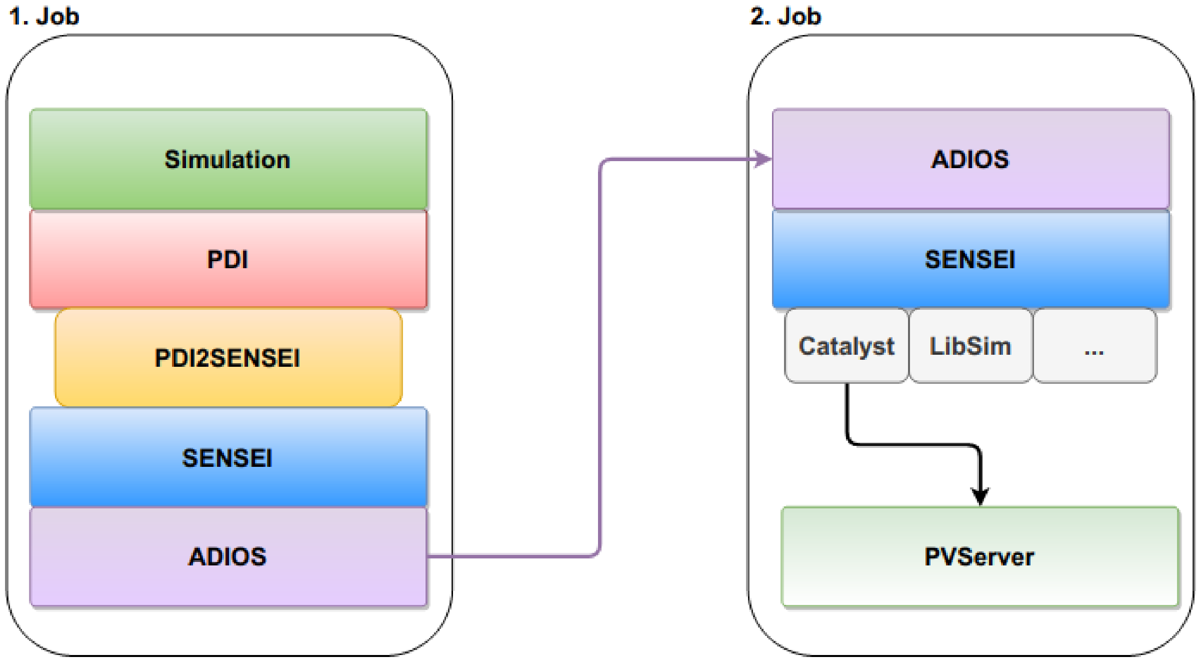


INCLUDING IN SITU VISUALIZATION AND ANALYSIS IN PDI

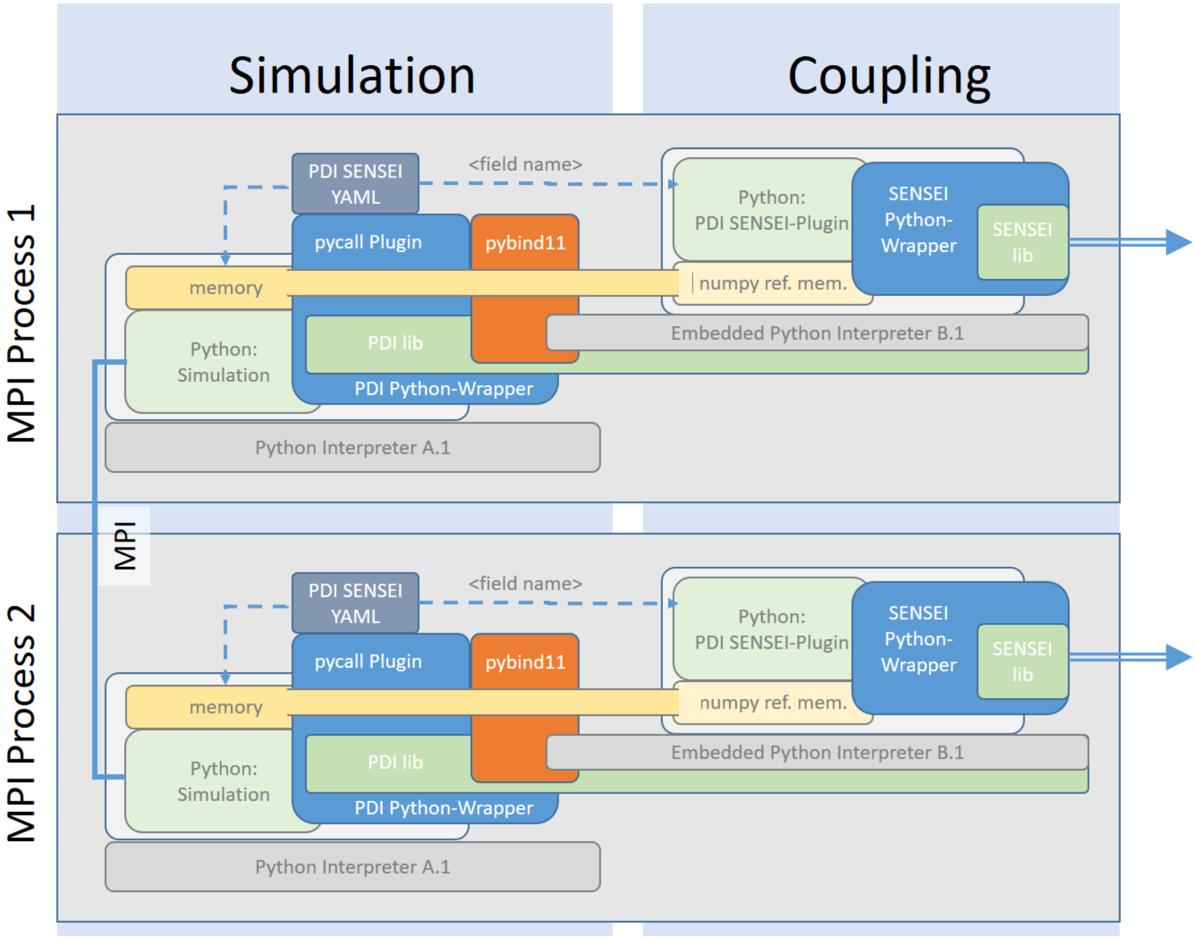
2021-07-02 I **CHRISTIAN WITZLER** (C.WITZLER@FZ-JUELICH.DE), J. MIGUEL ZAVALA-AKE (MIGUELZA@KTH.SE), KAROL SIEROCINSKI (KSIERO@MAN.POZNAN.PL), HERBERT OWEN (HERBERT.OWEN@BSC.ES)

