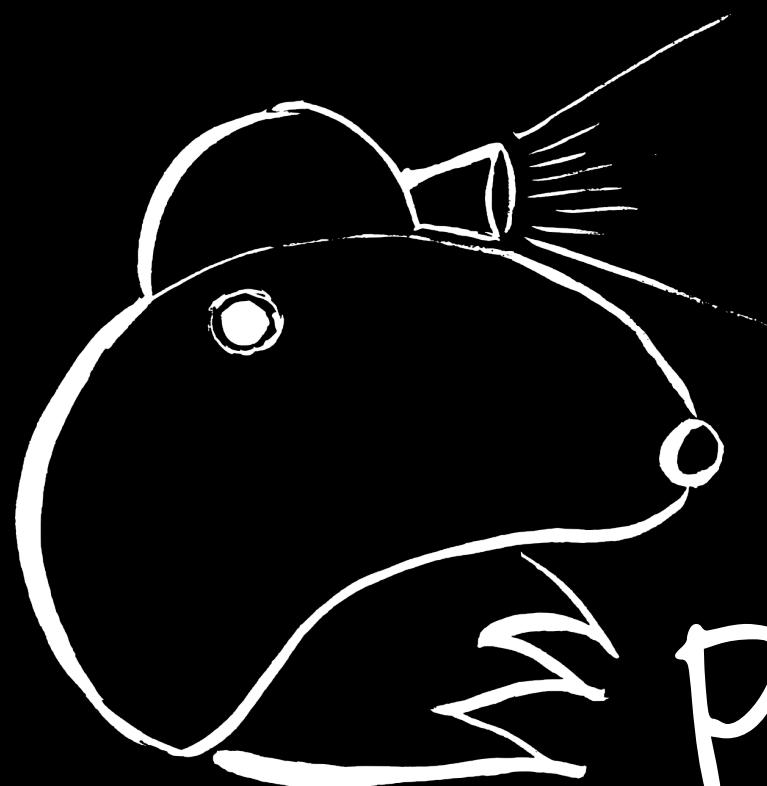


# Distributing a (Linux) native application with



penMOLE

# Prerequisite

You should have:

- OpenMOLE (5.4)
- A Linux computer  
(to practice packaging)
- Python

# Get the archives

Go to [www.openmole.org/files/csdcc](http://www.openmole.org/files/csdcc)

Get packaging.tgz and Native.tgz

Extract packaging.tgz

# A native application

```
reuillon@simplet: /home/reuillon/Documents/Recherche/Work/OpenMOLE/python
import sys
import numpy
from numpy import *
from array import *
import csv

input = open(sys.argv[1], 'r')
n = float(sys.argv[2])

print("reading the matrix")
data = csv.reader(input)

array = numpy.array(list(data)).astype(float)

print(array)
print(n)
mult = array * n

print("saving the matrix")
numpy.savetxt(sys.argv[3], mult, fmt='%g')
```

2,1

33%

Content of "matrix.py"

# Run it

```
[reuillon:~/Documents ... ond/application] 2s $ python matrix.py data.csv 2 out.csv
reading the matrix
[[ 8.  8.  9.]
 [ 7.  6.  8.]
 [ 5.  7.  8.]]
2.0
saving the matrix
[reuillon:~/Documents ... ond/application] 6s $ ls
care data.csv matrix.py out.csv
[reuillon:~/Documents ... ond/application] 5s $ █
```

Reads data.csv

Multiply it by 2

Write out.csv

# Make it portable

This application depends on:

- python
- numpy
- libc...

The solution: use CARE

<http://reproducible.io/>

# Make it portable

```
./care -o python.bin python matrix.py data.csv 2 out.csv
care info: concealed path: $HOME /home/reuillon
care info: concealed path: /tmp
care info: revealed path: $PWD /home/reuillon/Documents/Recherche/Work/OpenMOLE/python/second/application
care info: revealed path: /usr/bin/python2.7
care info: -----
reading the matrix
[[ 8.  8.  9.]
 [ 7.  6.  8.]
 [ 5.  7.  8.]]
2.0
saving the matrix
care info: -----
care info: Hints:
care info:   - search for "conceal" in `care -h` if the execution didn't go as expected.
care info:   - run `./python.bin` or `care -x python.bin` to extract the output archive correctly.
$ls
care data.csv matrix.py out.csv python.bin
$
```

# Make it portable

What is python.bin?

- An auto-extractible archive
- Contains every single files accessed by the program (and a bit more)

How do I use it?

- Extract: ./python.bin
- Run: ./python/re-execute.sh

# Make it portable

What's inside `python.bin`?

```
reuillon@simpler:~/home/reuillon/Documents/Recherche/Projects/open... x | reuillon@simpler:~/home/reuillon/Documents/Recherche/Work/OpenM... x | reuillon@simpler:~/home/reuillon/Documents/Recherche/Work/OpenM... x
$tree python
python
├── concealed-accesses.txt
├── proot ← Emulates your system
├── README.txt
└── re-execute.sh ← Re-executes your program
rootfs
├── dev
└── etc
    ├── alternatives
    │   ├── libblas.so.3 -> /usr/lib/libblas/libblas.so.3
    │   └── liblapack.so.3 -> /usr/lib/lapack/liblapack.so.3
    ├── ld.so.cache
    ├── localtime
    └── python2.7
        └── sitecustomize.py
home
└── reuillon
    └── Documents
        └── Recherche
            └── Work
                └── OpenMOLE
                    └── python
                        ├── data.csv
                        ├── matrix.py
                        └── out.csv
```

rootfs/ is your file system

# Make it portable

```
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/core.pyc
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/extras.py
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/extras.pyc
info: extracted: python/rootfs/usr/lib/python2.7/csv.py
info: extracted: python/rootfs/usr/lib/python2.7/csv.pyc
info: extracted: python/rootfs/usr/lib/python2.7/lib-dynload/_csv.x86_64-linux-gnu.so
info: extracted: python/rootfs/home/reuillon/Documents/Recherche/Work/OpenMOLE/python/second/application/data.csv
info: extracted: python/rootfs/home/reuillon/Documents/Recherche/Work/OpenMOLE/python/second/application/out.csv
info: extracted: python/re-execute.sh
info: extracted: python/concealed-accesses.txt
info: extracted: python/README.txt
info: extracted: python/proot
$ ./python/re-execute.sh
reading the matrix
[[ 8.  8.  9.]
 [ 7.  6.  8.]
 [ 5.  7.  8.]]
2.0
saving the matrix
$
```

