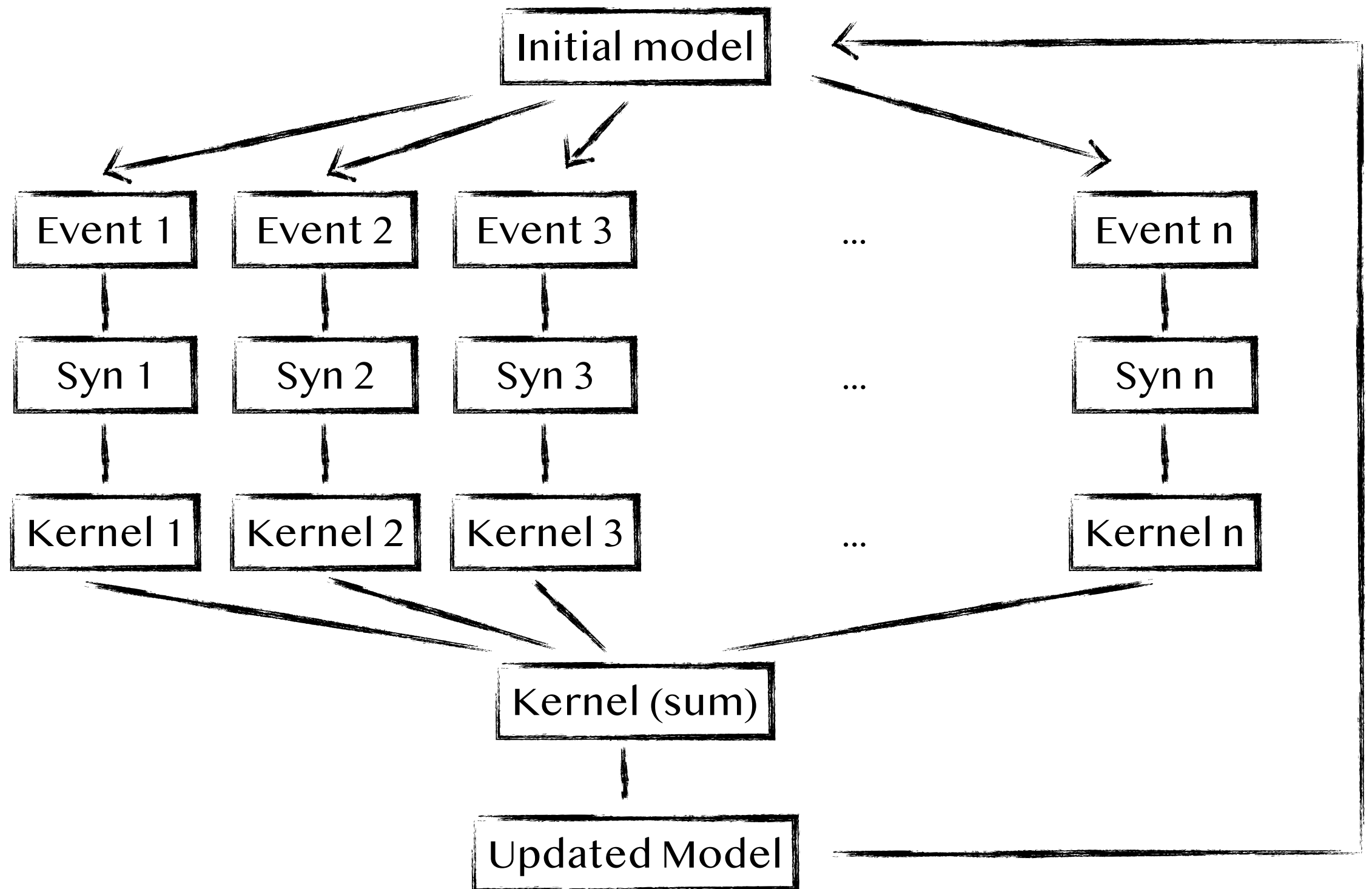


nnodes

A workflow manager for Python functions and executables
<https://github.com/icui/nnodes>

Motivation

Full waveform inversion



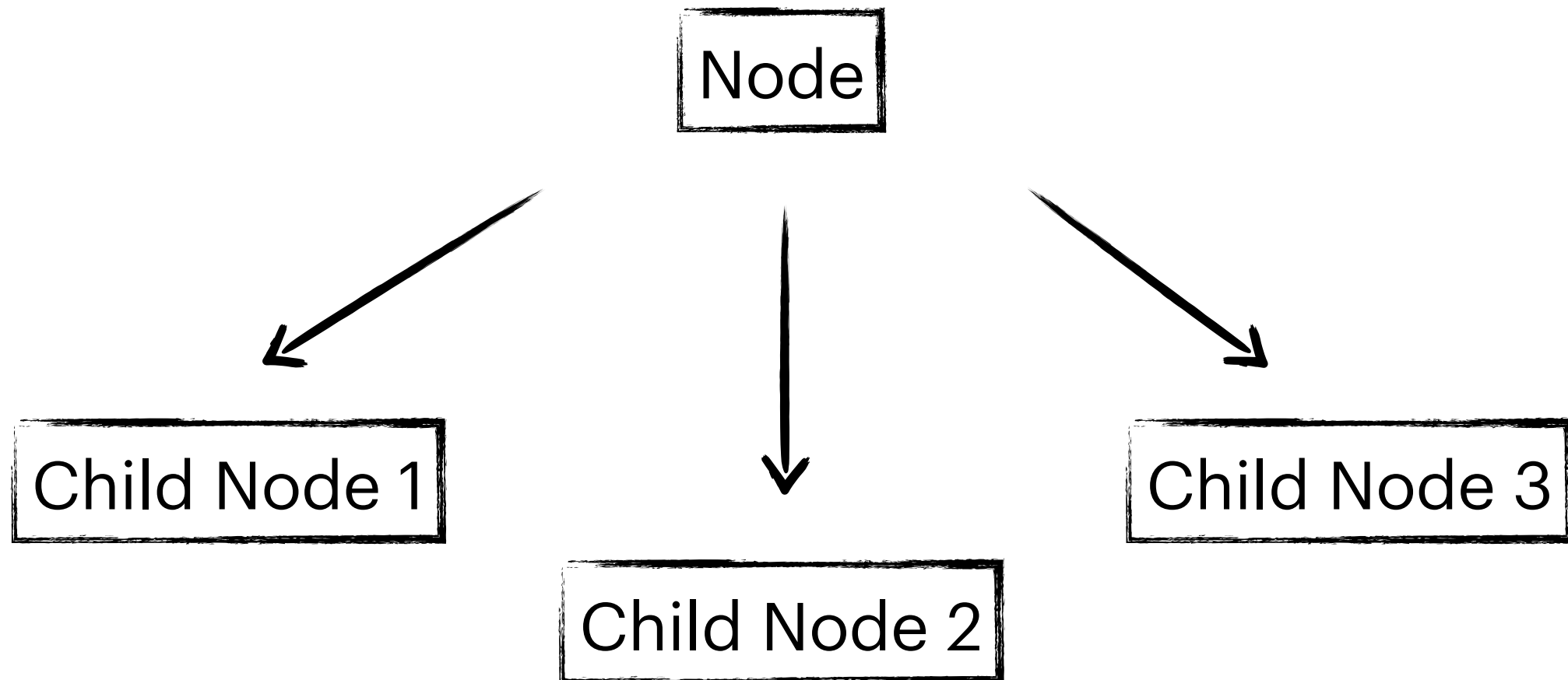
Motivation

Full waveform inversion

- Easy to run partial workflow and assemble multiple workflows into one.
- Backup and restore progress.
- Run multiple expensive functions at the same time.
- Manage directories and parameters.

