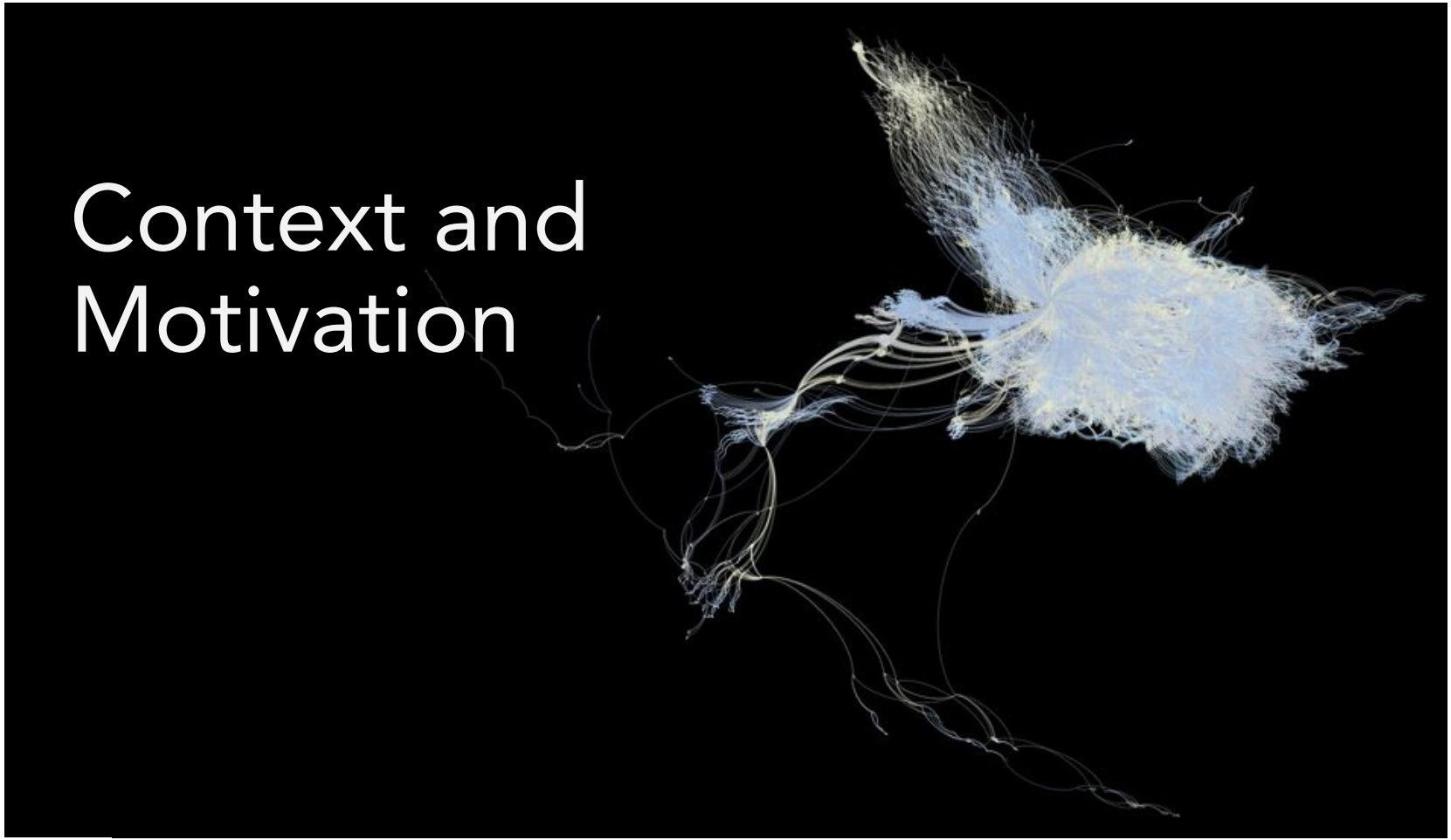
Introduction to AiiDA

Francisco Ramirez

MARVEL NATIONAL CENTRE OF COMPETENCE IN RESEARCH · MaX DRIVING THE EXASCALE TRANSITION · swissuniversities · FNSNF Swiss National Science Foundation · INTERSECT · MarketPlace · EPFL

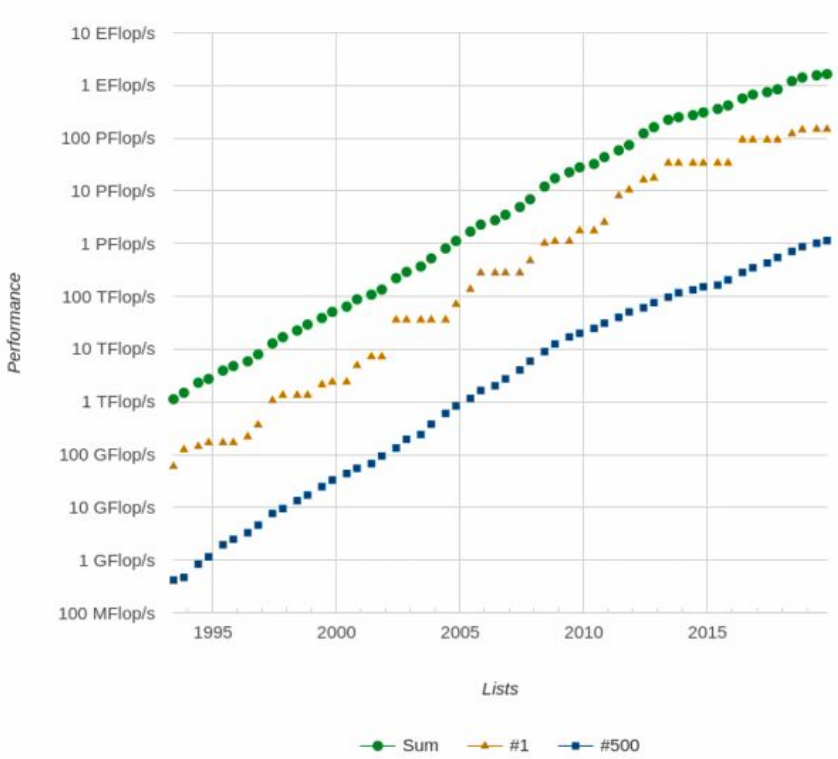
Context and Motivation



Francisco Ramirez

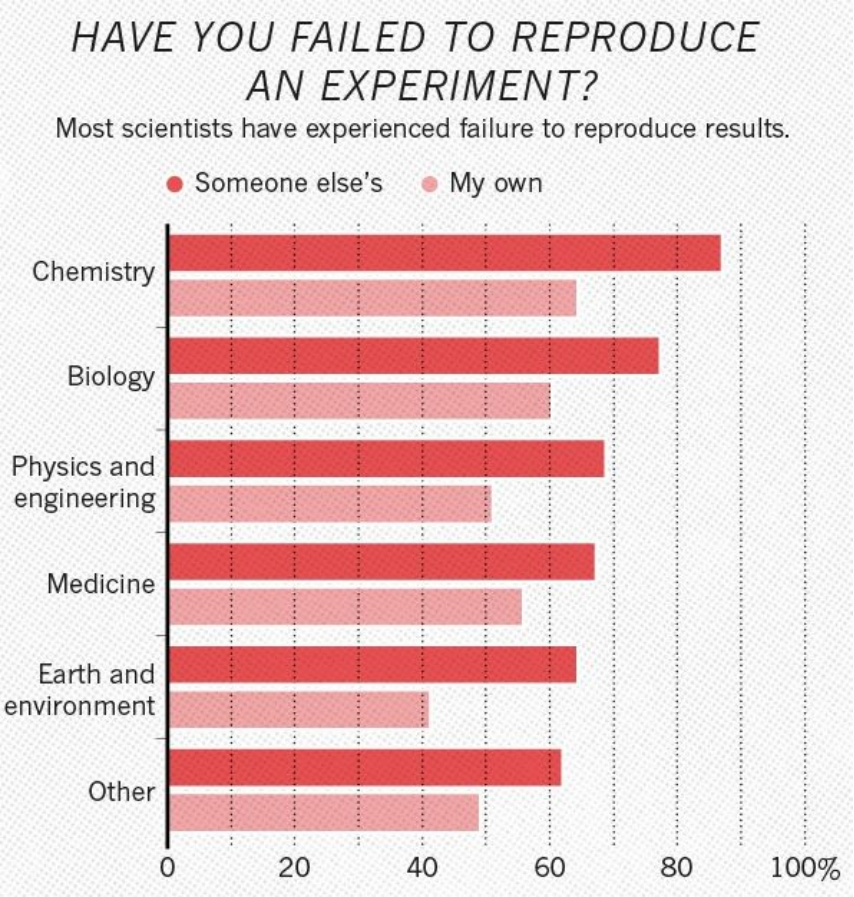
The rise of HTC and HPC

- Computational resources keep becoming more powerful: new tools needed to efficiently utilize them.
- This allows to push the boundaries of research:
  - more complex and precise calculations (HPC)
  - more extensive studies of larger samples (HTC)
- Community of material sciences: one of the most invested in this endeavor.