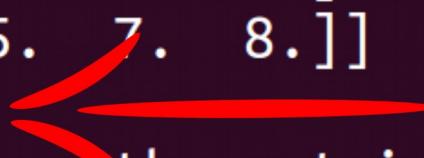
It will work on any other Linux machine

# Make it portable

And you can change the parameters

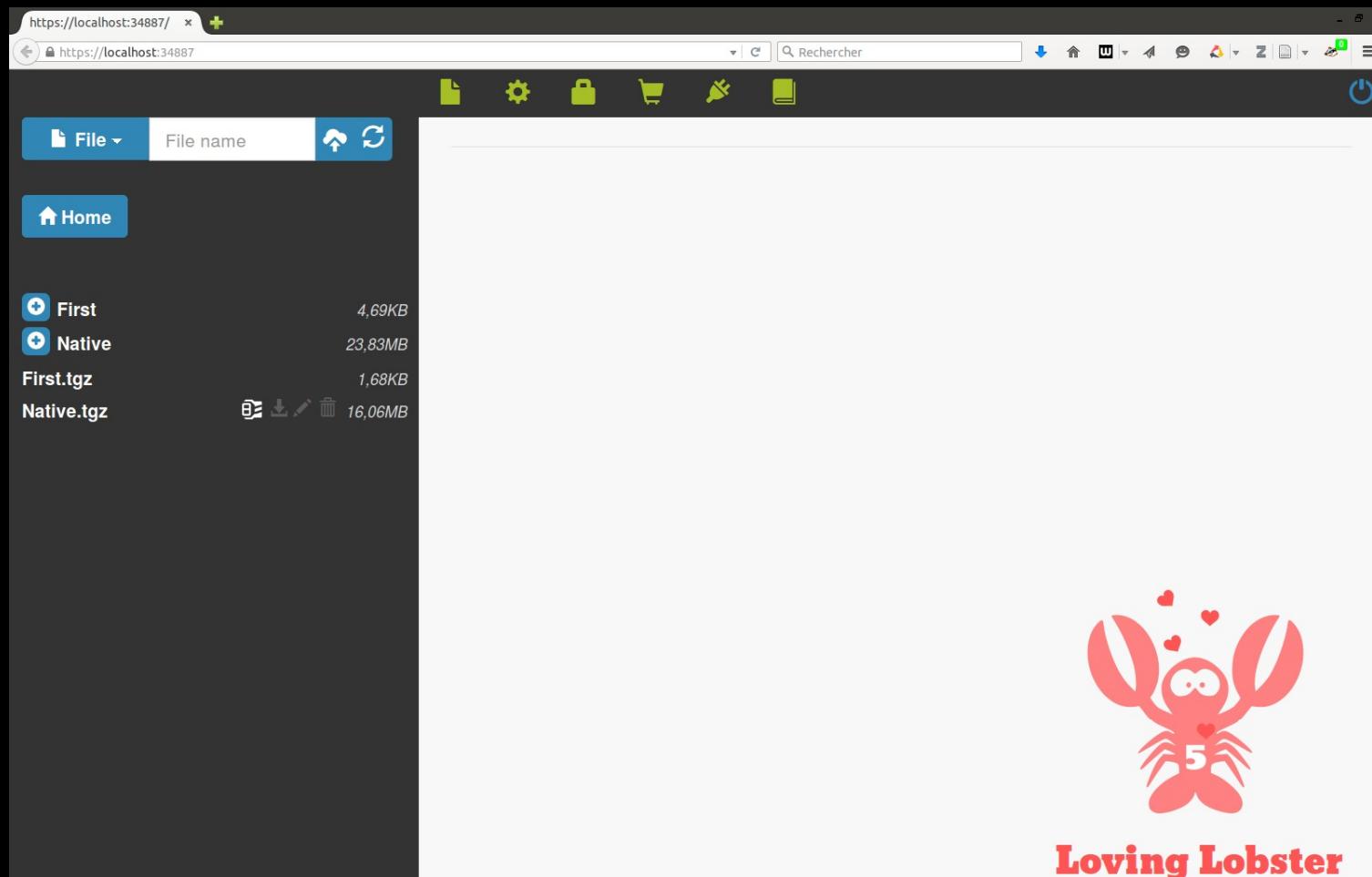
```
$./python/re-execute.sh python matrix.py data.csv 4 out.csv
reading the matrix
[[ 8.  8.  9.]
 [ 7.  6.  8.]
 [ 5.  7.  8.]]
4.0
saving the matrix
$
```

