Automating electronic structure calculations with AiiDA

Thor Wikfeldt
kthw@kth.se
PDC Center for HPC, Stockholm

Why AiiDA?

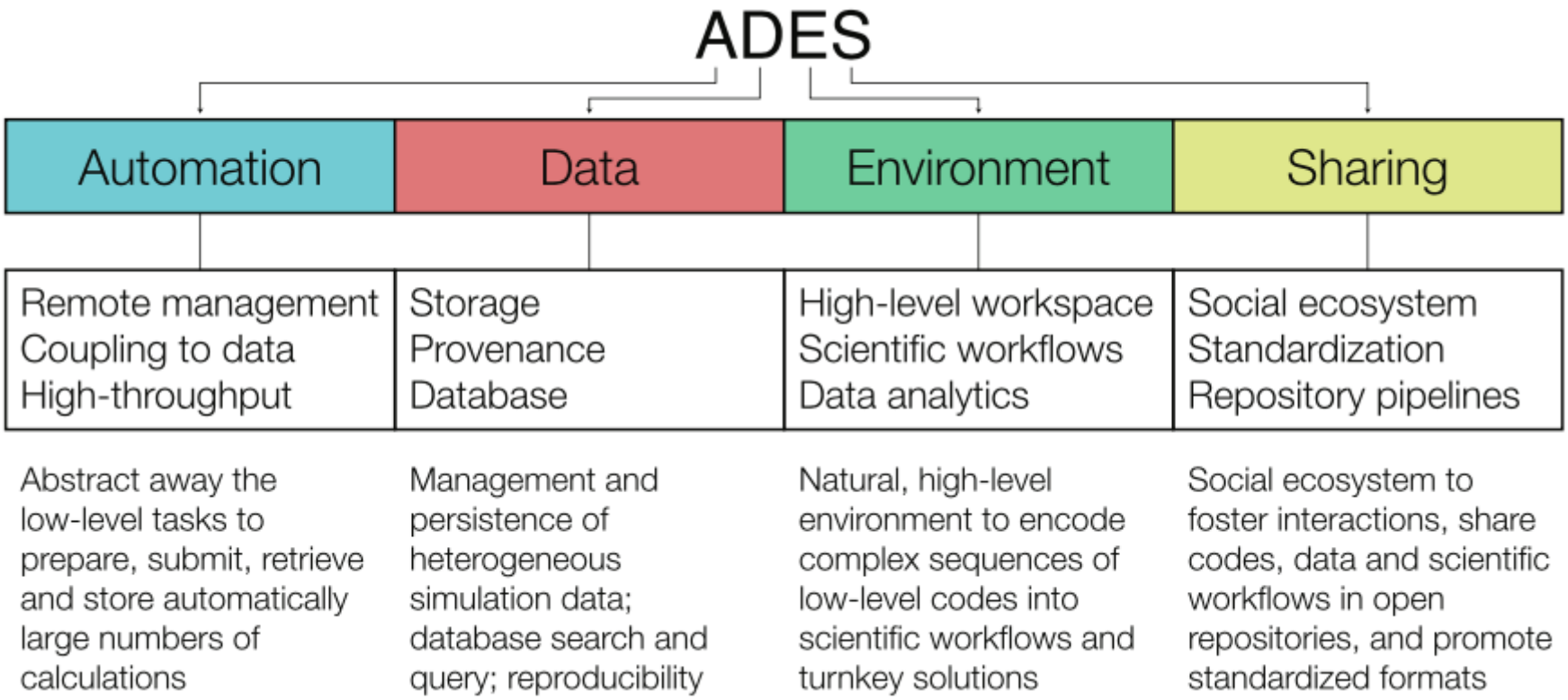
- Computation has emerged as third pillar of science, bridging experiment and theory
- Traditional approach of computational science:
  - create input, copy to cluster, submit, copy back results, analyze, repeat...
  - manual steps, focus on one or few systems at a time
- But we live in the age of automation, big data and sharing. Some research can be generalized beyond the traditional approach

