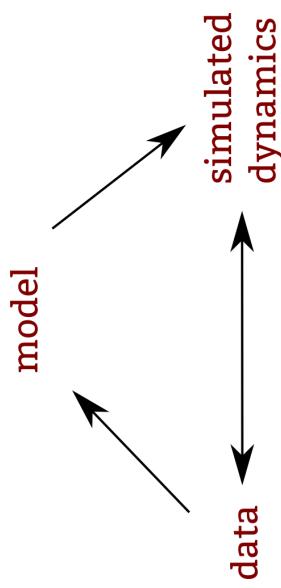
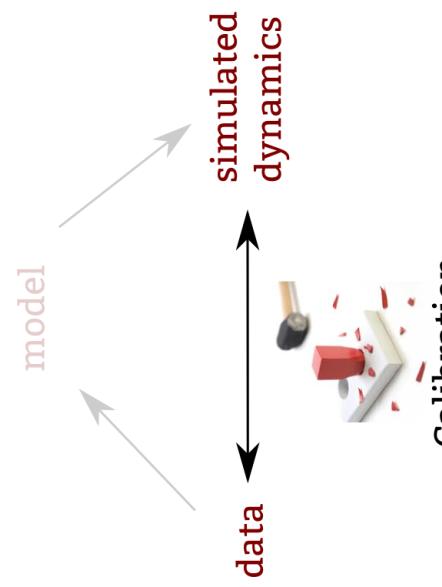
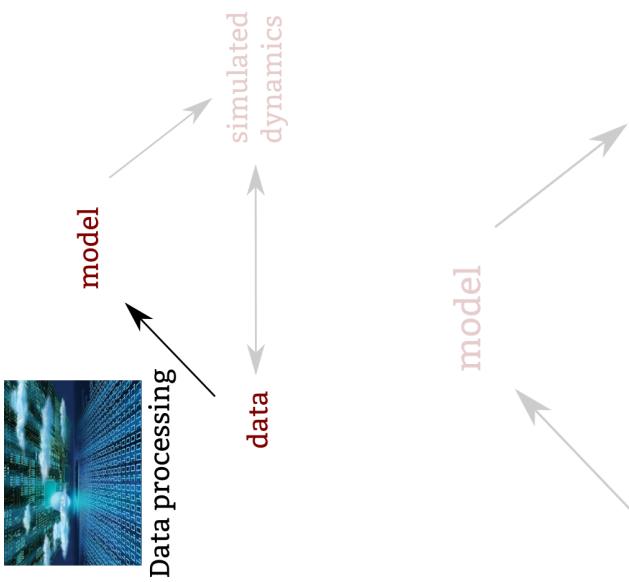
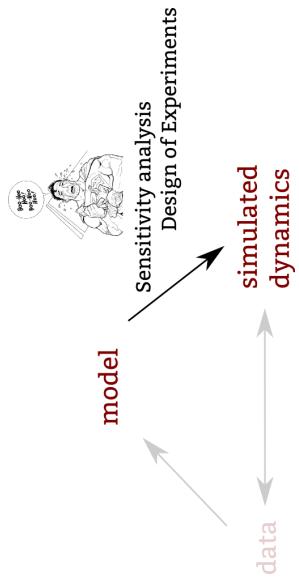




