Workflows general concepts and Application

Akshay krishna Ammothum kandy

IR engineer, IMPMC
Sorbonne University

Outline

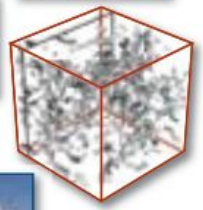
- Introduce workflow management concepts.
- Survey established workflow systems
- Give a hands-on demo running an **AiiDA workflow** on google colab.

Introduction

Science Paradigms

- Thousand years ago:
  science was **empirical**
  describing natural phenomena
- Last few hundred years:
  theoretical branch
  using models, generalizations
- Last few decades:
  a **computational** branch
  simulating complex phenomena
- Today: **data exploration** (eScience)
  unify theory, experiment, and simulation
  - Data captured by instruments or generated by simulator
  - Processed by software
  - Information/knowledge stored in computer
  - Scientist analyzes database/files using data management and statistics

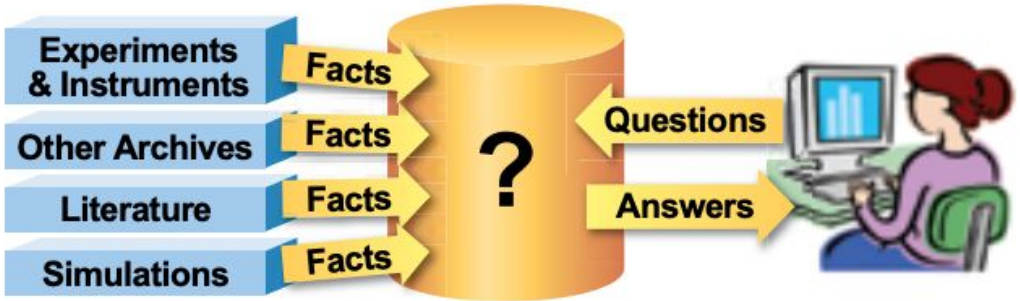
$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{4\pi G p}{3} - \mathrm{K}\frac{c^2}{a^2}$$



Fourth paradigm is using the computational power to generate, curate, analyze, archive a massive scientific data

Introduction

- How to codify and represent our knowledge

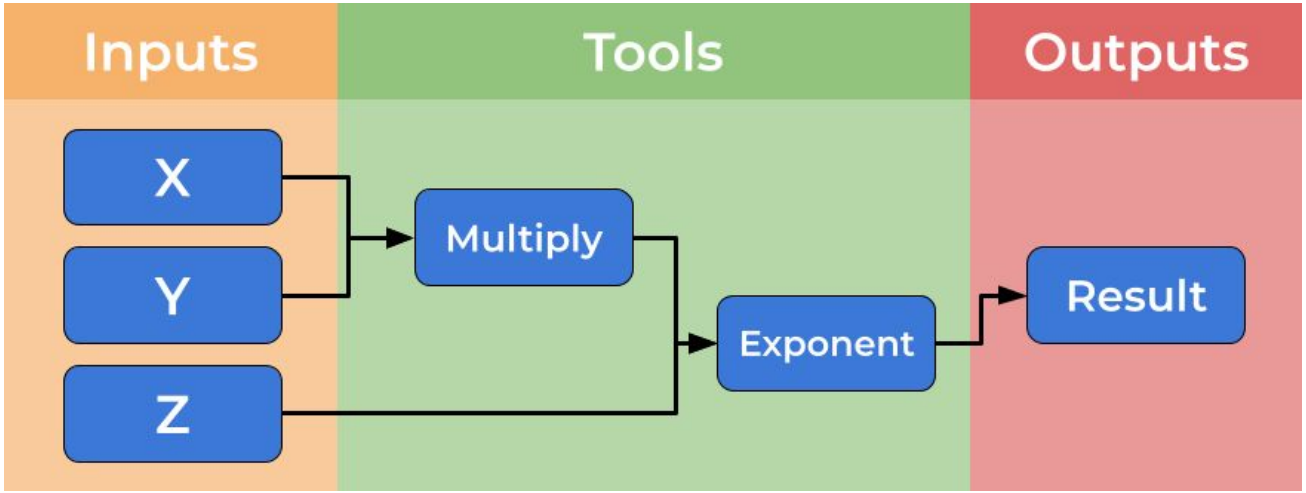


The Generic Problems

- Data ingest
- Managing a petabyte
- Common schema
- How to organize it
- How to reorganize it
- How to share it with others
- Query and Vis tools
- Building and executing models
- Integrating data and literature
- Documenting experiments
- Curation and long-term preservation

