

MOTEUR: a data intensive service-based workflow engine

Tristan Glatard, I3S, Rainbow + INRIA, Asclepios teams

Johan Montagnat, PhD, I3S, Rainbow team

Xavier Pennec, PhD, INRIA-Sophia, projet Asclepios

- **Workflow composition: Task-based vs service-based approach**
- **The MOTEUR workflow enactor**
- **Generic WS wrapper**
- **Performance tests**
- **Conclusion**

- **Workflow description**

- Business workflows languages (e.g. BPEL)
 - Control-centric
- Scientific workflows languages (e.g. Scuffl)
 - Data-centric

- **Workflow execution**

- **Task-based** workflows (e.g. DAGMan)
 - Explicit mention of data dependencies
 - Complex workflow, simple optimisation

← CS friendly

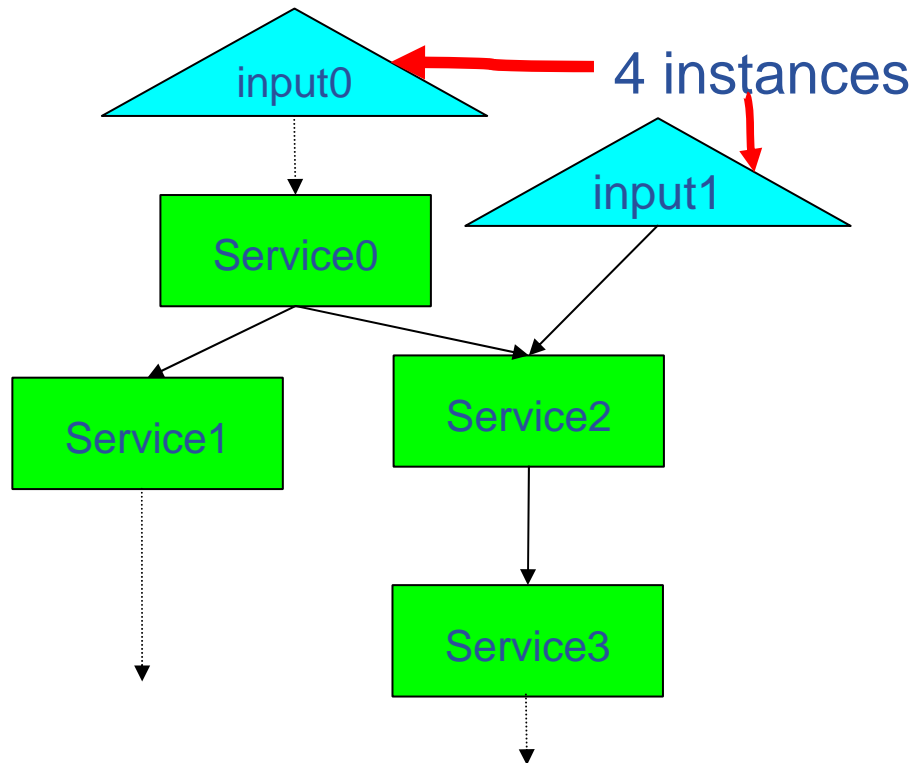
- **Service-based** workflows (e.g. Taverna, Triana, Kepler)

← user friendly

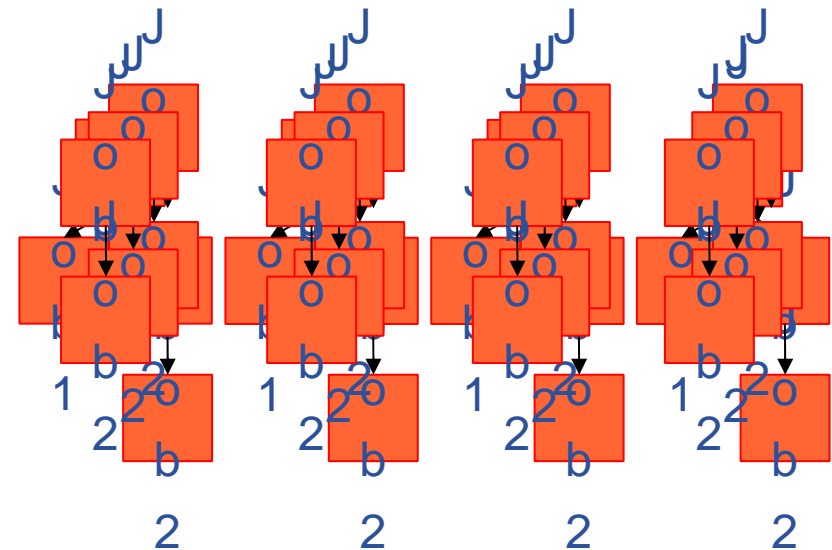
- Independent expression of processors and input data-sets
- Simple workflows, complex optimisation