Now it's 4.0



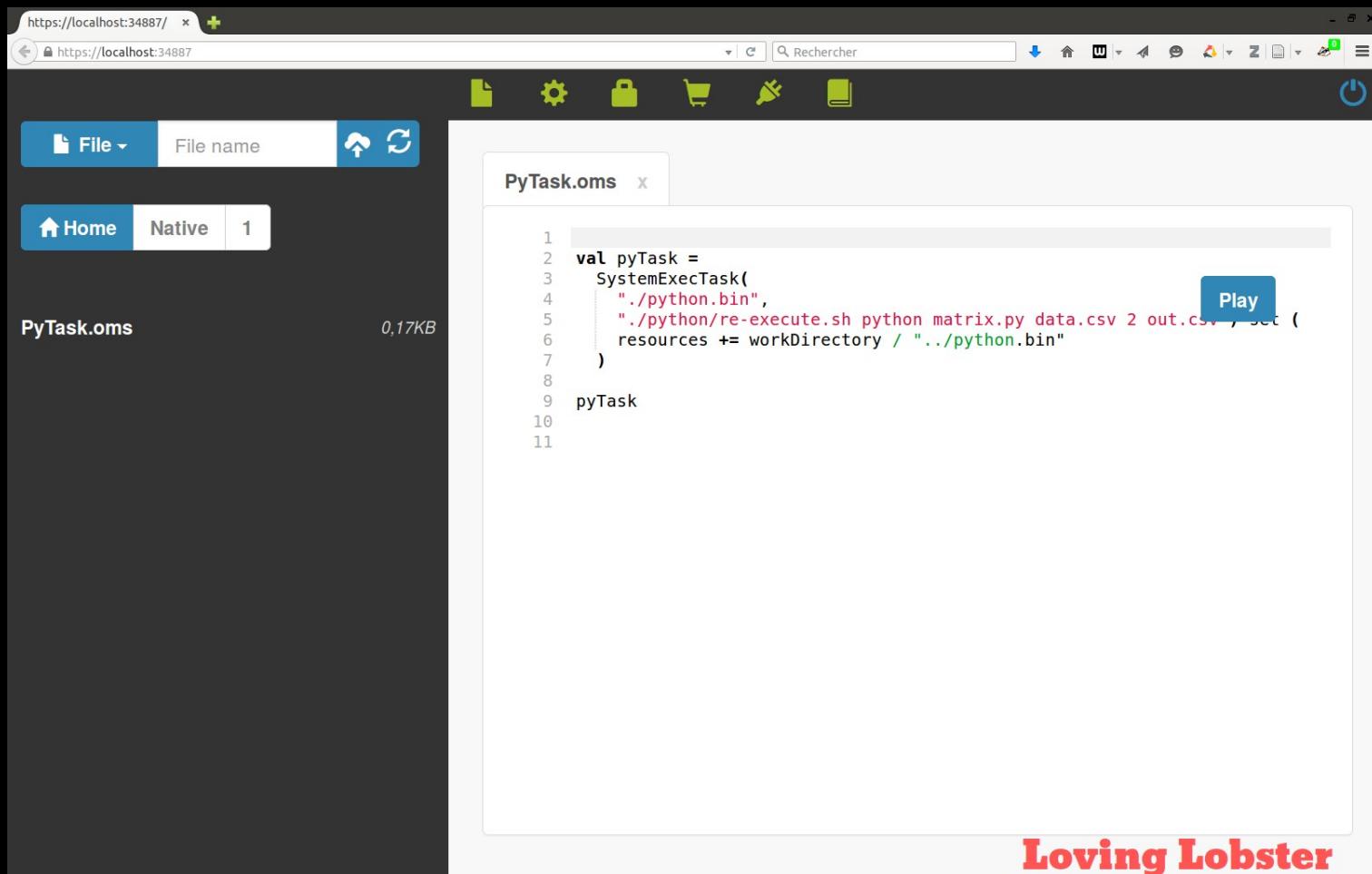
# Run it in OpenMOLE

Launch OpenMOLE, upload native.tgz and extract it.



# Run it in OpenMOLE

Go to "Native/1" and edit PyTask.oms



The screenshot shows the OpenMOLE web interface. At the top, there's a browser bar with the URL <https://localhost:34887/>. Below the browser bar is a toolbar with various icons. The main area has tabs for "File", "Native", and "1". The "Native" tab is selected. On the left, there's a list of files: "PyTask.oms" (0,17KB). The right side is a code editor titled "PyTask.oms" containing the following Scala code:

```
1 val pyTask =
2   SystemExecTask(
3     "./python.bin",
4     "./python/re-execute.sh python matrix.py data.csv 2 out.csv", set(
5       resources += workDirectory / "../python.bin"
6     )
7   )
8
9 pyTask
10
11
```

A blue "Play" button is located to the right of the code editor. At the bottom right of the interface, the text "Loving Lobster" is displayed.

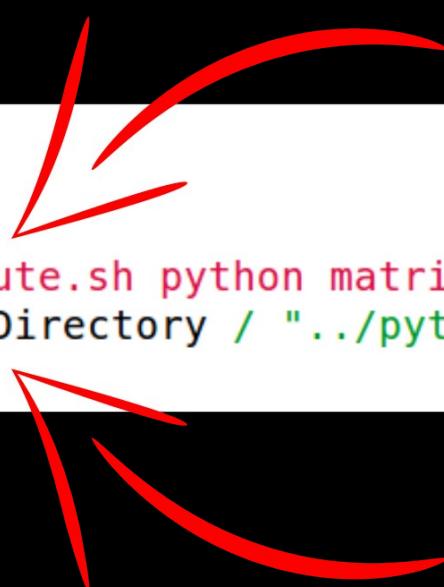
# Run it in OpenMOLE

Use a SystemExecTask:

```
2 val pyTask =  
3   SystemExecTask(  
4     "./python.bin",  
5     "./python/re-execute.sh python matrix.py data.csv 2 out.csv") set (  
6       resources += workDirectory / "../python.bin"  
7     )
```

Commands

Required files



# Run it in OpenMOLE

The screenshot shows the OpenMOLE web interface at <https://localhost:34887/>. The main title bar says "VOMS Admin > vo.com...". The browser address bar also shows <https://localhost:34887>. The search bar contains "Rechercher". The top navigation bar includes icons for file, settings, security, cart, and help.

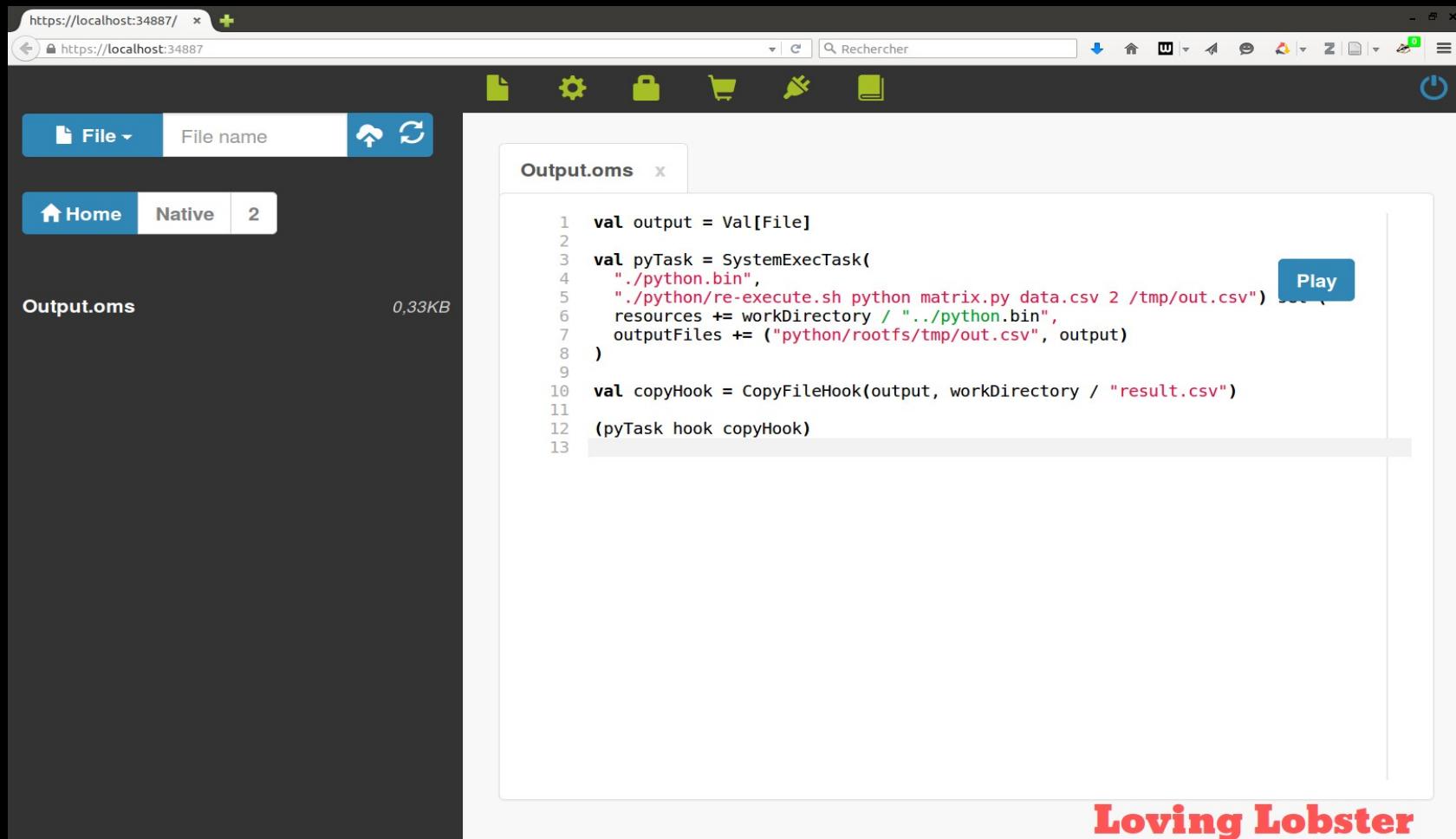
The main area displays an "Executions" card for a task named "PyTask.oms" from 30/9/2015, 14:33:20. The card shows 0 errors, 1 / 1 steps completed, 0:00:00 duration, and a status of "finished". It includes buttons for environment, output history (500 entries), and other controls.