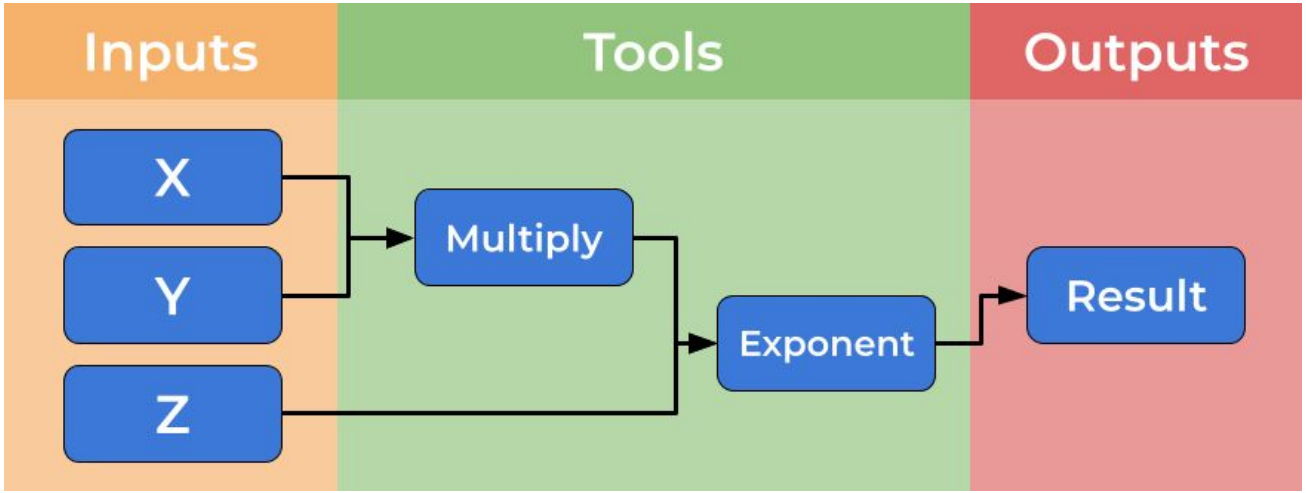
What is a workflow?

Computational workflows explicitly create a divide between a user's dataflow and the computational details which underpin the chain of tools, placing these elements in separate files



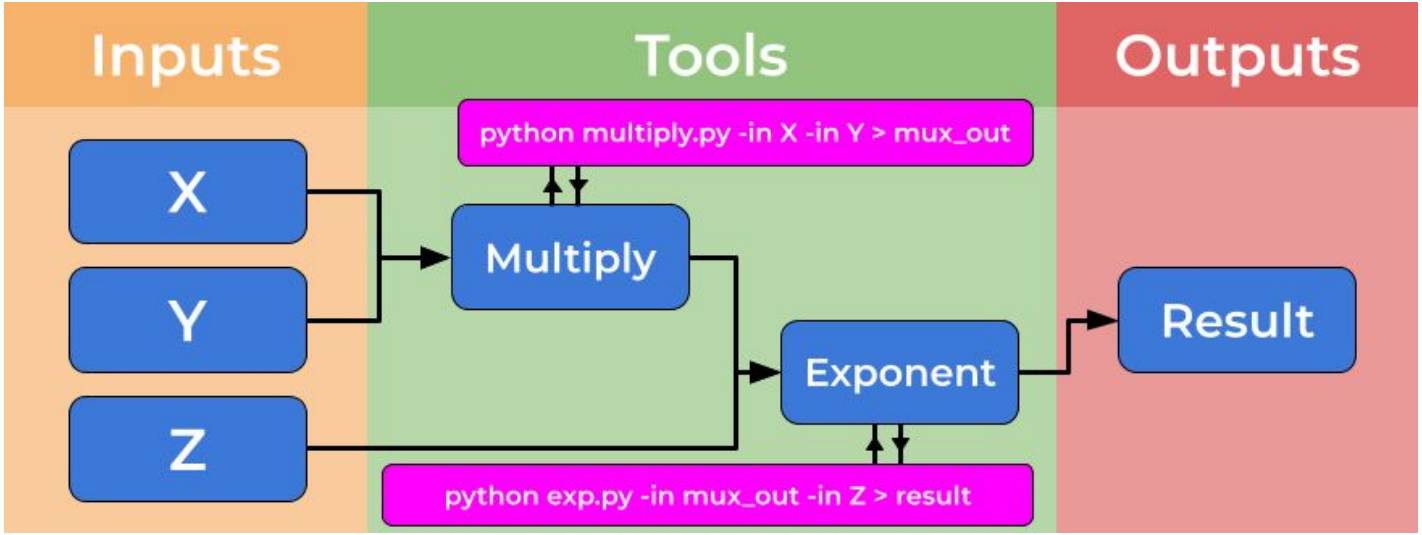
What is a workflow?