- But we live in the age of automation, big data and sharing. Some research can be generalized beyond the traditional approach
- AiiDA: Automated Interactive Infrastructure and Database for Computational Science

Computational science should be

- **Reproducible** (often not possible from the data reported in papers)
- **Searchable** (find existing calculations, reuse and data-mine results)
- **Reliable** (automated procedures to reduce errors and verify results)
- **Shareable** (community to share results, cross-validate them, and boost scientific discovery)

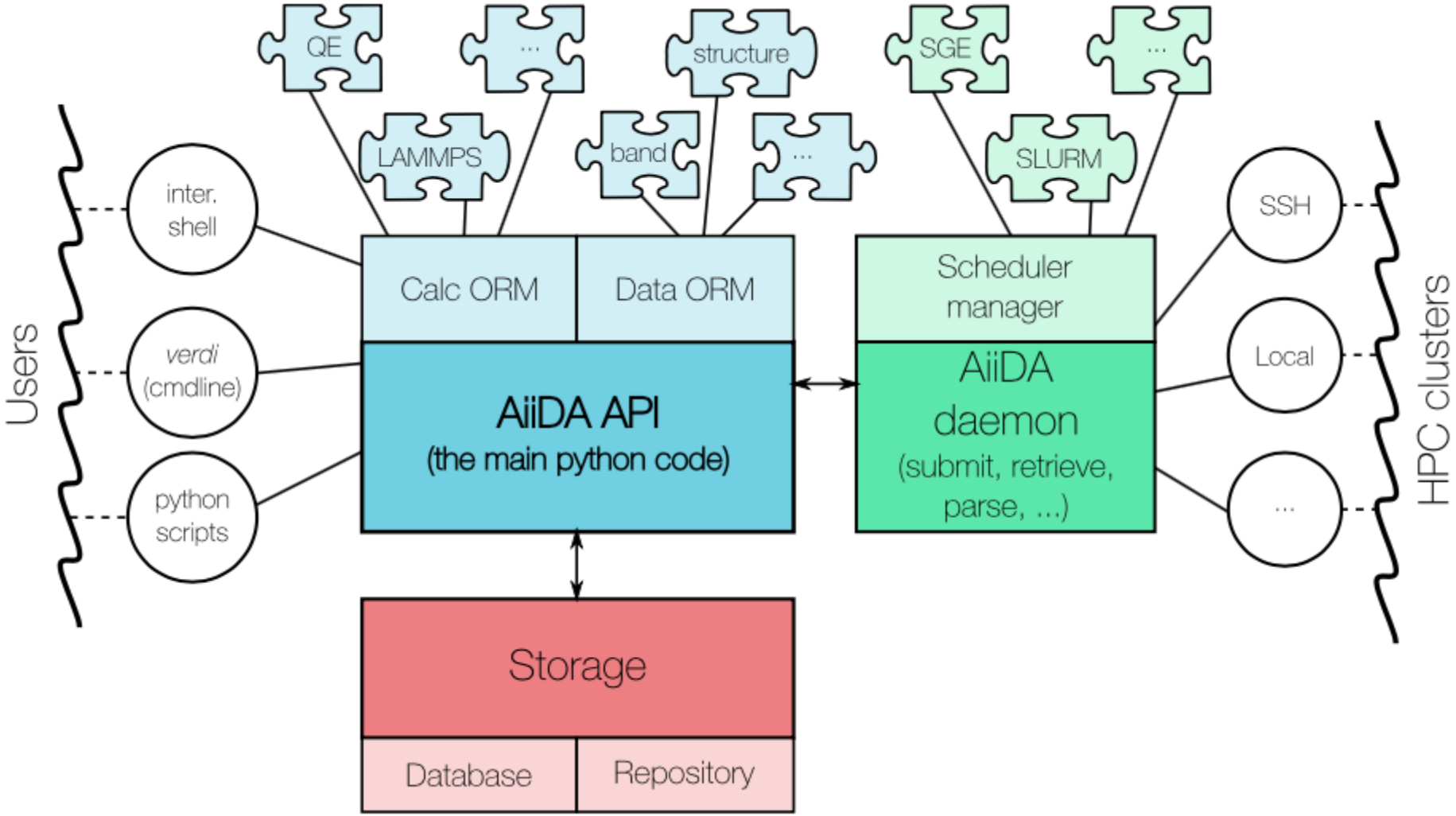
AiiDA: Automated Interactive Infrastructure and Database for Computational Science