[1] https://www.top500.org/

A Crisis of Reproducibility



- Reproducibility: key property of the scientific method.
- Computational sciences: best equipped to address this problem (high levels of control over variables VS real life experiments).
- Even so, reproducibility is also difficult to achieve for them: why is that?

Nature 533, 452–454 (26 May 2016) doi:10.1038/533452a

The Provenance Model

- In order to see how this two elements interact, we need first to understand the complexity of scientific research.
- We need to model the procedures involved in research work...

The Provenance Model

- In order to see how this two elements interact, we need first to understand the complexity of scientific research.
- We need to model the procedures involved in research work...
  - Data nodes: pieces of information (a list of atomic positions, set of input parameters for a calculation, etc.)

The Provenance Model

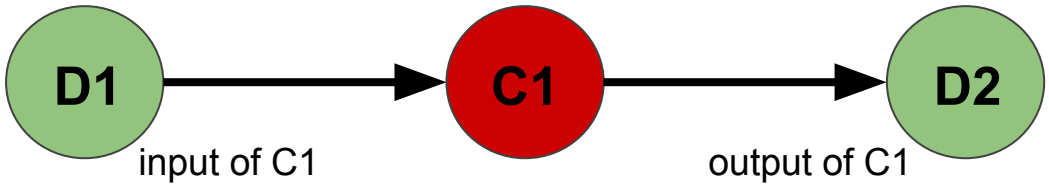
Francisco Ramirez

- In order to see how this two elements interact, we need first to understand the complexity of scientific research.
- We need to model the procedures involved in research work...
  - Data nodes: pieces of information (a list of atomic positions, set of input parameters for a calculation, etc.)
  - Calculation Nodes: processes that transform data.

The Provenance Model

Francisco Ramirez

- In order to see how this two elements interact, we need first to understand the complexity of scientific research.
- We need to model the procedures involved in research work...
  - Data nodes: pieces of information (a list of atomic positions, set of input parameters for a calculation, etc.)
  - Calculation Nodes: processes that transform data.

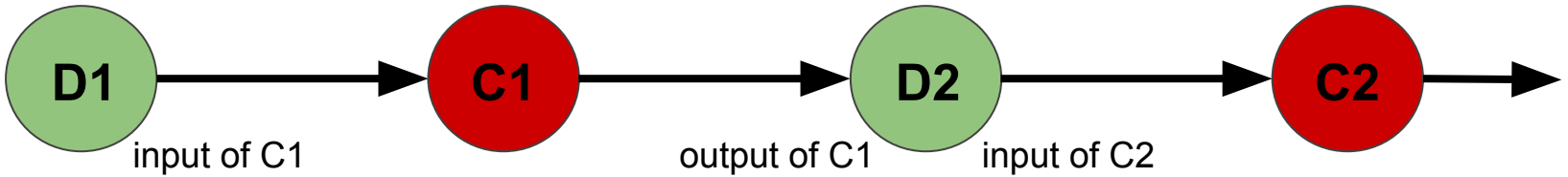


Information not only on the nodes but also on the links!

The Provenance Model

Francisco Ramirez

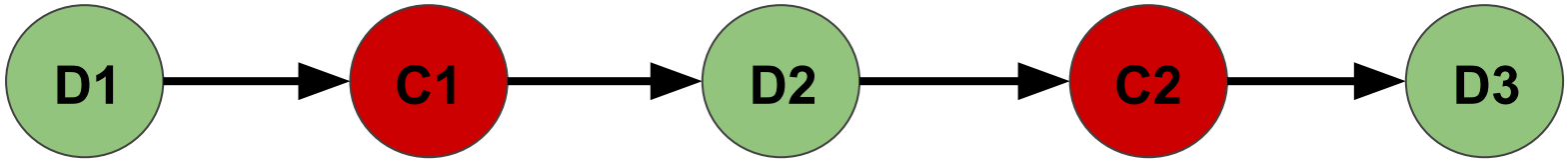
- In order to see how this two elements interact, we need first to understand the complexity of scientific research.
- We need to model the procedures involved in research work...
  - Data nodes: pieces of information (a list of atomic positions, set of input parameters for a calculation, etc.)
  - Calculation Nodes: processes that transform data.



Organizing Information

Francisco Ramirez

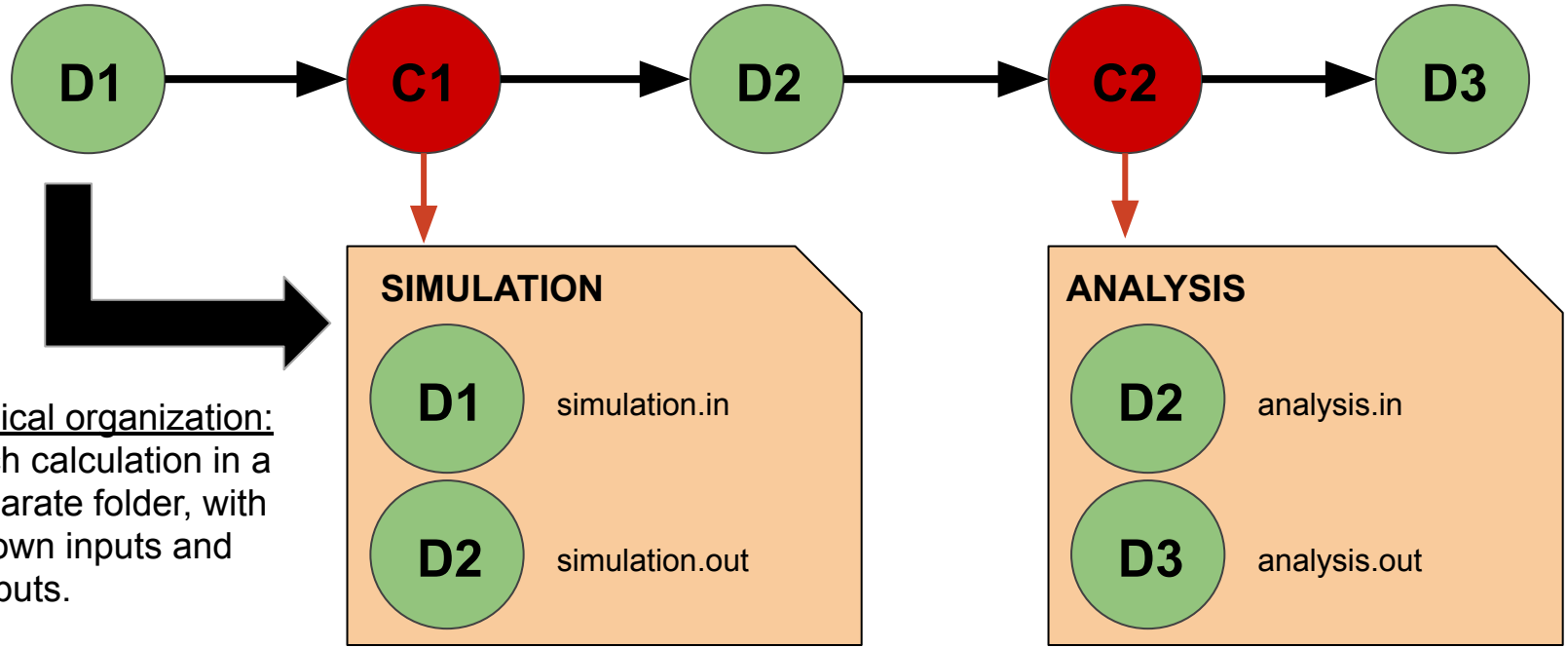
- Typical organizational system: folder structure (not well suited to represent these workflows).



Organizing Information

Francisco Ramirez

- Typical organizational system: folder structure (not well suited to represent these workflows).

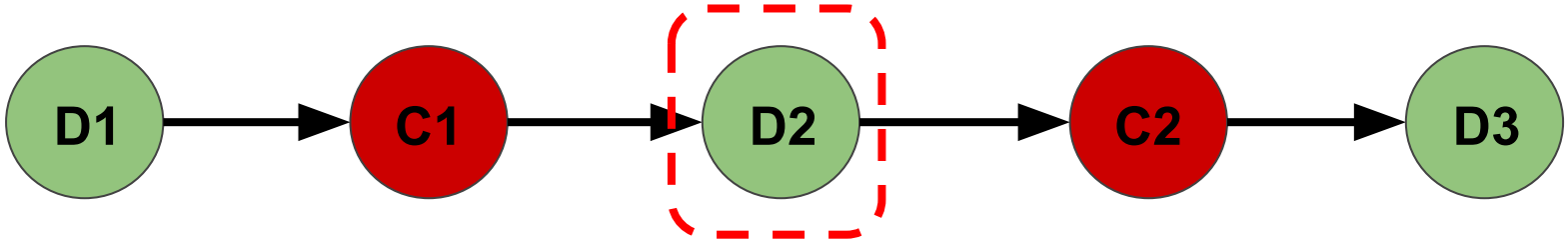


Typical organization: each calculation in a separate folder, with its own inputs and outputs.