Control flow: Dictate the sequence in which steps are executed (Analogous to traffic lights & intersections)
Data flow: *what* moves between steps (eg: files, data) (Analogous to cars)



What is a workflow?

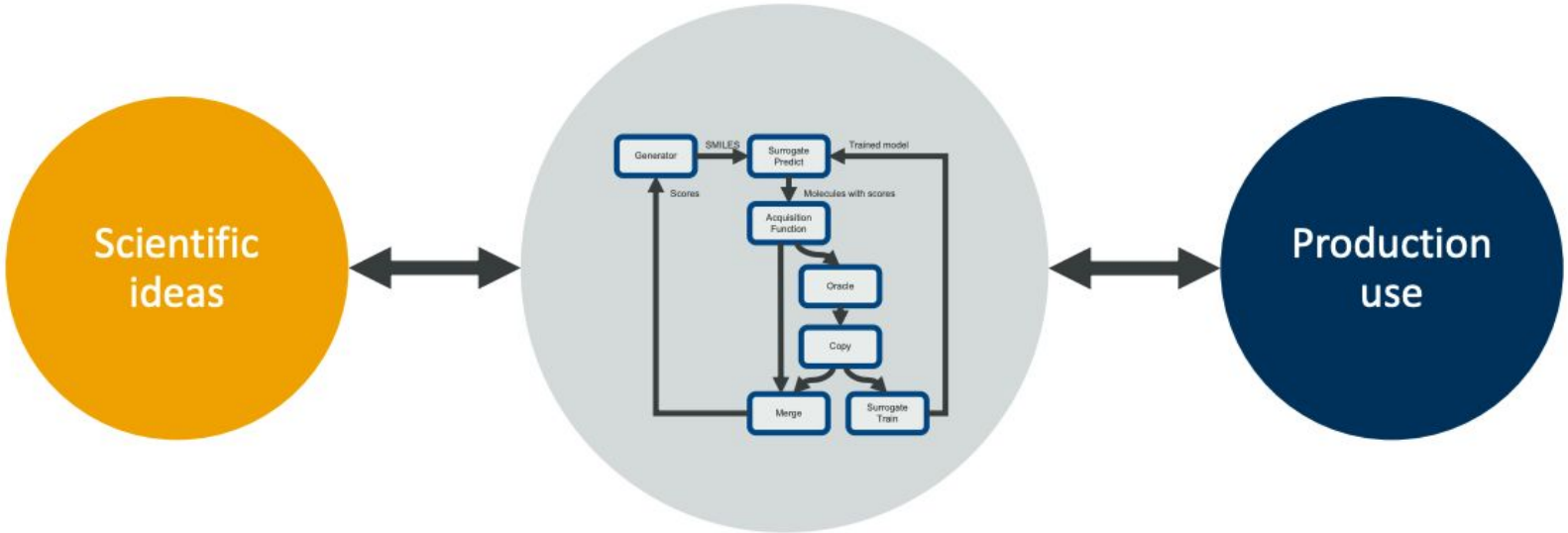
Abstraction: components can be thought of as black boxes



What are workflow managers?