Founded by:
- EoCoE II, grant agreement No 824158
- CoEC, grant agreement No 952181

Member of the Helmholtz Association

OUTLINE

- Motivation
- PDI
- Design decisions
  - Sensei
  - Adios2
  - ParaView Catalyst
- Usage
  - PDI integration
  - YAML file
  - Catalyst
- Example Bolund



Simulation

Coupling

MPI Process 1

PDI SENSEI YAML

<field name>

pycall Plugin

pybind11

Python: PDI SENSEI-Plugin

SENSEI Python-Wrapper

SENSEI lib

memory

numpy ref. mem.

Python: Simulation

PDI lib

Embedded Python Interpreter B.1

PDI Python-Wrapper

Python Interpreter A.1

MPI

MPI Process 2

PDI SENSEI YAML

<field name>

pycall Plugin

pybind11

Python: PDI SENSEI-Plugin

SENSEI Python-Wrapper

SENSEI lib

memory

numpy ref. mem.

Python: Simulation

PDI lib

Embedded Python Interpreter B.1

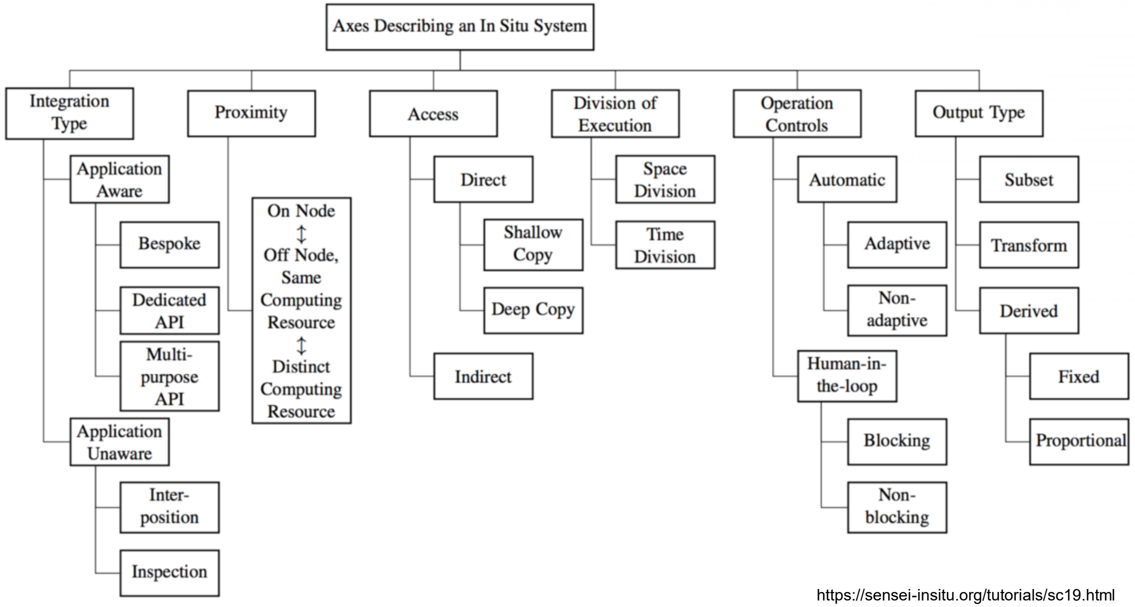
PDI Python-Wrapper

Python Interpreter A.1

MOTIVATION

What do we need?

- Minimal dependencies for simulation code
- Error resistant visualization
- Minimal influence on simulation
- Easy to use