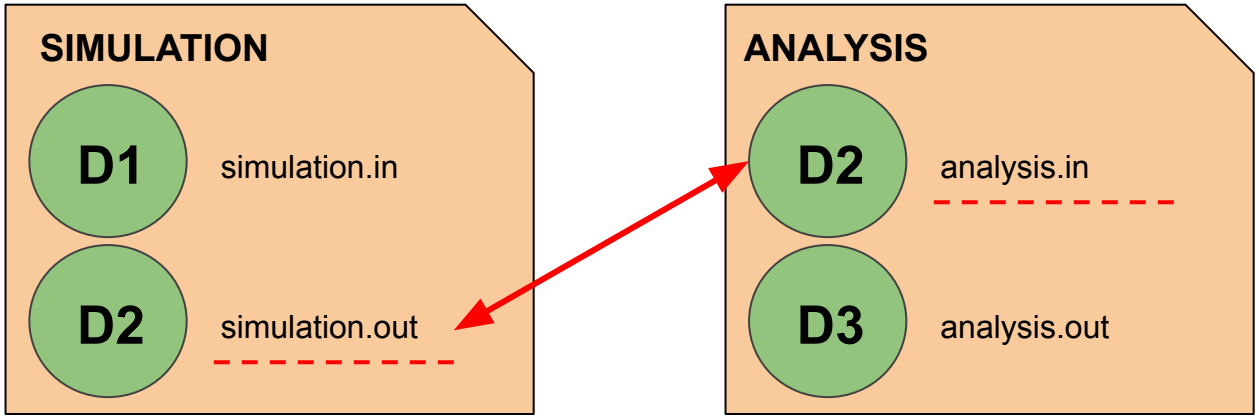
Organizing Information

Francisco Ramirez

- Typical organizational system: folder structure (not well suited to represent these workflows).



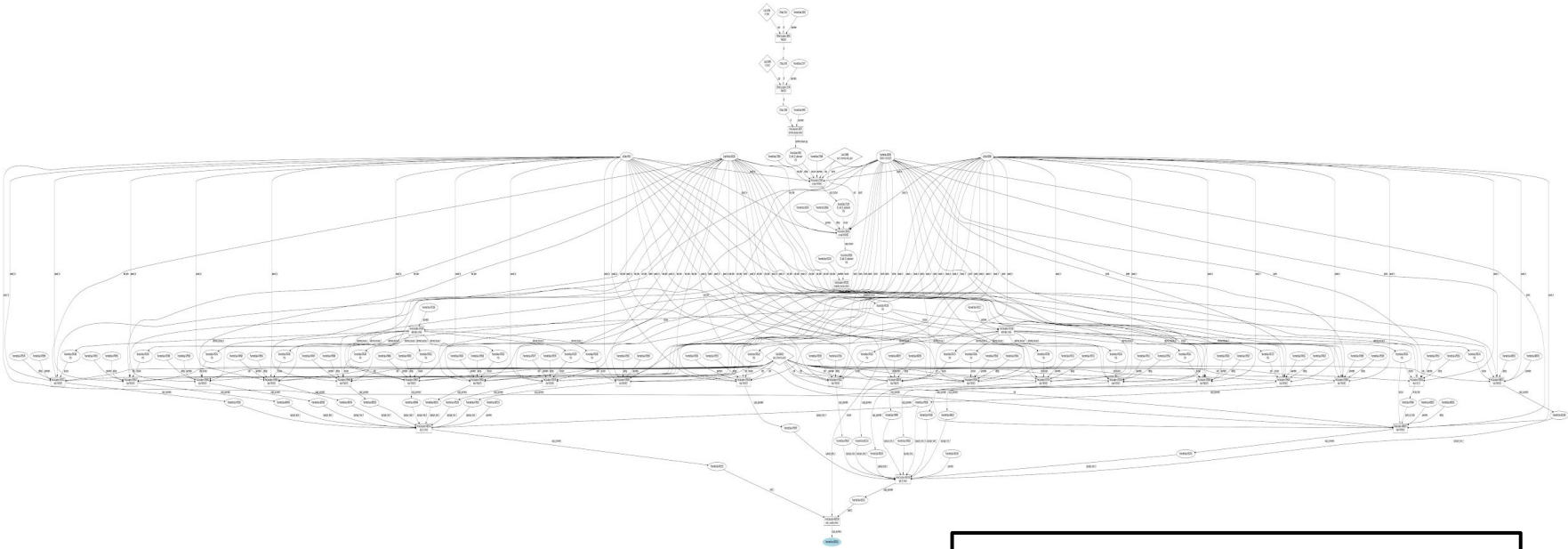
Problem:
The connections between pieces of information gets lost.
Other approaches possible: you always sever links.



The Complexity of Research

Francisco Ramirez

- Real scientific workflows can be much more complex…

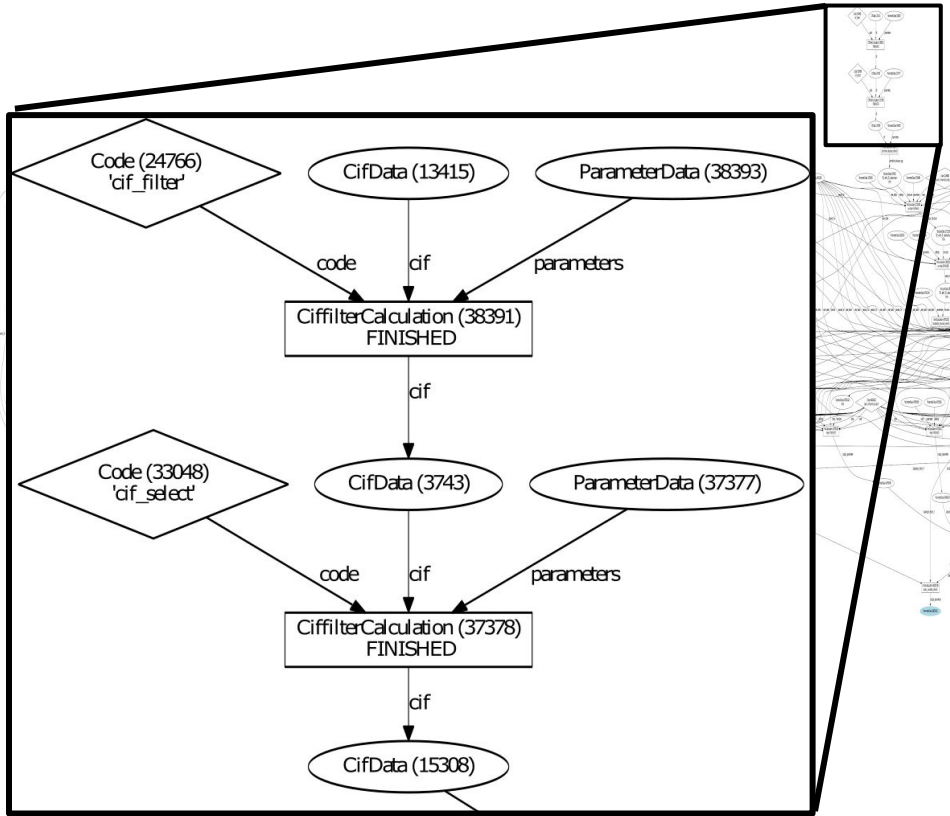


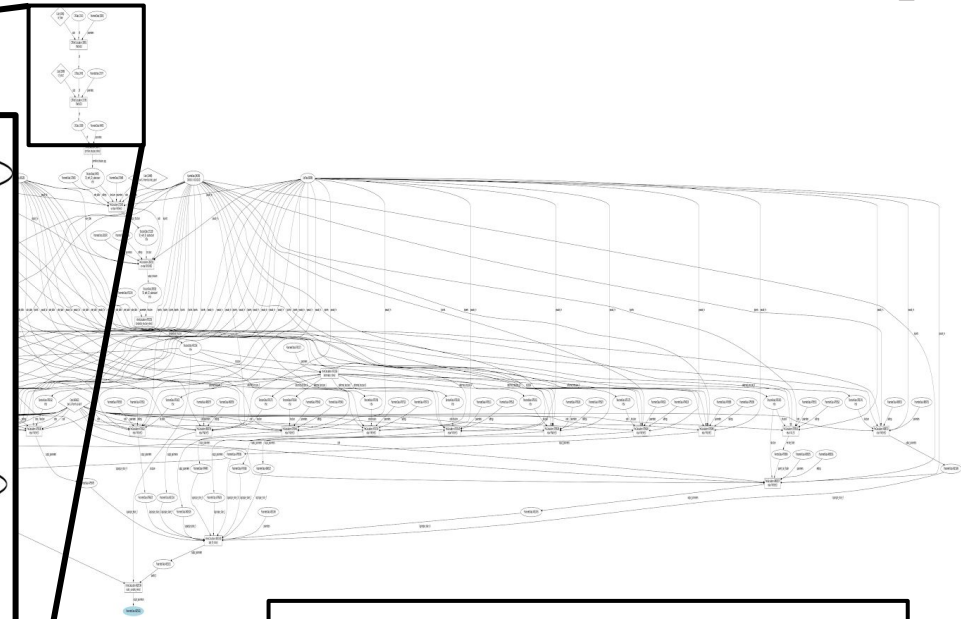
Calculation of the elastic constant for 1 material