Workflow managers are a tool for abstraction!

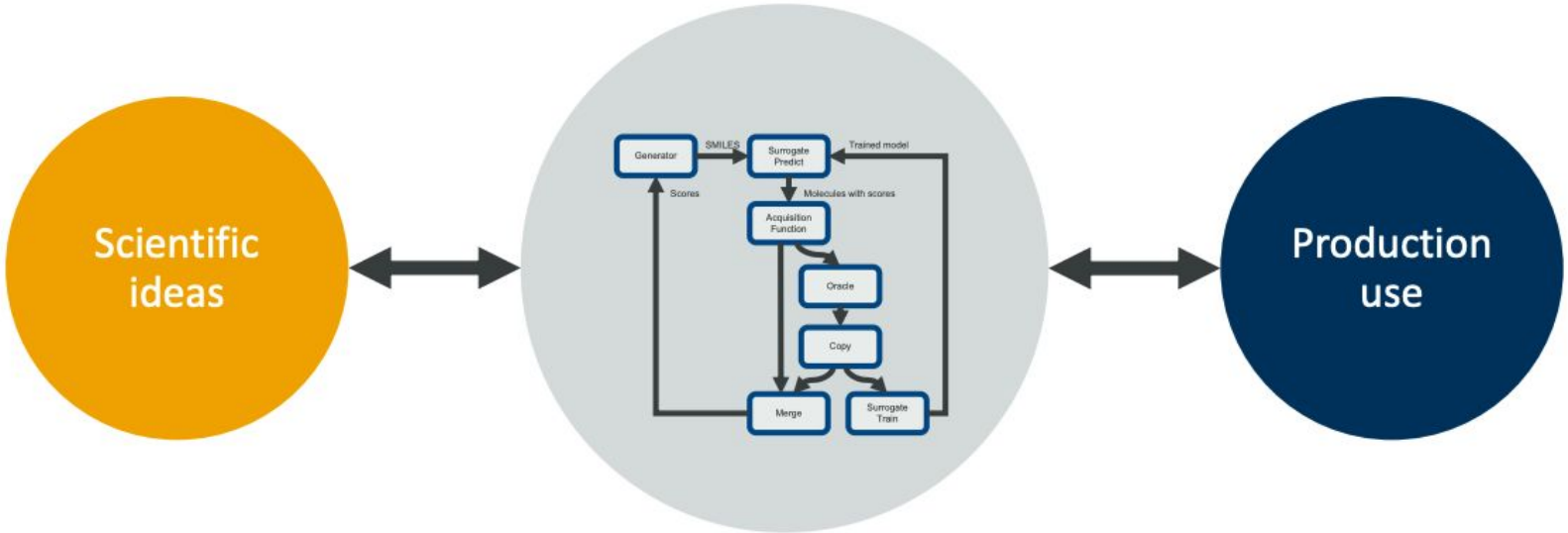
you write “what” to run; the system handles “how/where/when”



https://ai-dd.eu/sites/default/files/slides/Thomas_Loehr-maize-talk.pdf

What are workflow managers?

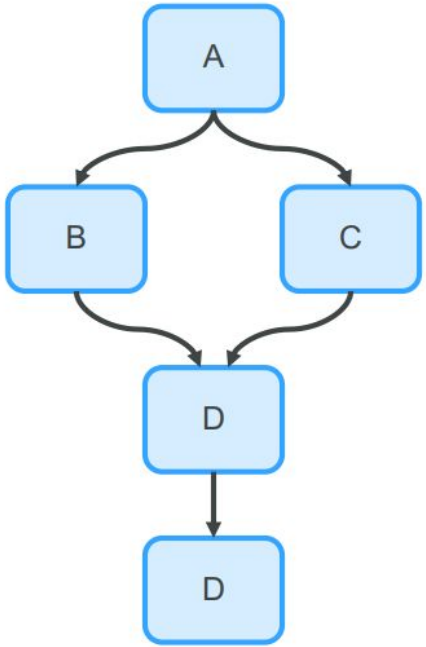
relax → phonons → DOS over ~200 structures, with automatic plane-wave cutoff