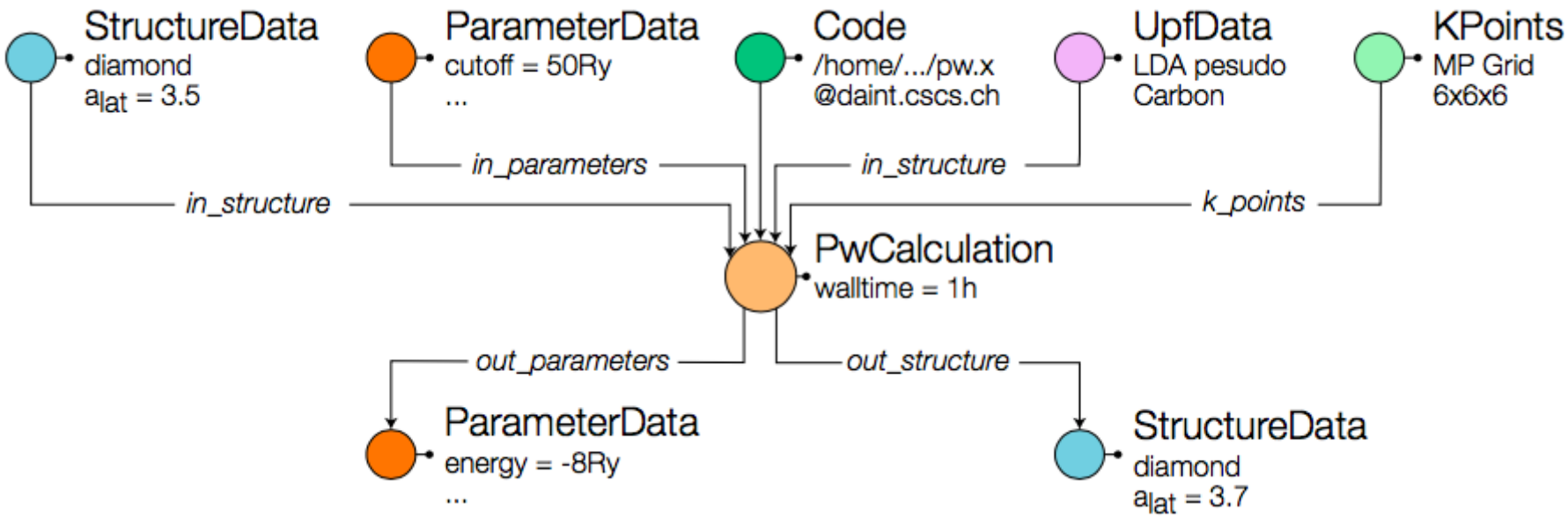
G. Pizzi, A. Cepellotti, R. Sabatini, N. Marzari, and B. Kozinsky, AiiDA: automated interactive infrastructure and database for computational science, Comp. Mat. Sci. 111, 218-230 (2016)

