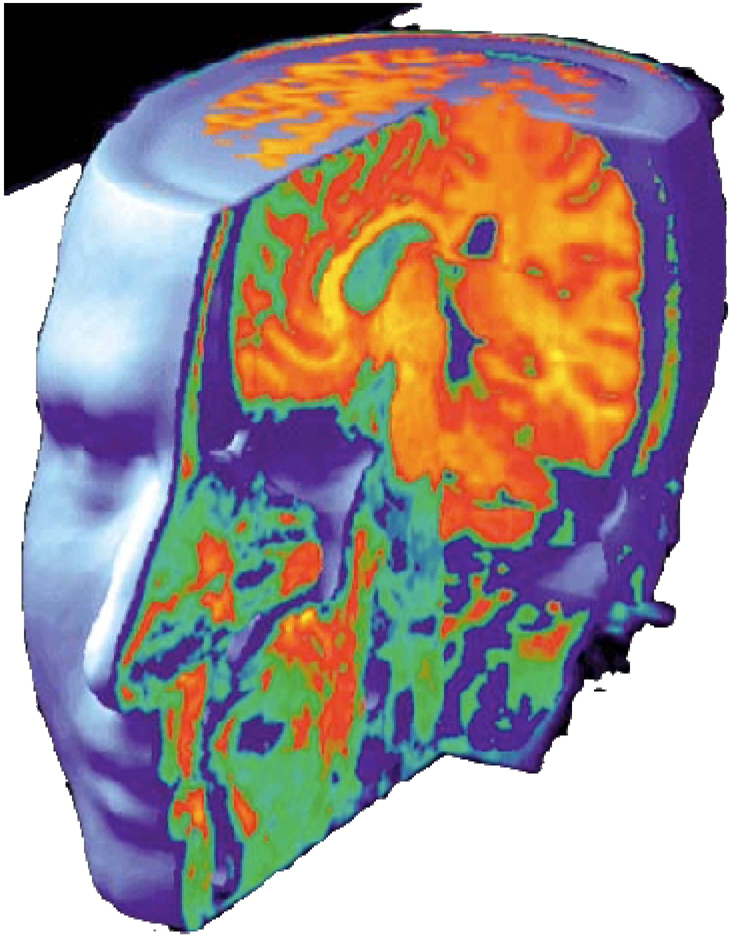
Workflows Description, Enactment and Monitoring in SAGA

Ashiq Anjum, UWE Bristol
Shantenu Jha, LSU

neuGrid

- Recent progress in neuroimaging techniques and data formats has led to an explosive growth in neuroimaging data
- Analysis of this data can facilitate research in neuro-degenerative diseases.