Design

Inspired by HTML document

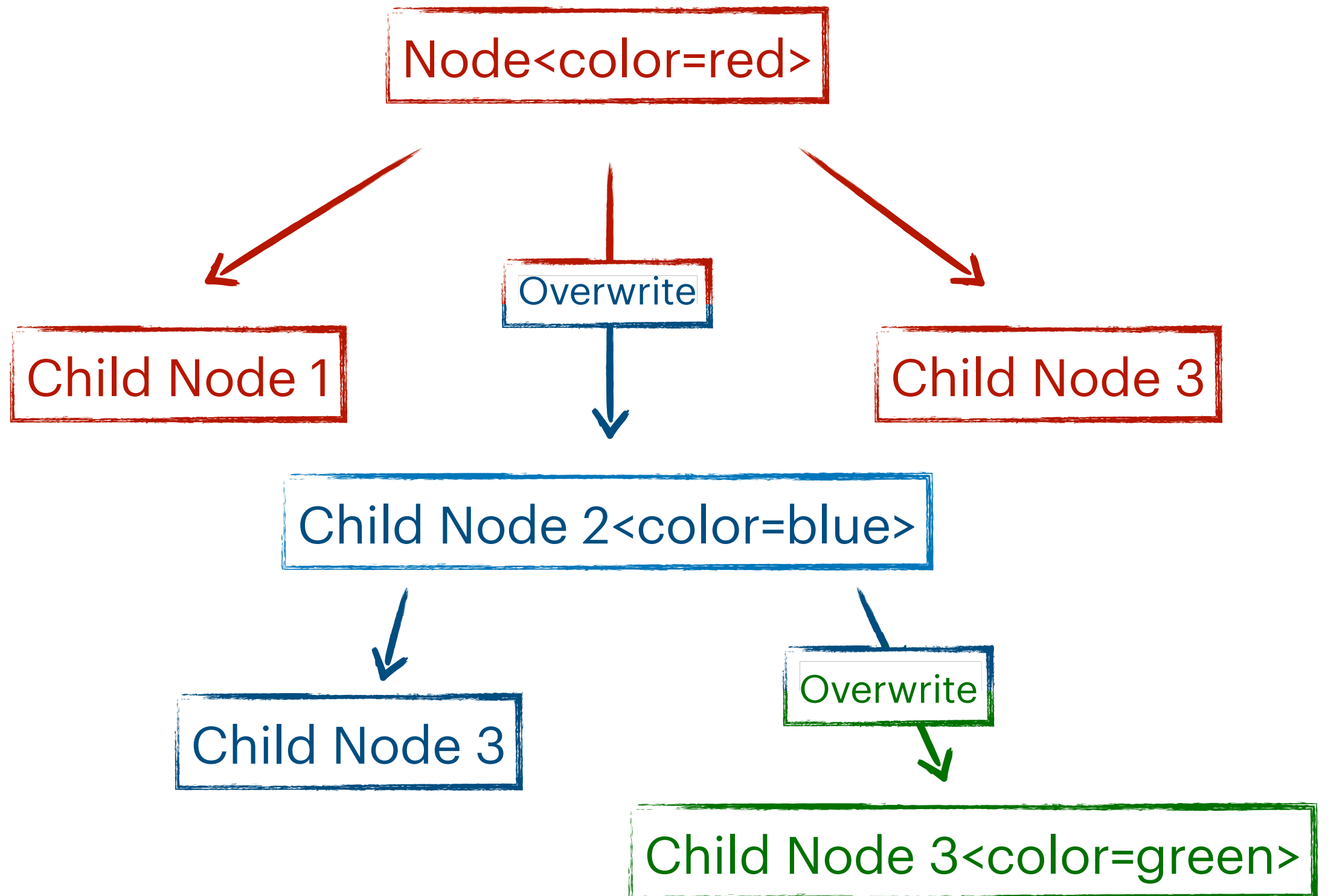


Node function -> Task function or executables in the workflow

Node style -> Task properties or directories

Design

Property inheritance

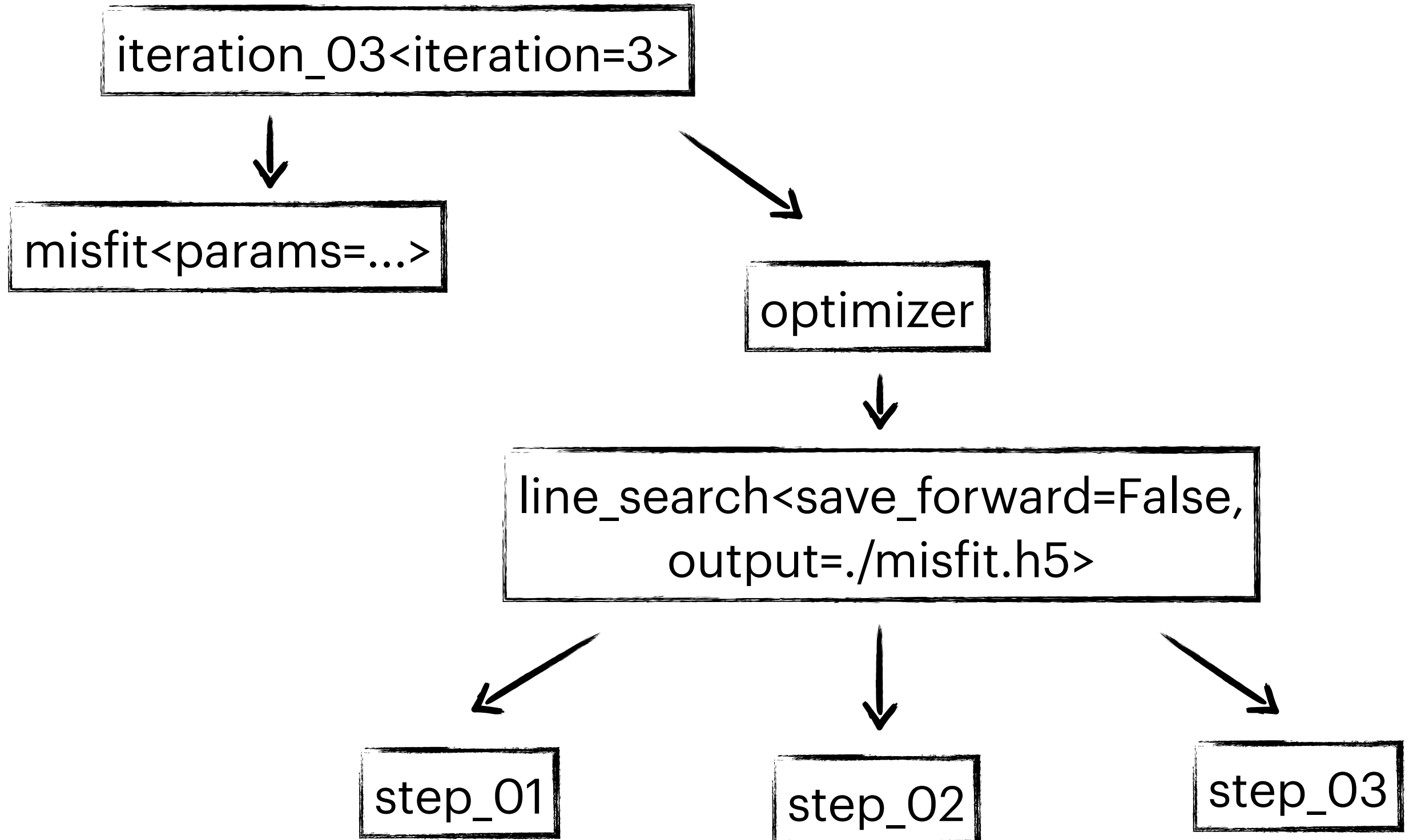


Design

- Node is the single execution unit that can contain a task and any number of child nodes.
- Each node points to a directory, which is also its child nodes' default directory.
- Tasks (e.g. `compute_kernels`, `compute_misfit`) changed from object-based to function-based.
- Properties can be propagated.

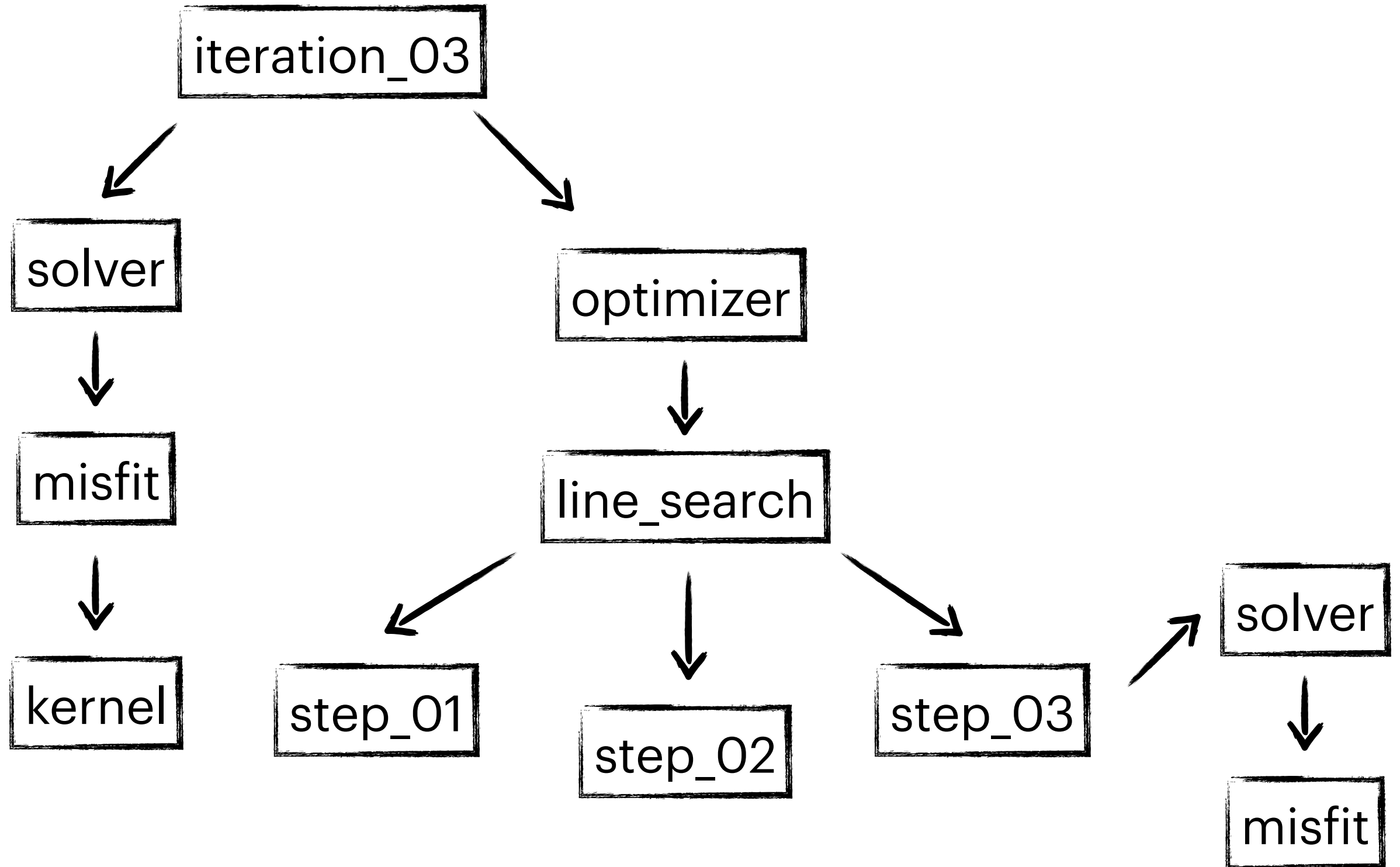
Design

Property and directory management



Design

Code example



Design

Code example

```
def main(node):  
    node.add(iterate, cwd='iter_03', iteration=3)
```

```
def iterate(node):  
    node.add(solver, cwd='solver')  
    node.add(optimizer)
```

```
def solver(node):  
    node.add(specfem)  
    node.add(misfit)
```

```
if not node.misfit_only:  
    node.add(adjoint)
```



```
def optimizer(node):  
    node.add(line_search)
```

```
def line_search(node):  
    node.add(step, step=0, cwd='step_00')  
    node.add(step, step=1, cwd='step_01')  
    node.add(step, step=2, cwd='step_02')
```