The "Output history" section shows the following log entries:

```
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/__init__.py
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/_init_.pyc
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/core.py
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/core.pyc
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/extras.py
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/extras.pyc
info: extracted: python/rootfs/usr/lib/python2.7/csv.py
info: extracted: python/rootfs/usr/lib/python2.7/csv.pyc
info: extracted: python/rootfs/usr/lib/python2.7/lib-dynload/_csv.x86_64-linux-gnu.so
info: extracted: python/rootfs/home/reuillon/Documents/Recherche/Work/OpenMOLE/python/data.csv
info: extracted: python/rootfs/home/reuillon/Documents/Recherche/Work/OpenMOLE/python/out.csv
info: extracted: python/re-execute.sh
info: extracted: python/concealed-accesses.txt
info: extracted: python/README.txt
info: extracted: python/proot
reading the matrix
[[ 8.  8.  9.]
 [ 7.  6.  8.]
 [ 5.  7.  8.]]
2.0
saving the matrix
```

# Get the output file

Go to "2" and edit Output.oms



The screenshot shows a web browser window with a dark theme. The address bar displays `https://localhost:34887/`. The main content area is a code editor titled "Output.oms". The code is as follows:

```
1 val output = Val[File]
2
3 val pyTask = SystemExecTask(
4     "./python.bin",
5     "./python/re-execute.sh python matrix.py data.csv 2 /tmp/out.csv")
6 resources += workDirectory / "./python.bin",
7 outputFiles += ("python/rootfs/tmp/out.csv", output)
8
9
10 val copyHook = CopyFileHook(output, workDirectory / "result.csv")
11
12 (pyTask hook copyHook)
13
```

On the right side of the code editor, there is a blue "Play" button. The status bar at the bottom right of the browser window shows the text "Loving Lobster".

# Get the output file

```
1 val output = Val[File] ← A file variable
2
3 val pyTask = SystemExecTask(
4   "./python.bin",
5   "./python/re-execute.sh python matrix.py data.csv 2 /tmp/out.csv") set(
6     resources += workDirectory / "../python.bin",
7     outputFiles += ("python/rootfs/tmp/out.csv", output))
8 )
9
10 val copyHook = CopyFileHook(output, workDirectory / "result.csv")
11 (pyTask hook copyHook)
12
13 |
```

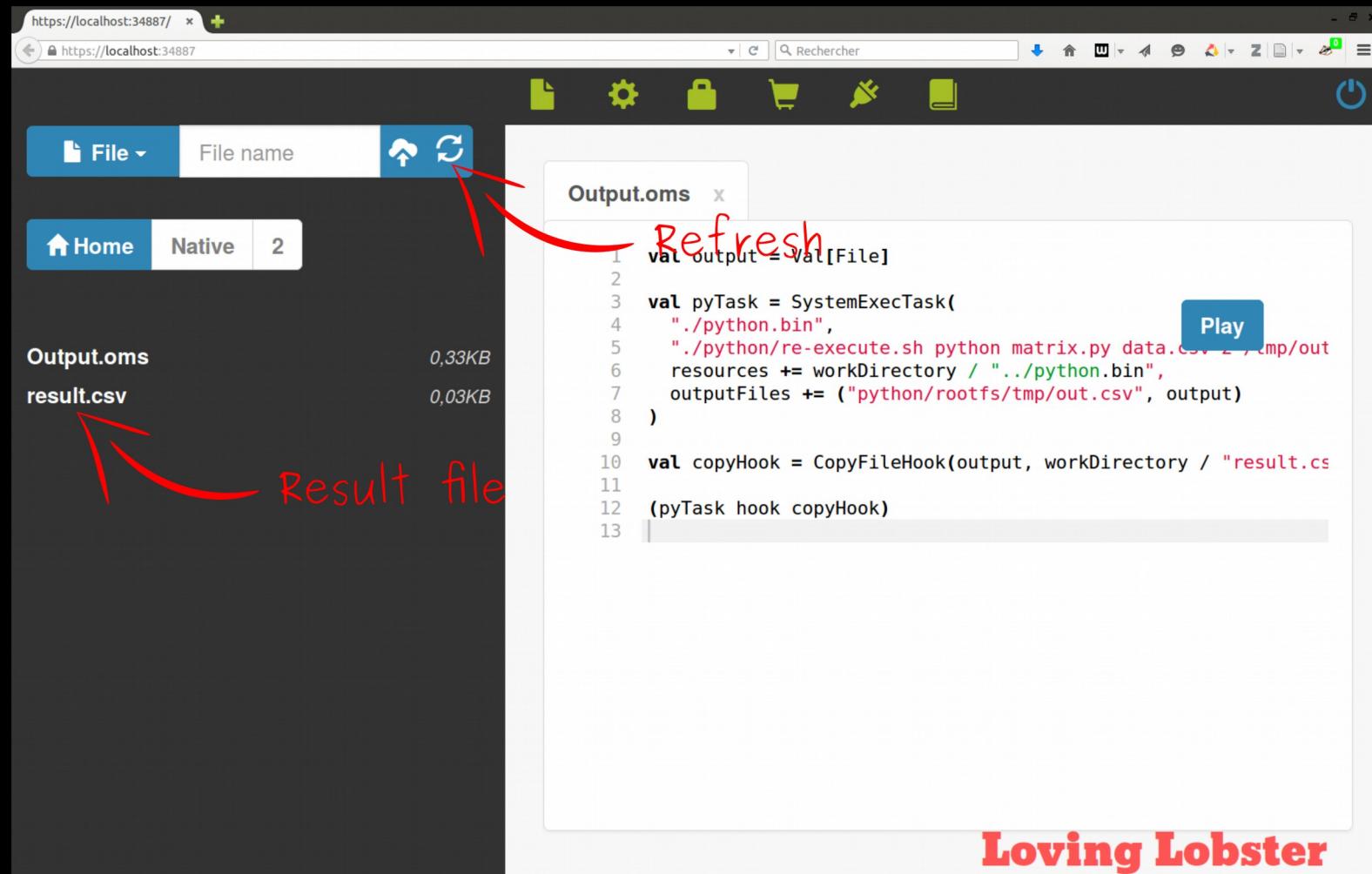
Use a hook to extract  
the file from the dataflow