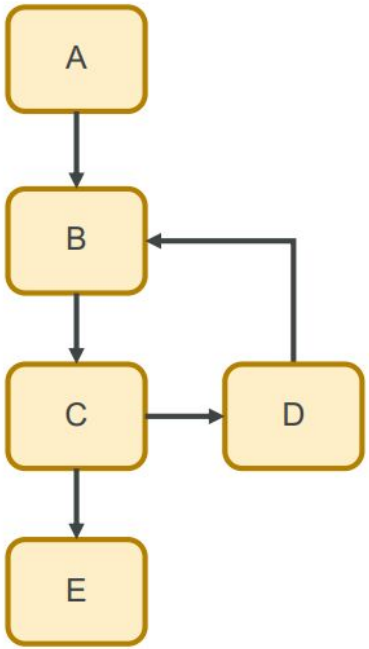
https://ai-dd.eu/sites/default/files/slides/Thomas_Loehr-maize-talk.pdf

Complex workflows

Directed Acyclic Graph (DAG)



Directed Cyclic Graph (DCG)



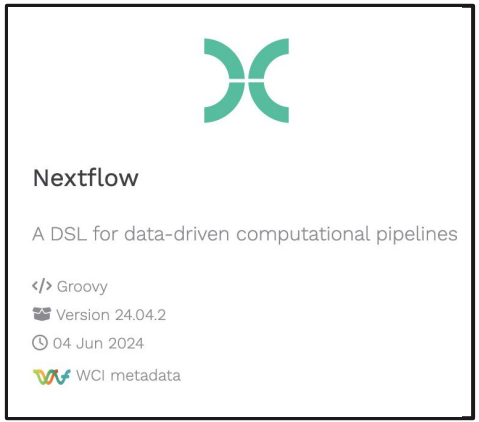
DAGs are the basis for most workflow engines:

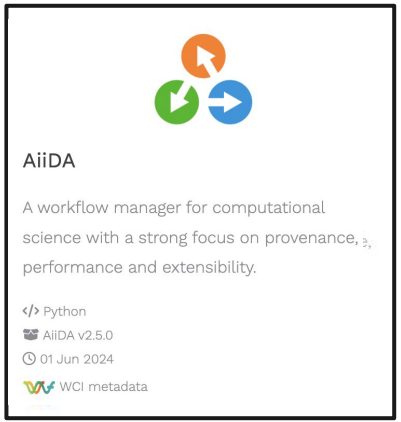
- Apache Airflow
- Luigi
- DAGster
- ...

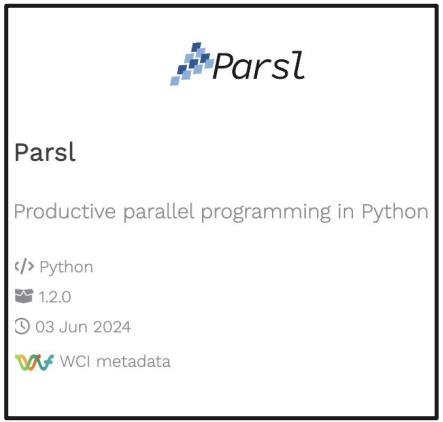
DCGs allow iteration, control flow, but have limited support:

- Akka (Scala)
- NoFlo (JavaScript)

Commonly used workflow managers







100+ workflow managers available…

Why not use python or bash scripts?

- **Reproducibility** – not just for others, but for yourself too!
- **Configuration** – no searching through shell history for used parameters
- **Modularization** – easily exchange components to make experiments easier
- **Automation** – easier to integrate into routine systems
- **Abstraction** – components can be thought of as black boxes
- **Performance** – some workflows allow parallelization

