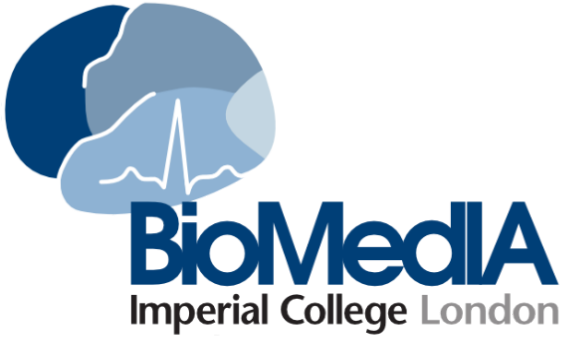
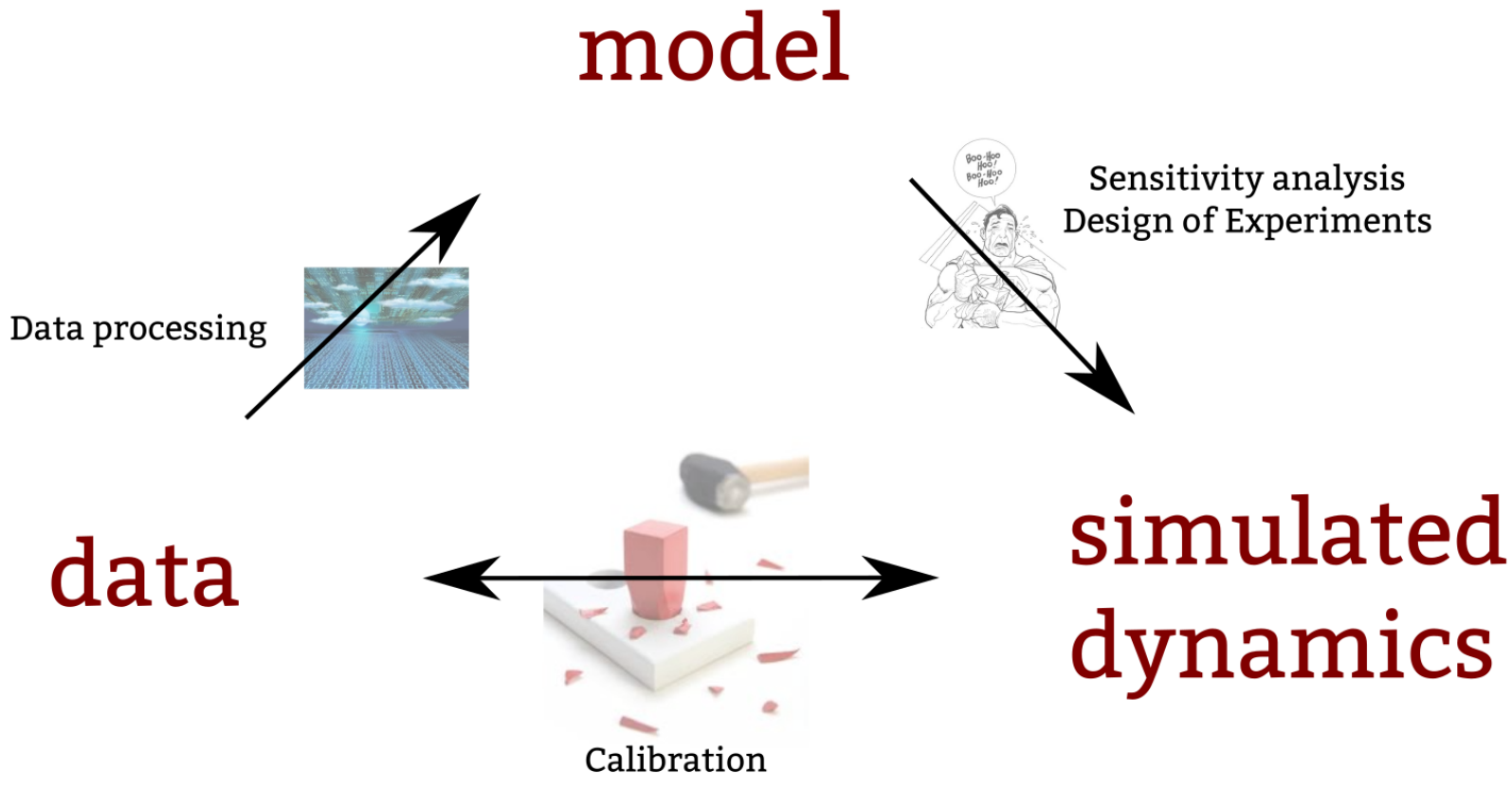
OPEN MODEL EXPERIMENT

Mathieu Leclaire
Romain Reuillon
Jonathan Passerat-Palmbach



European Research Council
Established by the European Commission

developing Human
Connectome Project



Model construction steps

Many scientists use **naturally parallel methods** daily:

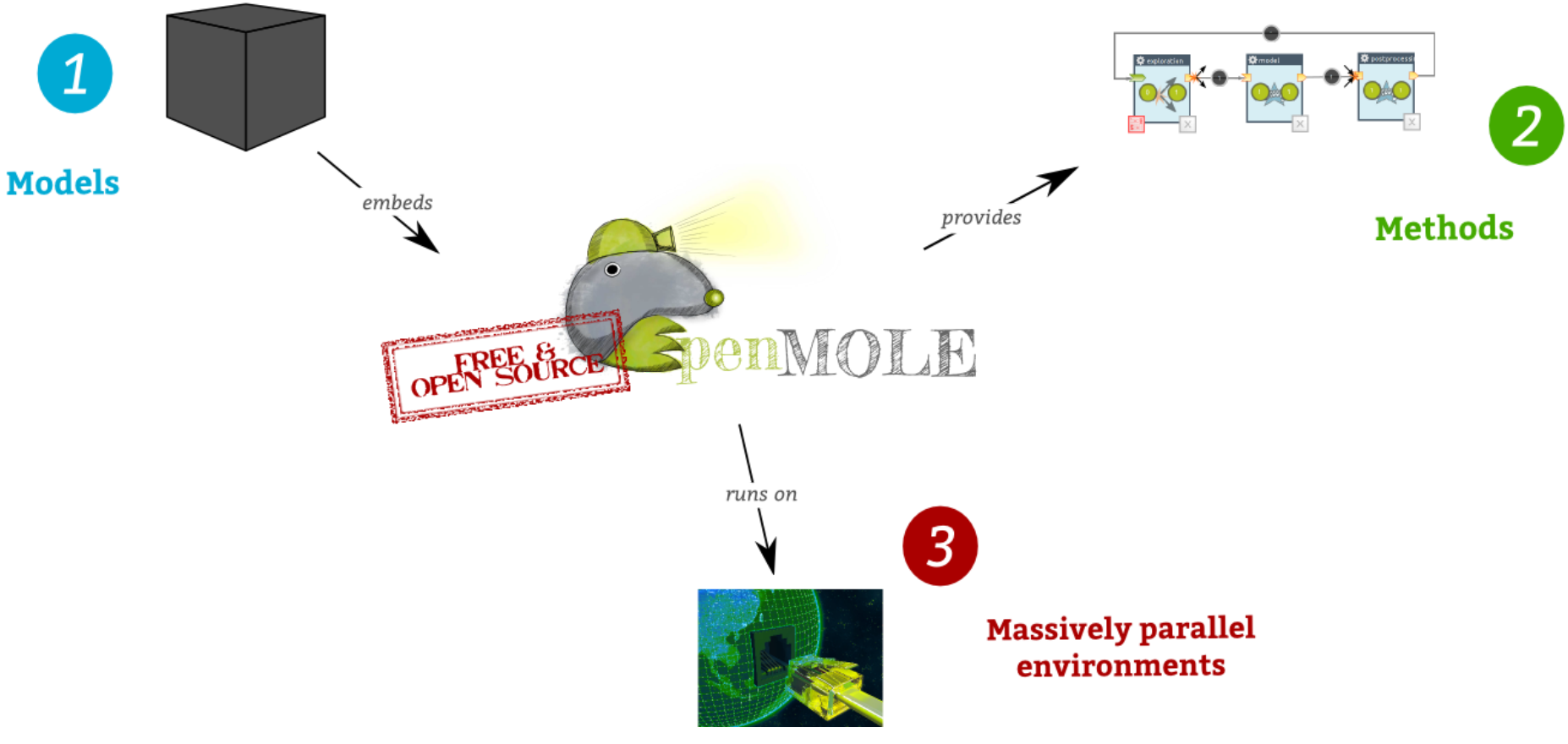
- Data reconstruction
- Parameter estimation
- Sensitivity analysis
- Optimisation
- Replication
- ...