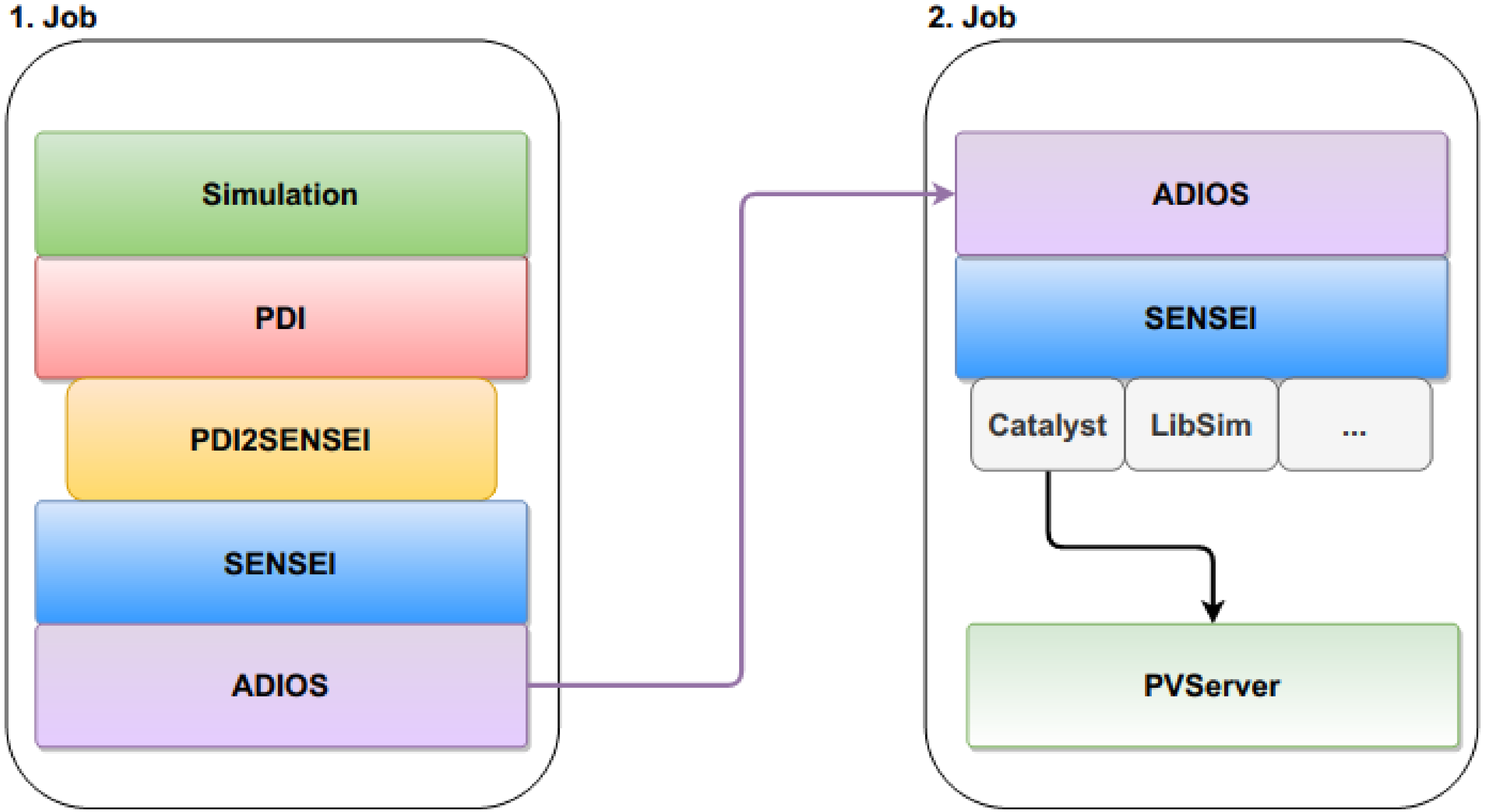


Axes Describing an In Situ System

- Integration Type
  - Application Aware
    - Bespoke
    - Dedicated API
    - Multi-purpose API
  - Application Unaware
    - Inter-position
    - Inspection
- Proximity
  - On Node ↕ Off Node, Same Computing Resource ↕ Distinct Computing Resource
- Access
  - Direct
    - Shallow Copy
    - Deep Copy
  - Indirect
- Division of Execution
  - Space Division
  - Time Division
- Operation Controls
  - Automatic
    - Adaptive
    - Non-adaptive
  - Human-in-the-loop
    - Blocking
    - Non-blocking
- Output Type
  - Subset
  - Transform
  - Derived
    - Fixed
    - Proportional

