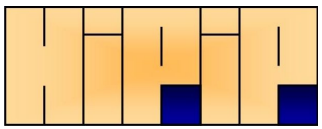
Python in Neuroscience August 29-30 2011
Soizic Laguitton

SOMA workflow

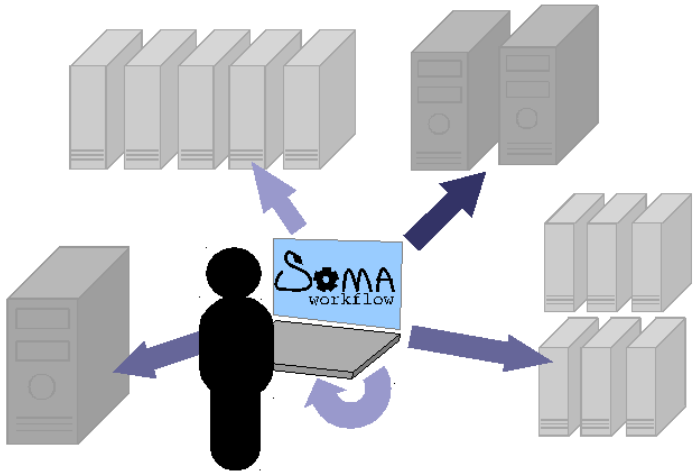
A unified and simple interface
to parallel computing resources



Motivation

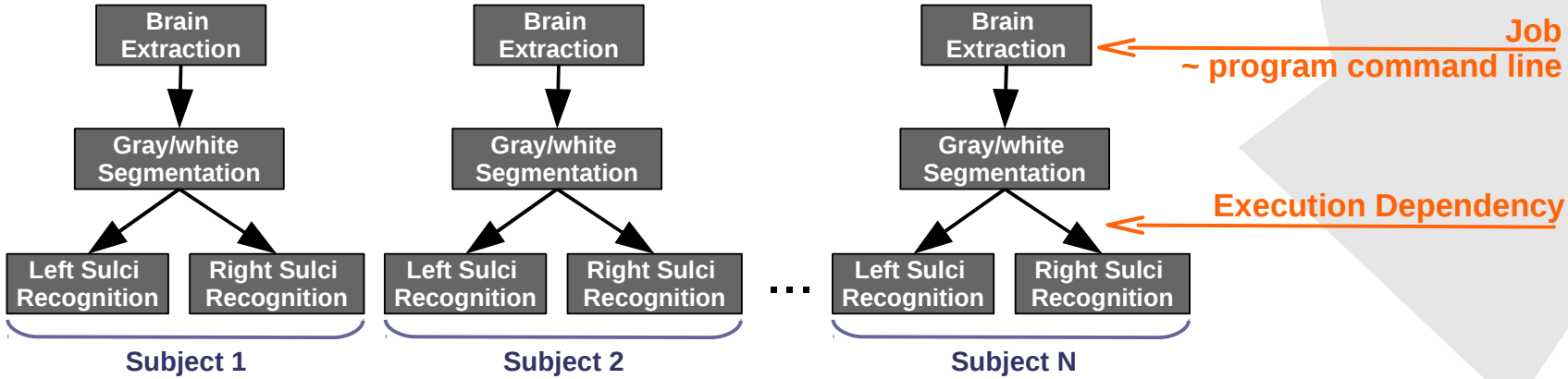
- Parallel computing resources are...
  - highly available
  - valuable asset to accelerate, test, validate and compare methods
- Coarse-grained parallelism is well adapted in many use cases:
  - Many long sub processes (a minute to several days or weeks)
  - No communication between running sub processes

- However, parallel computing resources are...
  - not user friendly
  - heterogeneous
  - (generally) lacking tools dedicated to coarse-grained parallelization