Execution on the **same program** with **different parameters** and/or **datasets**.

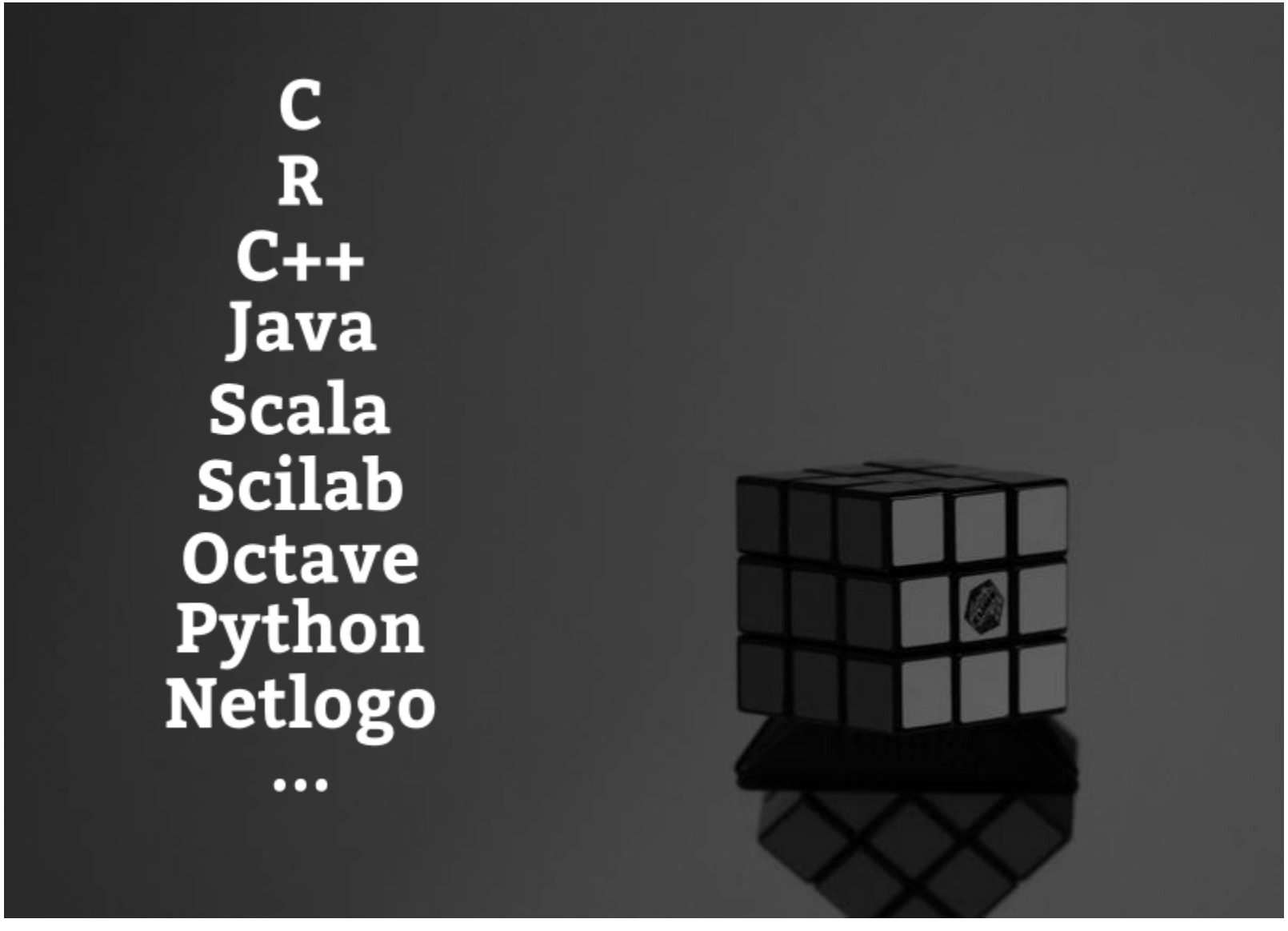
These methods are **time consuming**, but generally executed in a sequential manner.

OpenMOLE fills the gap. It provides an easy way to **describe** and **distribute** naturally parallel processes.

OPENMOLE IN 3 STEPS



EMBED YOUR MODEL AS A BLACK BOX



ZERO DEPLOYMENT APPROACH

- The code is the user's code, not a web service
- User code is automatically deployed at runtime
- Ships to remote environment
- No prior knowledge of remote environment needed
- No installation required on any machine

PORT (ALMOST) ANY PROGRAM TO THE GRID IN 3 SIMPLE STEPS

- Archive it with CARE* <=> execute it on linux
- Write your OpenMOLE workflow
- Click the run button