## OPEN MODEL EXPERIMENT



*The Complex system modellisation*

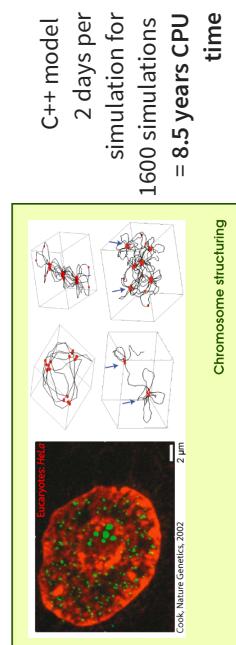


Mathieu Leclaire  
Romain Reuillon  
Jonathan Passerat-Palmbach





## CHROMOSOME STRUCTURING

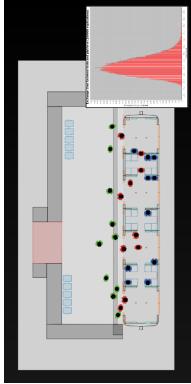


*Naturally parallel algorithms permit to leverage parallelism*



## THE SIMTRAP PROJECT

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



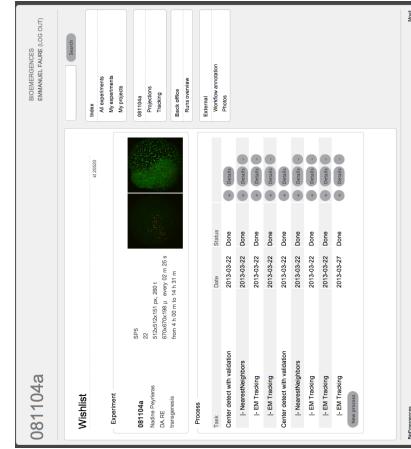
## THE SIMPOP PROJECT

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

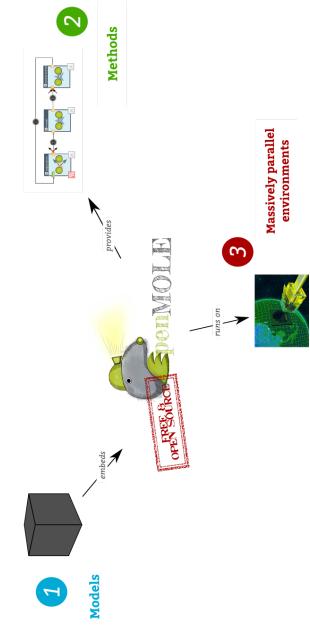


## THE BIOEMERGENCES PROJECT

C model - Portal access  
Daily production 10,000 hours / day



## OPENMOLE IN 3 STEPS

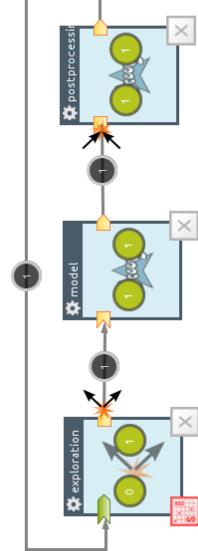


## EMBED YOUR MODEL AS A BLACK BOX



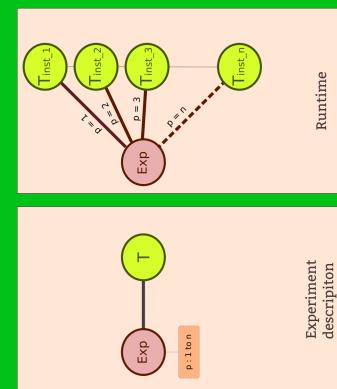
## ZERO DEPLOYMENT APPROACH

- Code is embedded with runtime
- Ships to remote environment
- No prior knowledge of remote environment needed
- No installation required on any machine



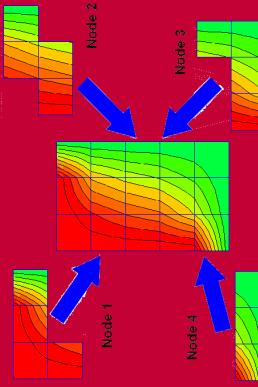
## NATURALLY PARALLEL FORMALISM TO DESIGN EXPERIMENTS

## WHAT OPENMOLE DOES



Experiment description  
Runtime  
Data parallelism

## WHAT OPENMOLE DOES NOT



Parallelisation by message / Task Parallelism  
Example: Spark, MPI, ...