https://sensei-insitu.org/tutorials/sc19.html

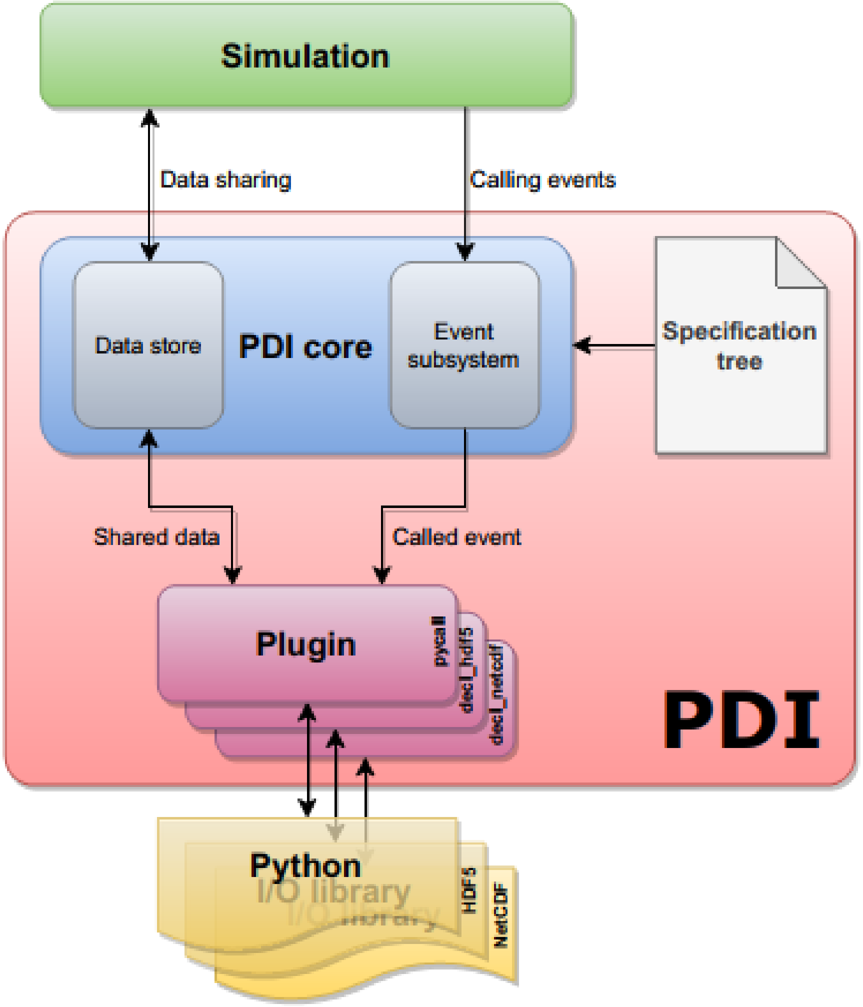
DESIGN


1. Job
Simulation
PDI
PDI2SENSEI
SENSEI
ADIOS
2. Job
ADIOS
SENSEI
Catalyst
LibSim
...
PVServer

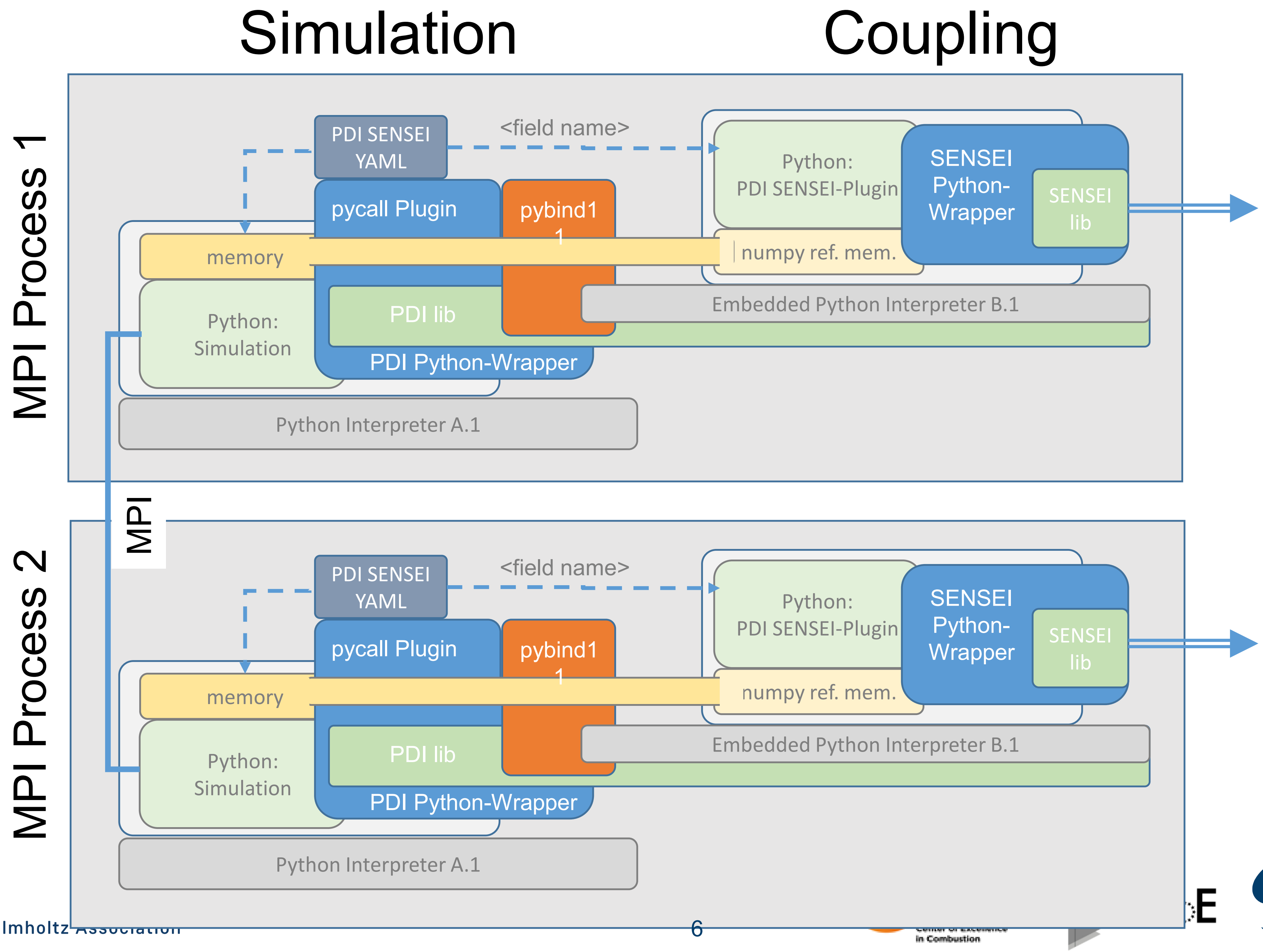
PDI DATA INTERFACE

- Decouple I/O from HPC codes

- 3 parts:
  - PDI core
  - Specification tree
  - Plugins



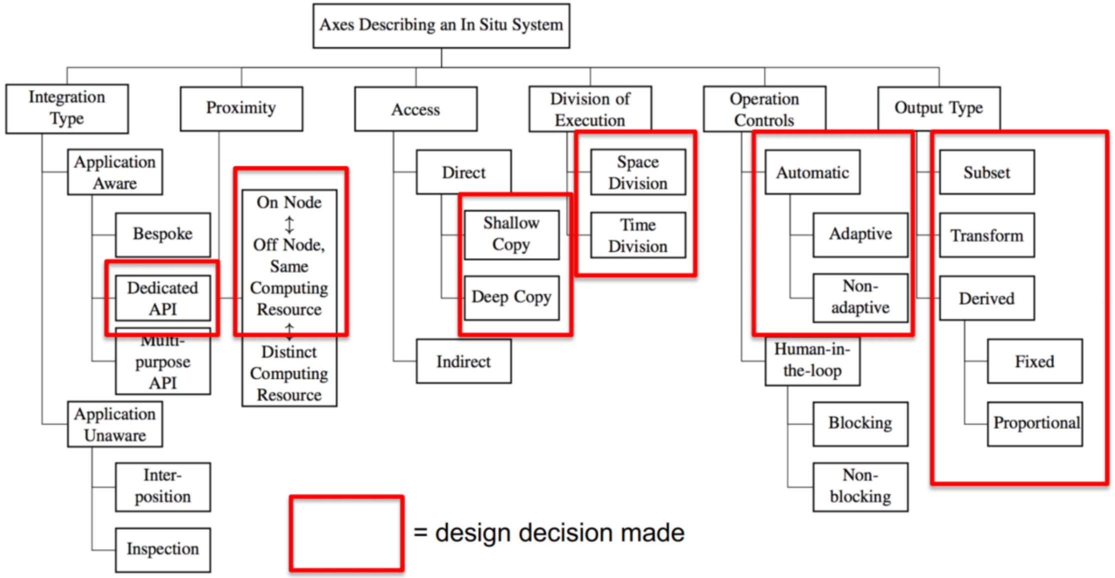
https://pdi.julien-bigot.fr/1.2/
https://gitlab.maisondelasimulation.fr/pdidev/pdi
https://hal.archives-ouvertes.fr/hal-01587075