* http://reproducible.io

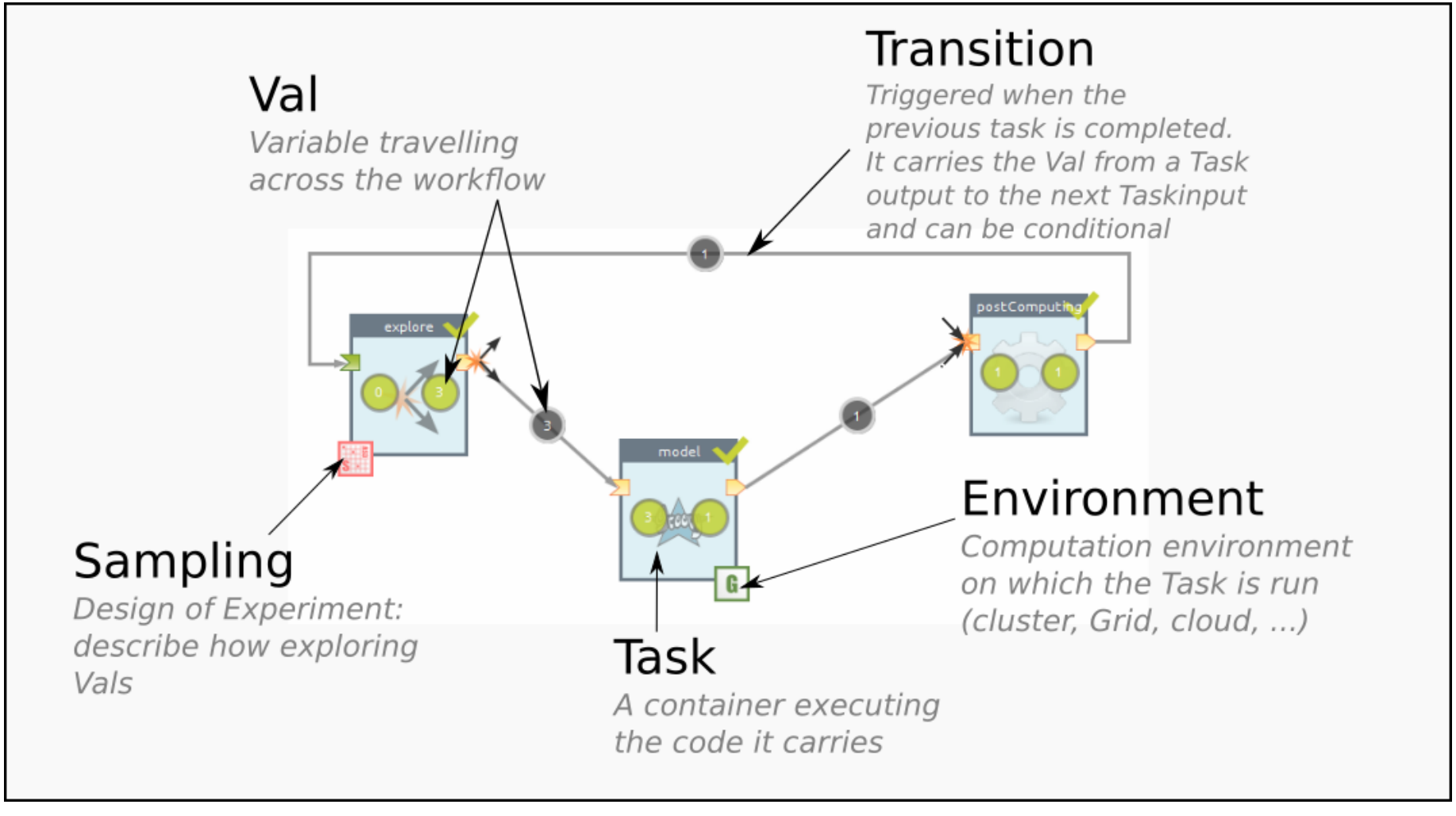
PACKAGING AN APPLICATION WITH CARE

Applications have dependencies:

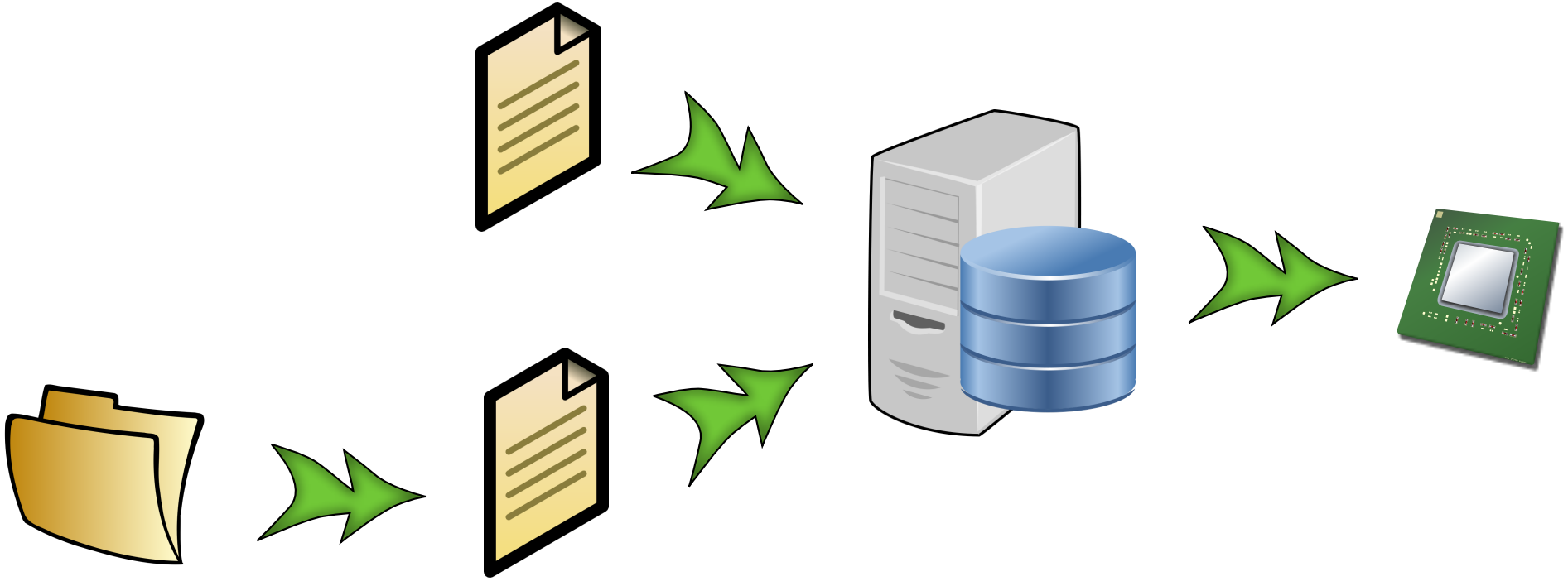
- Shared libraries
- Packages (Python, R, ...)
- Low level system calls
- Environment variables
- ...

Capture these dependencies and **transfer** along with the application from Linux to Linux

