress tensor, formation energies, band structures, ghost states. Verification against all-electron Δ. Internal consistency.

I. E. Castelli, N. Mounet, A. Marrazzo, G. Prandini, and N. Marzari, in preparation (2016)

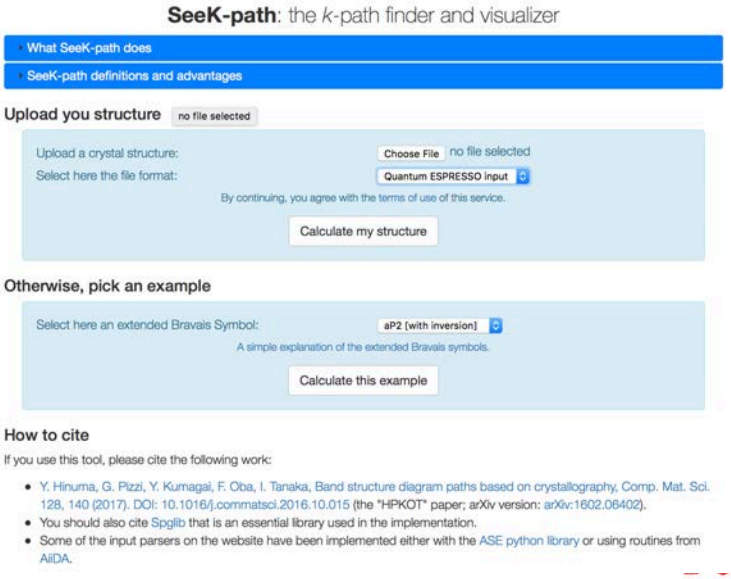
AS ACCURATE AS ALL-ELECTRON

AE	average <Δ>	Elk	exciting	FHI-aims/tier2	FLEUR	FPLO/T+F+s	RSPt	WIEN2k/acc
Elk	0.6		0.3	0.3	0.6	1.0	0.9	0.3
exciting	0.5	0.3		0.1	0.5	0.9	0.8	0.2
FHI-aims/tier2	0.5	0.3	0.1		0.5	0.9	0.8	0.2
FLEUR	0.6	0.6	0.5	0.5		0.8	0.6	0.4
FPLO/T+F+s	0.9	1.0	0.9	0.9	0.8		0.9	0.9
RSPt	0.8	0.9	0.8	0.8	0.6	0.9		0.8
WIEN2k/acc	0.5	0.3	0.2	0.2	0.4	0.9	0.8	
SSSP/QE	0.5	0.4	0.3	0.3	0.5	0.9	0.8	0.3

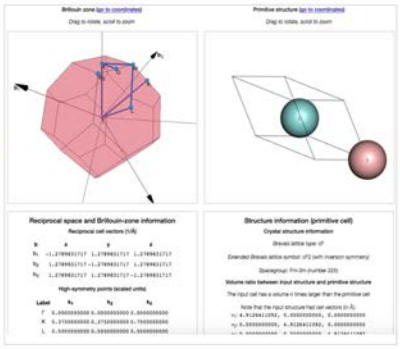
K. Lejaeghere *et al.*, Science (2016)

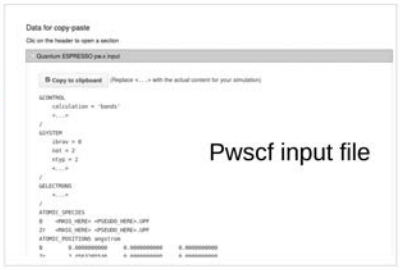
K. Lejaeghere and S. Cottenier - https://molmod.ugent.be/deltacodesdft

http://materialscloud.org/tools/seekpath



SeeK-path: a crystallography-aware tool to obtain and visualize band paths





- Python code, for programmatic use/high-throughput: GitHub: https://github.com/giovannipizzi/seekpath
- Web interface for learning/trying out: MaterialsCloud: http://materialscloud.org/tools/seekpath/ (interactive visualization, primitive cell, list of k points, copy-paste content for QE input, etc.)

Y. Hinuma, G. Pizzi, Y. Kumagai, F. Oba, I. Tanaka, Comp. Mat. Sci. ***128****, 140 (2016)*

SeeK-path improvements on existing literature work

Starting point: paper by Setyawan and Curtarolo, Comp. Mat. Sci. 2010 (SC2010)

Main aims/improvements of SeeK-path:

1. Ensure **compliance with crystallographic standards** (*International Tables of Crystallography*):

- use of the crystallographic cell (e.g. monoclinic cells always *b*-axis unique);
- use the standard setting (e.g. in orthorhombic Pmm2, the third axis *c* is fixed by symmetry (2 rather than m) and a rule for standardization can be imposed only for *a*<*b*, not for *c*);
- use the same letters of the International Tables where available, and non-colliding letters where new ones must be defined.

2. Improve over shortcomings in SC2010, where standardization is based *only* on Bravais lattice:

- the point group **must** be taken into account (in some cases) to *fully sample* all the high-symmetry lines of the BZ;
  - e.g. *primitive cubic cells*, *in SC2010 there is a missing high-symmetry line* in the suggested path based only on the Bravais lattice for point group is Pm-3 (instead the path is complete for point group Pm-3m).

http://nccr-marvel.ch/en/events/aiida-tutorial-may-2017

AiiDA tutorial — May 2017

from 29 to May 31 2017 at EPFL, Lausanne

A MARVEL/MaX/Psi-k *Tutorial on high-throughput computations: general methods and applications using AiiDA* will be held on May 29 to 31, 2017 at EPFL (Lausanne, Switzerland).

The tutorial is targeted at about 50 students, postdocs and researchers interested in applying high-throughput computations in their research, and in particular to those interested in learning how to use the AiiDA platform.

Programme and location

The programme includes a tutorial on the AiiDA code, and four invited highlight talks from experts in the field of high-throughput computations:

- Prof. Thomas Bligaard (Stanford University, USA)
- Prof. Marco Fornari (Central Michigan University, USA)
- Prof. Chris J. Pickard (Univ. of Cambridge, UK)
- Prof. Stefano Sanvito (Trinity College Dublin, IRL)

THE AiiDA AND MATERIALS CLOUD TEAM


Giovanni Pizzi (EPFL)


Nicolas Mounet (EPFL)


Andrea Cepellotti (Berkeley)


Leonid Kahle (EPFL)


Andrius Merkys (Vilnius)


Riccardo Sabatini (Hum. Longevity)


Spyros Zoupanos (EPFL)


Boris Kozinsky (BOSCH)


Anton Kozhevnikov (CSCS)


Davide Campi (EPFL)


Sebastiaan Huber (EPFL)


Martin Uhrin (EPFL)


Fernando Gargiulo (EPFL)


Nicola Varini (EPFL)


Snehal Waychal (EPFL)

Georgy Samsonidze, Prateek Mehta, Andrea Greco @ Bosch

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 (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)