- Reusable workflow: input data sets to be processed

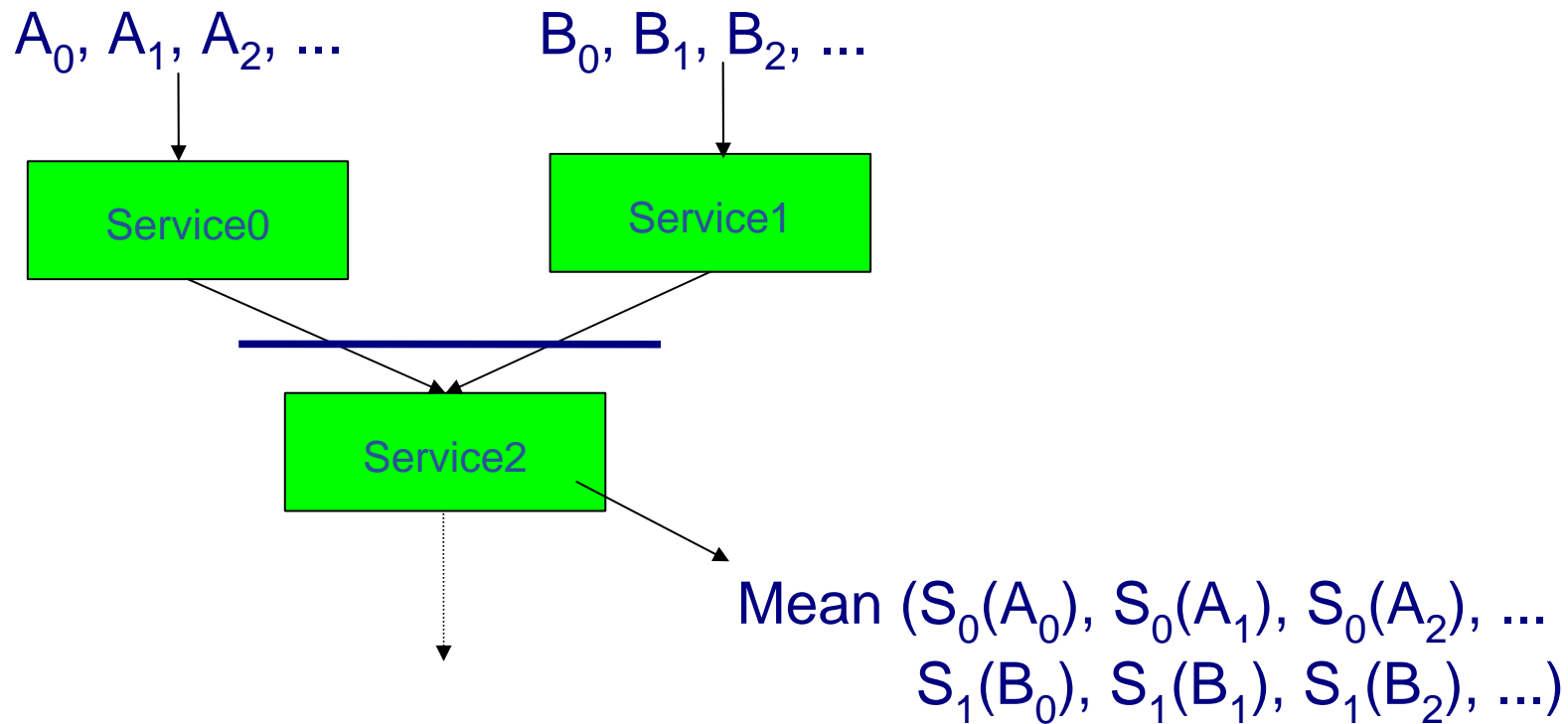
Service-based approach



Task-based approach

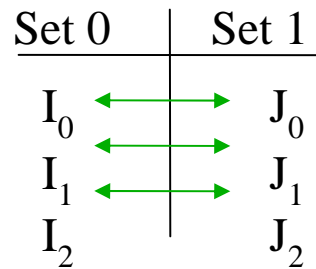


- Data synchronization are difficult to describe with the task-based approach
 - Example: computing a mean on results

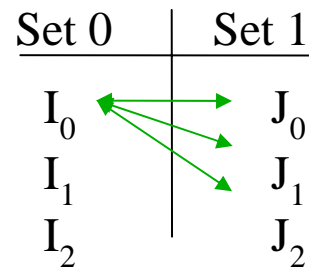


- **Service-based approach allows data iteration strategies**
 - Basic ones (from Taverna community):