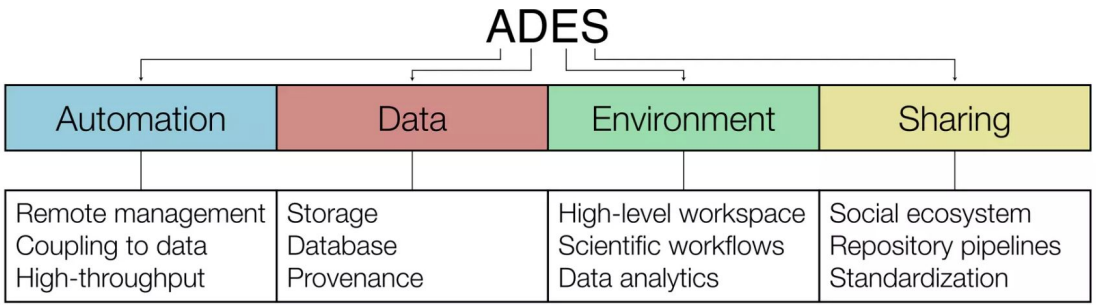
https://ai-dd.eu/sites/default/files/slides/Thomas_Loehr-maize-talk.pdf

Part 2

Introduction to AiiDA: automated interactive infrastructure and database for computational science

AiiDA

Automated Interactive Infrastructure and Database for Computational Science



Language: implemented and API in Python

License: MIT open source **http://www.aiida.net/**

Source: https://github.com/aiidateam/aiida-core

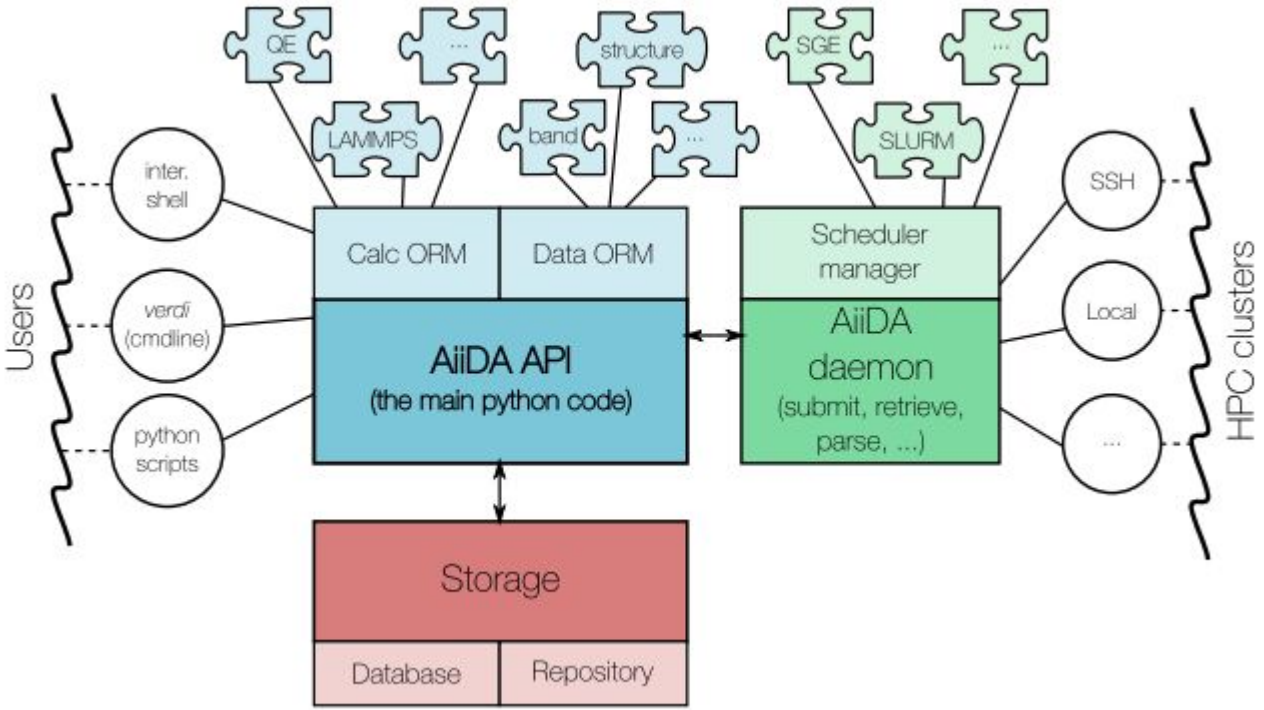
MIT LICENSED

open source initiative

AiiDA plugins

raspa2, zeo++, ...

AiiDA schema



Object–Relational Mapping: a layer that maps **classes/objects** in code to **tables/rows** in a relational database.

Plugin interface: supports any computational code and data type via extensible plugins.