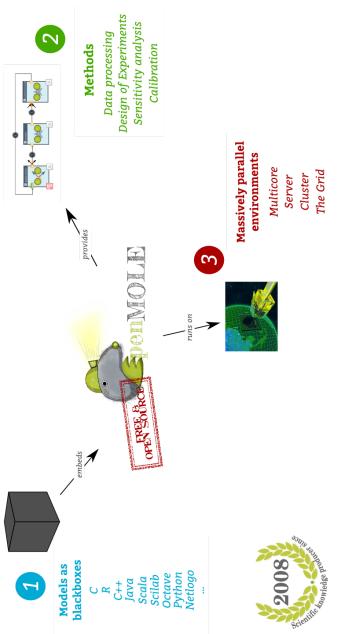
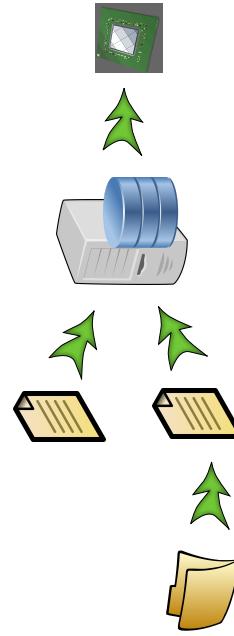
## POWERED BY EGI (AND OTHERS)



SO HARD TO USE...



## AUTOMATIC DATA TRANSFERS AND REPLICA MANAGEMENT



## A WORKFLOW

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

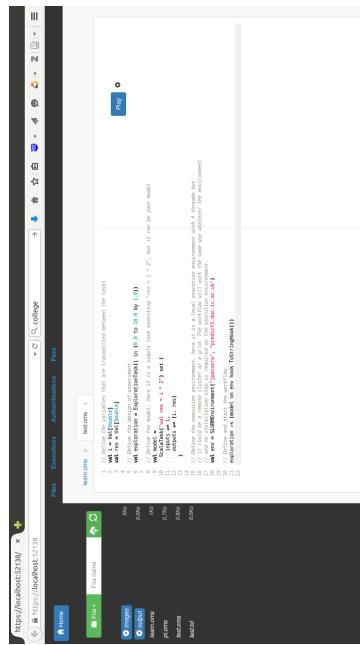
val exploration = ExplorationTask( i in (0.0 to 10.0 by 1.0) )

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

val env = LocalEnvironment(5)
val ex = exploration <- (model on env) start
```

The code block shows a Scala snippet defining an exploration task. It uses a local environment with 5 cores and performs a scalar task where each core calculates the result of multiplying its input by 2.

## WEB APPLICATION



## CONSOLE INTERFACE

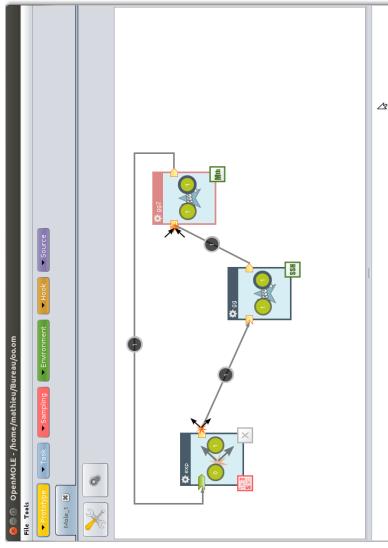
```
[reuililon@simplet: /home/reuililon]$ openmole --c
picked up JAVA_TOOL_OPTIONS: -javaagent:/usr/share/java/jayatanaag.jar
(Type 'q' to quit)
Enter your OpenMole password (for preferences encryption): *****

```

## TOWARDS A MULTI-USER WEB PLATFORM



## TOWARDS GRAPHICAL WORKFLOWS



## USEFUL LINKS

Documentation	<a href="http://www.openmole.org">www.openmole.org</a>
Mailing-list	<a href="http://list.openmole.org">list.openmole.org</a>
Development version	<a href="http://next.openmole.org">next.openmole.org</a>
Source code	<a href="http://github.com/openmole">github.com/openmole</a>
Market place	<a href="http://github.com/openmole-market">github.com/openmole-market</a>

## SETUP YOUR ENVIRONMENT

Download openmole: [next.openmole.org](http://next.openmole.org)

*Don't close the tab you'll need it for the documentation;)*

## HANDS-ON

[openmole-market](http://openmole-market)

Full examples available