The Complexity of Research

Francisco Ramirez

- Real scientific workflows can be much more complex…



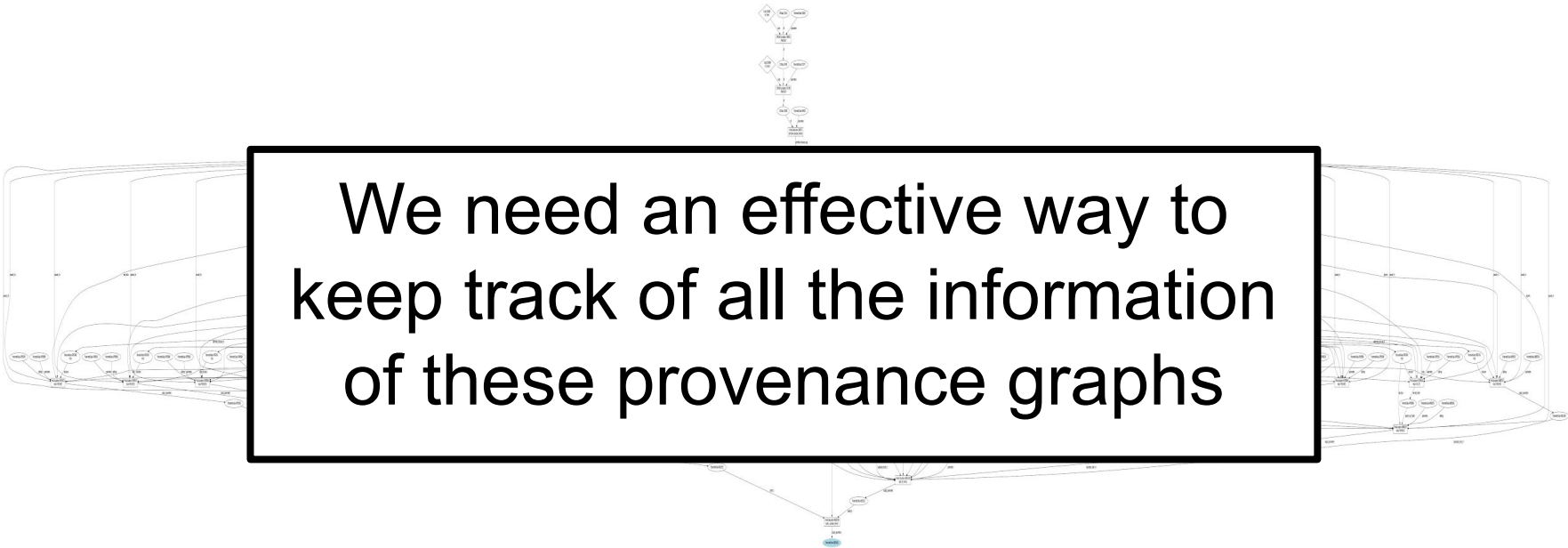


Calculation of the elastic constant for 1 material

The Complexity of Research

Francisco Ramirez

- Real scientific workflows can be much more complex…



We need an effective way to keep track of all the information of these provenance graphs

Francisco Ramirez

The Platform

AiiDA

General Overview

Francisco Ramirez

What AiiDA is...

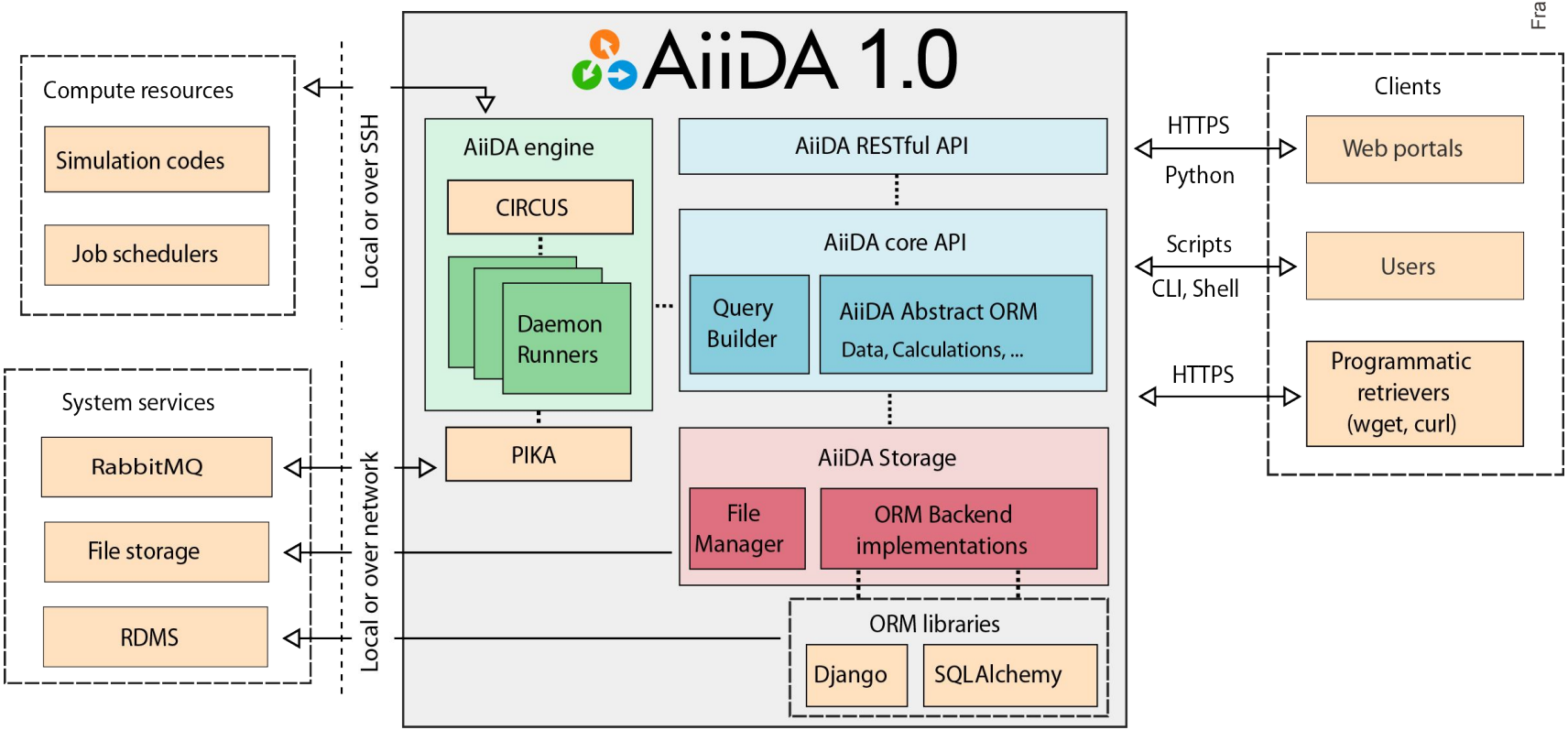
- Python Library (pip-installable)
- ORM for scripting
- Command line interface

What AiiDA does...

- Controls remote resources and manages calculation jobs
- Tracks the provenance and organizes the data.

General Overview

Francisco Ramirez



How it works

Francisco Ramirez

```
cmd$
```

procedure_tracked.py

```
def times2( number ):
    return 2 * number


def plus1( number ):
    return 1 + number


data0 = 10
data1 = times2(number=data0)
data2 = plus1(number=data1)
```

How it works

Francisco Ramirez

```
cmd$ python procedure.py

     21
```

procedure_tracked.py

```
def times2( number ):
    return 2 * number


def plus1( number ):
    return 1 + number


data0 = 10
data1 = times2(number=data0)
data2 = plus1(number=data1)
```

How it works

Francisco Ramirez

```
cmd$
```

procedure_tracked.py

```python
@calcfunction
def times2( number ):
    return 2 * number

@calcfunction
def plus1( number ):
    return 1 + number


data0 = orm.Int( 10 )
data1 = times2(number=data0)
data2 = plus1(number=data1)
```

Database:

How it works

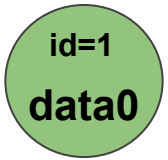
Francisco Ramirez

```
cmd$ verdi run procedure_tracked.py
```

procedure_tracked.py

```
@calcfunction
def times2( number ):
    return 2 * number

@calcfunction
def plus1( number ):
    return 1 + number


data0 = orm.Int( 10 )
data1 = times2(number=data0)
data2 = plus1(number=data1)
```

Database:



How it works

Francisco Ramirez

```
cmd$ verdi run procedure_tracked.py
```

procedure_tracked.py

```
@calcfunction
def times2( number ):
    return 2 * number

@calcfunction
def plus1( number ):
    return 1 + number


data0 = orm.Int( 10 )
data1 = times2(number=data0)
data2 = plus1(number=data1)
```

Database:

How it works

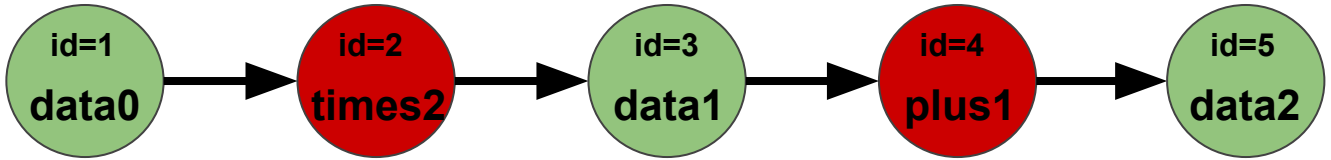
Francisco Ramirez

```
cmd$ verdi run procedure_tracked.py
```

procedure_tracked.py

```
@calcfunction
def times2( number ):
    return 2 * number

@calcfunction
def plus1( number ):
    return 1 + number


data0 = orm.Int( 10 )
data1 = times2(number=data0)
data2 = plus1(number=data1)
```