Simulation
Coupling
MPI Process 1
PDI SENSEI YAML
<field name>
pycall Plugin
pybind1 1
memory
Python: Simulation
PDI lib
PDI Python-Wrapper
Python: PDI SENSEI-Plugin
SENSEI Python-Wrapper
SENSEI lib
numpy ref. mem.
Embedded Python Interpreter B.1
Python Interpreter A.1
MPI
MPI Process 2
PDI SENSEI YAML
<field name>
pycall Plugin
pybind1 1
memory
Python: Simulation
PDI lib
PDI Python-Wrapper
Python: PDI SENSEI-Plugin
SENSEI Python-Wrapper
SENSEI lib
numpy ref. mem.
Embedded Python Interpreter B.1
Python Interpreter A.1

DESIGN DECISIONS

SENSEI

- Flexible connection to different frameworks
  - Adios, Catalyst, LibSim
  - Allows in-transit
- But:
  - No build in data transport



Axes Describing an In Situ System

Integration Type
- Application Aware
  - Bespoke
  - Dedicated API
  - Multi-purpose API
- Application Unaware
  - Inter-position
  - Inspection

Proximity
- On Node ↕ Off Node, Same Computing Resource ↕ Distinct Computing Resource

Access
- Direct
  - Shallow Copy
  - Deep Copy
- Indirect

Division of Execution
- Space Division
- Time Division

Operation Controls
- Automatic
  - Adaptive
  - Non-adaptive
- Human-in-the-loop
  - Blocking
  - Non-blocking

Output Type
- Subset
- Transform
- Derived
  - Fixed
  - Proportional

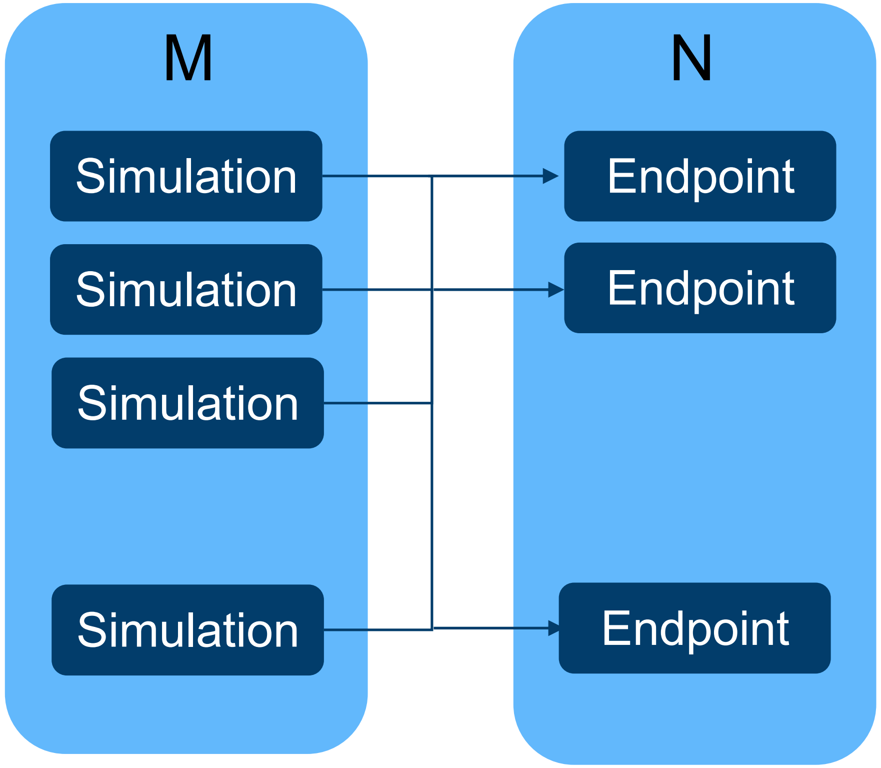
□ = design decision made

https://sensei-insitu.org/tutorials/sc19.html

DESIGN DECISIONS

ADIOS2

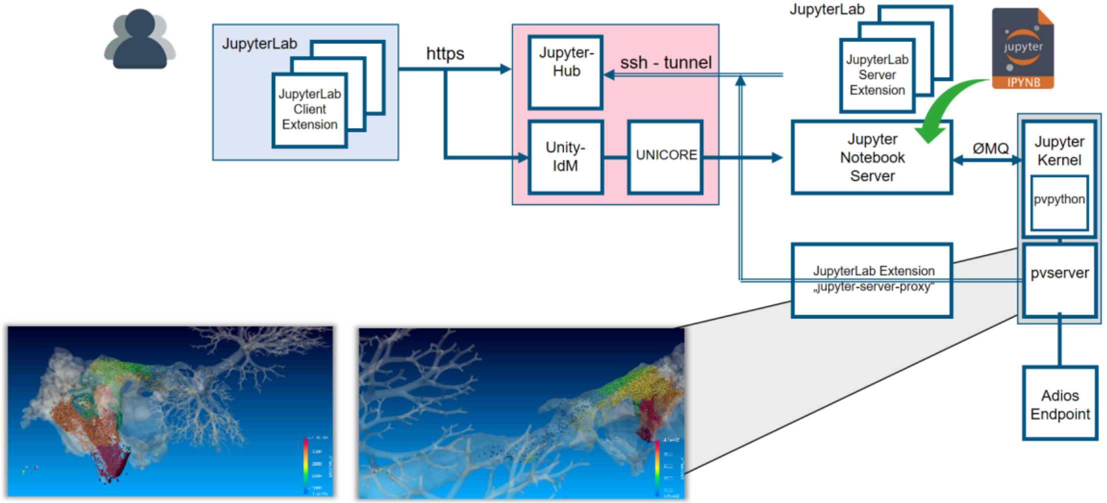
- Using Adios2 SST for data transport
  - Simple to use
  - Scales well
  - Allows M2N communication
  - Uses RDMA
- But:
  - Creates configuration file
    - Needs shared file system