Daemon (background engine): automates job submission, monitoring, file retrieval, and parsing.

Schedulers: talks to SLURM/PBS/… through scheduler plugins; abstracts queue details.

Transports: connects to resources locally or via SSH (and other channels).

Remote interaction: submits jobs, checks states, and fetches results from clusters automatically.

Basic concepts

Data Types

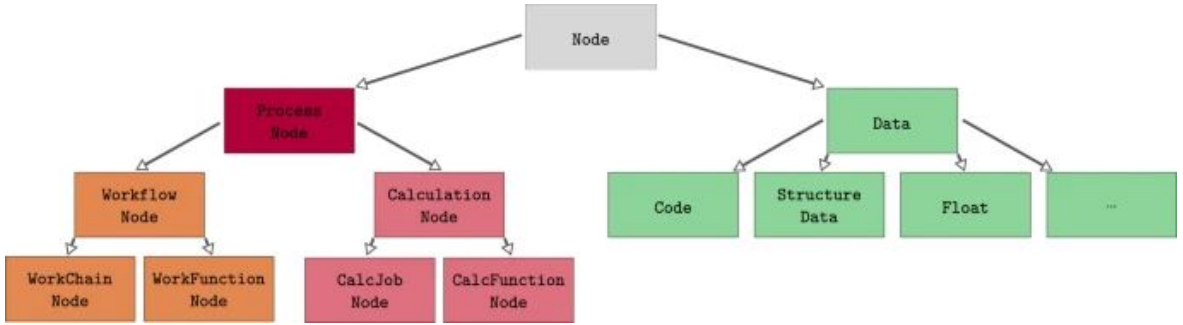
Python classes hosting data, allow storing in the database and provenance.

Simple wrappers of python types.

```
from aiida.orm import Float
vols = Float(7)
```

Material science related objects.

```
from aiida.orm import StructureData
struct = StructureData(ase=ase_struct)
```



Basic concepts

Data Types

Python classes hosting data, allow storing in the database and provenance.

Simple wrappers of python types.

```
from aiida.orm import Float
vols = Float(7)
```

Material science related objects.

```
from aiida.orm import StructureData
struct = StructureData(ase=ase_struct)
```

Calculations plugins

Interface with codes

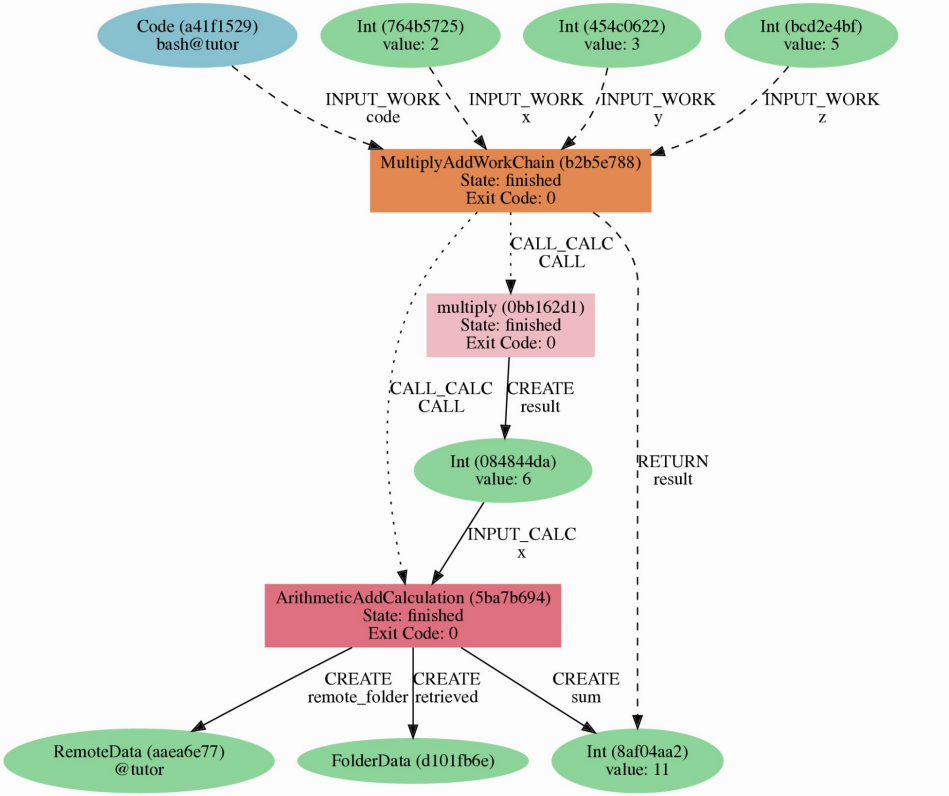
```
from aiida.plugins import CalculationFactory
from aiida.engine import submit
my_calc = CalculationFactory("plugin.name")
submit(my_calc, **inputs)
```