NATURALLY PARALLEL FORMALISM TO DESIGN EXPERIMENTS: A WORKFLOW

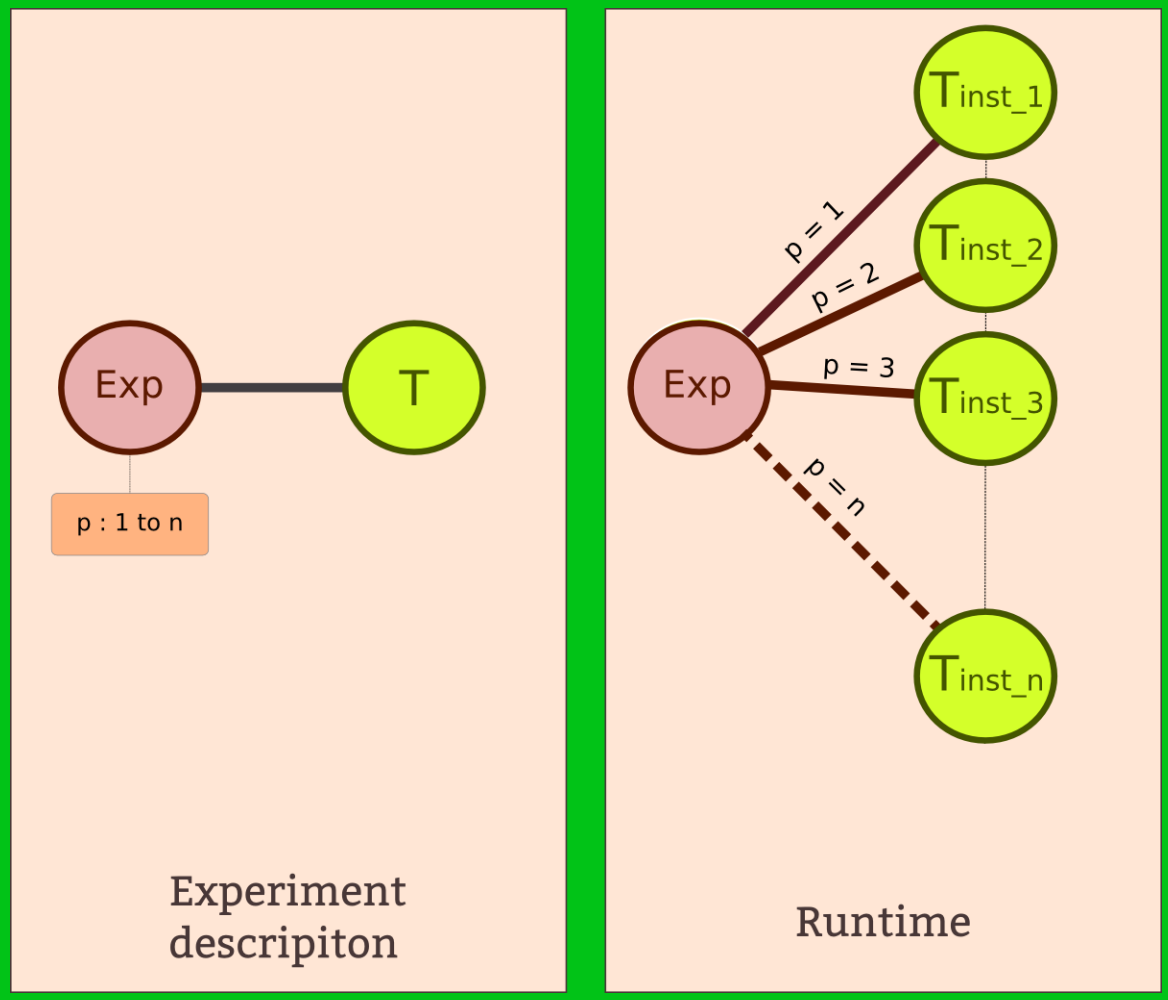


AUTOMATIC DATA TRANSFERS AND REPLICA MANAGEMENT



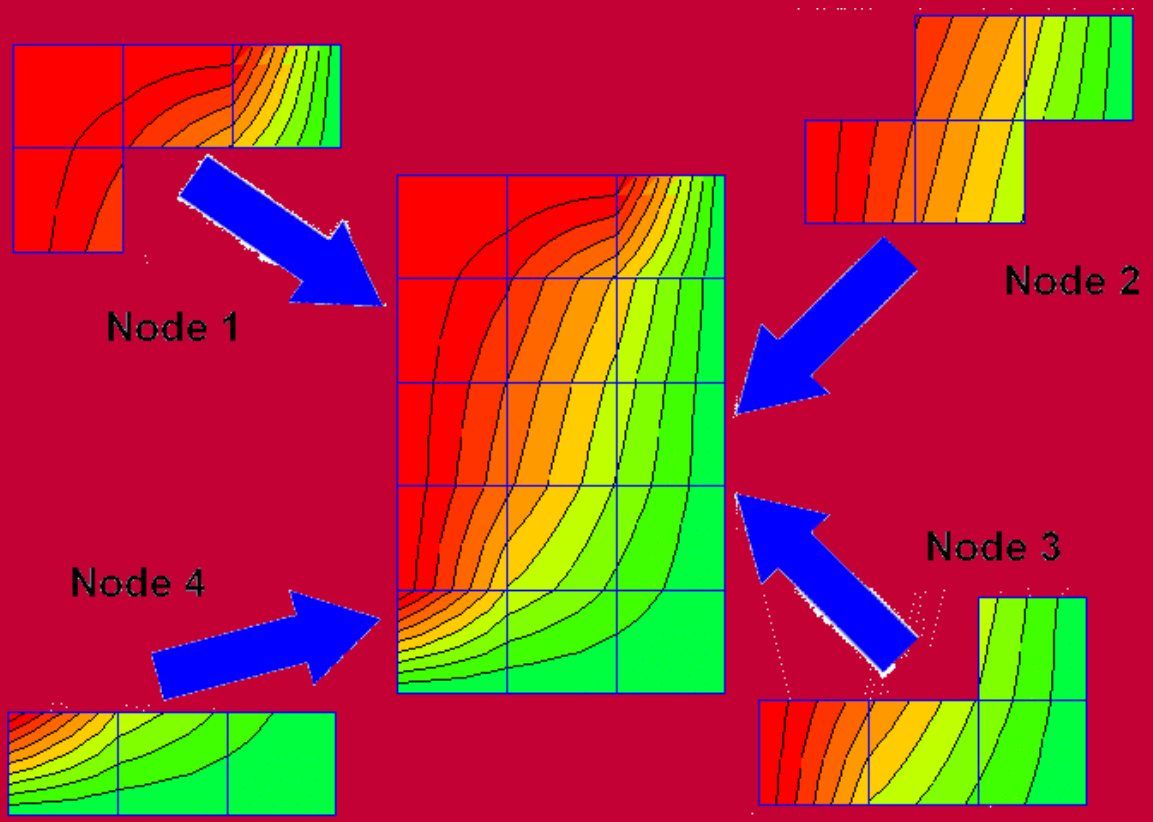
Files and folders transfers are **handled transparently** by OpenMOLE