</small>

AiiDA components and their interactions

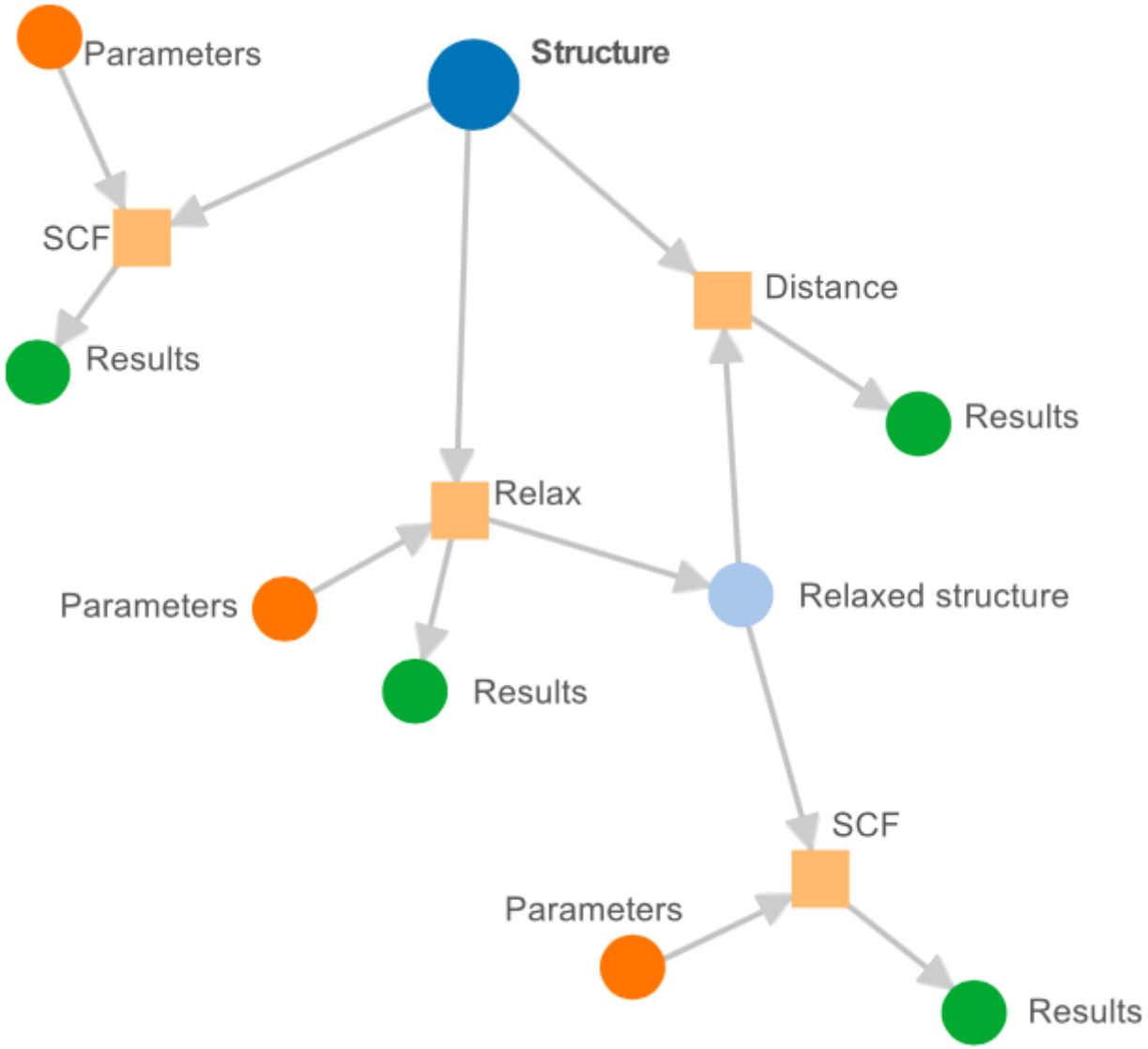


One calculation, executable code and input/output data



- Nodes organized into directed acyclic graph (DAG)
- All `Calculation` objects act as functions, direct links between `Data` objects impossible