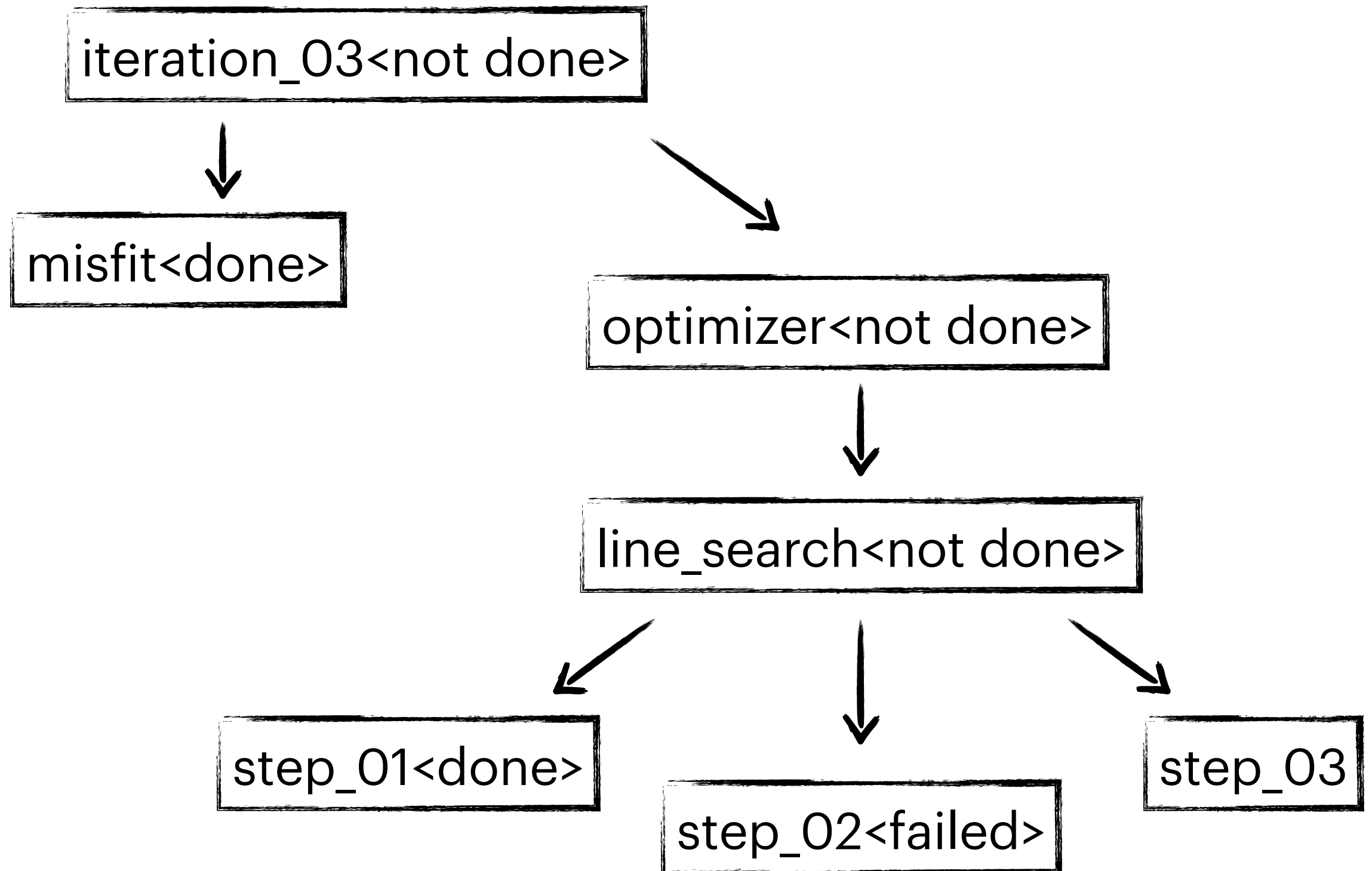
```
def step(node):  
    node.add(solver, misfit_only=True)
```

Motivation

Full waveform inversion

- Easy to run partial workflow and assemble multiple workflows into one. 
- Manage directories and parameters 
- Backup and restore progress
- Run multiple expensive functions at the same time

Backup and restore






Backup and restore

Strategy

- The workflow is saved every time a task finishes or fails
- A node is <done> if:
 1. Its task function or executable finished without error
 2. All child nodes are <done>
- When error occurs:
 - The workflow will try to resubmit if any task fails
 - The workflow will abort if any task fails twice in a row.

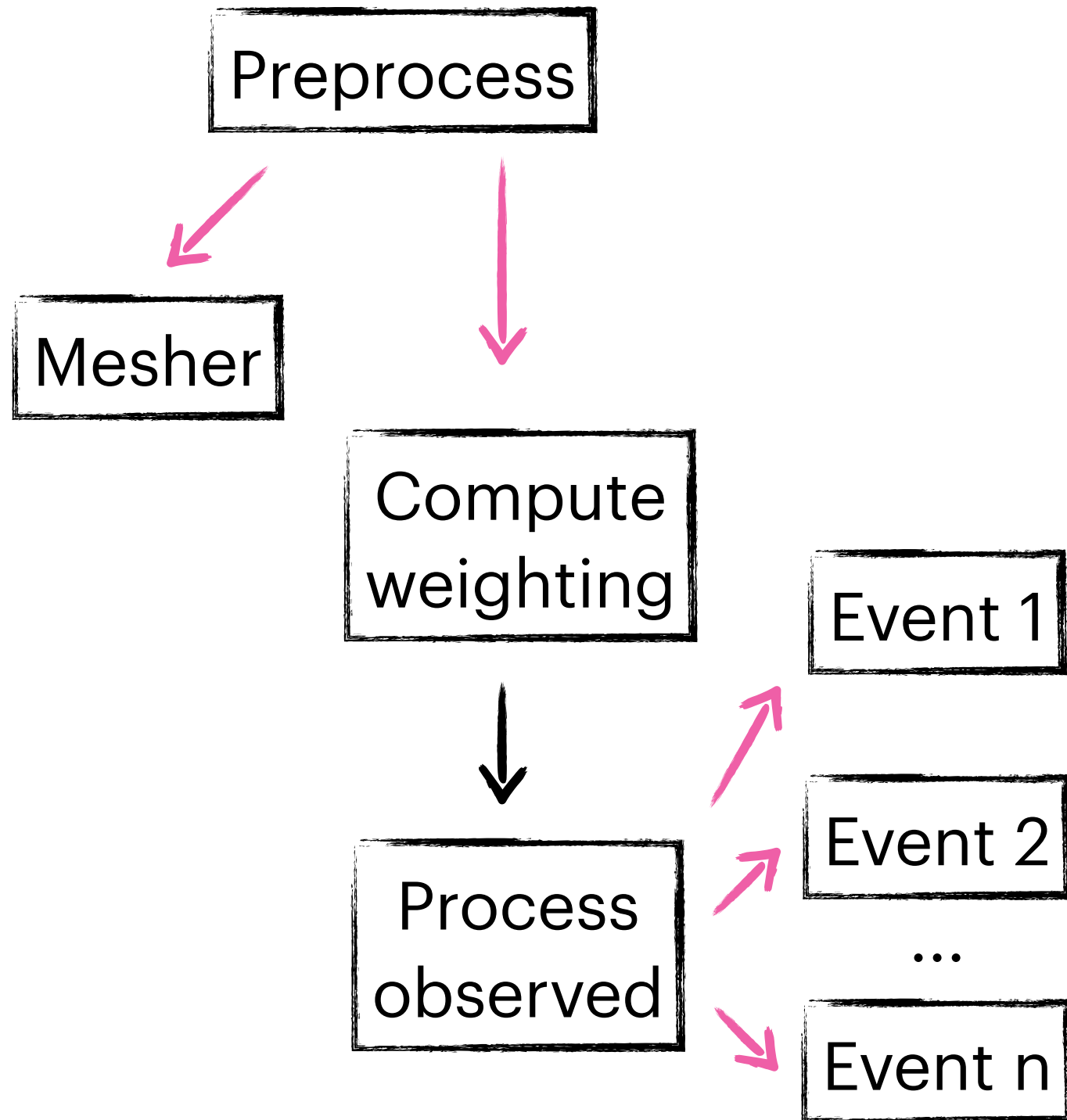
Motivation

Full waveform inversion

- Define a workflow that includes all tasks 
- Manage directories and parameters 
- Backup and restore progress 
- Run multiple expensive functions at the same time