Soma-workflow overview

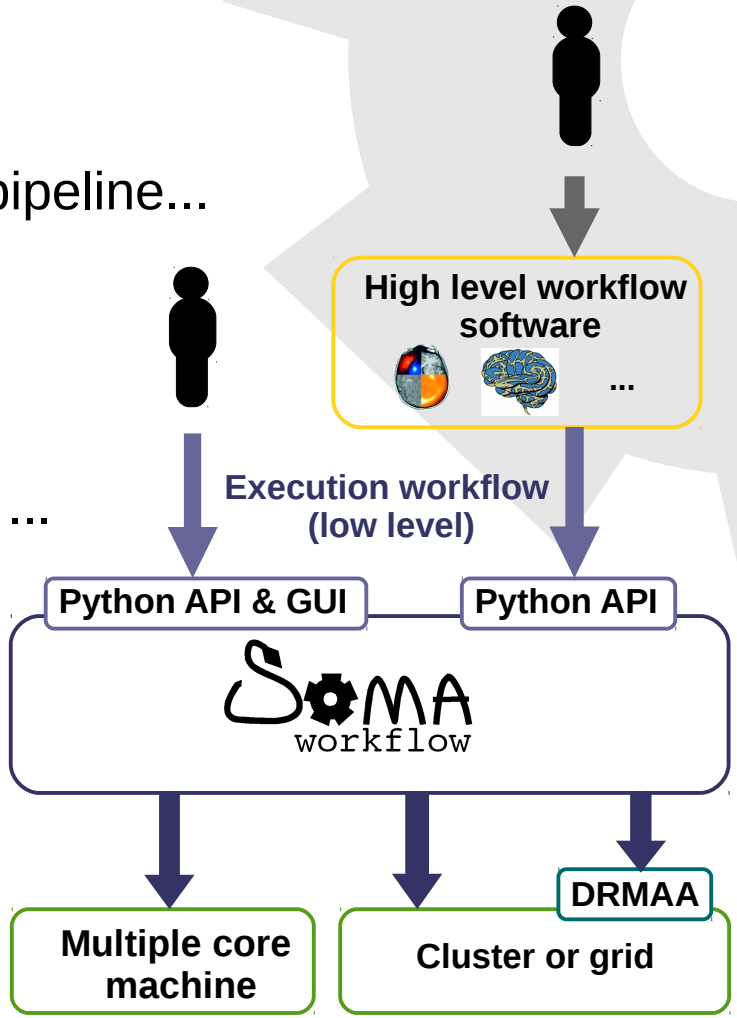
- Coarse-grained parallelized processes => workflows
  - Workflow example:



- Submission, control and monitoring of workflows on different resources:
  - Python API
  - GUI => especially valuable for workflow monitoring
- Remote computing resource:
  - Transparent remote access to computing resource
  - Disconnections
  - Files and file paths management tool

Soma-workflow position

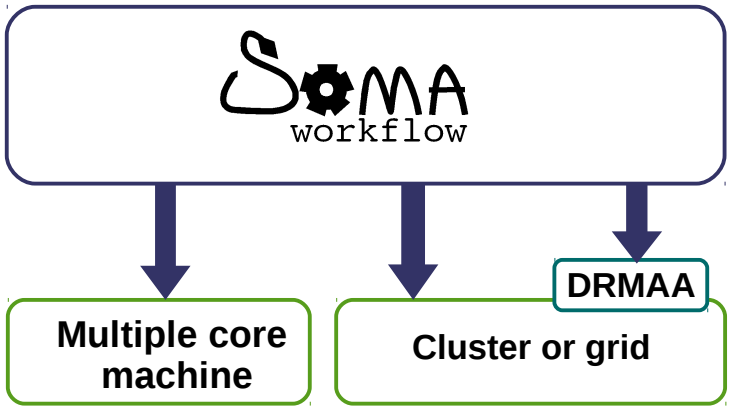
- Soma-workflow deals with execution workflows (low level)
- ≠ from high level workflow software
  - In neuroimaging: BrainVISA, NiPype, LONI pipeline...
  - Higher level description of workflow
  - Higher level features
- Bridges the gap between parallel resources and ...
  - Non expert user
    - Documentation
    - Python API made to be simple
    - GUI
  - High level workflow software