Database:



How it works

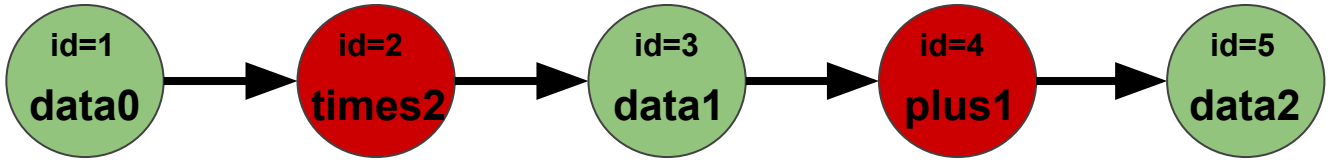
Francisco Ramirez

```
cmd$ verdi run procedure_tracked.py

cmd$
```

procedure_tracked.py

```python
@calcfunction
def times2( number ):
    return 2 * number

@calcfunction
def plus1( number ):
    return 1 + number


data0 = orm.Int( 10 )
data1 = times2(number=data0)
data2 = plus1(number=data1)
```

Database:



How it works

Francisco Ramirez

```
cmd$ verdi run procedure_tracked.py

cmd$
```

procedure_tracked.py

```
@calcfunction
def times2( number ):
    return 2 * number

@calcfunction
def plus1( number ):
    return 1 + number


data0 = orm.Int( 10 )
data1 = times2(number=data0)
data2 = plus1(number=data1)
```

Database:

How it works

Francisco Ramirez

```
cmd$ verdi run procedure_tracked.py

cmd$ verdi process list -a

                  Process  Process
  Id    Created   Label    State
 ---- --------- -------- ----------
  2    2m ago    times2   Finished [0]
  4    2m ago    plus1    Finished [0]

cmd$
```

procedure_tracked.py

```
@calcfunction
def times2( number ):
    return 2 * number

@calcfunction
def plus1( number ):
    return 1 + number


data0 = orm.Int( 10 )
data1 = times2(number=data0)
data2 = plus1(number=data1)
```

Database:

How it works

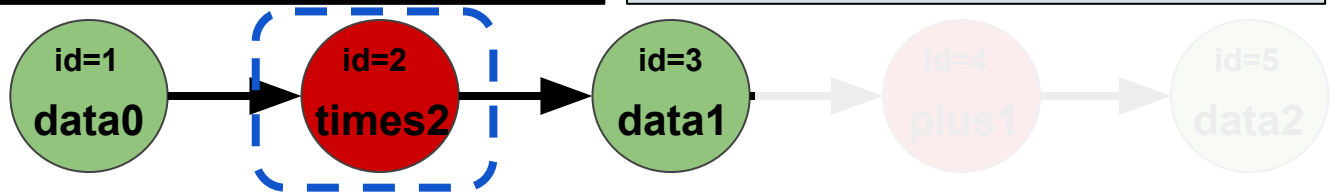
Francisco Ramirez

```
cmd$ verdi process show 2

   Property        Value
  ------------    -----------------
    type            times2
    state           Finished [0]
    (...)           (...)

   Inputs          Id       Type
  ------------    ------   --------
    number          1        Int

   Outputs         Id       Type
  ------------    ------   --------
    result          3        Int
```

procedure_tracked.py

```
@calcfunction
def times2( number ):
    return 2 * number

@calcfunction
def plus1( number ):
    return 1 + number


data0 = orm.Int( 10 )
data1 = times2(number=data0)
data2 = plus1(number=data1)
```

Database:



How it works

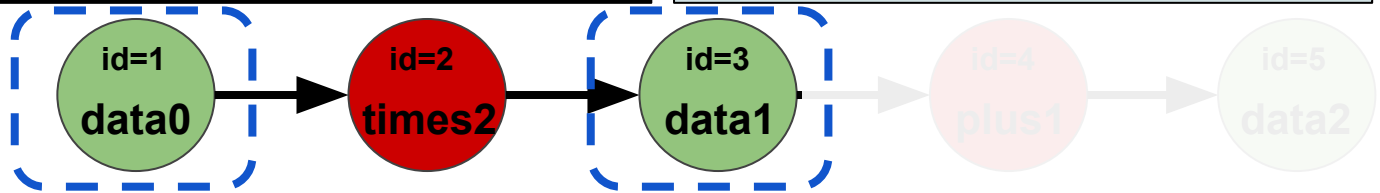
Francisco Ramirez

```
cmd$ verdi node attributes 1

     “value”: 10

cmd$ verdi node attributes 3

     “value”: 20
```

procedure_tracked.py

```
@calcfunction
def times2( number ):
     return 2 * number

@calcfunction
def plus1( number ):
     return 1 + number


data0 = orm.Int( 10 )
data1 = times2(number=data0)
data2 = plus1(number=data1)
```

Database:



How it works

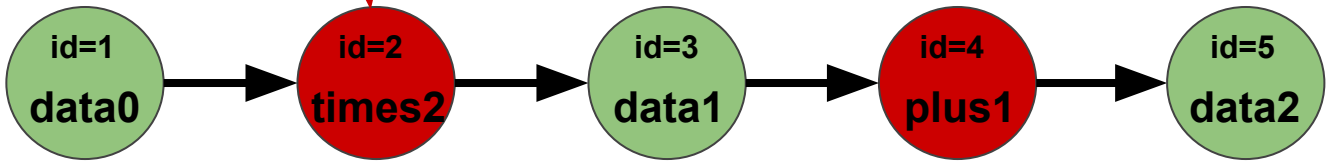
Francisco Ramirez

```
cmd$ verdi node repo cat 2 src

     @calcfunction
     def times2( number ):
         return 2 * number

     @calcfunction
     def plus1( number ):
         return 1 + number

     (...)

cmd$
```

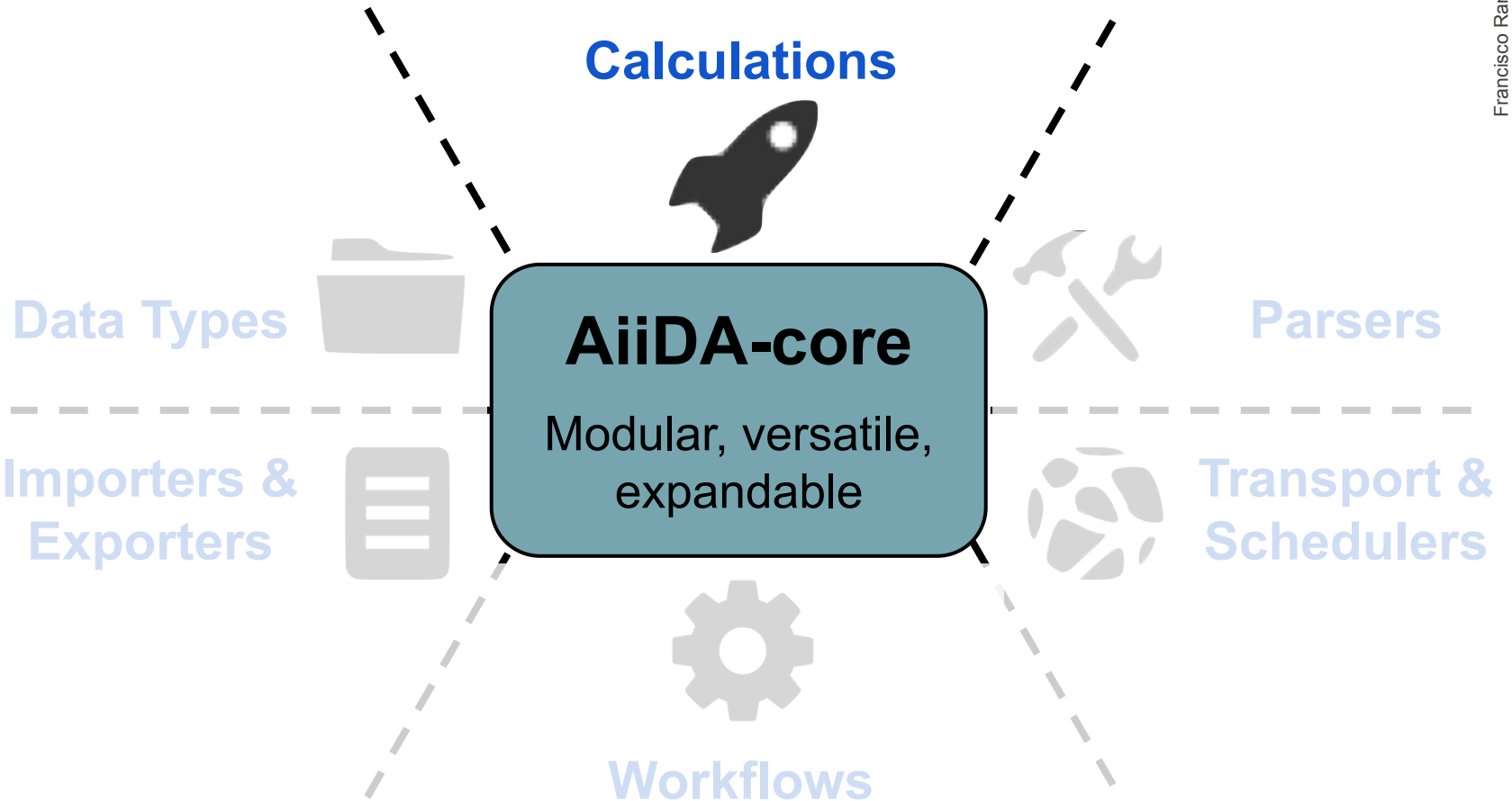
procedure_tracked.py

```
@calcfunction
def times2( number ):
    return 2 * number

@calcfunction
def plus1( number ):
    return 1 + number


data0 = orm.Int( 10 )
data1 = times2(number=data0)
data2 = plus1(number=data1)
```