WHAT OPENMOLE DOES



Data parallelism

WHAT OPENMOLE DOES NOT

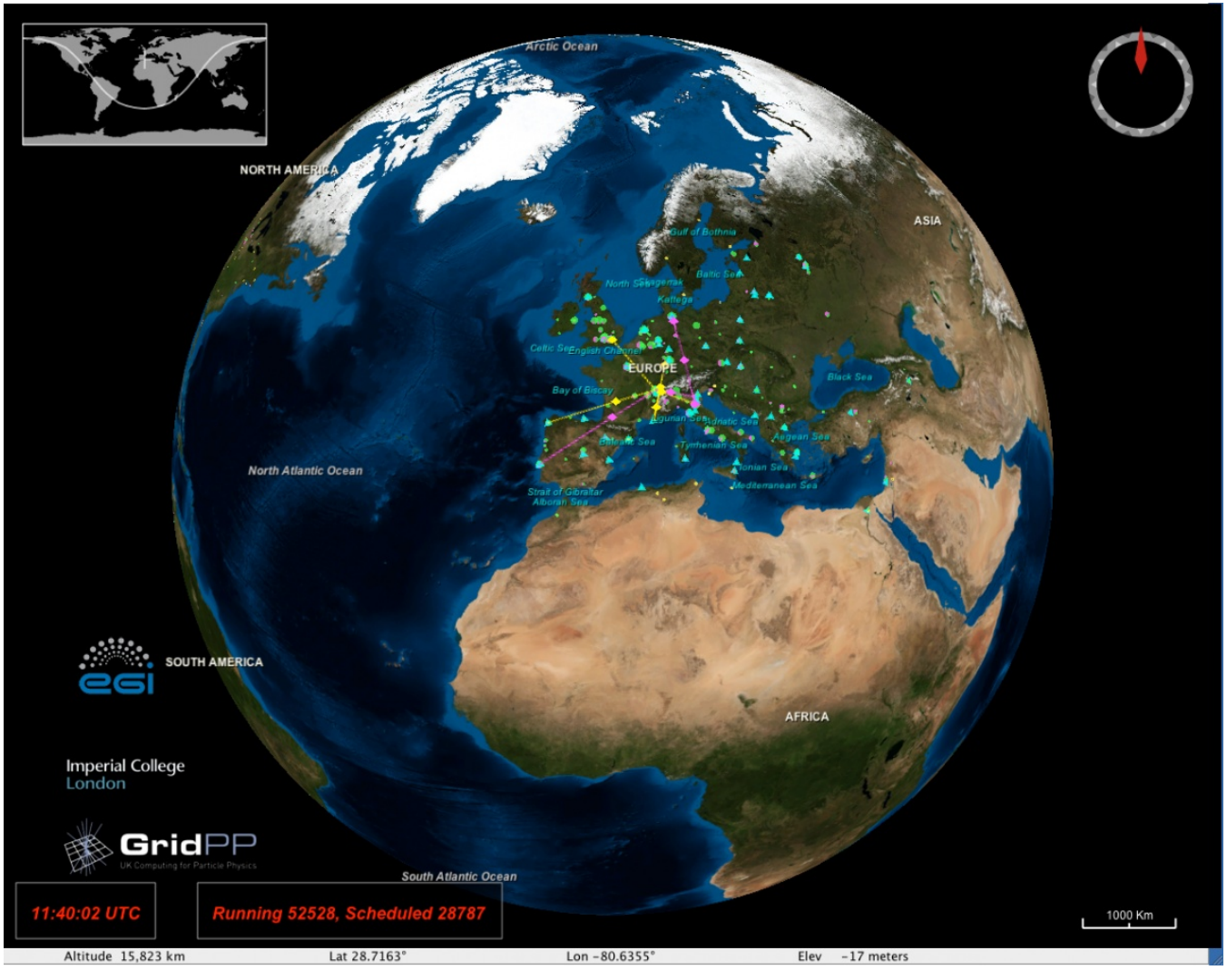


Parallelisation by message / Task Parallelism

Example: Spark, MPI, ...

POWERED BY EGI

(AND OTHERS)



A WORKFLOW

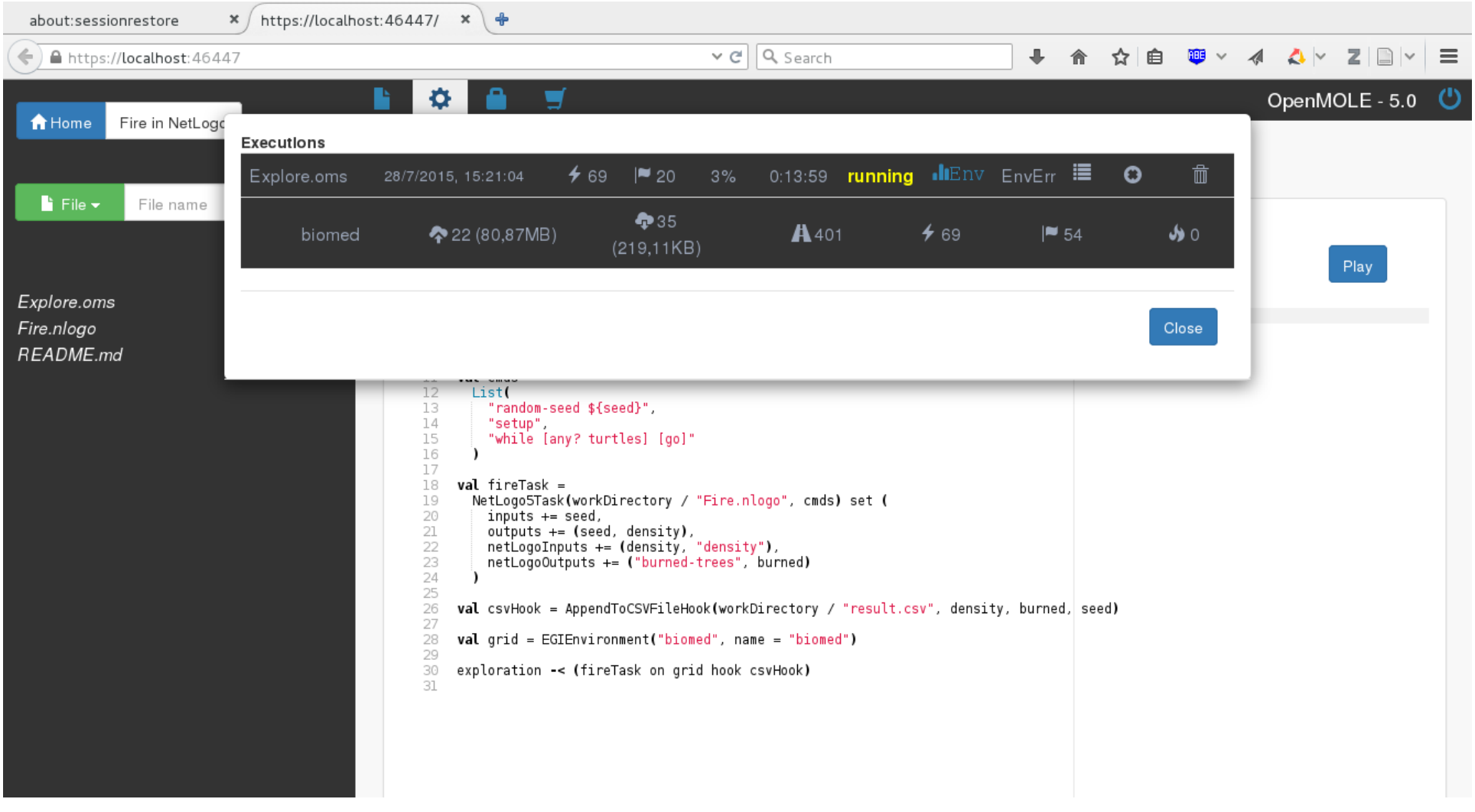
```
val i   = Val[Double]
val res = Val[Double]

val exploration = ExplorationTask ( i in (0.0 to 10.0 by 1.0) )

val model =
  ScalaTask ("val res = i * 2") set (
    inputs  += i,
    outputs += (i, res)
  )

val env = LocalEnvironment(5)

val ex = exploration -< (model on env) start
```

THE SAME WORKFLOW ON THE GRID !