Interface with computing resources

- Uses only very basic computing resource functionalities
  - ➔ Compatibility with a wide range of resources
- Creation of an interface with a new resource
  - ➔ Implementation of 4 Python methods
- Built in interfaces:
  - Ready to use on multiple core machines
  - Interface with the Distributed Resource Management Application API (DRMAA)
    - Software standard developed by the Open Grid Forum
    - S-w was used with success on clusters with the systems:
      - Oracle Grid Engine
      - Torque
      - LSF
      - Condor

- Job submission
- Job suppression
- Job status
- Job exit information



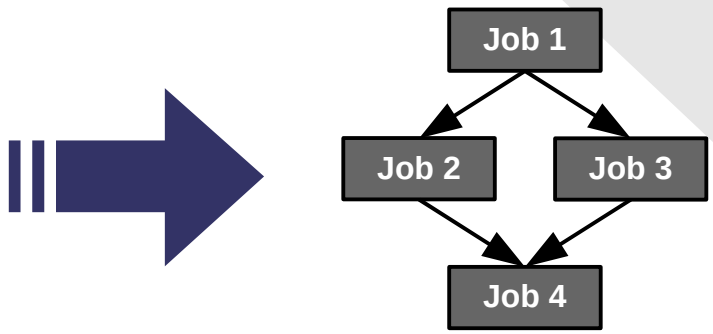
API overview

```python
from soma.workflow.client import Job, Workflow
from soma.workflow.client import WorkflowController

# create the workflow:
job_1 = Job(command=["sleep", "60"], name="job 1")
job_2 = Job(command=["sleep", "60"], name="job 2")
job_3 = Job(command=["sleep", "60"], name="job 3")
job_4 = Job(command=["sleep", "60"], name="job 4")

jobs = [job_1, job_2, job_3, job_4]
dependencies = [(job_1, job_2),
                (job_1, job_3),
                (job_2, job_4),
                (job_3, job_4)]
workflow = Workflow(jobs=jobs,
                    dependencies=dependencies)

# submit the workflow:
controller = WorkflowController("Resource_name",
                                login,
                                password)
controller.submit_workflow(workflow=workflow,
                           name="simple example")
```