Database:

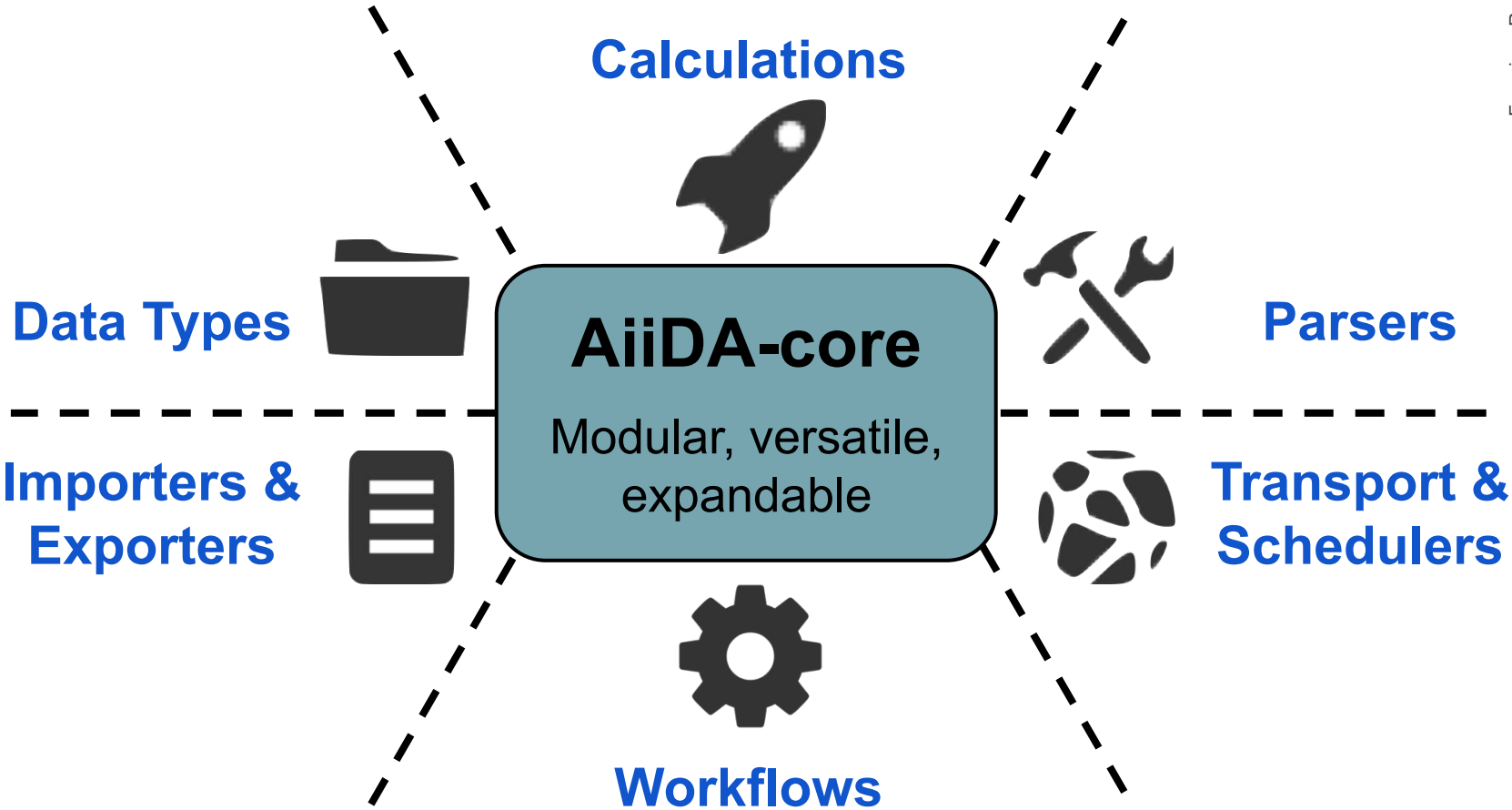


Extensibility & Plugins



Francisco Ramirez

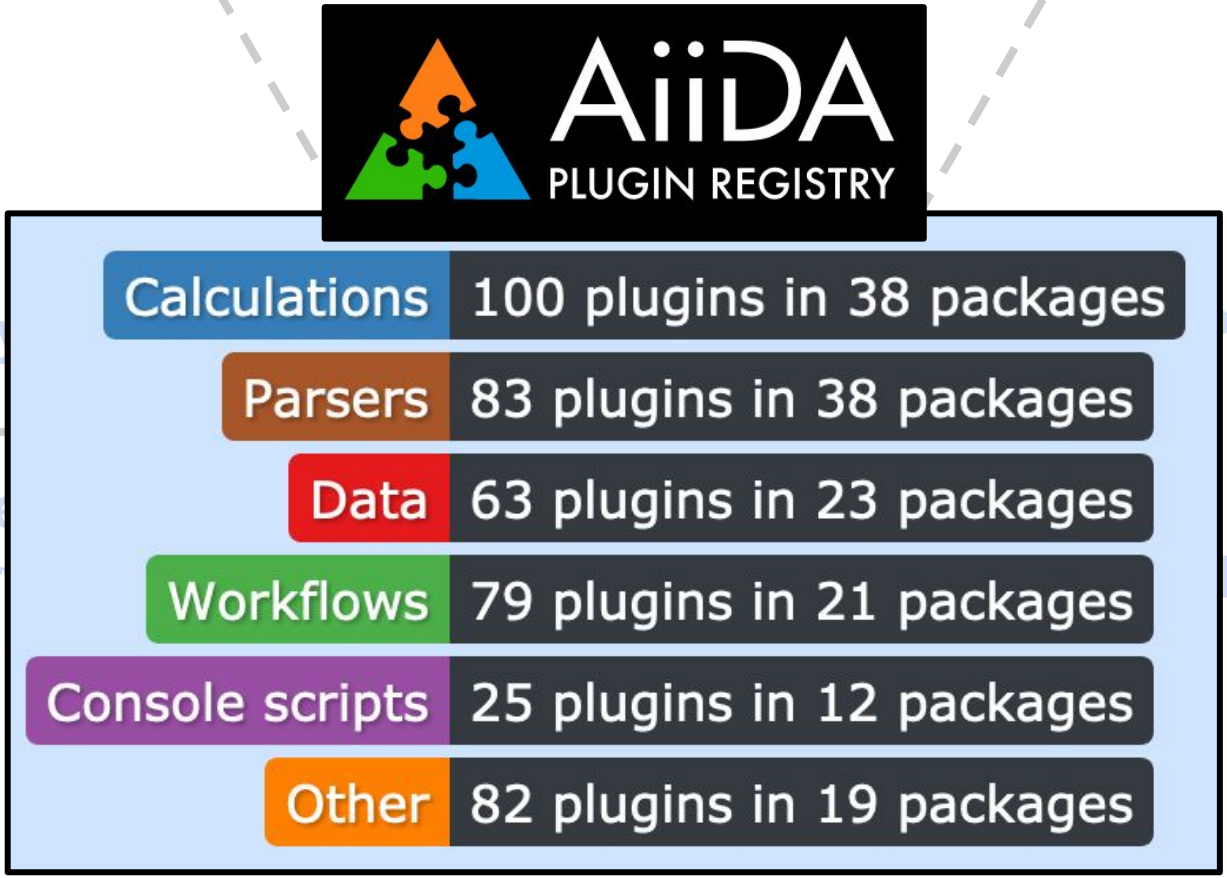
Extensibility & Plugins



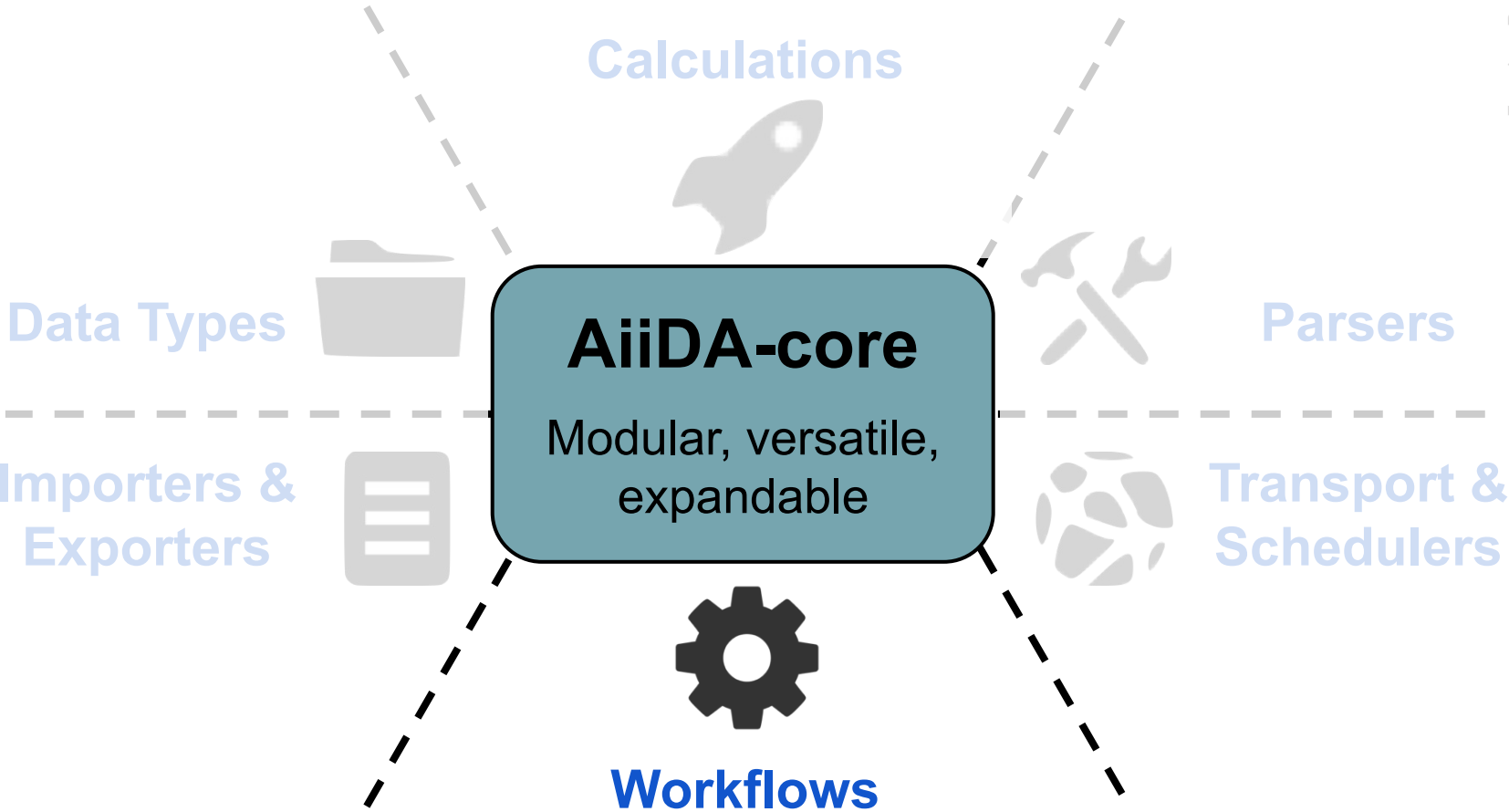
Francisco Ramirez

Extensibility & Plugins

Francisco Ramirez

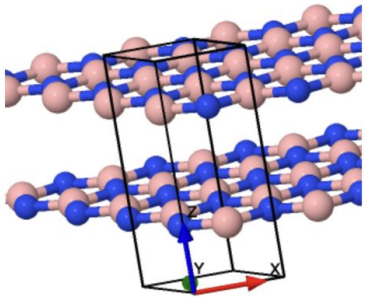


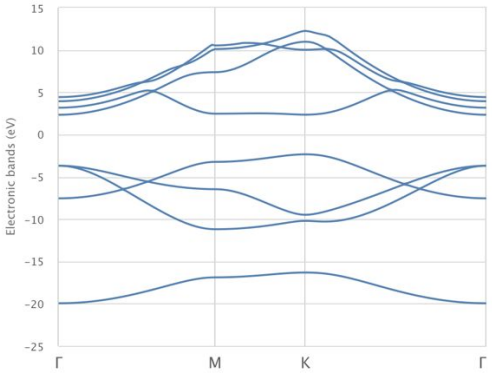
Extensibility & Plugins



Francisco Ramirez

Introducing Workflows

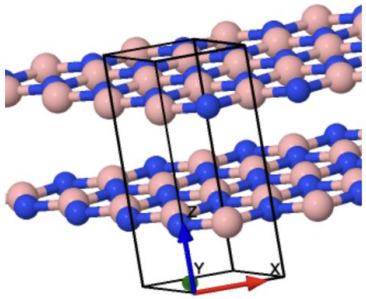


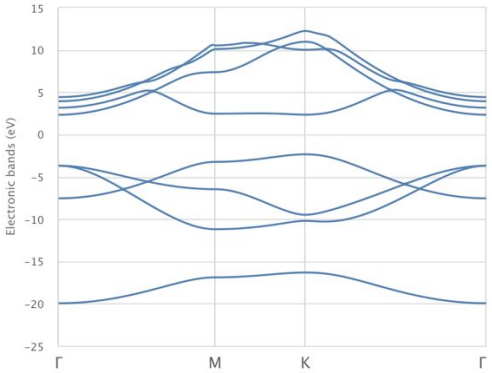