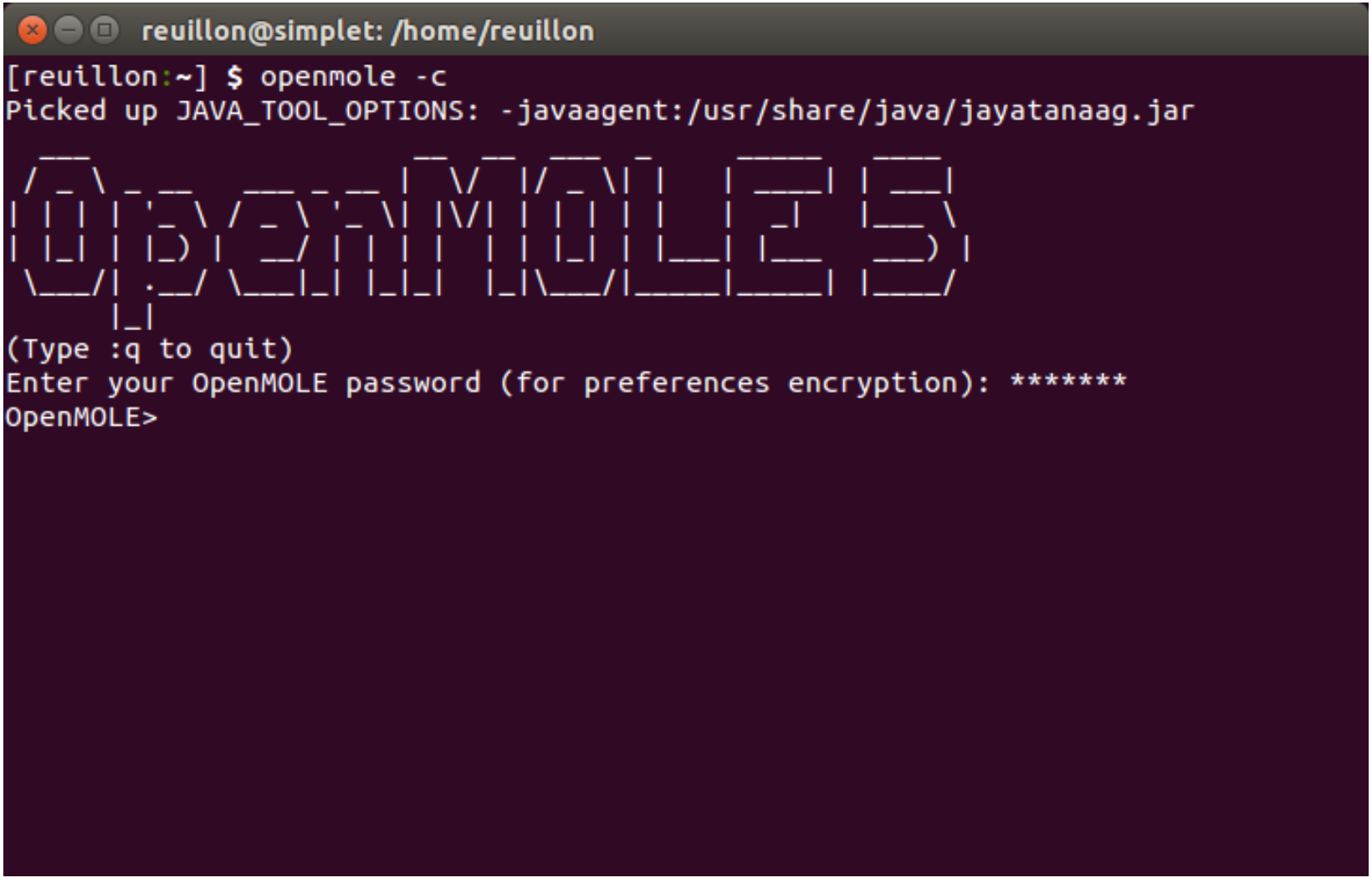
```
val i   = Val[Double]
val res = Val[Double]

val exploration = ExplorationTask ( i in (0.0 to 10.0 by 1.0) )

val model =
  ScalaTask ("val res = i * 2") set (
    inputs  += i,
    outputs += (i, res)
  )

val env = EGIEnvironment("biomed")

val ex = exploration -< (model on env) start
```

Switching to the **Grid Environment** is so easy !

WEB APPLICATION

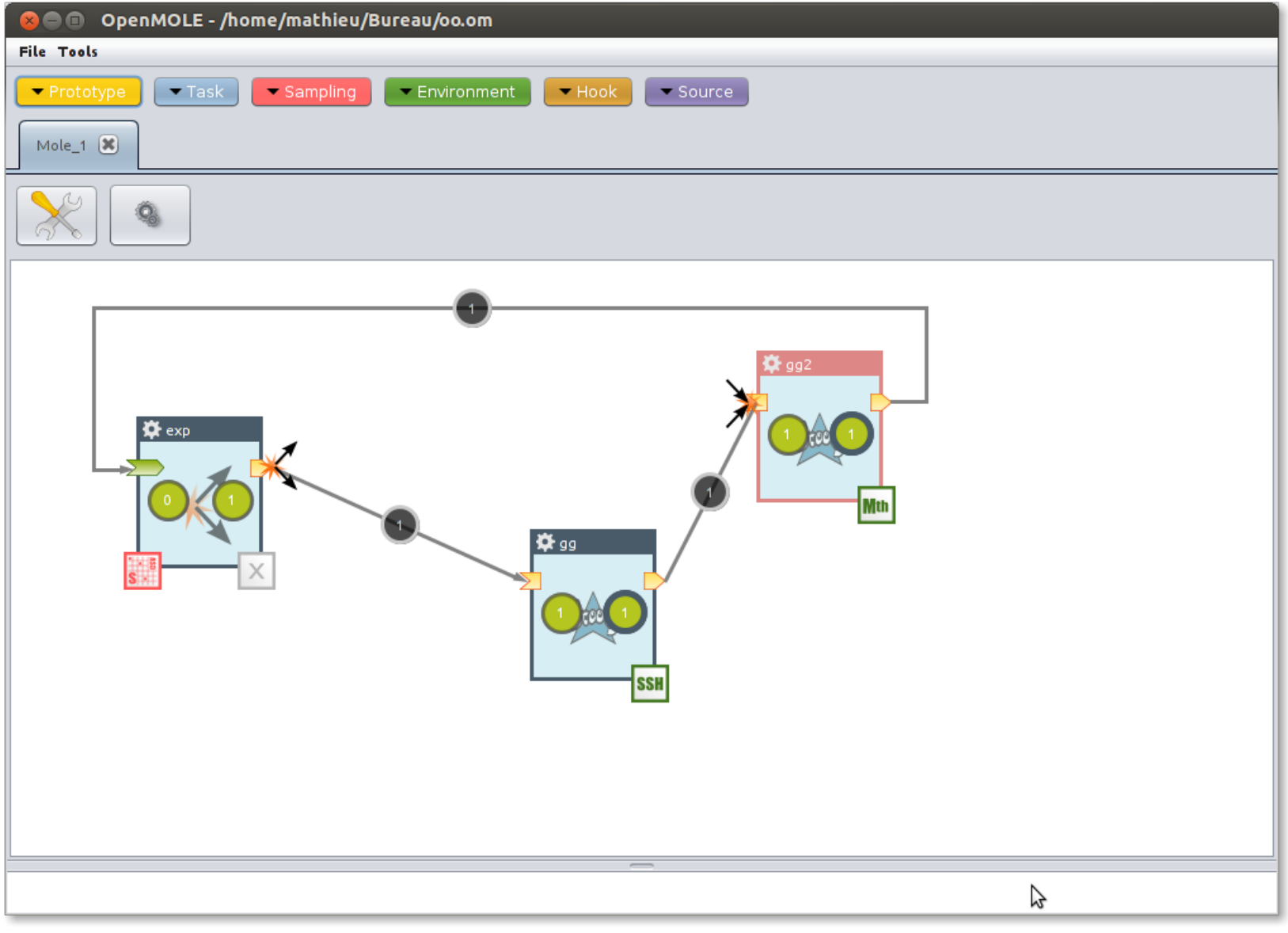


CONSOLE INTERFACE



TOWARDS A MULTI-USER WEB PLATFORM

TOWARDS GRAPHICAL WORKFLOWS



MARKET PLACE

- Full workflows describing various experiments from **real cases**.
- It containts scripts and all the required resources. Just **download** and **run** it!
- **Community**: Users can **upload** their own workflow through **github**

MARKET PLACE

Market place

Filter

Pi Computation		Stochastic Simulation
Random Forest		Stochastic Machine Learning Native Code Data Python
Hello World in R		R Data Native Code
Fire in NetLogo		NetLogo Stochastic Simulation
Hello World in Java		Java
Calibration of Ants	Download	NetLogo Genetic Algorithm Simulation Calibration

Calibration of a NetLogo model

This market entry calibrates the Ants NetLogo model with an Evolutionary/Genetic Algorithm (EA/GA) in OpenMOLE. It proposes 2 aproach, the first on uses a generationnal NSGA2 which is very classical. The second one is better fit for distributed computing and uses an island model distribution strategy.