GUI overview

List of the configured resources

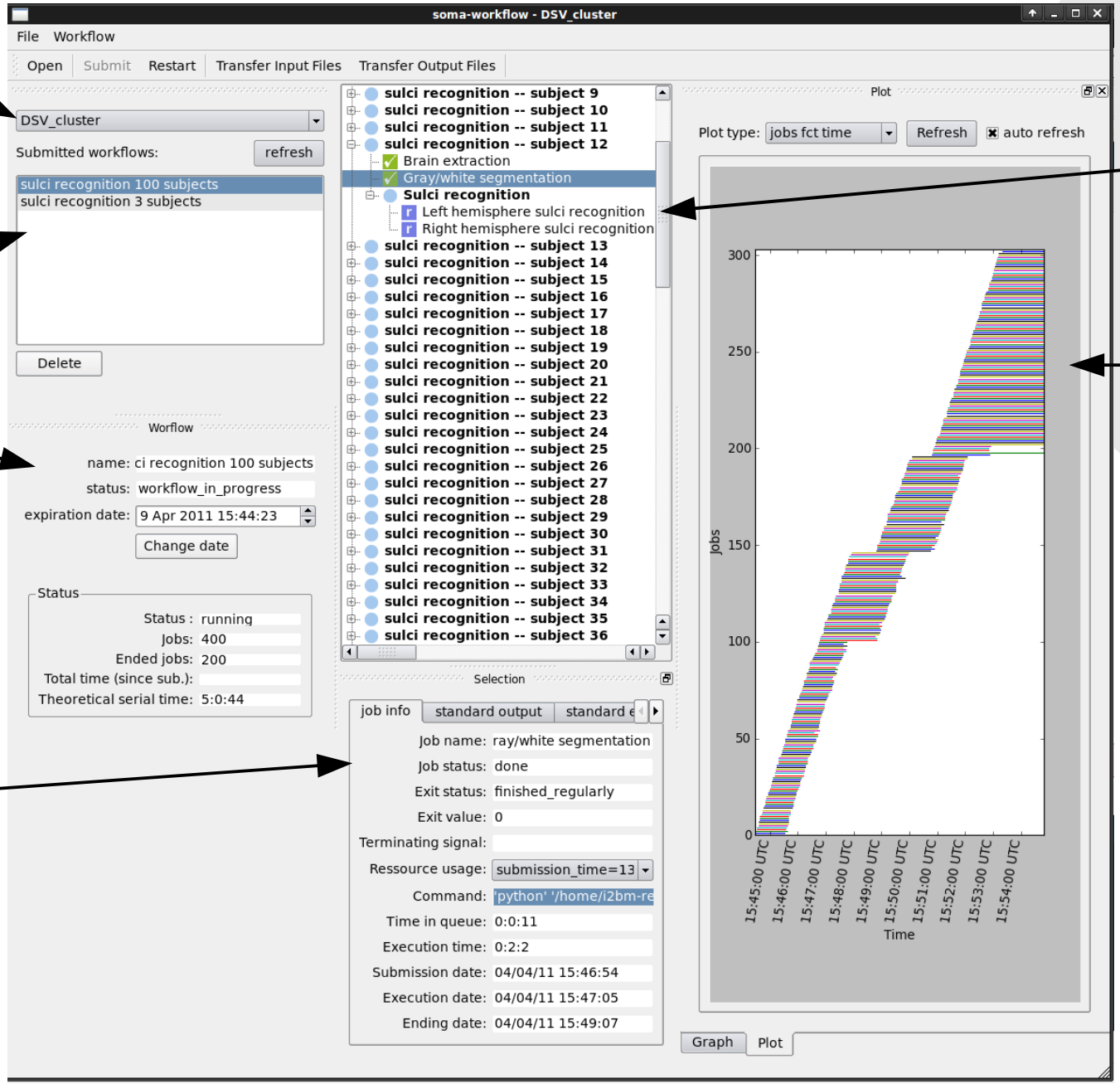
Submitted workflows

Current workflow information

Current selection Information (here a ended job)