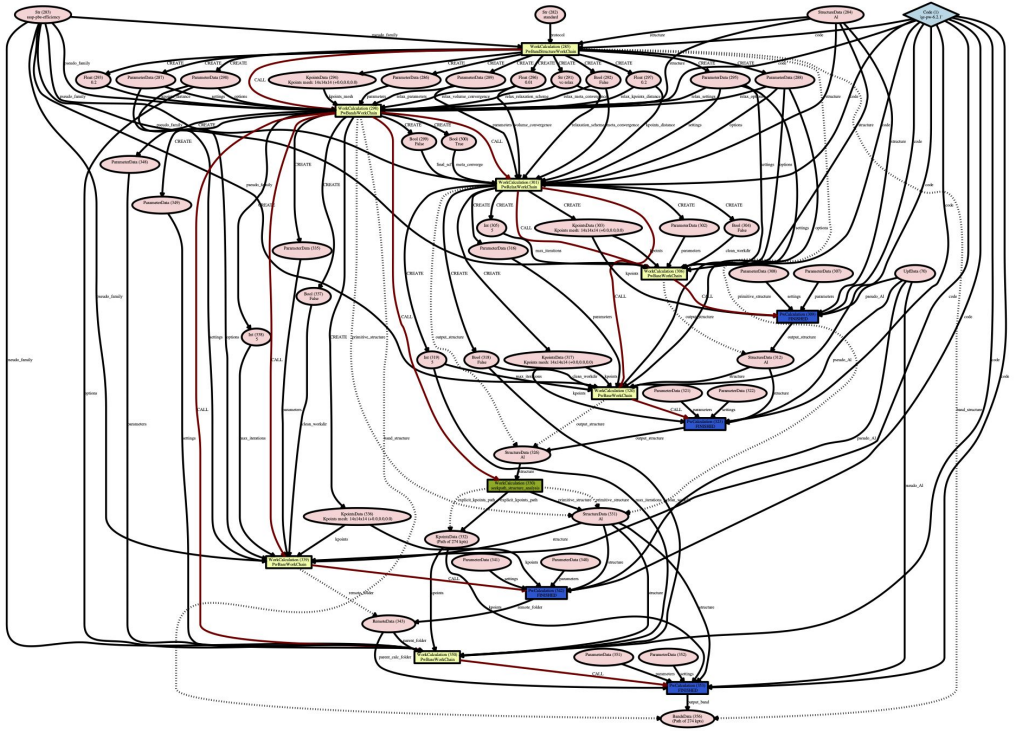
Francisco Ramirez

Introducing Workflows

Francisco Ramirez

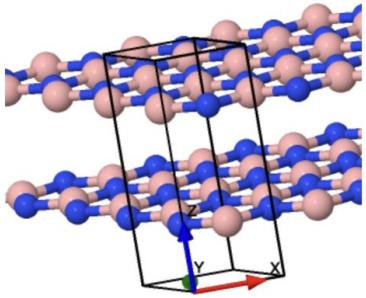


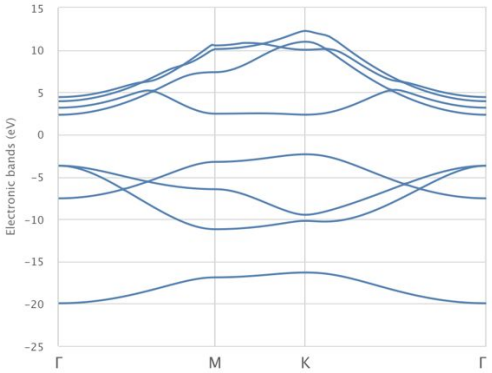




Introducing Workflows

Francisco Ramirez





- Scripting environment: allows to automate the full process involved (optimizing a geometry, calculating a property, etc).
- Hide complexity (computational and scientific) from the end user, makes tools available for a wider audience.
- AiiDA supports Workflows nodes and the concept of logical provenance.