```
java -version
java version "1.7.0_75"
OpenJDK Runtime Environment (IcedTea 2.5.4) (7u75-2.5.4-4-2)
OpenJDK 64-Bit Server VM (build 24.75-b04, mixed mode)
```

## ~/.OPENMOLE

Stores your preferences

Maintains a table of file locations

Can grow rapidly in number of files

=> move it to a folder without quota restrictions

## LAUNCH OPENMOLE

Command-line console: openmole -c

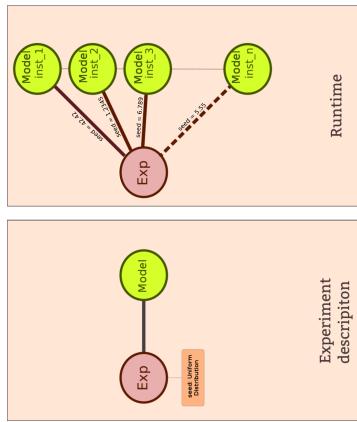
Web UI: openmole

## 1ST EXAMPLE Pi Monte-Carlo calculation

[www.doc.ic.ac.uk/~jpassera/openmole-tutorial-material](http://www.doc.ic.ac.uk/~jpassera/openmole-tutorial-material)

## 1ST EXAMPLE

DoE to achieve



## 1ST EXAMPLE

Remote environment?

- PBS
- Slurm
- Condor
- EGI
- ...

Define Authentication

## 1ST EXAMPLE

Add variables declaration

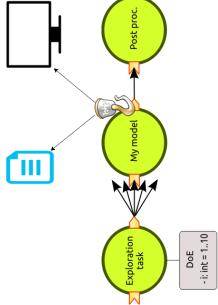
Define exploration

Define environment

# EXTRA NOTIONS

## HOOKS

Tasks are mute  
Hooks extract content from the dataflow



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

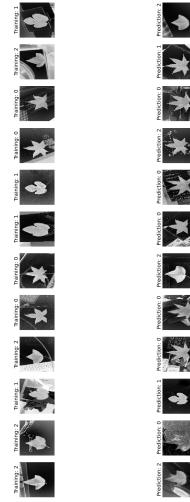
## GROUPING

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())
```

## OUTPUT

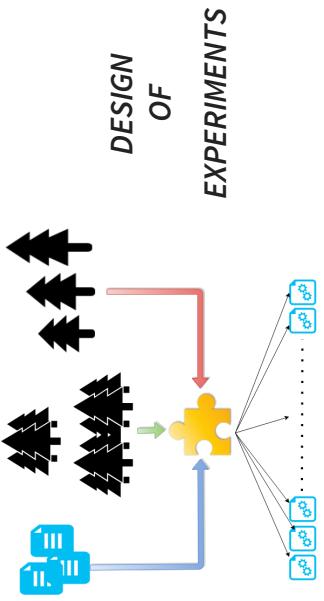


## 2ND EXAMPLE

Tree leaves classifier

[www.doc.ic.ac.uk/~jpassera/openmole-tutorial-material](http://www.doc.ic.ac.uk/~jpassera/openmole-tutorial-material)

## RANDOM FOREST



## PACKAGING AN APPLICATION

Applications have dependencies:

- Shared libraries
- Packages
- Low level system calls
- Environment variables
- ...

Capture these dependencies and transfer along with the application

## CARE

<http://reproducible.io/>

- Run against your application
  - Transfer archive
  - Re-execute
- Change the parameters

```
care -x -o archive.bin python script.py 4.2
```

Only from Linux to Linux

## 2ND EXAMPLE

- Add variables declaration
- Define exploration
- Define environment
- Collect results as CSV

## RESULTS VISUALISATION?

- Add another task
  - Running locally
  - Example: Python + Matplotlib scatter plot

## GOING FURTHER

- Sources
- Advanced DoE
- Desktop Grid
- Plugins
- Strainer Pattern

# THANKS!

[romain.reuillon@iscpif.fr](mailto:romain.reuillon@iscpif.fr)

[mathieu.leclaire@iscpif.fr](mailto:mathieu.leclaire@iscpif.fr)

[j.passerat-palmbach@imperial.ac.uk](mailto:j.passerat-palmbach@imperial.ac.uk)