Representation of the workflow as a tree

workflow execution plots

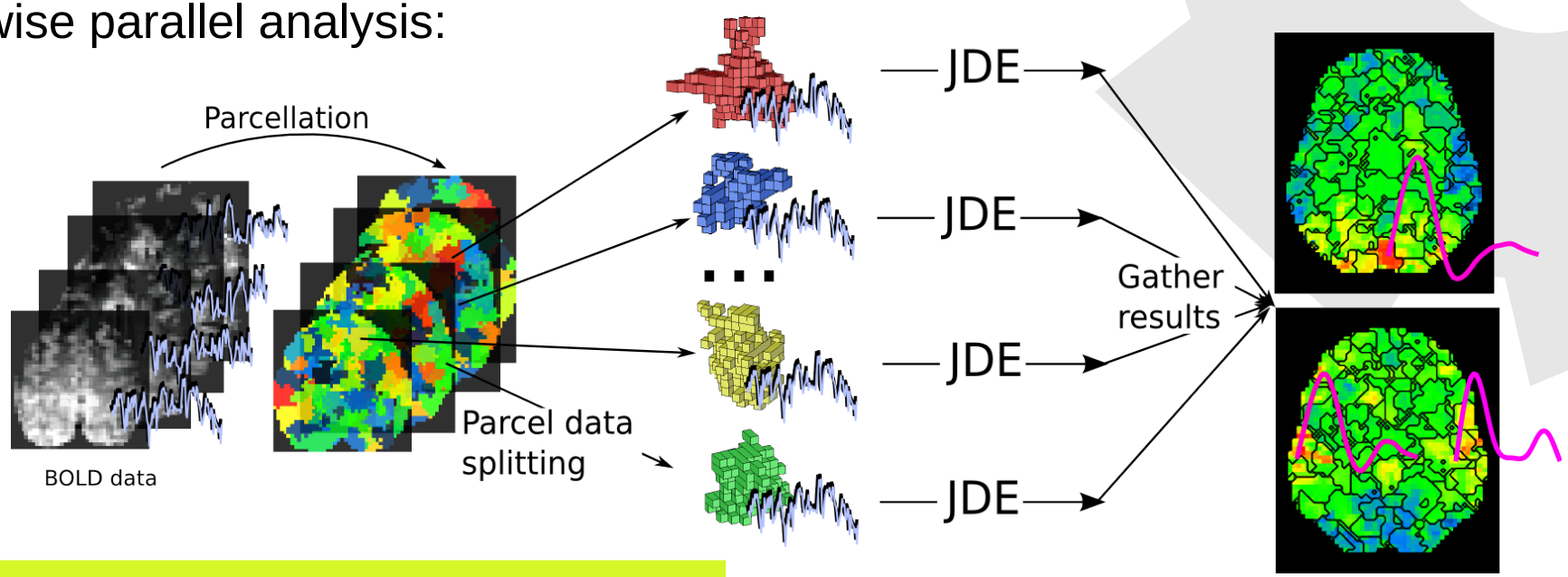


Concrete use cases in neuroimaging

- Computing resource:
  - cluster with 192 CPU
  - Shared with a group of user
- Three concrete use cases => 3 steps in the research process:
  - Acceleration of a single data analysis
  - Robustness improvement with regression tests
  - Extensive validation

Case 1: Acceleration of a single data analysis

- Functional neuroimaging application: Joint detection-estimation (JDE)[1]
  - Detection of the parts of the brain which are involved in a given stimulus
  - Estimation of the Hemodynamic Response Function (HRF)
- Parcel-wise parallel analysis:

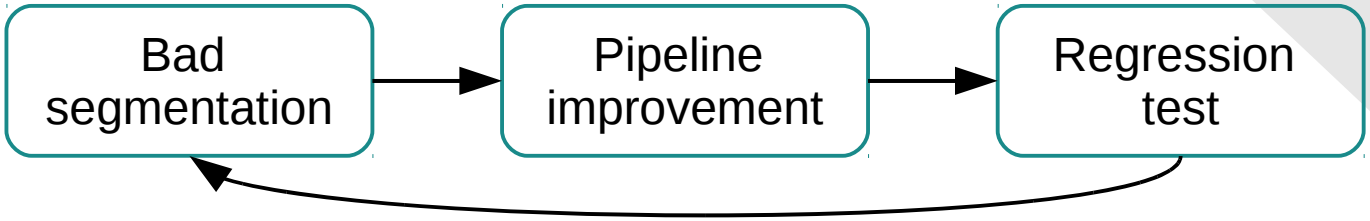


➔ Wrapping of existing code in a workflow
➔ A whole brain processing:
  - 10 hours on 1 CPU
  - 15 mins on the cluster
➔ A group study of 20 subjects: ~ 1 day

[1] Vincent, T., Risser, L., Ciuciu, P. Spatially adaptive mixture modeling for analysis of within-subject fMRI time series IEEE Trans. Med. Imag. 29, 1059–1074 (2010)

Case 2: Robustness improvement with regression tests

- Objective: to reduce the sensitivity of the Morphologist pipeline of BrainVISA [2]
- Morphologist: extraction of the main brain structures from T1 MRI
  - hemispheres, gray/white matter, cortical surface, cortical folds, etc.
- Step by step morphologist was tested on about 1000 T1 MR images.