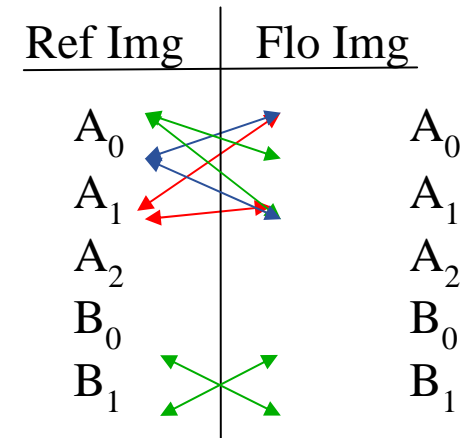
One-to-one



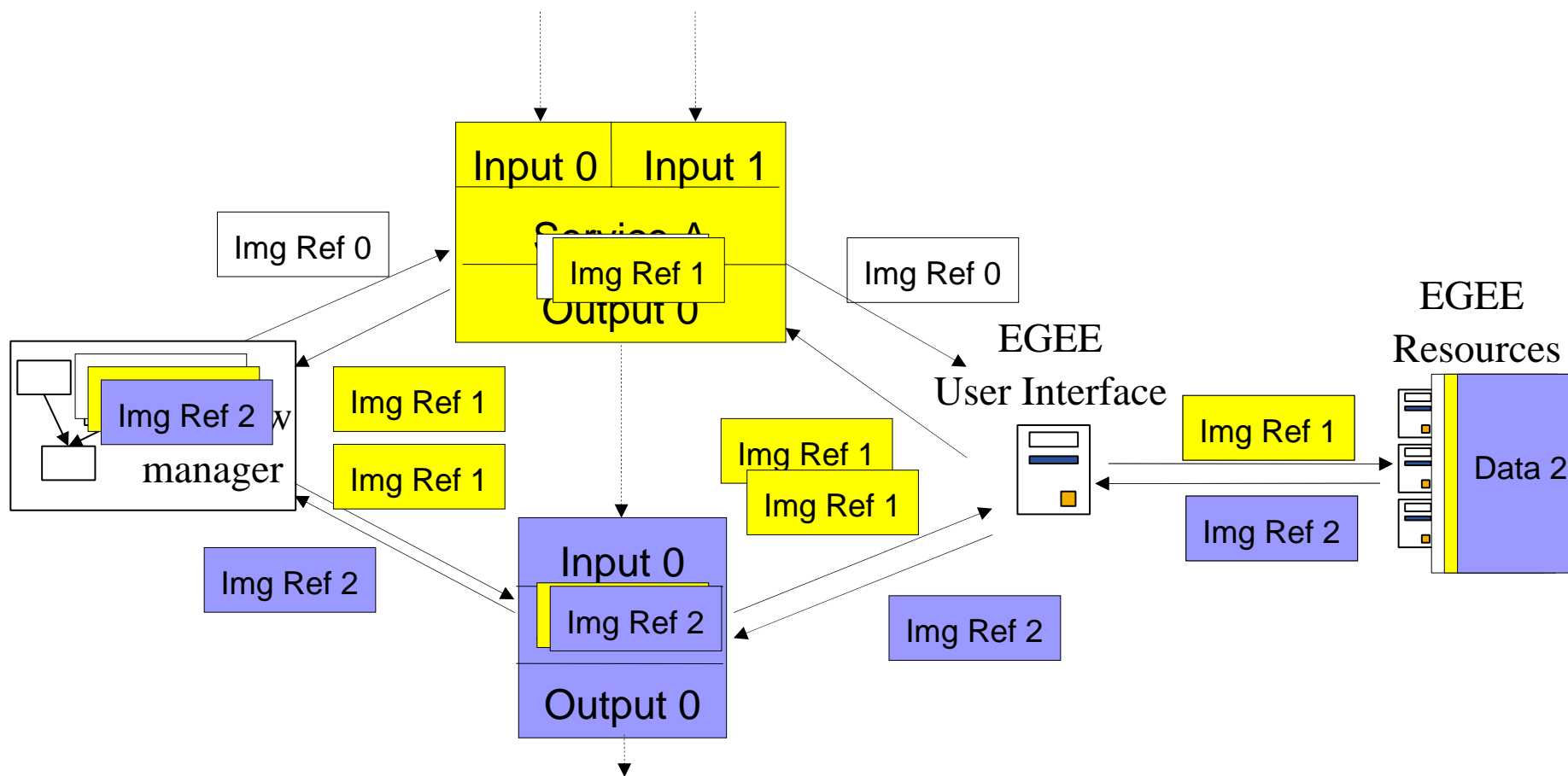
All-to-all



- Composed into more elaborated ones:
 “register all images of the same patient of the same modality at different exam dates”



- In the service-based approach, the workflow manager is isolated from the grid:

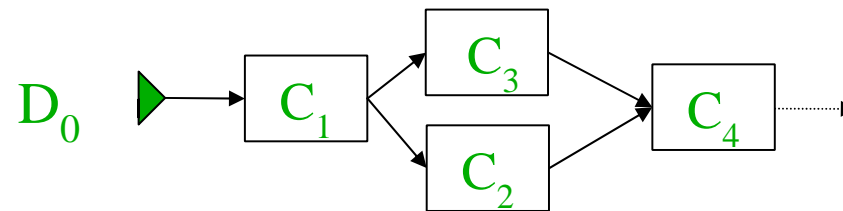


- Optimizing the execution is more difficult

MOTEUR

Optimizing service-based workflows on the grid

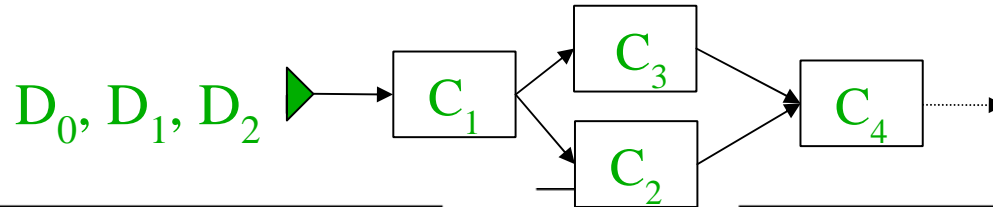
- **Intrinsic parallelism**
 - C_2 and C_3 run in independent threads



$C_2 \parallel C_3$

- **Implemented by all workflow managers**

- **Data + services parallelism (streaming, pipelining)**



C_1	D_0	D_1	D_2	-	-	-	-
C_2	-	-	-	$C_1 * D_0$	$C_1 * D_1$	$C_1 * D_2$	-
C_3	-	-	-	$C_1 * D_0$	$C_1 * D_1$	$C_1 * D_2$	-
C_4	-	-	-	-	-	-	Mean

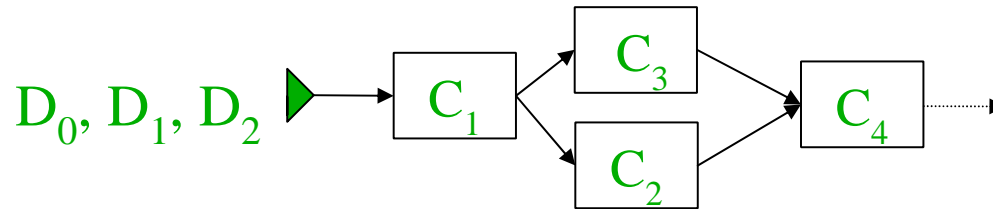
Without service parallelism

C_1	D_0	D_1	D_2	-	-
C_2	-	$C_1 * D_0$	$C_1 * D_1$	$C_1 * D_2$	-
C_3	-	$C_1 * D_0$	$C_1 * D_1$	$C_1 * D_2$	-
C_4	-	-	-	-	Mean

With service parallelism

- **Impact on the total execution time raises when:**
 - Data segments are numerous
 - Processing times are not constant
- **Not implemented in Taverna**

- Parallel processing of data:



C ₁	D ₀	D ₁	D ₂						
C ₂				D ₀	D ₁	D ₂			
C ₃				D ₀	D ₁	D ₂			
C ₄							D ₀	D ₁	D ₂

Without data parallelism

C ₁	D ₀	D ₁	D ₂			
C ₂			D ₀	D ₁	D ₂	
C ₃			D ₀	D ₁	D ₂	
C ₄				D ₀	D ₁	D ₂

With data parallelism

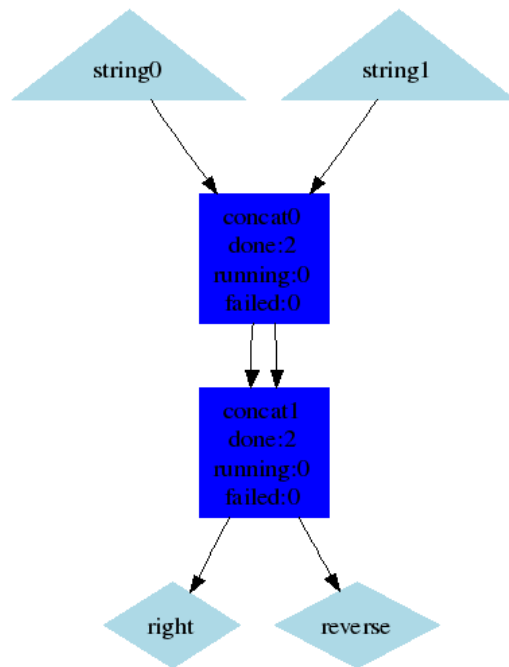
- Consequences:

- Change the order of data segments
- When coupled with service parallelism, reordering data is difficult

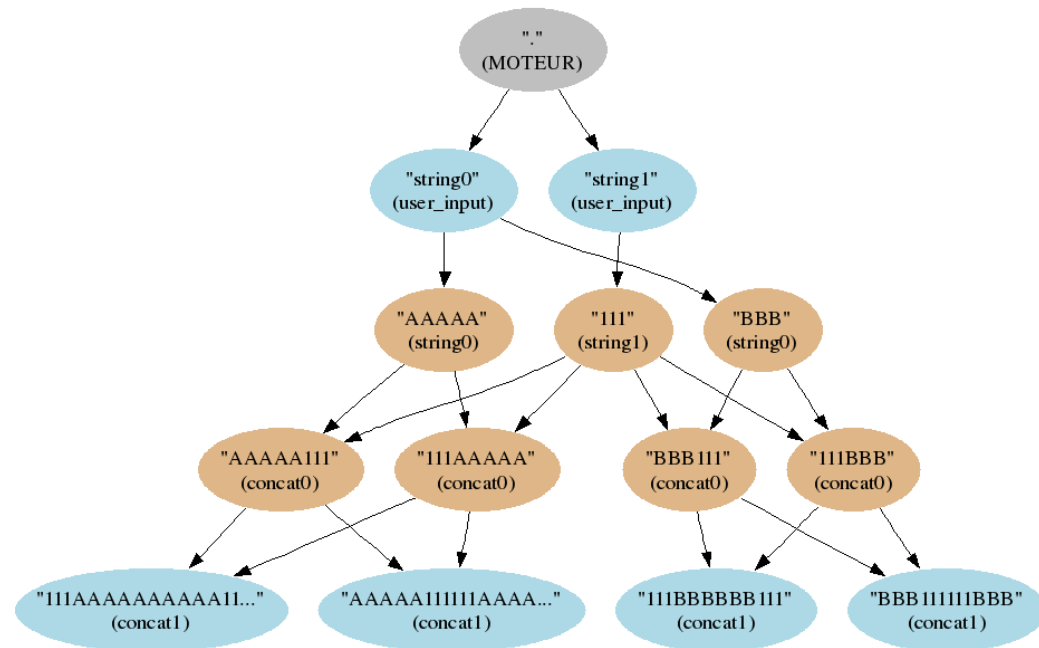
- Not implemented in Kepler

- In MOTEUR, data segments are stored within a tree:

Services representation

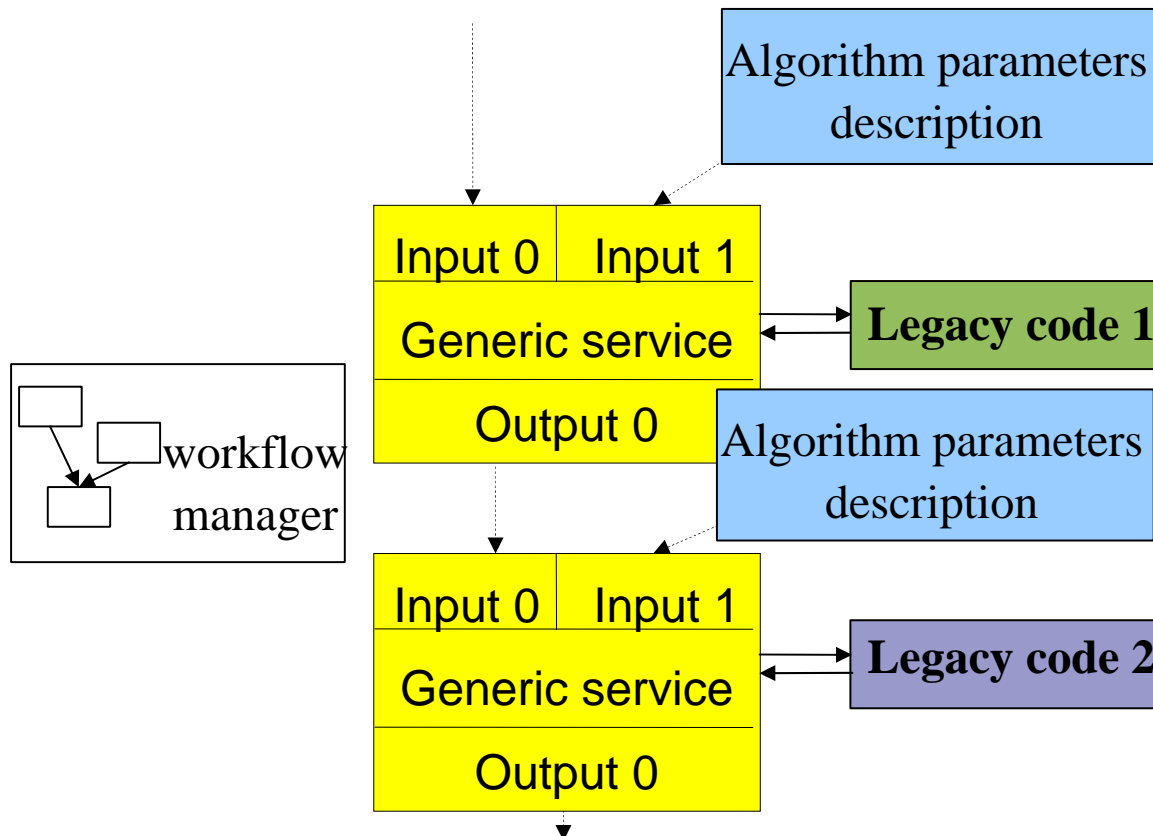


Data representation



- This data representation allows to:
 - Retrieve results provenance
 - Handle *one-to-one* iterations strategies if data segments are puzzled

- Problem: wrapping binaries into WS is time consuming
- Usage of a generic Web service



```

<description>
  <executable name="CrestLines.pl">
    <access type="URL">
      <path value="http://colors.unice.fr:80/">
    </access>
    <value value="CrestLines.pl"/>
    <input name="image" option="-im1">
      <access type="LFN" />
    </input>
    <input name="scale" option="-s"/>
    <output name="crest_lines" option="-c2">
      <access type="LFN" />
    </output>
    <sandbox name="convert8bits">
      <access type="URL">
        <path
value="http://colors.unice.fr:80/">
        </access>
        <value value="Convert8bits.pl"/>
      </sandbox>
    </executable>
  </description>
  
```

- **Workflow language**
 - Scufi
- **Services**
 - Web Services
 - GridRPC (DIET middleware)
- **Infrastructures**