Hello with OpenMOLE plugin	Scala Java Plugin
SimpopLocal	Stochastic Simulation Genetic Algorithm Scala Calibration
Metamimetic Networks	Stochastic Simulation NetLogo

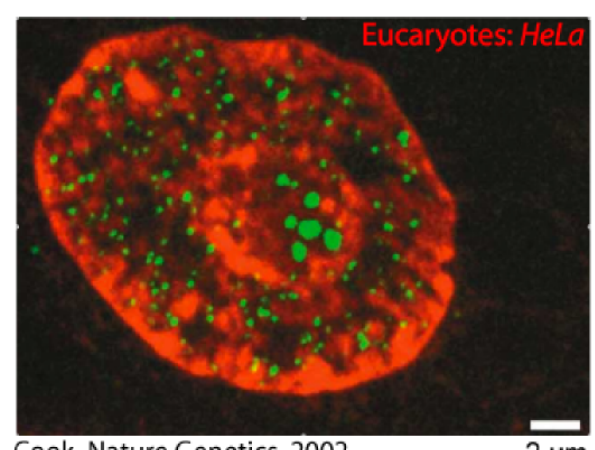
Close

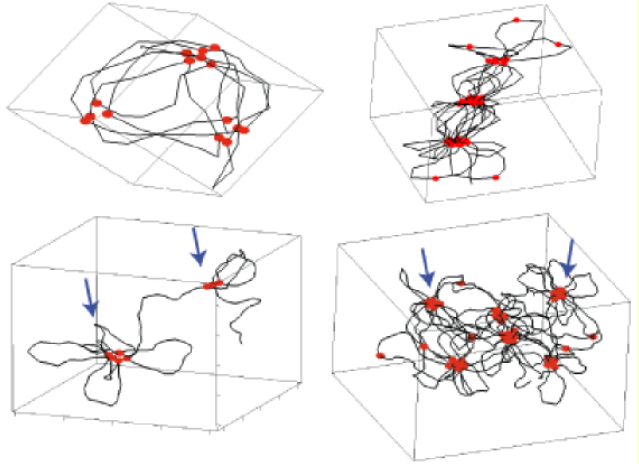
OPENMOLE THROUGH 4 EXAMPLES

OpenMOLE is neither dedicated to a scientific field nor to a language

- Chromosome structuring: Neuro Sciences, C++
- The SimTRAP project: Social Sciences, Netlogo
- The SimPOP project: Geography, Scala
- The BioEmergence project: Biology, C

CHROMOSOME STRUCTURING

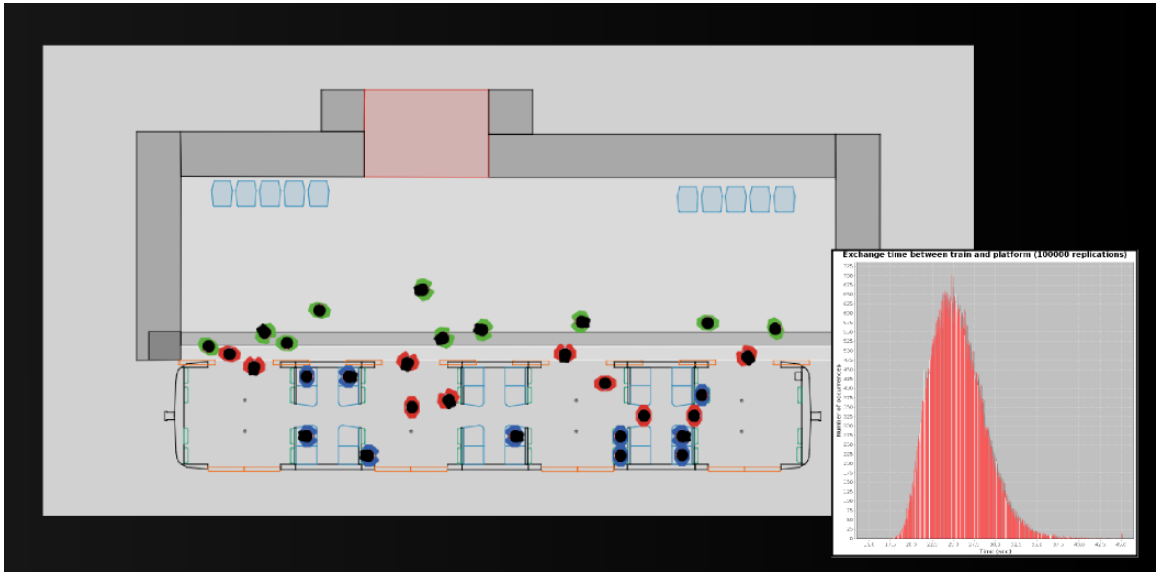




Chromosome structuring

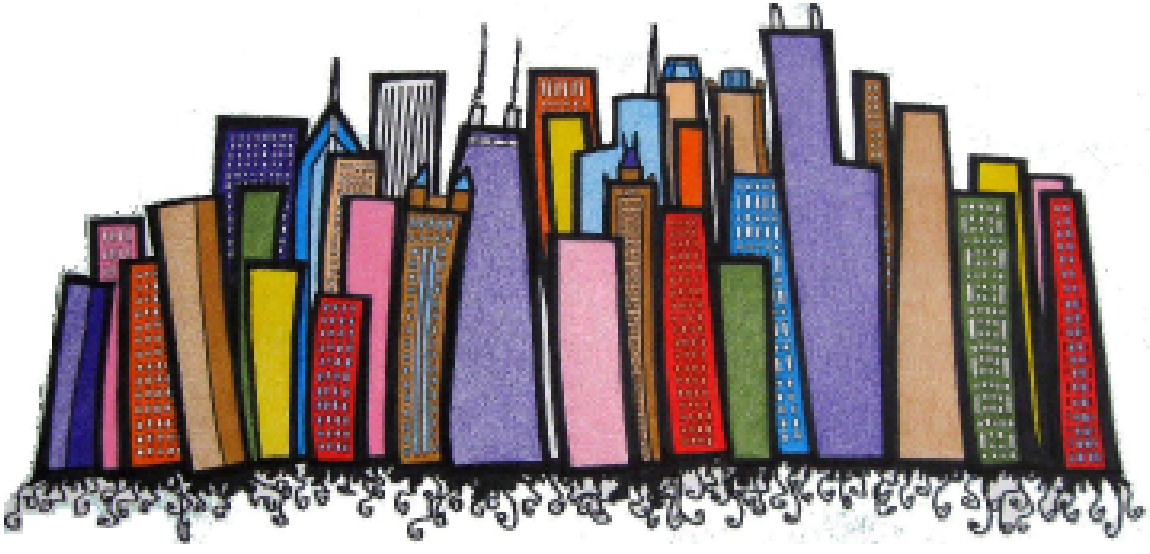
C++ model
2 days per simulation for 1,600 simulations
= **8.5 years CPU time**