- Regression test are frequently done on a database made of 80 T1 MRI

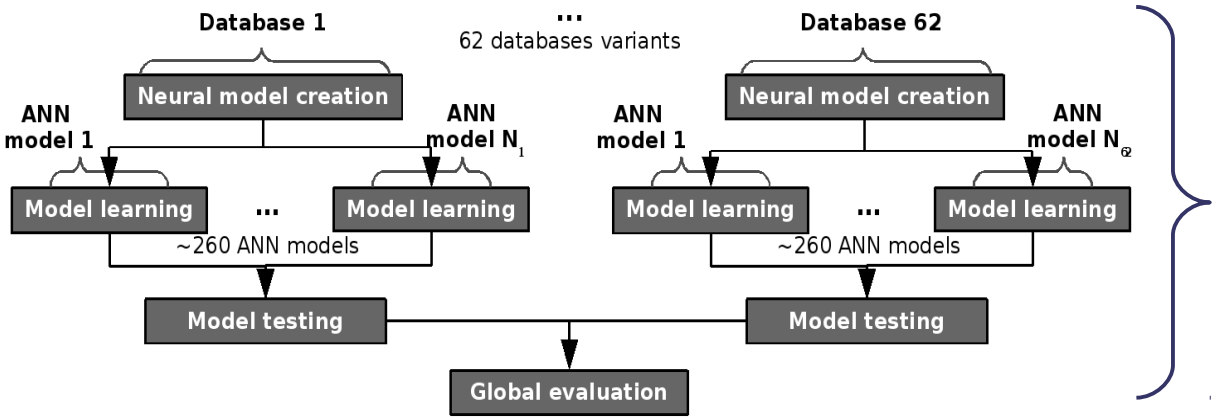
➔ workflows generated by BrainVISA
➔ 23 hours on 1 CPU
➔ ~ 1h on the cluster

→ More frequent regression tests
Tests on larger dataset

[2] Cointepas, Y. The BrainVISA project: a shared software development infrastructure for biomedical imaging research. Proceedings 16th HBM (2010)

Case 3: Extensive validation

- Extensive cross-validation of cortical sulci identification models [3] and comparison with the newer method by Perrot [4]
- Each neural model is trained using a supervised learning scheme, based on a learning database of 62 manually identified brains
- Leave-one-out cross-validation of the models on the learning database



➔ 70000 individual jobs
➔ > 7 months on 1 CPU
➔ ~ 3 days on ~ 100 CPU

[3] Rivière, D., Mangin, J.-F., Papadopoulos-Orfanos, D., Martinez, J.-M, Frouin, V., Régis, J. Automatic recognition of cortical sulci of the Human Brain using a congregation of neural networks. Medical Image Analysis. vol. 6, no. 2, pp. 77–92 (2002)

[4] Perrot, M., Rivière, D., Mangin, J.-F. Cortical sulci recognition and spatial normalization. In: Medical Image Analysis. In press (2011)

Conclusion & Future Work

- Soma-workflow aims at bridging the gap between:
  - Computing resources and researcher (non expert)
  - Computing resources and software
- Dedicated tools for coarse-grained parallelized processes
- Its relevance at different steps in the research process was demonstrated

- Future work:
  - Dynamic workflows
  - Use of soma-workflow on low throughput clusters
  - Job priority to determine order of submission of ready jobs
  - ...

Download, extensive documentation, examples:
http://brainvisa.info/soma-workflow

Remote access to computing resources

- Soma-workflow can be used as a client-server application :
  - Same Python API and same GUI
  - Disconnection at any time
  - The remote communication done with Pyro in a ssh tunnel
- If no shared file system between the user machine and the resource:
  - File path mapping tool
  - File and directory transfer tool
    - Soma-workflow takes into account the state of file transfer when executing a workflow.