DESIGN DECISIONS

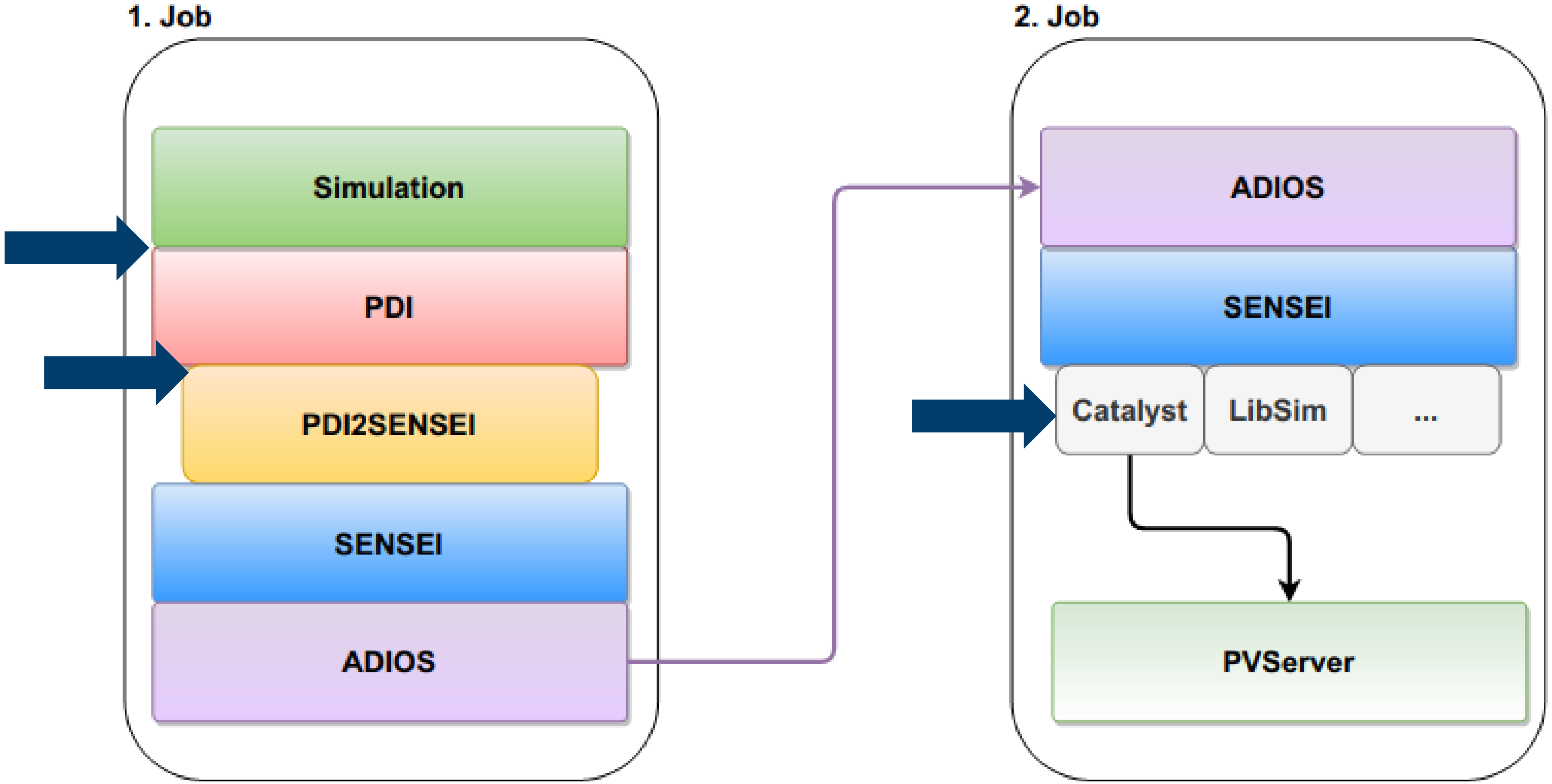
ParaView Catalyst

- Used for visualization
- Can be run in parallel
- Predefined visualization scripts
- Interaktive live visualization
- Is integrated in Jupyter-JSC



https://www.paraview.org/in-situ/
https://cfp.jupytercon.com/2020/schedule/presentation/123/jupyter-for-interactive-in-situ-visualization-with-paraviewcatalyst/