Location of  
the output file

Put the output file  
in the dataflow

# Get the output file

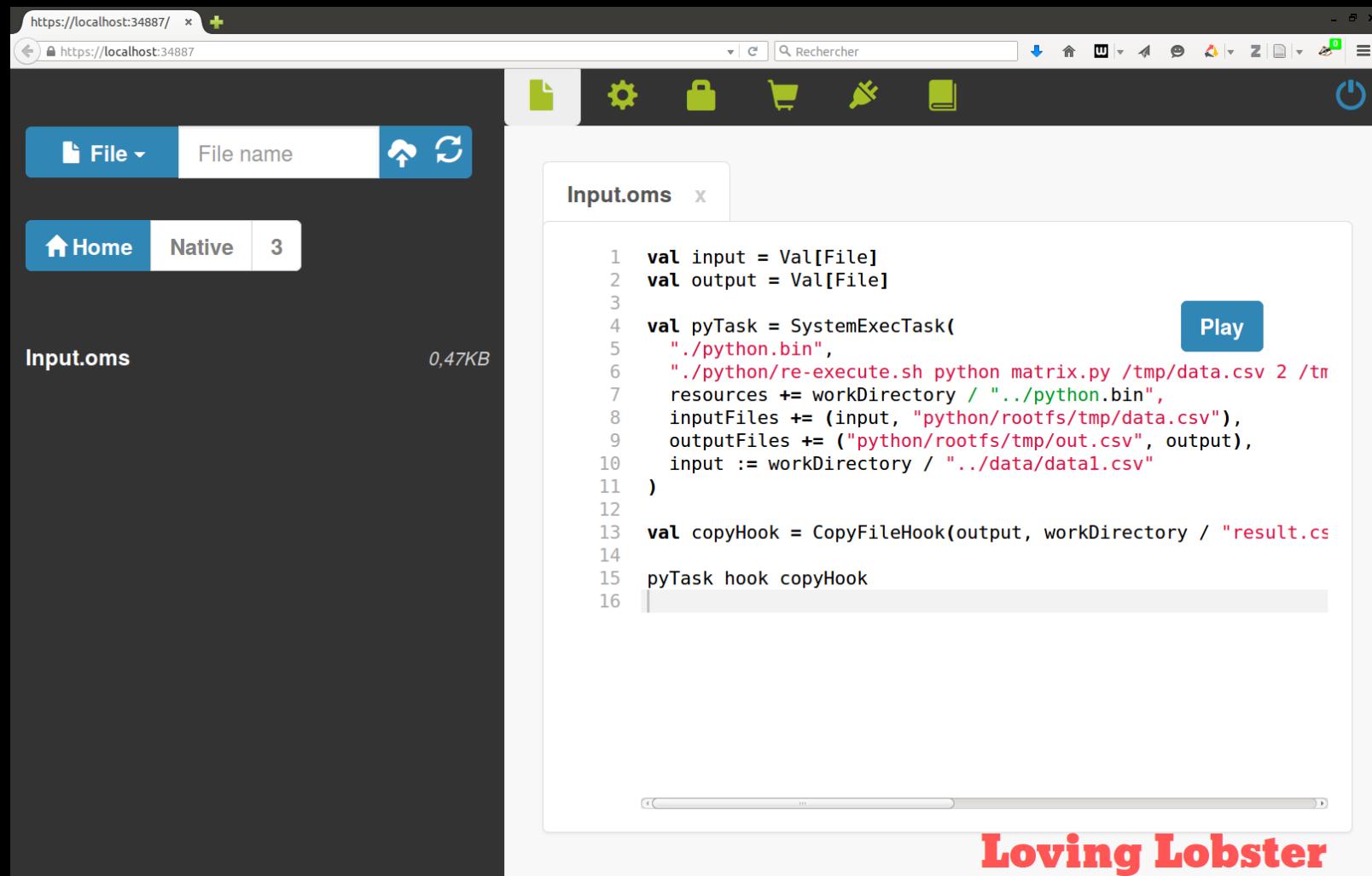
Run it and refresh.



Loving Lobster

# Inject an input file

Go to "3" and edit Input.oms



The screenshot shows a web-based interface for editing code. The URL is <https://localhost:34887/>. The main area displays a file named `Input.oms` with the following content:

```
1 val input = Val[File]
2 val output = Val[File]
3
4 val pyTask = SystemExecTask(
5   "./python/bin",
6   "./python/re-execute.sh python matrix.py /tmp/data.csv 2 /tmp/
resources += workDirectory / "../python/bin",
8   inputFiles += (input, "python/rootfs/tmp/data.csv"),
9   outputFiles += ("python/rootfs/tmp/out.csv", output),
10  input := workDirectory / "../data/data1.csv"
11 )
12
13 val copyHook = CopyFileHook(output, workDirectory / "result.cs"
14
15 pyTask hook copyHook
16
```

The interface includes a toolbar with various icons (File, Settings, etc.), a search bar, and a navigation bar with tabs for Home, Native, and a selected tab labeled "3". A "Play" button is located on the right side of the code editor. The footer of the page features the text "Loving Lobster" in red.

# Inject an input file

```
1 val input = Val[File]
2 val| output = Val[File]
3
4 val pyTask = SystemExecTask(
5   "./python.bin",
6   "./python/re-execute.sh python matrix.py /tmp/data.csv 2 /tmp/out.csv") set (
7   resources += workDirectory / "../python.bin",
8   inputFiles += (input, "python/rootfs/tmp/data.csv"),
9   outputFiles += ("python/rootfs/tmp/out.csv", output),
10  input := workDirectory / "../data/data1.csv"
11 )
12
13 val copyHook = CopyFileHook(output, workDirectory / "result.csv")
14
15 pyTask hook copyHook
16
```