Workflows

Encode complex steps of scientific workflow

```
from aiida.plugins import WorkflowFactory
from aiida.engine import submit
my_wc = WorkflowFactory("workflow.name")
submit(my_wc, **inputs)
```

Basic concept: Provenance

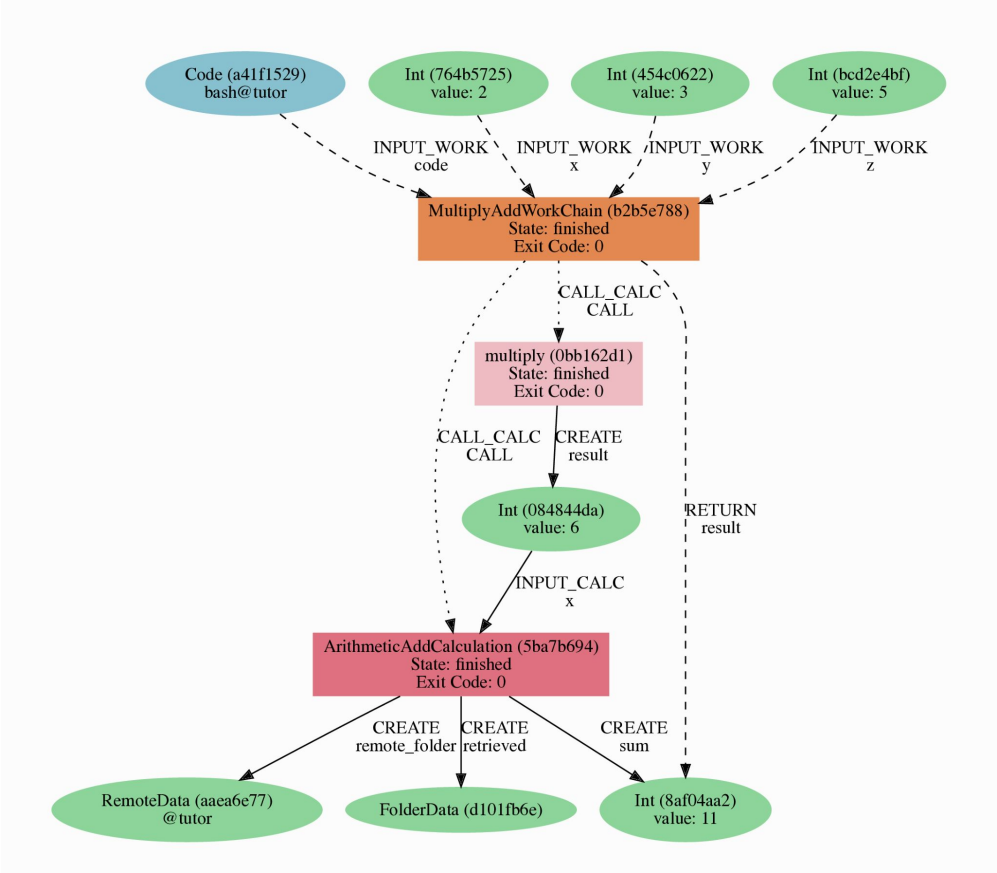


AiiDA stores **results, inputs, and every execution step** of your calculations.

All this information is captured as a **Directed Acyclic Graph (DAG)**.

Provenance lets you **fully retrace how any data was produced → reproducibility**.

Hands on



- Learn to run simple AiIDA workflow
- Generate provenance graph for the workflow

AiiDA schema