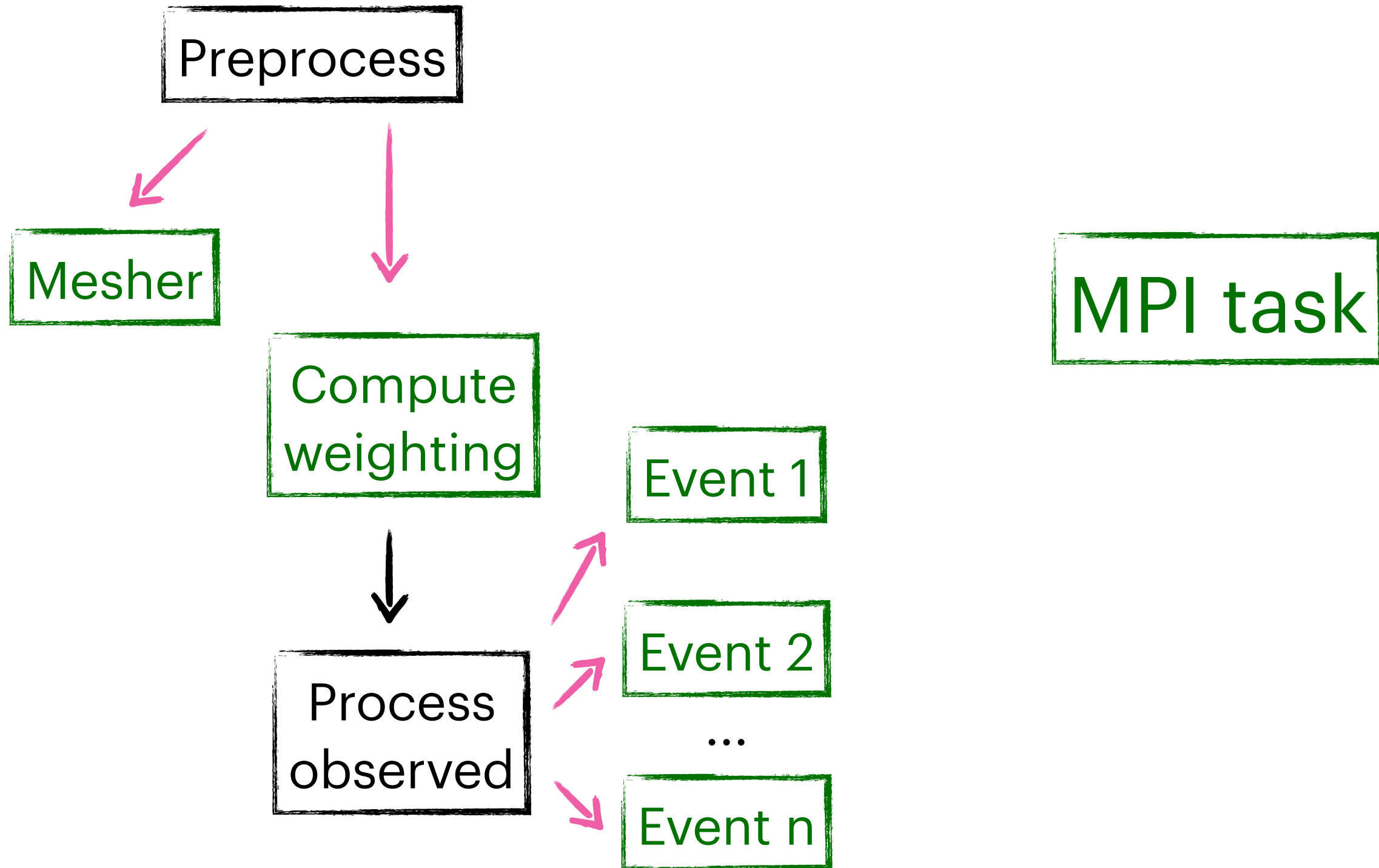
Task management

Manage multiple MPI tasks



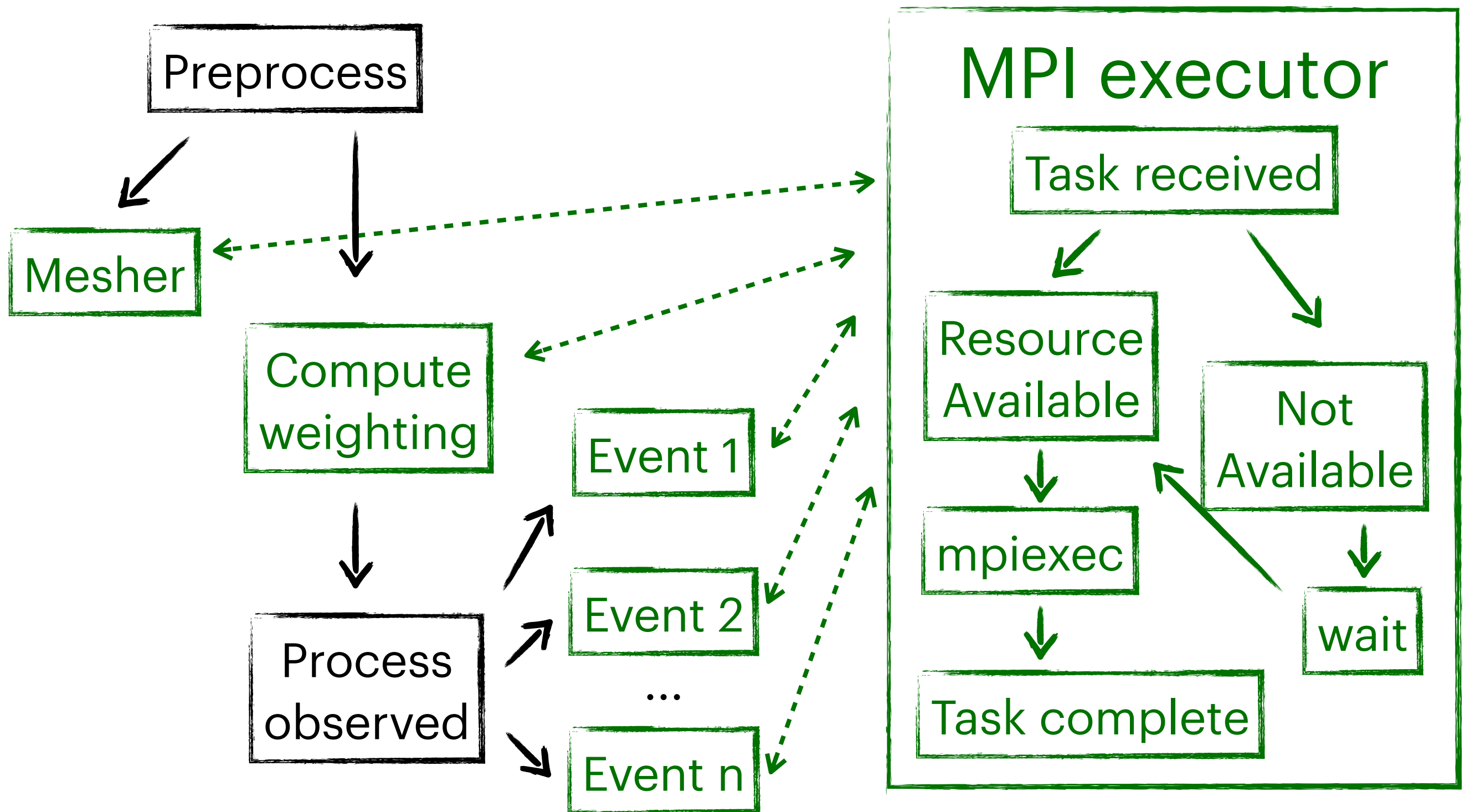
Task management

Manage multiple MPI tasks



Task management

Manage multiple MPI tasks



Task management

Concurrency

- Main process is single-threaded
 - guaranteed to behave exactly the same on any CPU architecture
 - Unlimited number of concurrent process