A new file variable

Changed input file location

Copy the file in the task execution directory

Set the default value of the input variable

# Inject an input file

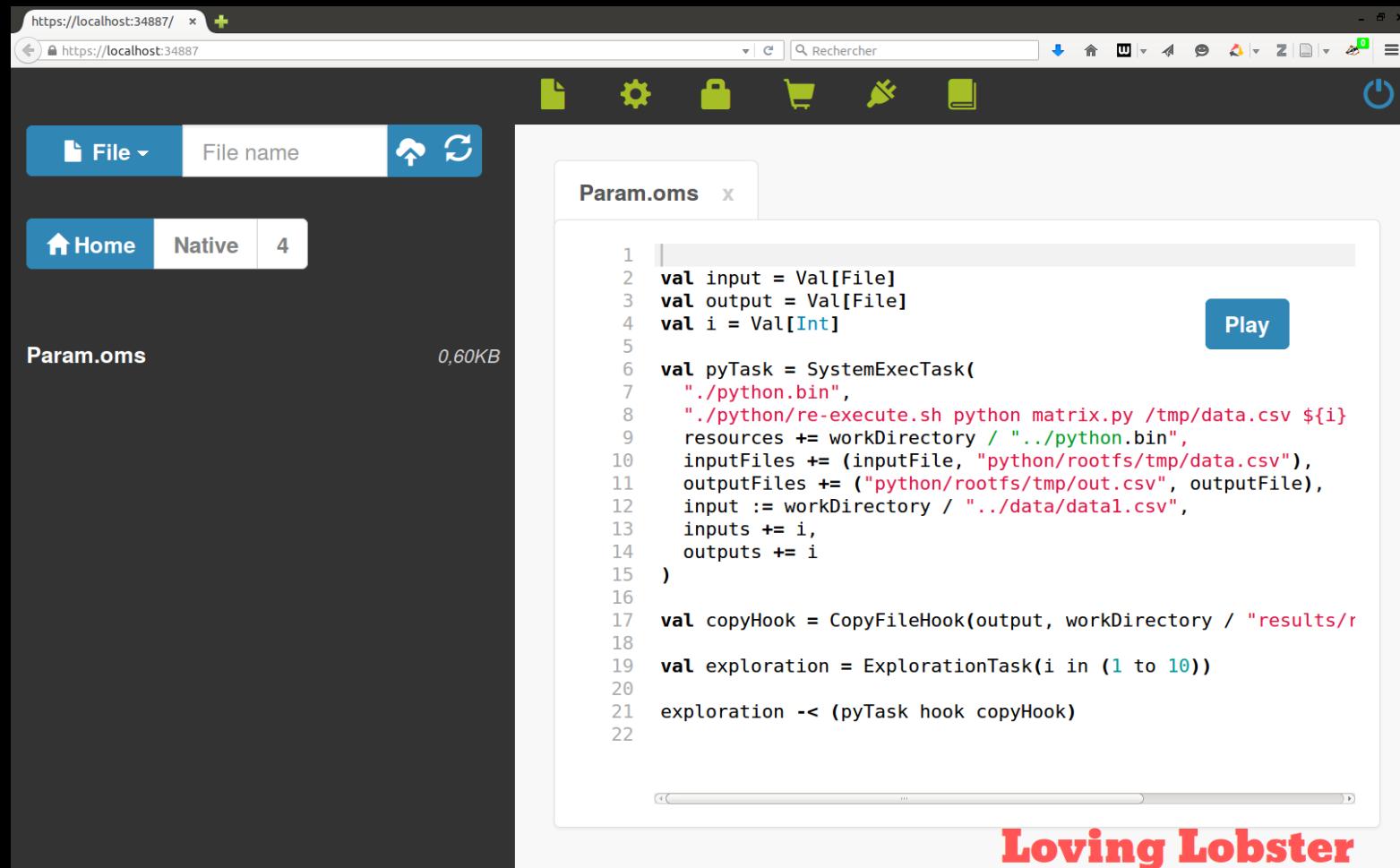
Run it!

You can change the file manually:

- data1.csv
- data2.csv
- data3.csv

# Explore a parameter

Go to "4" and edit Param.oms



The screenshot shows a web-based interface for editing code. The URL is https://localhost:34887/. The main window displays a file named Param.oms with the following content:

```
1
2 val input = Val[File]
3 val output = Val[File]
4 val i = Val[Int]
5
6 val pyTask = SystemExecTask(
7   "./python.bin",
8   "./python/re-execute.sh python matrix.py /tmp/data.csv ${i}
9   resources += workDirectory / "../python.bin",
10  inputFiles += (inputFile, "python/rootfs/tmp/data.csv"),
11  outputFiles += ("python/rootfs/tmp/out.csv", outputFile),
12  input := workDirectory / "../data/data1.csv",
13  inputs += i,
14  outputs += i
15 )
16
17 val copyHook = CopyFileHook(output, workDirectory / "results/r
18
19 val exploration = ExplorationTask(i in (1 to 10))
20
21 exploration -< (pyTask hook copyHook)
22
```

The interface includes a toolbar with icons for file operations, a search bar, and a navigation bar with tabs for Home, Native, and a selected tab labeled '4'. A 'Play' button is located on the right side of the code editor. At the bottom right of the window, the text 'Loving Lobster' is displayed.

# Explore a parameter

```
2 val input = Val[File]
3 val output = Val[File]
4 val i = Val[Int] A new variable
5
6 val pyTask = SystemExecTask(
7   "./python.bin",
8   "./python/re-execute.sh python matrix.py /tmp/data.csv ${i} /tmp/out.csv") set (
9   resources += workDirectory / "../python.bin",
10  inputFiles += (input, "python/rootfs/tmp/data.csv"),
11  outputFiles += ("python/rootfs/tmp/out.csv", output),
12  input := workDirectory / "../data/datal.csv",
13  inputs += i,
14  outputs += i "i" is an input and an output
15 )
16
17 val copyHook = CopyFileHook(output, workDirectory / "results/result${i}.csv")
18
19 val exploration = ExplorationTask(i in (1 to 10))
20
21 exploration -<- (pyTask hook copyHook)
22
```

*"exploration"*  
is used in  
the workflow

*"i"* varies from  
1 to 10

The value of "i"  
is used in the  
command line

The value of "i"  
is used in the hook

# Explore a parameter

Run it, refresh and go to "results" folder.