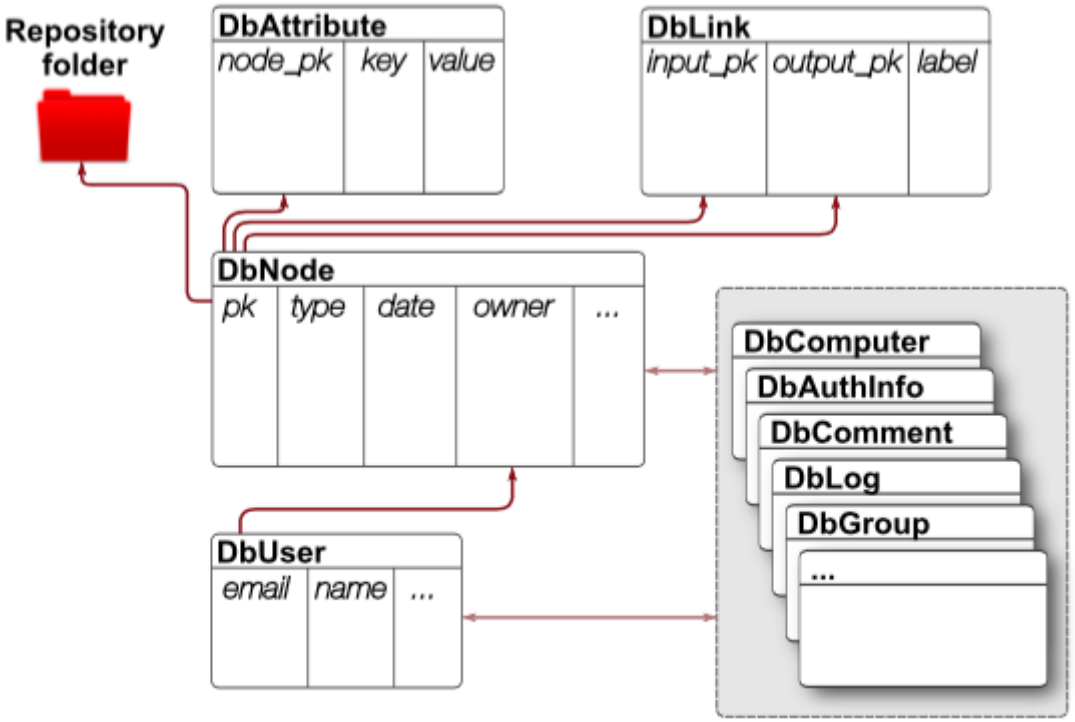
A possible DAG inside AiiDA database



Database representation of DAGs

- *Each* ***node***: row in a SQL table
  - Additional data:
    - key-value **attributes**
    - **Files/folders**

- **Links** also stored in a SQL table
  ⇒ *jobs provenance*



Further learning

- G. Pizzi, A. Cepellotti, R. Sabatini, N. Marzari, and B. Kozinsky, AiiDA: automated interactive infrastructure and database for computational science, Comp. Mat. Sci. 111, 218-230 (2016)
- aiidausers@googlegroups.com
- http://aiida-core.readthedocs.io/en/stable/ (http://aiida-core.readthedocs.io/en/stable/)
- http://www.aiida.net/tutorials/ (http://www.aiida.net/tutorials/)
- Regular workshops organized. Next one is 29-31 May 2017 at EPFL, Lausanne

Live demo

- Interacting with AiiDA via verdi commands (typically in terminal)
  - users, computers, codes, calculations
  - graphs and nodes
  - inspecting `Calculation`, `Data` and `Code` nodes
  - groups of calculations
- Interacting with AiiDA objects
  - via Jupyter Notebook or verdi shell
  - inspect or set up pseudopotentials, k-points, input parameters and structures

- Submit, monitor and debug calculations
  - the AiiDA daemon
  - creating a new calculation
  - submit calculation (from notebook or through `verdi run filename`)
- Queries in AiiDA
  - building a query
  - a high-throughput example

Workflows

- Allow the user to define one or more processes that (optionally) take some inputs and (optionally) produce some outputs.
  - **Workfunctions**: python function with a decorator and a couple of constraints on its inputs and return value
  - **Workchains**: a series of instructions used to carry out a process with checkpoints being taken between each instruction such that the process can be paused/stopped/resumed

- Workfunctions can be nested to create complicated workflows
- Example applications: Equation of state, phonon calculations, etc.