- The main process will progress if a task is marked as pending, running or complete.

Task management

Concurrency

```
import asyncio

async def task1():
    await asyncio.sleep(2)

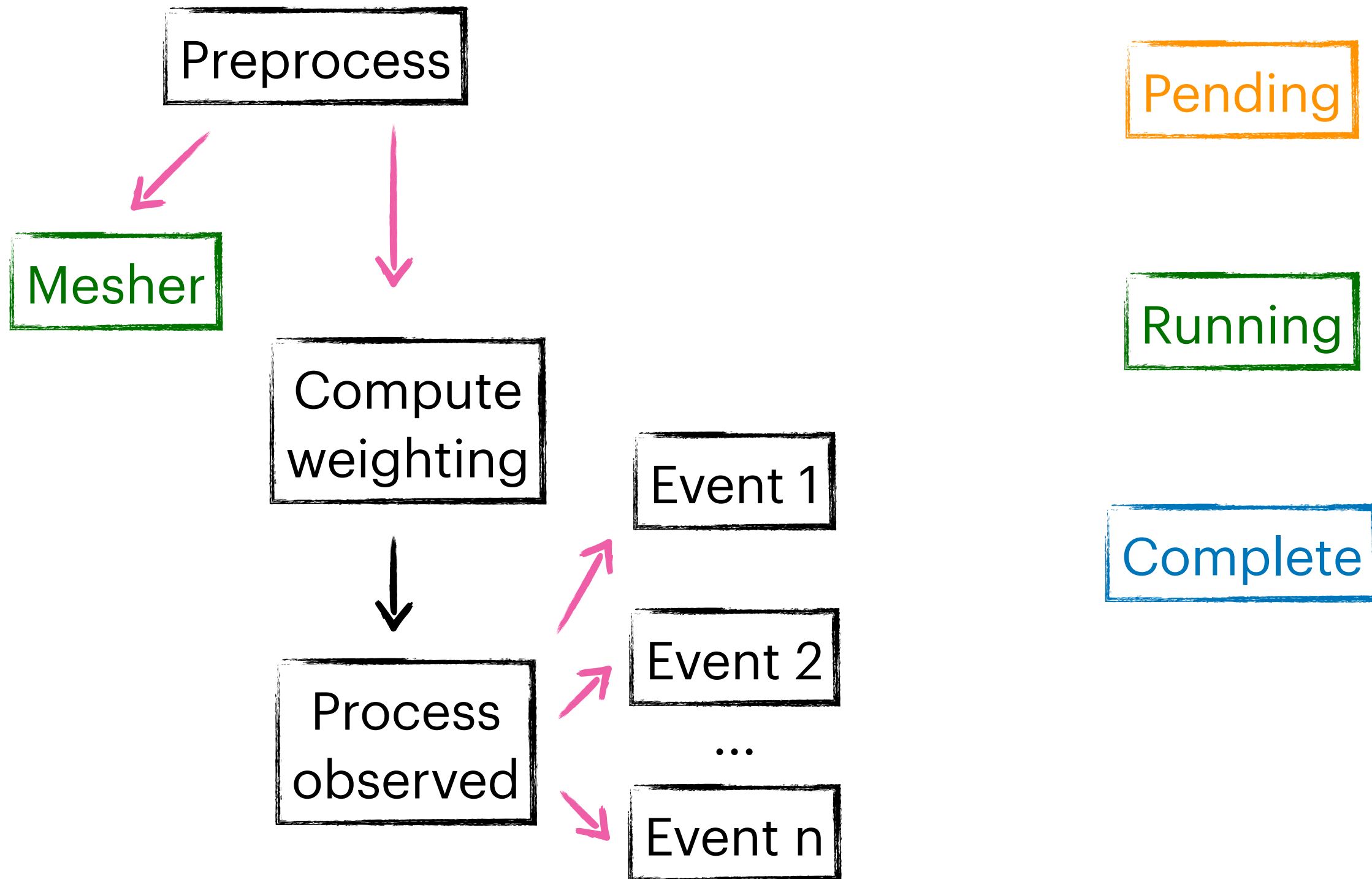
async def task2():
    await asyncio.sleep(2)

async def main():
    await asyncio.gather(task1(), task2())

asyncio.run(main()) # takes 2 seconds
```