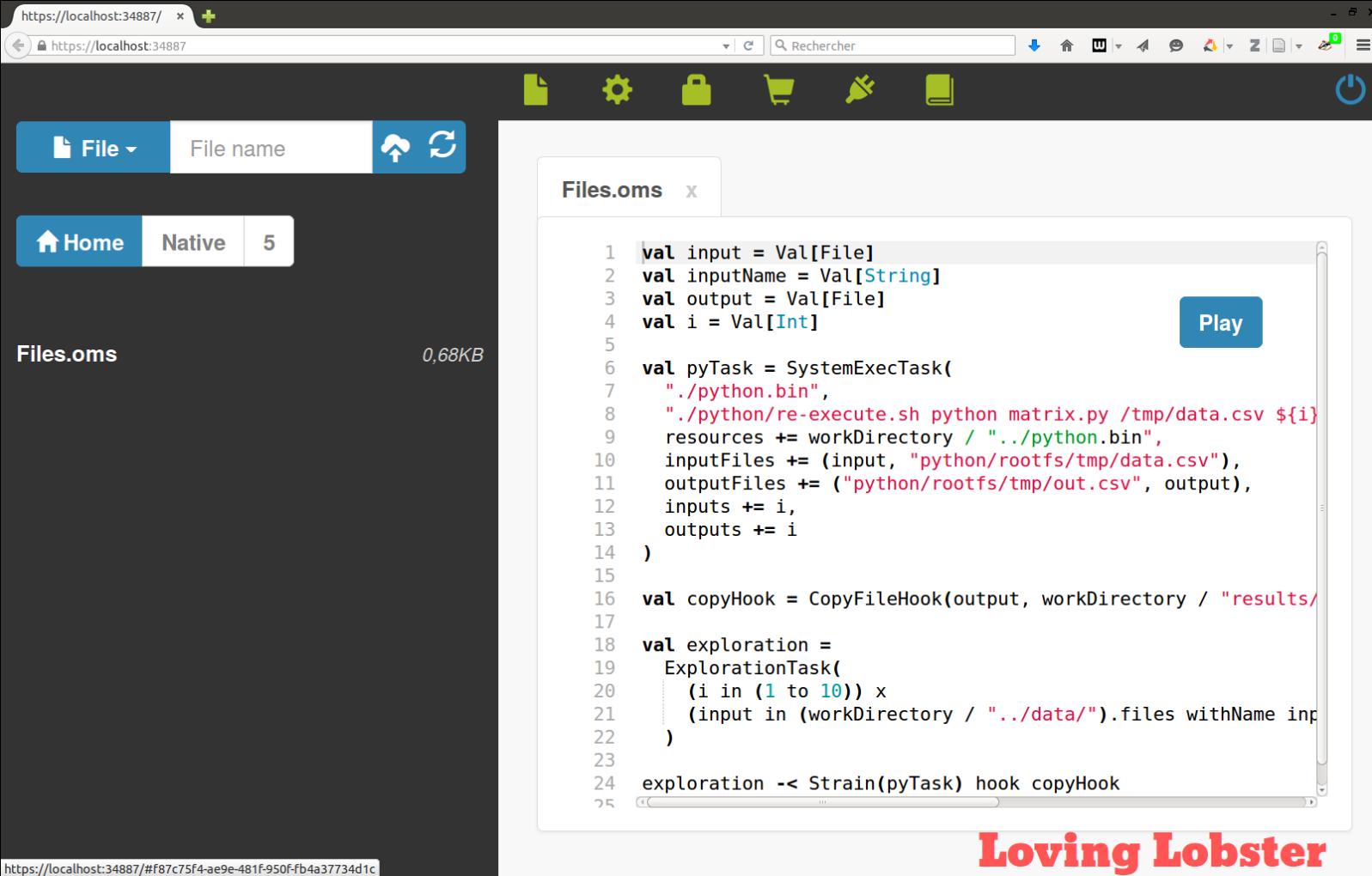
The screenshot shows a web browser window with the URL <https://localhost:34887>. The interface is a dark-themed file manager. On the left, there's a sidebar with a navigation bar including 'Home', '...', '4', and 'results'. Below this are ten entries for CSV files: 'result1.csv', 'result10.csv', 'result2.csv', 'result3.csv', 'result4.csv', 'result5.csv', 'result6.csv', 'result7.csv', 'result8.csv', and 'result9.csv', each with a size of '0,02KB' or '0,03KB'. The main area shows a file named 'Param.oms' with a tab for 'result1.csv'. The content of 'result1.csv' is a table with four rows:

	3	8	0
1	3	8	0
2	4	42	7
3	98	8	12

A blue edit icon is located in the bottom right corner of the 'result1.csv' view. At the bottom right of the entire interface, the text 'Loving Lobster' is displayed in red.

# Explore the input file

Go to "5" and edit Files.oms



The screenshot shows a web-based interface for editing code. At the top, there's a browser header with the URL <https://localhost:34887/>. Below the header is a toolbar with various icons: a folder, settings, a green bag, a shopping cart, a plug, and a document. On the left, there's a sidebar with a 'File' dropdown, a 'File name' input field containing 'Files.oms', and a file upload button. Below this are navigation links: 'Home' (highlighted), 'Native', and a number '5'. The main area is titled 'Files.oms' and contains the following Scala code:

```
1 val input = Val[File]
2 val inputName = Val[String]
3 val output = Val[File]
4 val i = Val[Int]
5
6 val pyTask = SystemExecTask(
7   "./python.bin",
8   "./python/re-execute.sh python matrix.py /tmp/data.csv ${i}"
9   resources += workDirectory / "./python.bin",
10  inputFiles += (input, "python/rootfs/tmp/data.csv"),
11  outputFiles += ("python/rootfs/tmp/out.csv", output),
12  inputs += i,
13  outputs += i
14 )
15
16 val copyHook = CopyFileHook(output, workDirectory / "results/"
17
18 val exploration =
19   ExplorationTask(
20     (i in (1 to 10)) x
21     (input in (workDirectory / "../data/").files withName input)
22   )
23
24 exploration -> Strain(pyTask) hook copyHook
25
```

To the right of the code editor is a large blue 'Play' button. At the bottom right of the main window, the text 'Loving Lobster' is displayed in red.

# Explore the input file

```
1 val input = Val[File]
2 val inputName = Val[String]
3 val output = Val[File]
4 val i = Val[Int]
5
6 val pyTask = SystemExecTask(
7   "./python.bin",
8   "./python/re-execute.sh python matrix.py /tmp/data.csv ${i} /tmp/out.csv") set (
9   resources += workDirectory / "../python.bin",
10  inputFiles += (input, "python/rootfs/tmp/data.csv"),
11  outputFiles += ("python/rootfs/tmp/out.csv", output),
12  inputs += i,
13  outputs += i
14 )
15
16 val copyHook = CopyFileHook(output, workDirectory / "results/${inputName.dropRight(4)}/result${i}.csv")
17
18 val exploration =
19   ExplorationTask(
20     (i in (1 to 10)) x
21     (input in (workDirectory / "../data/").files withName inputName)
22   )
23
24 exploration -> Strain(pyTask) hook copyHook
25
```