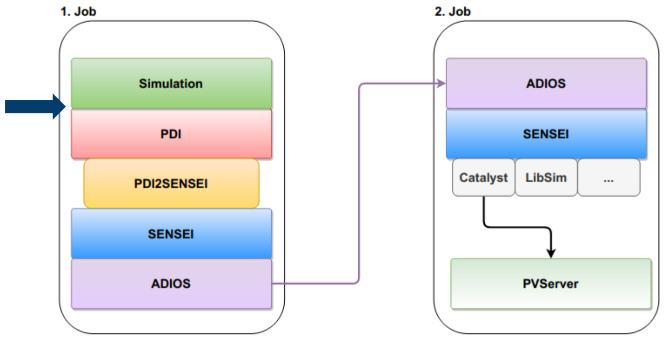
USAGE

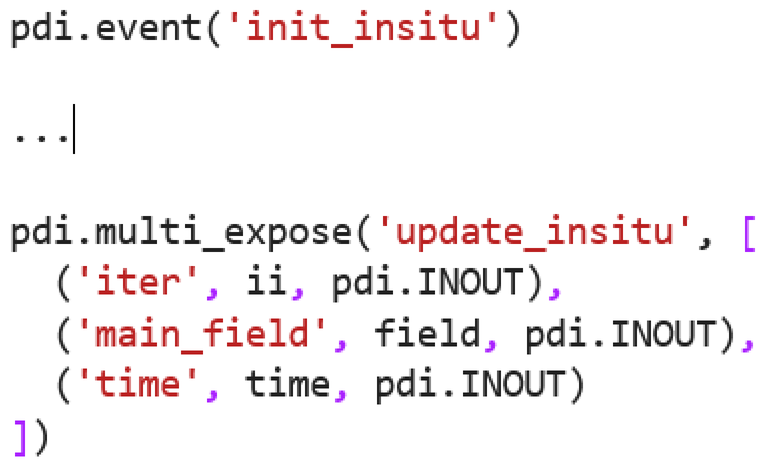

1. Job
Simulation
PDI
PDI2SENSEI
SENSEI
ADIOS
2. Job
ADIOS
SENSEI
Catalyst
LibSim
...
PVServer

USAGE

PDI integration

- PDI Events
  - Initialization
  - In-situ update
  - Finalization
- PDI Data:
  - Data share/expose



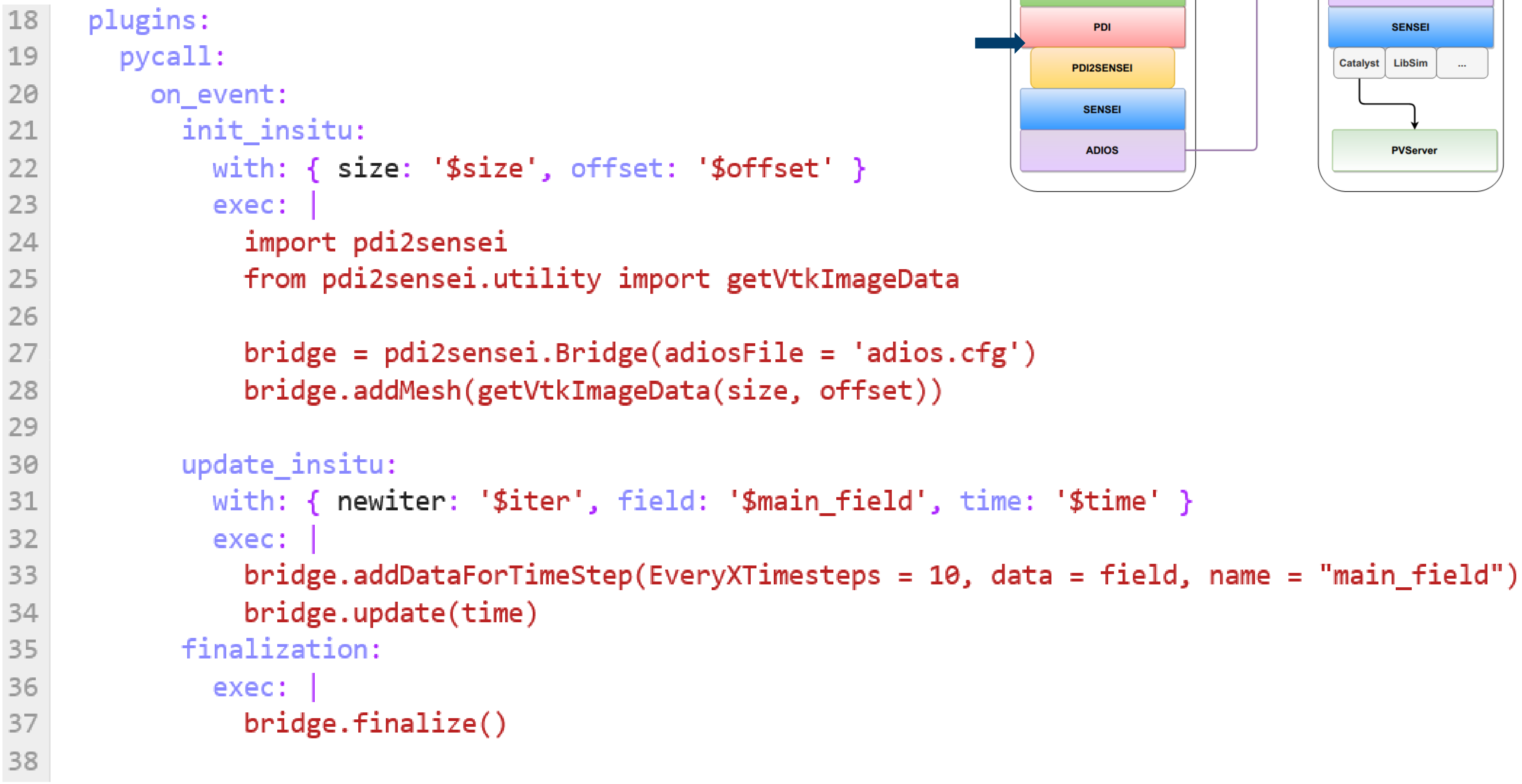


```
pdi.event('init_insitu')

...

pdi.multi_expose('update_insitu', [
  ('iter', ii, pdi.INOUT),
  ('main_field', field, pdi.INOUT),
  ('time', time, pdi.INOUT)
])
```