Introducing Workflows

Francisco Ramirez

workflow.py

```
def workflow_x2p1( data0 ):
        data1 = times2(number=data0)
        data2 = plus1(number=data1)
        return data2


data_inp = orm.Int( 10 )
data_out = workflow_x2p1(data0=data_inp)
```

procedure_tracked.py

```
data0 = orm.Int( 10 )
data1 = times2(number=data0)
data2 = plus1(number=data1)
```

Database:

Introducing Workflows

Francisco Ramirez

workflow.py

```
def workflow_x2p1( data0 ):
        data1 = times2(number=data0)
        data2 = plus1(number=data1)
        return data2


data_inp = orm.Int( 10 )
data_out = workflow_x2p1(data0=data_inp)
```

```
cmd$ verdi run workflow.py
```

Database:

Introducing Workflows

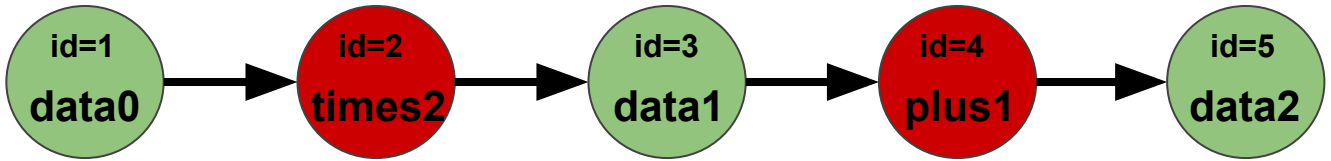
Francisco Ramirez

workflow.py

```
def workflow_x2p1( data0 ):
        data1 = times2(number=data0)
        data2 = plus1(number=data1)
        return data2


data_inp = orm.Int( 10 )
data_out = workflow_x2p1(data0=data_inp)
```

```
cmd$ verdi run workflow.py

cmd$
```

Database:



Introducing Workflows

Francisco Ramirez

workfunc.py

```
@workfunction
def workflow_x2p1( data0 ):
        data1 = times2(number=data0)
        data2 = plus1(number=data1)
        return data2


data_inp = orm.Int( 10 )
data_out = workflow_x2p1(data0=data_inp)
```

```
cmd$ verdi run workfunc.py
```

Database:

Introducing Workflows

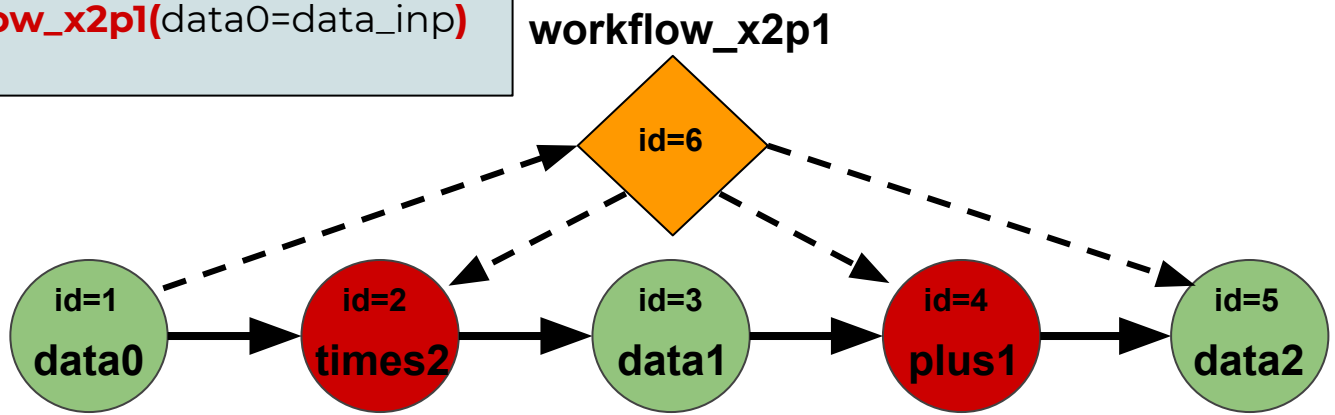
Francisco Ramirez

workfunc.py

```
@workfunction
def workflow_x2p1( data0 ):
        data1 = times2(number=data0)
        data2 = plus1(number=data1)
        return data2


data_inp = orm.Int( 10 )
data_out = workflow_x2p1(data0=data_inp)
```

```
cmd$ verdi run workfunc.py

cmd$
```

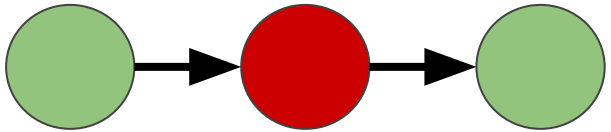
Database:



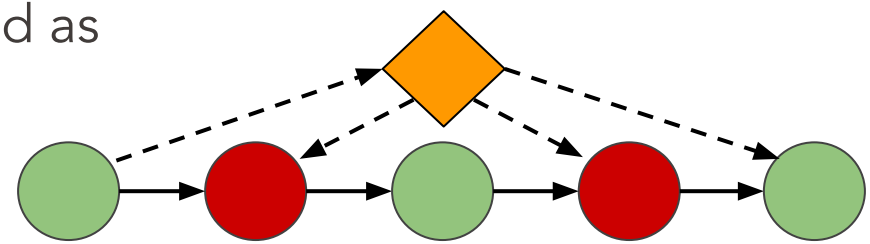
Summary of main features

- Locally controls remote resources, with an engine that centralizes its management.

- Automatically tracks the provenance and provides tools to explore it.



- Supports the implementation of high-level workflows to be used as turn-key solutions.



Funding and Online Resources

MARVEL

MARVEL National Centre for Competency in Research

MaX

European Centre of Excellence MaX

EPFL

École Polytechnique Fédérale de Lausanne

FNSNF

SWISS NATIONAL SCIENCE FOUNDATION

swissuniversities

MarketPlace

INTERSECT

erc

European Research Council

Established by the European Commission

European Research Council

pasc

Platform for Advanced Scientific Computing

PRACE

PARTNERSHIP FOR ADVANCED COMPUTING IN EUROPE

nffa.eu

nanoscience foundries & fine analysis

EMMC

The European Materials Modelling Council

AiiDA 1.x

S.P. Huber et al. arXiv:2003.12476 (2020)

Sebastiaan P. Huber[a,b], Spyros Zoupanos[a,b], Martin Uhrin[a,b], Leopold Talirz[a,b], Leonid Kahle[a,b], Rico Häuselmann[a,b], Dominik Gresch[c], Tiziano Müller[d], Aliaksandr V. Yakutovich[a,b], Casper W. Andersen[a,b], Francisco F. Ramirez[a,b], Carl S. Adorf[a,b], Snehal Kumbhar[a,b], Elsa Passaro[a,b], Conrad Johnston[a,b], Nicolas Mounet[a,b], Nicola Marzari[a,b] and Giovanni Pizzi[a,b]

WEBSITE
http://www.aiida.net

MATERIALS CLOUD
https://materialscloud.org

SOURCE CODE
github.com/aiidateam/aiida-core

DOCUMENTATION
https://aiida.readthedocs.io

Developers

Francisco Ramirez

The Materials Cloud And AiiDA teams


Carl Simon Adorf (EPFL)


Casper W. Andersen (EPFL)


Marnik Bercx (EPFL)


Marco Borelli (EPFL)


Valeria Granata (EPFL)


Sebastiaan P. Huber (EPFL)


Leonid Kahle (EPFL)


Snehal P. Kumbhar (EPFL)


Elsa Passaro (EPFL)


Francisco F. Ramirez (EPFL)


Leopold Talirz (EPFL)


Aliaksandr Yakutovich (EPFL)


Chris Sewell (EPFL)


Giovanni Pizzi (EPFL)


Berend Smit (EPFL)


Joost VandeVondele (ETHZ,CSCS)


Thomas Schulthess (ETHZ,CSCS)


Nicola Marzari (EPFL)

Contributors for the 40+ plugins: Quantum ESPRESSO, Wannier90, CP2K, FLEUR, YAMBO, SIESTA, VASP, CASTEP, CRYSTAL, …

Contributors to aiida-core and former AiiDA team members —
Oscar Arbelaez, Michael Atambo, Valentin Bersier, Marco Borelli, Jocelyn Boullier, Jens Bröder, Ivano E. Castelli, Andrea Cepellotti, Keija Cui, Vladimir Dikan, Marco Dorigo, Y.-W. Fang, Fernando Gargiulo, Marco Gibertini, Davide Grassano, Dominik Gresch, Conrad Johnston, Rico Häuselmann, Daniel Hollas, Eric Hontz, Jianxing Huang, Christoph Koch, Espen Flage-Larsen, Ian Lee, Daniel Marchand, Antimo Marrazzo, Andrius Merkys, Simon Pintarelli, Nicolas Mounet, Tiziano Müller, Gianluca Prandini, Philip Rüßmann, Riccardo Sabatini, Ole Schütt, Phillippe Schwaller, Andreas Stamminger, Atsushi Togo, Daniele Tomerini, Nicola Varini, Martin Uhrin, Jason Yu, Austin Zadoks, Bonan Zhu, Mario Zic, Spyros Zoupanos