A variable to store  
the name of the input file

The path of  
contains the

Combine

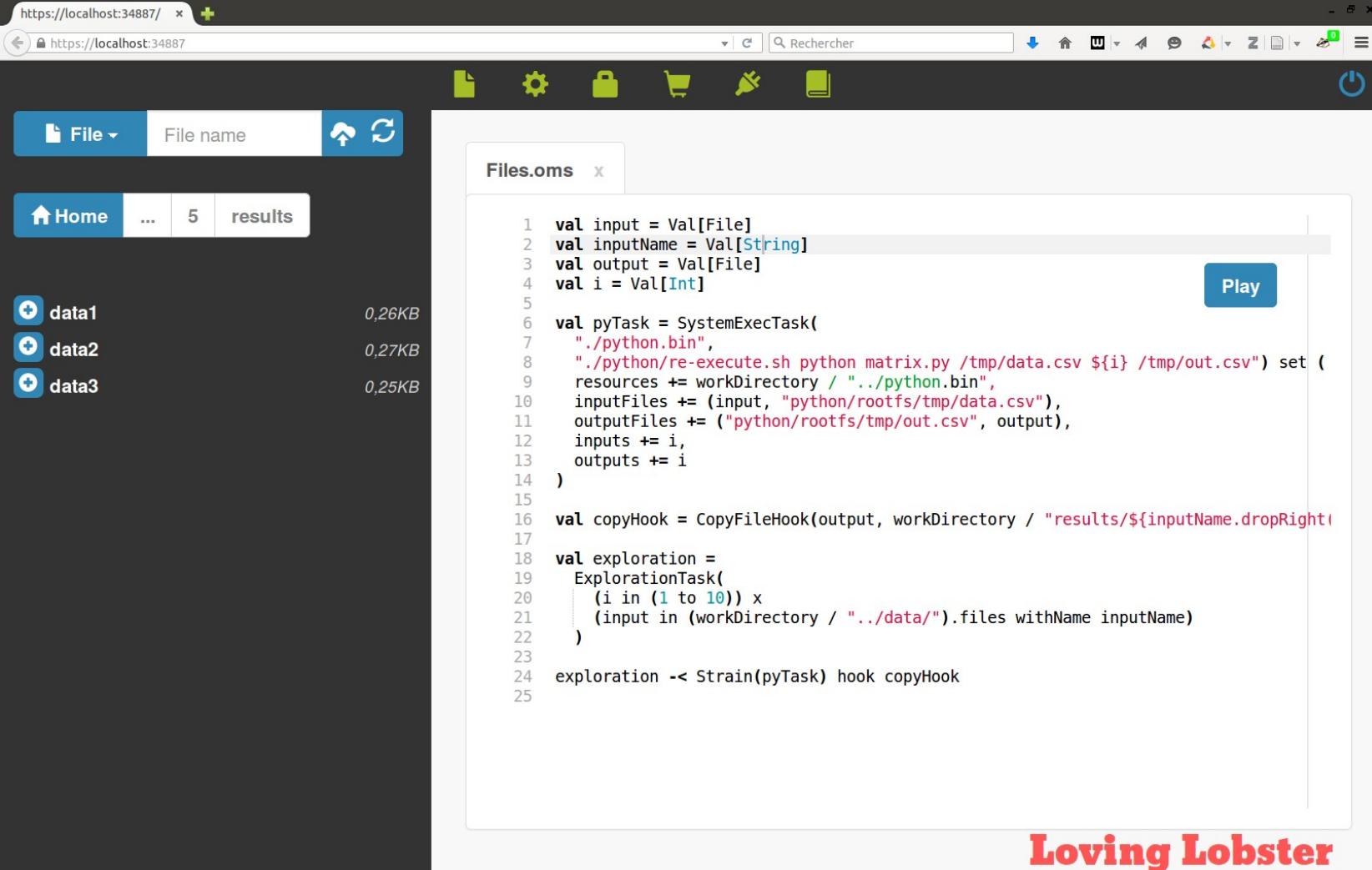
the result file  
name of the input file

Go through all the files of the folder  
"data", put the file in "input" and  
its name in "inputName"

All inputs are also outputs  
(they are used by the hook)

# Explore the input file

Run it and go in the "results" folder.



The screenshot shows a web browser window with the URL <https://localhost:34887/>. The interface has a dark theme with various icons at the top. On the left, there's a sidebar with a 'File' dropdown, a 'File name' input field, and a file list containing 'data1', 'data2', and 'data3' with sizes 0,26KB, 0,27KB, and 0,25KB respectively. The main area is titled 'Files.oms' and contains a code editor with the following Scala code:

```
1 val input = Val[File]
2 val inputName = Val[String]
3 val output = Val[File]
4 val i = Val[Int]
5
6 val pyTask = SystemExecTask(
7   "./python bin",
8   "./python/re-execute.sh python matrix.py /tmp/data.csv ${i} /tmp/out.csv") set (
9   resources += workDirectory / "./python bin",
10  inputFiles += (input, "python/rootfs/tmp/data.csv"),
11  outputFiles += ("python/rootfs/tmp/out.csv", output),
12  inputs += i,
13  outputs += i
14 )
15
16 val copyHook = CopyFileHook(output, workDirectory / "results/${inputName.dropRight(
17
18 val exploration =
19   ExplorationTask(
20     (i in (1 to 10)) x
21     (input in (workDirectory / "../data/").files withName inputName)
22   )
23
24 exploration -> Strain(pyTask) hook copyHook
25
```

A blue 'Play' button is located to the right of the code editor. At the bottom right of the main area, there is a red watermark that says 'Loving Lobster'.

# Delegate to the Grid

The complex-systems propose 5000+ cores of computing for researchers.

Go to: <http://iscpif.fr/vo/> to learn how to access it.

# Delegate to the Grid

## Configure OpenMOLE

The screenshot shows the OpenMOLE web interface at <https://localhost:34887/>. A modal dialog is open for 'Authentications' under the 'File' menu. It displays fields for 'EGI P12 certificate' (selected), 'Password' (redacted), and 'Key file' (set to 'egi.p12'). A 'Save' button is visible in the top right of the modal. In the bottom right corner of the modal, there is a snippet of Scala code:

```
 ${i} /tmp/out.csv") set (
```

Below the modal, a code editor window is open, showing the following Scala code:

```
10 val copyHook = CopyTaskHook(output, workDirectory / "results/${inputName.dropRight(1)}
```

```
17
```

```
18 val exploration =
```

```
19 ExplorationTask(
```

```
20   (i in (1 to 10)) x
```

```
21   (input in (workDirectory / "../data/").files withName inputName)
```

```
22 )
```

```
23
```

```
24 exploration -> Strain(pyTask) hook copyHook
```

```
25
```

A red banner at the bottom right of the slide reads 'Loving Lobster'.

# Delegate to the Grid

Add one line to the script

```
23  
24 val env = EGIEnvironment("vo.complex-systems.eu")  
25  
26 exploration -< (Strain(pyTask) hook copyHook on env)  
27
```

# Delegate to the Grid

It runs!

The screenshot shows a web browser window with the URL <https://localhost:46447>. The page displays the OpenMOLE interface for managing executions. A modal window is open, titled "Executions", showing details for a workflow named "Files.oms" submitted on 30/9/2015, 17:55:19. The workflow is currently "running". The modal also contains a code editor showing the Scala code for the workflow:

```
15 val copyHook = CopyFileHook(output, workDirectory / "results/${inputName.dropRight(1)}")
16
17
18 val exploration =
19   ExplorationTask(
20     (i in (1 to 10)) x
21     (input in (workDirectory / "../data/").files withName inputName)
22   )
23
24 val env = EGIEnvironment("vo.complex-systems.eu")
25
26 exploration -< (Strain(pyTask) hook copyHook on env)
27
```

At the bottom right of the modal, there is a red "Loving Lobster" watermark.