THE SIMTRAP PROJECT



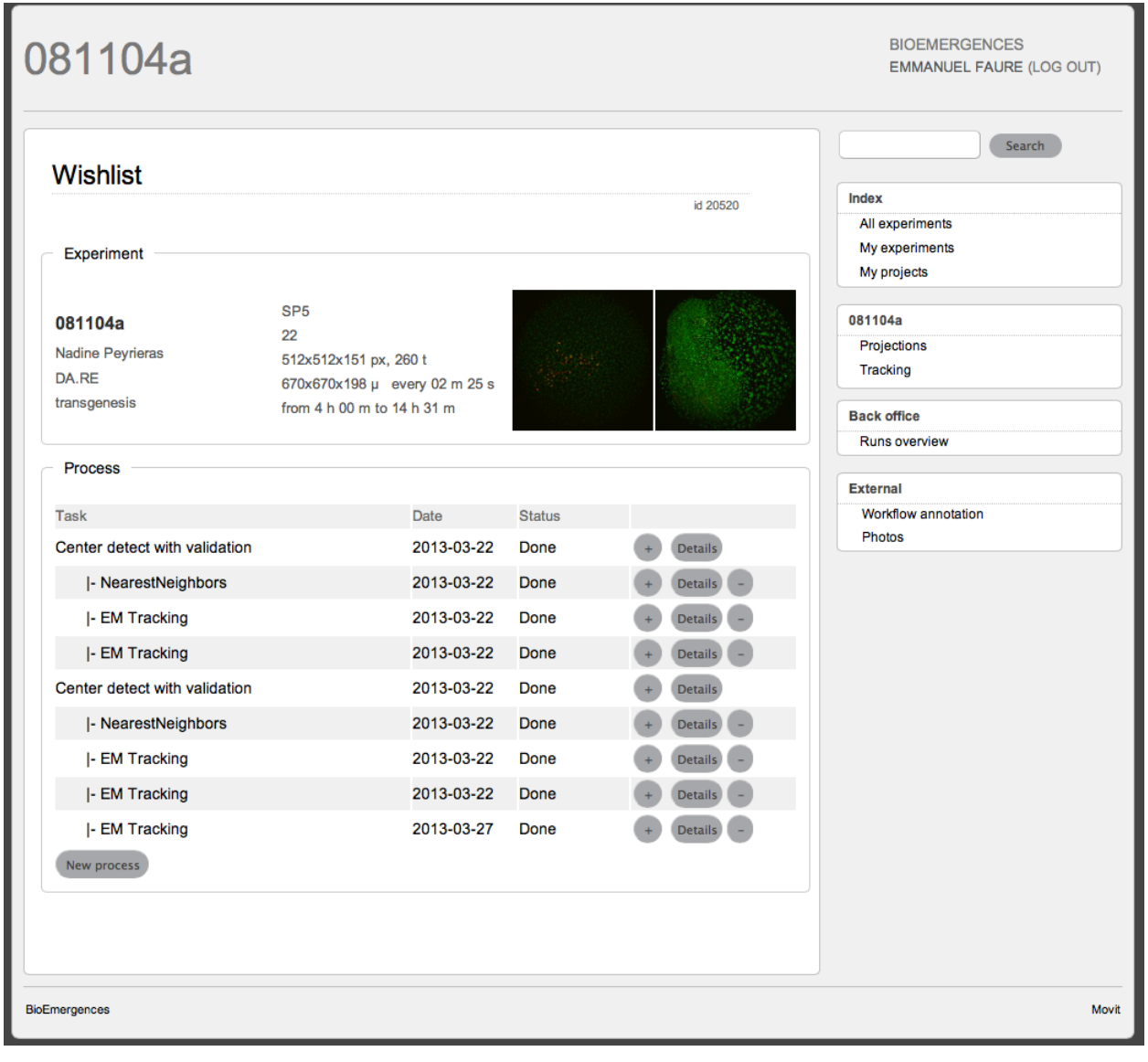
Netlogo model
5 mins per simulation
for 100,000 simulations
= **1 year CPU time**

THE SIMPOP PROJECT



Scala model
2 secs per simulation
for
500,000,000 simulations
= **30 years CPU time**

THE BIOEMERGENCES PROJECT



C model - Portal access

Daily production

10,000 hours / day

USEFUL LINKS

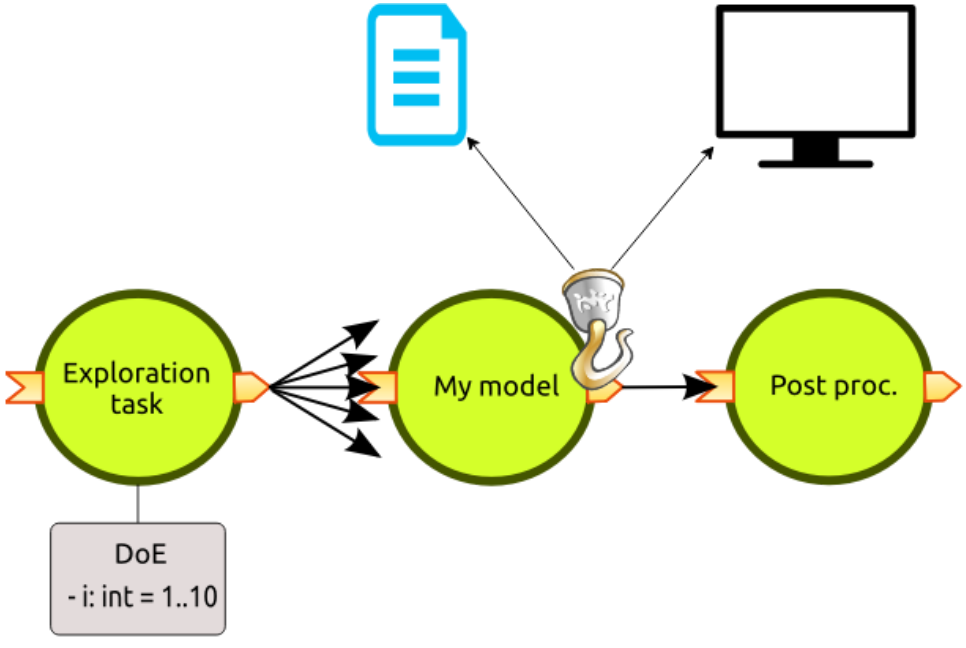
Documentation	www.openmole.org
Mailing-list	list.openmole.org
Development version	next.openmole.org
Source code	github.com/openmole
Market place	github.com/openmole-market

EXTRA NOTIONS

HOOKS

Tasks are mute

Hooks **extract content** from the dataflow




```
exploration -< (model on env) >- (average hook ToStringHook())
```

GROUPING

Short jobs don't cope well with distributed computing

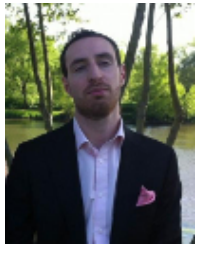
Group **multiple tasks** in the **same batch job**

```
exploration -< (model on env by 10) >- (average hook ToStringHook())
```

THANKS!

romain.reuillon@iscpif.fr

mathieu.leclaire@iscpif.fr



j.passerat-palmbach@imperial.ac.uk