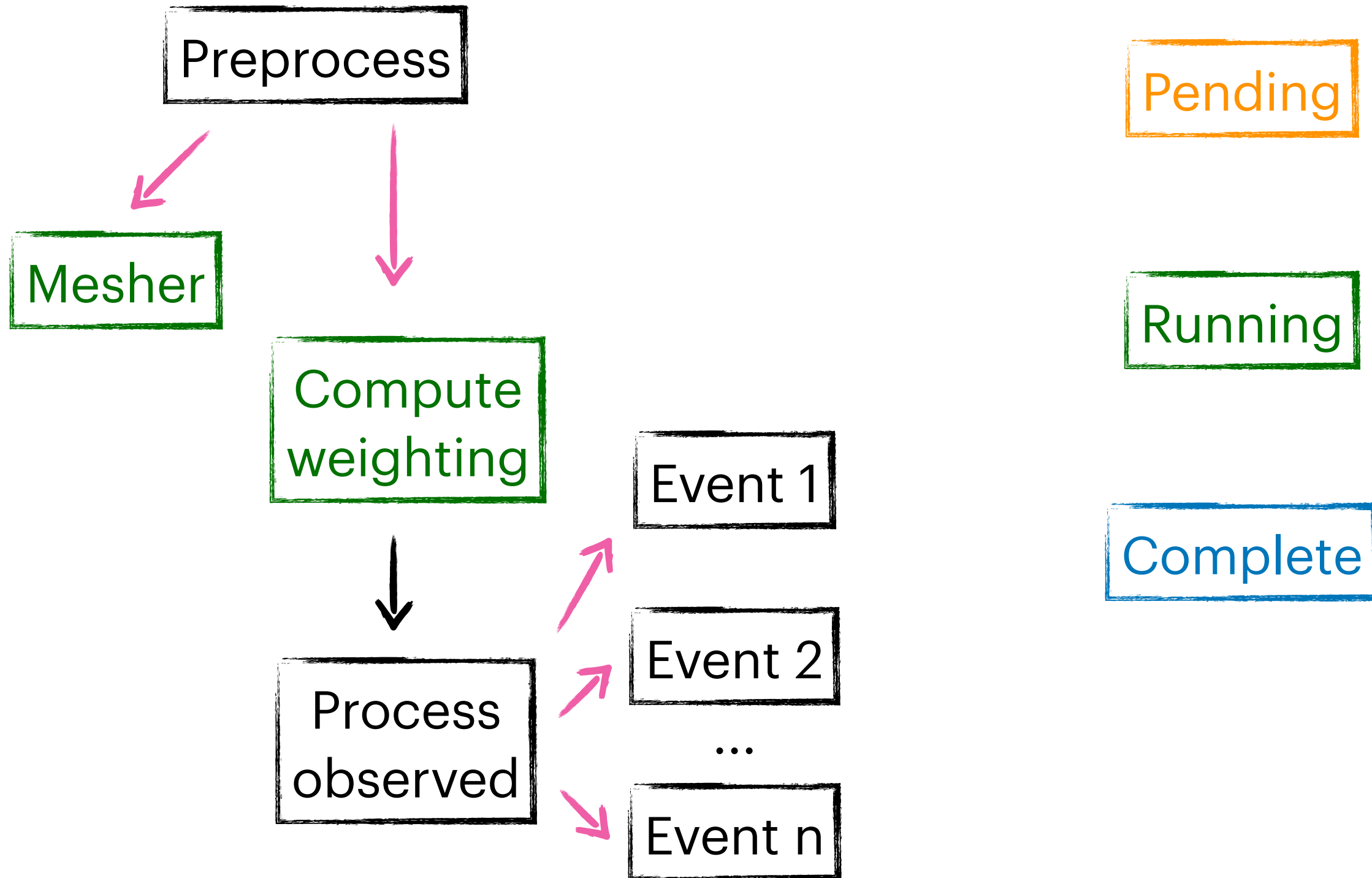
Task management

Manage multiple MPI tasks



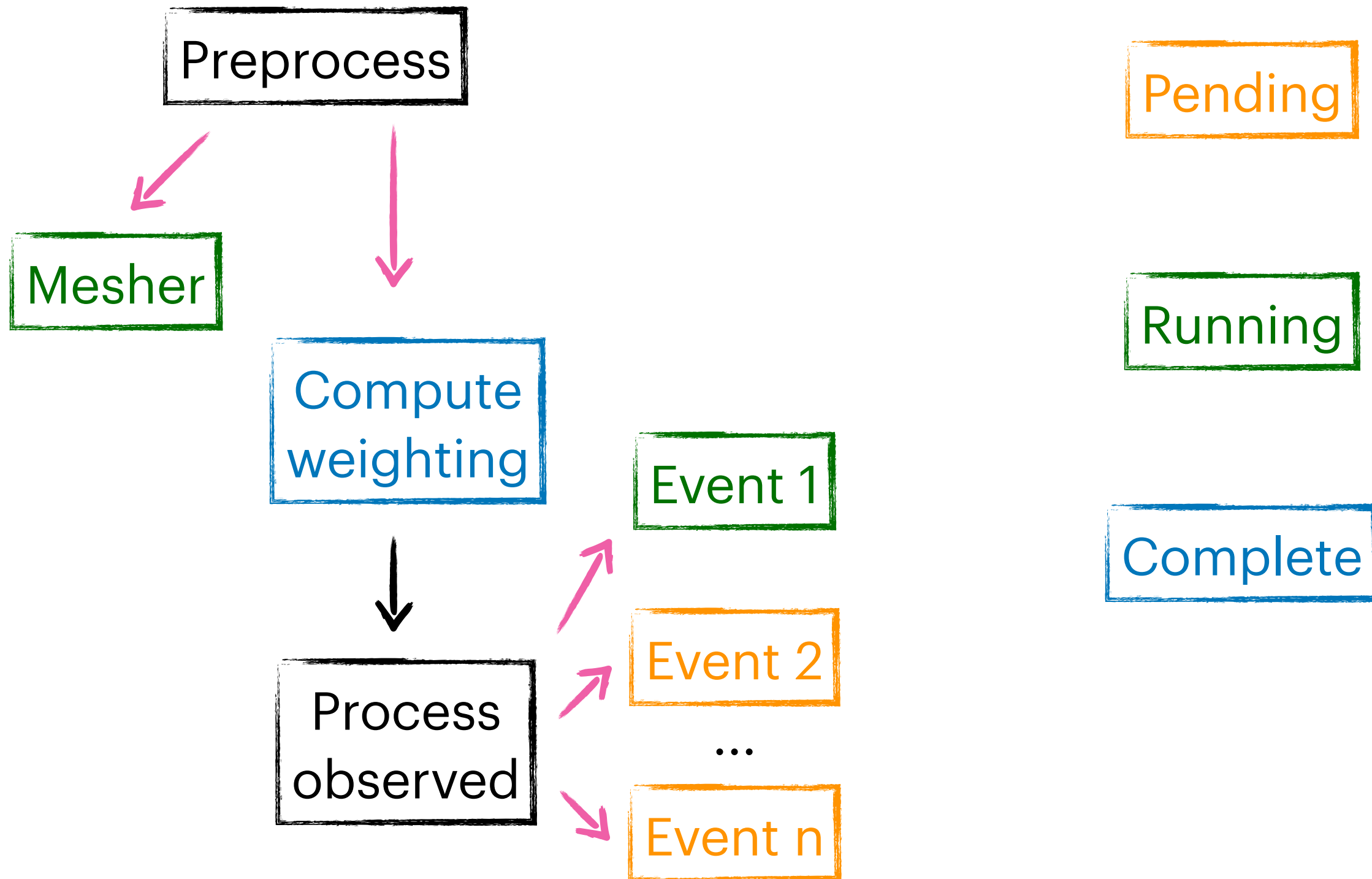
Task management

Manage multiple MPI tasks



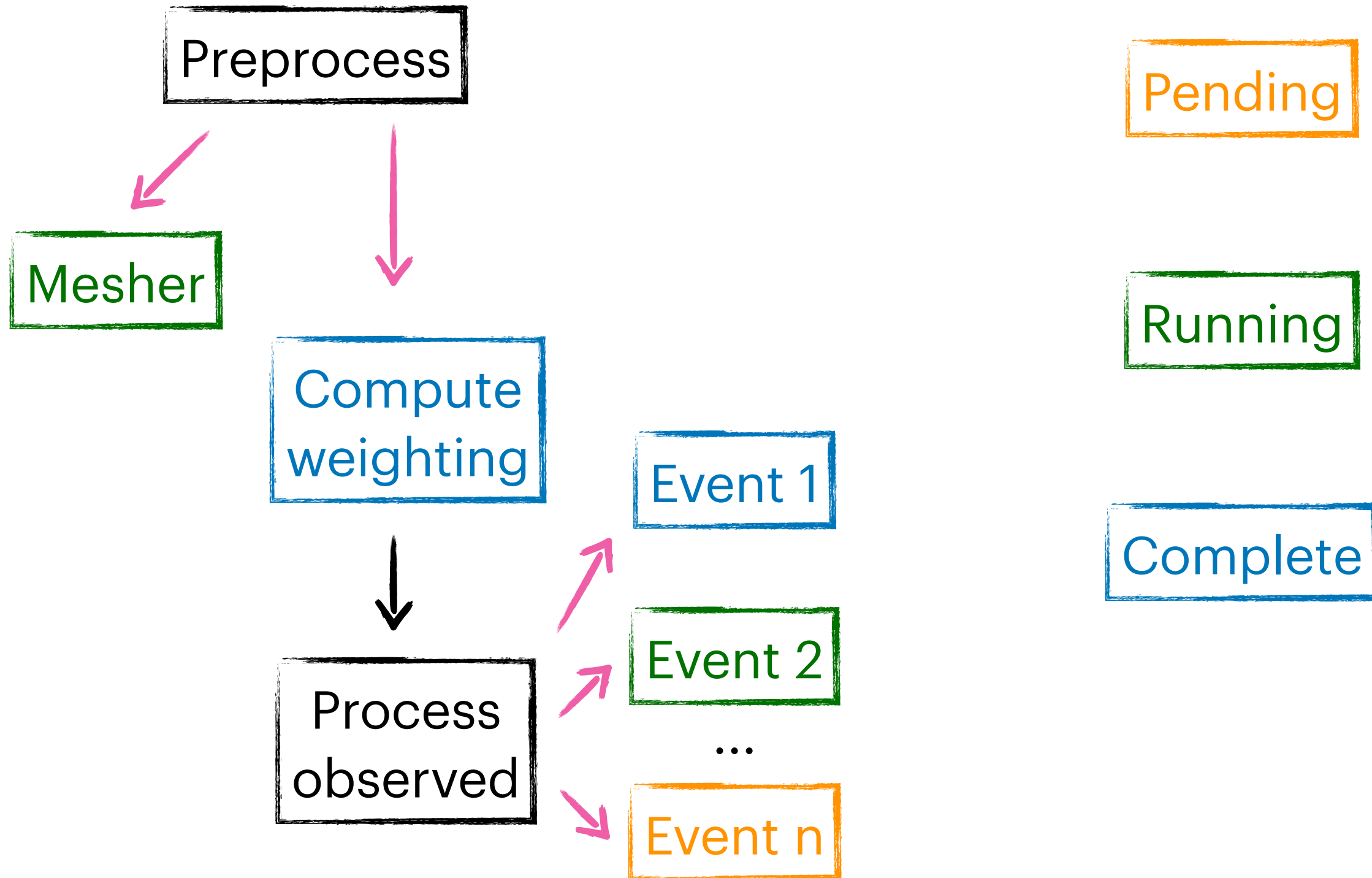
Task management

Manage multiple MPI tasks



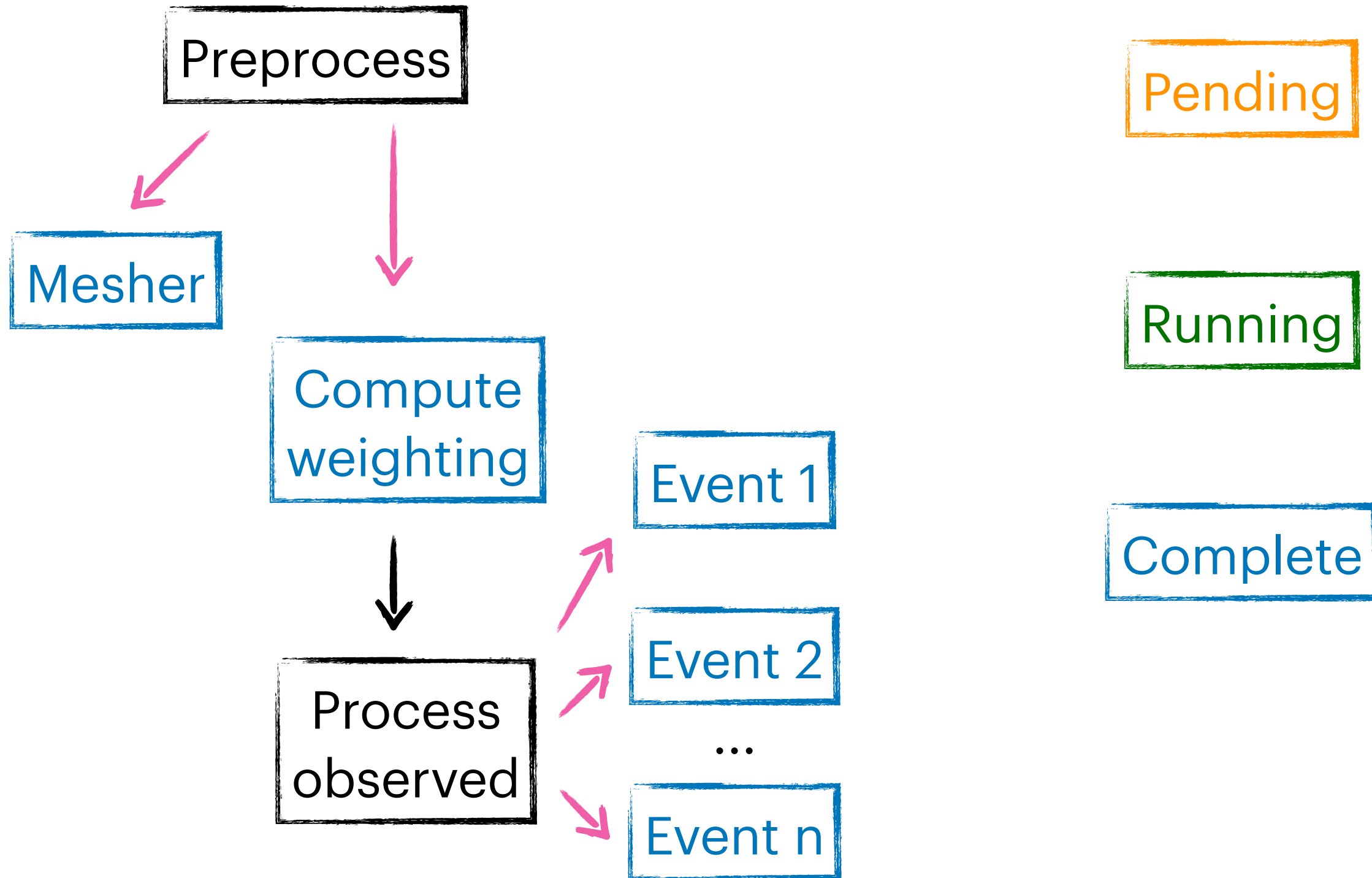
Task management

Manage multiple MPI tasks







Task management

Manage multiple MPI tasks



Motivation

Full waveform inversion

- Define a workflow that includes all tasks 
- Manage directories and parameters 
- Backup and restore progress 
- Run multiple expensive functions at the same time 

100%