> 1000 procs

OAR batch submitter

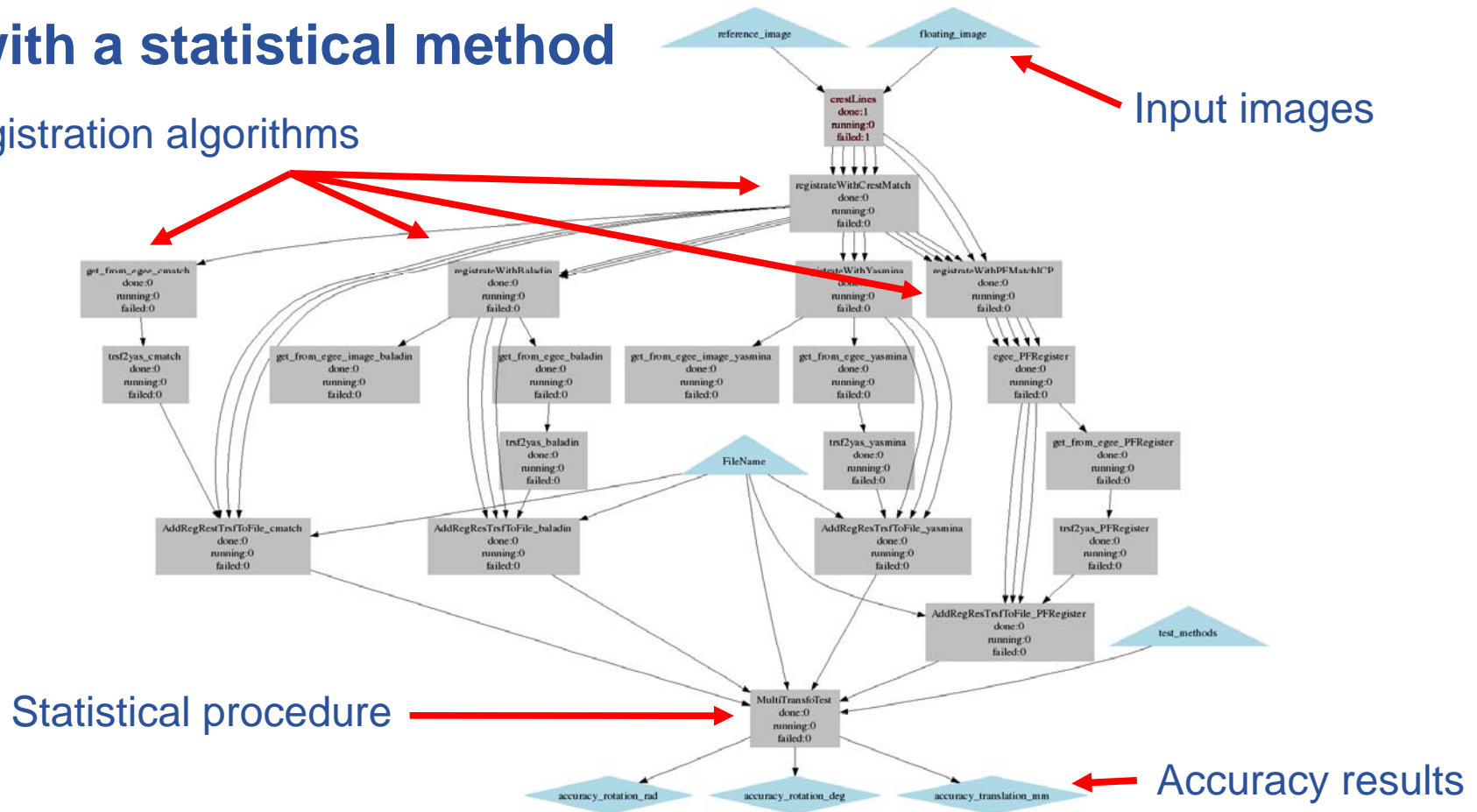
research infrastructure

LCG2 middleware

gLite

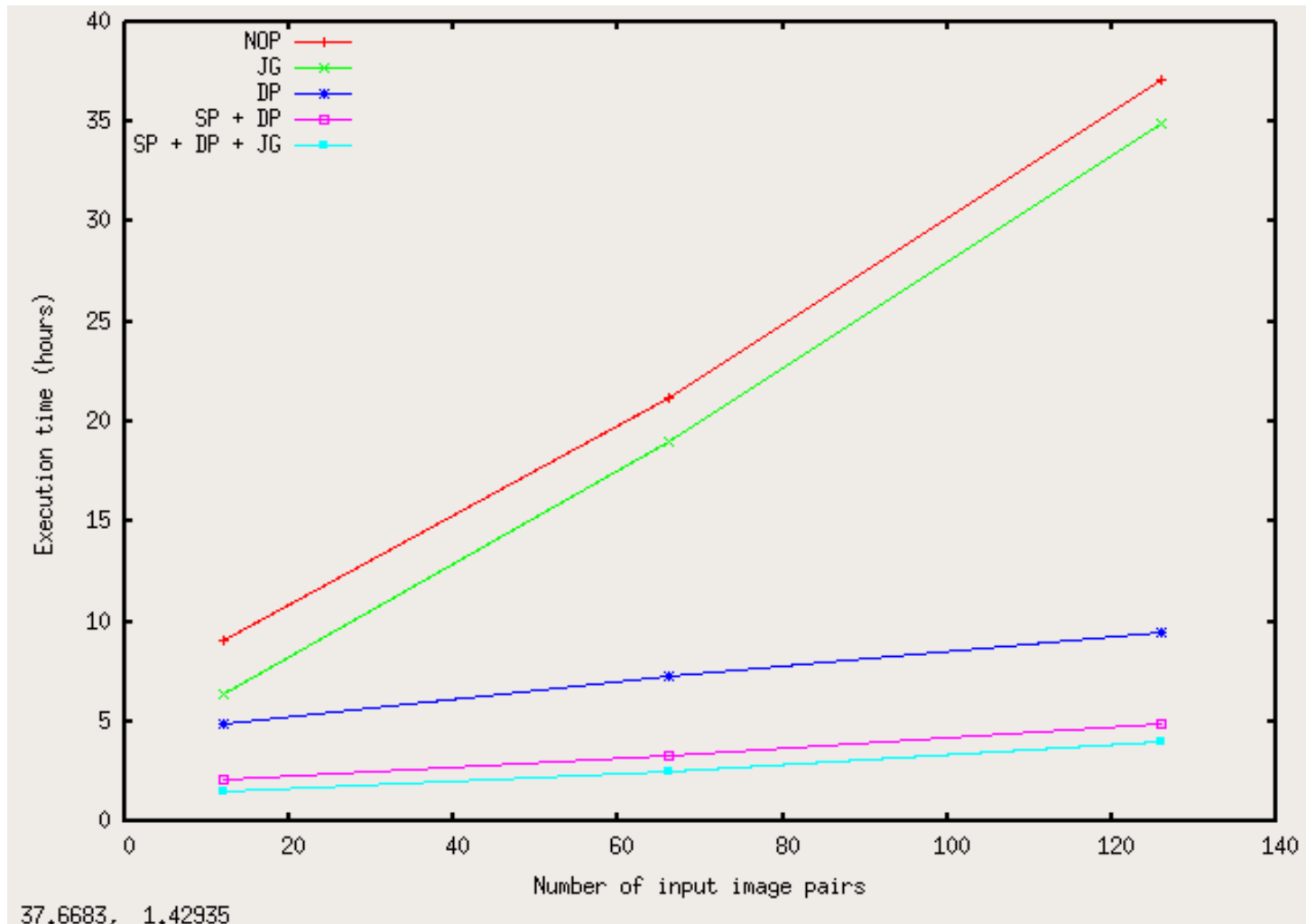
- Evaluation of the accuracy of registration algorithms with a statistical method