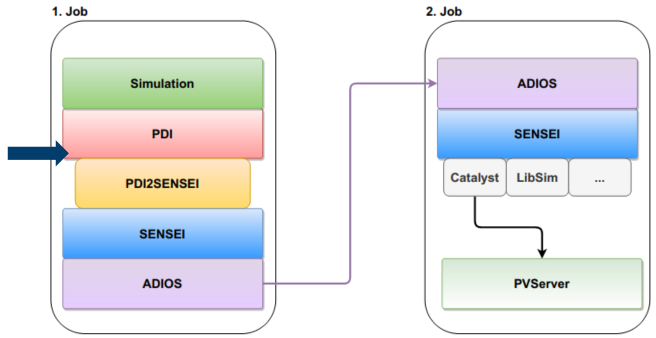
USAGE

YAML file

- Connect the Events and data shares



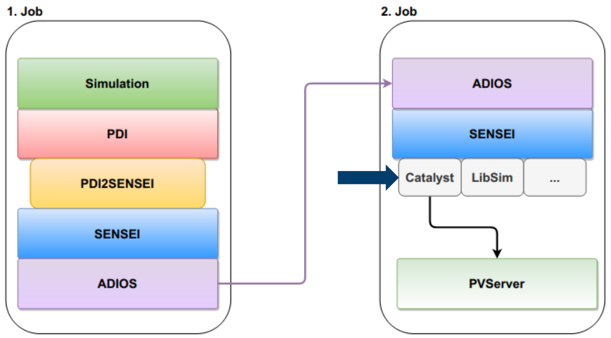
```yaml
plugins:
  pycall:
    on_event:
      init_insitu:
        with: { size: '$size', offset: '$offset' }
        exec: |
          import pdi2sensei
          from pdi2sensei.utility import getVtkImageData

          bridge = pdi2sensei.Bridge(adiosFile = 'adios.cfg')
          bridge.addMesh(getVtkImageData(size, offset))

      update_insitu:
        with: { newiter: '$iter', field: '$main_field', time: '$time' }
        exec: |
          bridge.addDataForTimeStep(EveryXTimesteps = 10, data = field, name = "main_field")
          bridge.update(time)
      finalization:
        exec: |
          bridge.finalize()

```

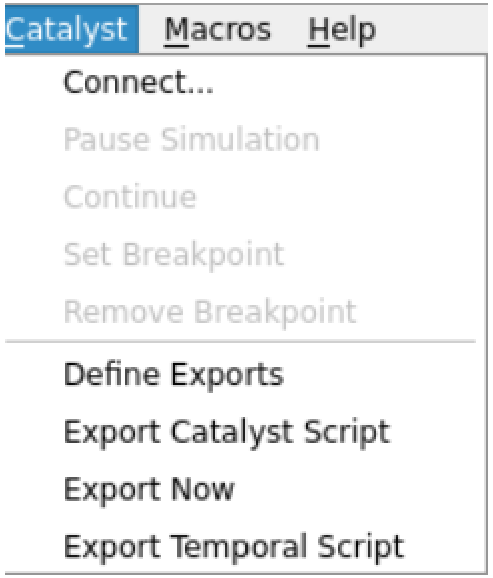


USAGE

Catalyst

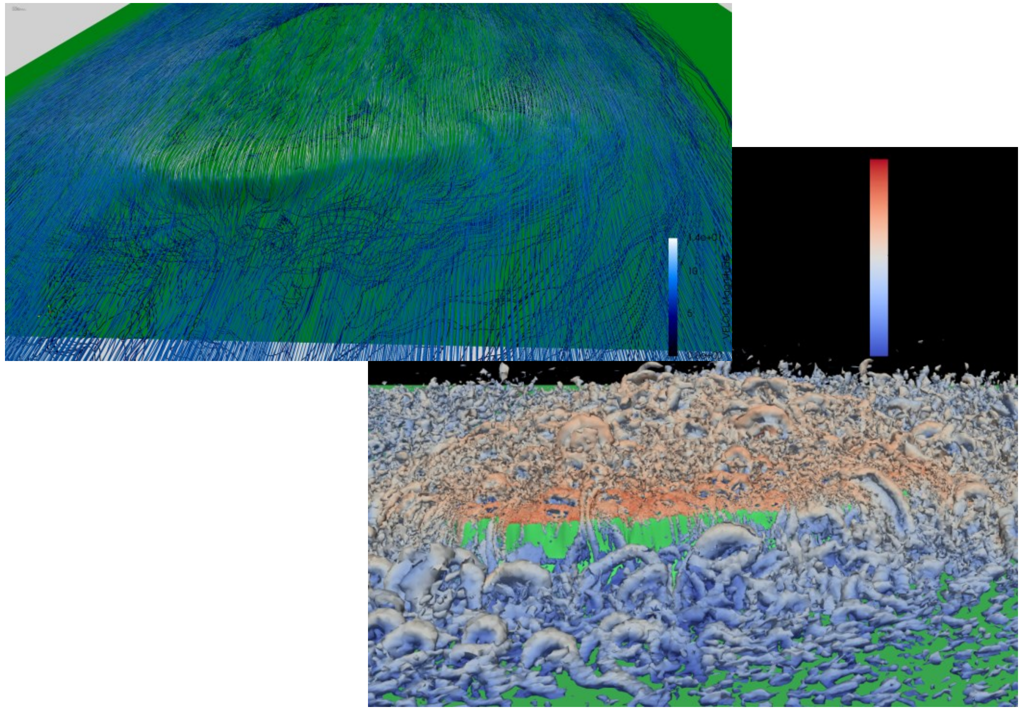
- Connect with a small simulation or load a sample file
- Setup your visualisation pipeline
- Export catalyst Script





EXAMPLE CASE: BOLUND

- ALYA
  - parallel multi-physics/multi-scale simulation code
  - Developed at BSC
- Example case Bolund
  - Incrompressible flow
  - Large Eddy Simulation



SUMMARY

- Minimal dependencies on simulation site
  - Only pdi needed
    - **All other dependencies are indirekt**
- In-transit visualization
  - 2nd (3rd) visualization job
    - **Independent visualization**
- Easy configuration
  - YAML file
    - **Simple functions to setup your in-transit workflow**

QUESTIONS?

Thank you for your attention

Acknowledgments:
This work was supported by the European Council under the Horizon 2020 Project Energy oriented Center of Excellence for computing applications - EoCoE II, grant agreement No 824158 and CoEC, grant agreement No 952181

Additional thanks to the PDI-developer team as well as the ATML Vis group from JSC.