Registration algorithms



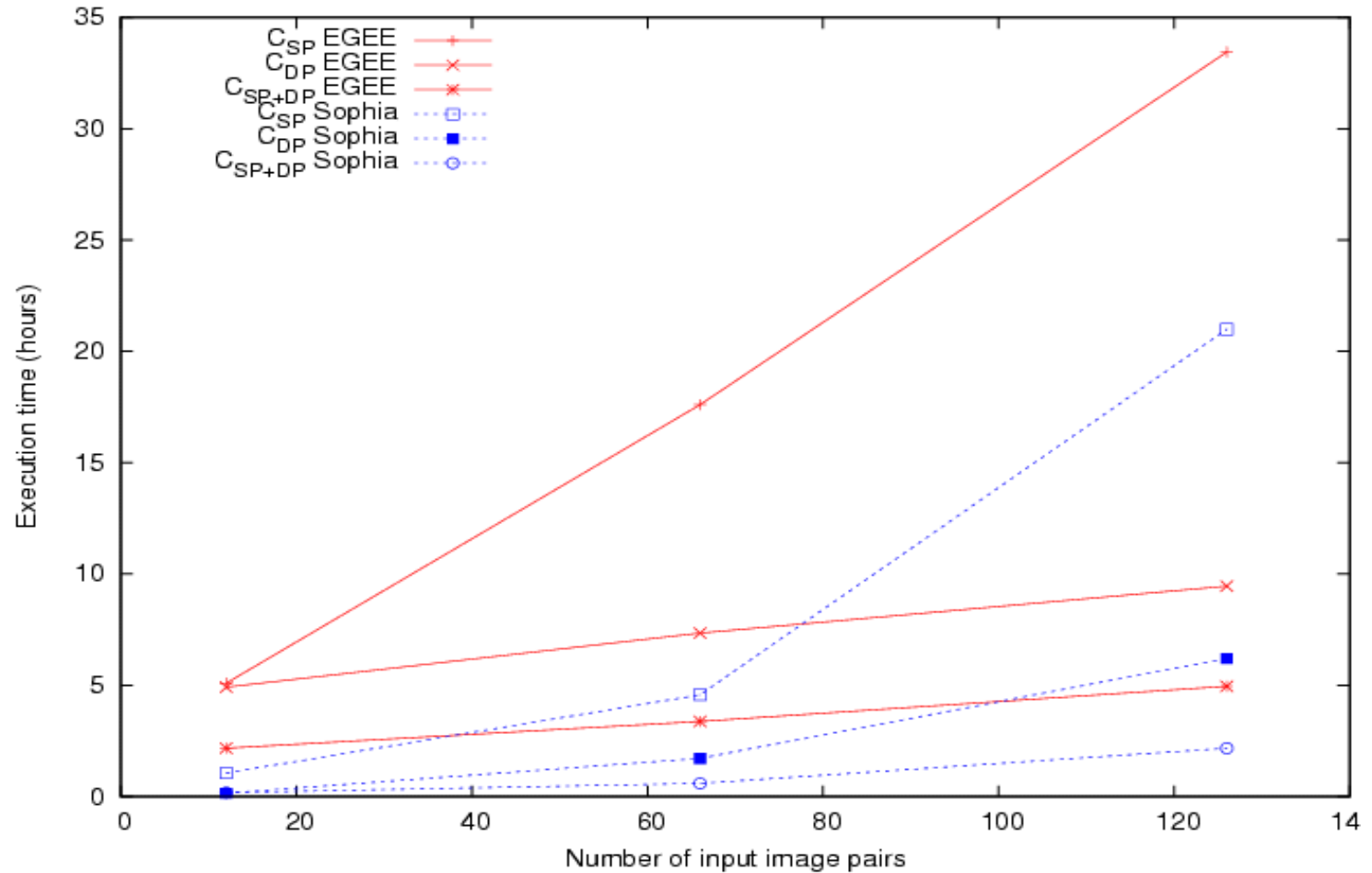
- Demo tonight at 18:30

- Impact of parallelisms:



- EGEE production infrastructure VS Grid5000 Sophia

cluster:



- **Source distribution of MOTEUR available at**
<http://www.i3s.unice.fr/~glatard/software.html>
- ***Demo tonight 18h30 -> 19h30***
- ***Collaboration with P-Grade portal for combining task-based and service-based approaches***



Enabling Grids for E-science