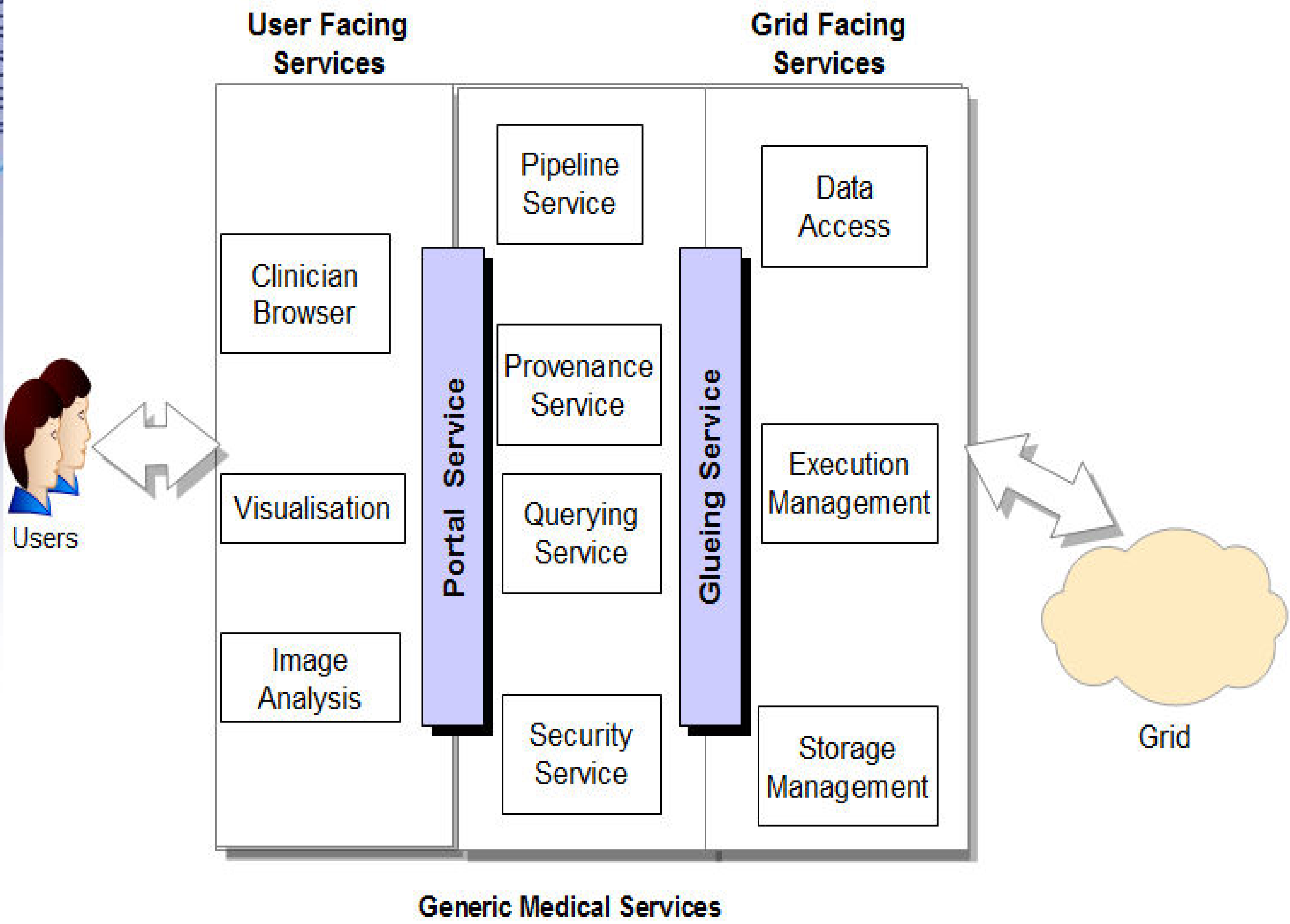
Commercial Partners

Academic Partners

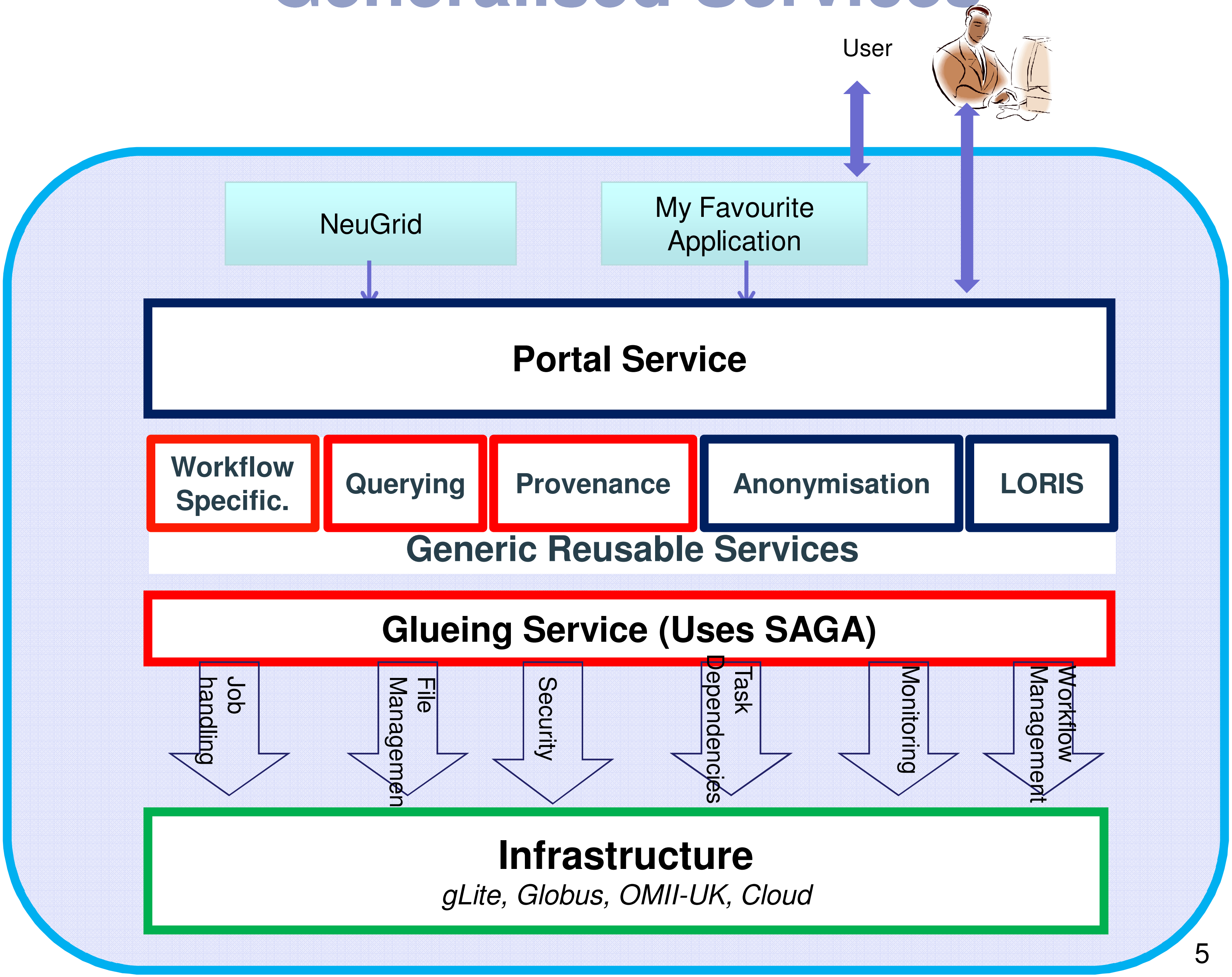
Clinical Users

http://www.neugrid.eu

Services in neuGRID



Generalised Services



User

NeuGrid

My Favourite Application

Portal Service

Workflow Specific.

Querying

Provenance

Anonymisation

LORIS

Generic Reusable Services

Glueing Service (Uses SAGA)

Job handling

File Managemen

Security

Task Dependencies

Monitoring

Workflow Management

Infrastructure

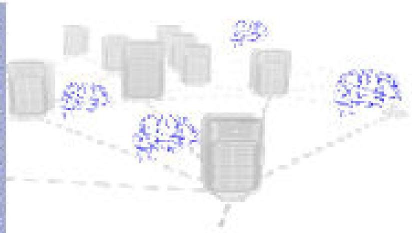
gLite, Globus, OMII-UK, Cloud

Neuroimaging datasets are generally processed through Neuroimaging pipelines

CIVET produces 1100% more data than it consumes, and intermediate data usage is more than 4000%.

Without optimisation runtime of a single workflow on single image is around 8 hrs

10,000 brain images in neuGrid by the end of this year, each Image between 70 to 120 MB

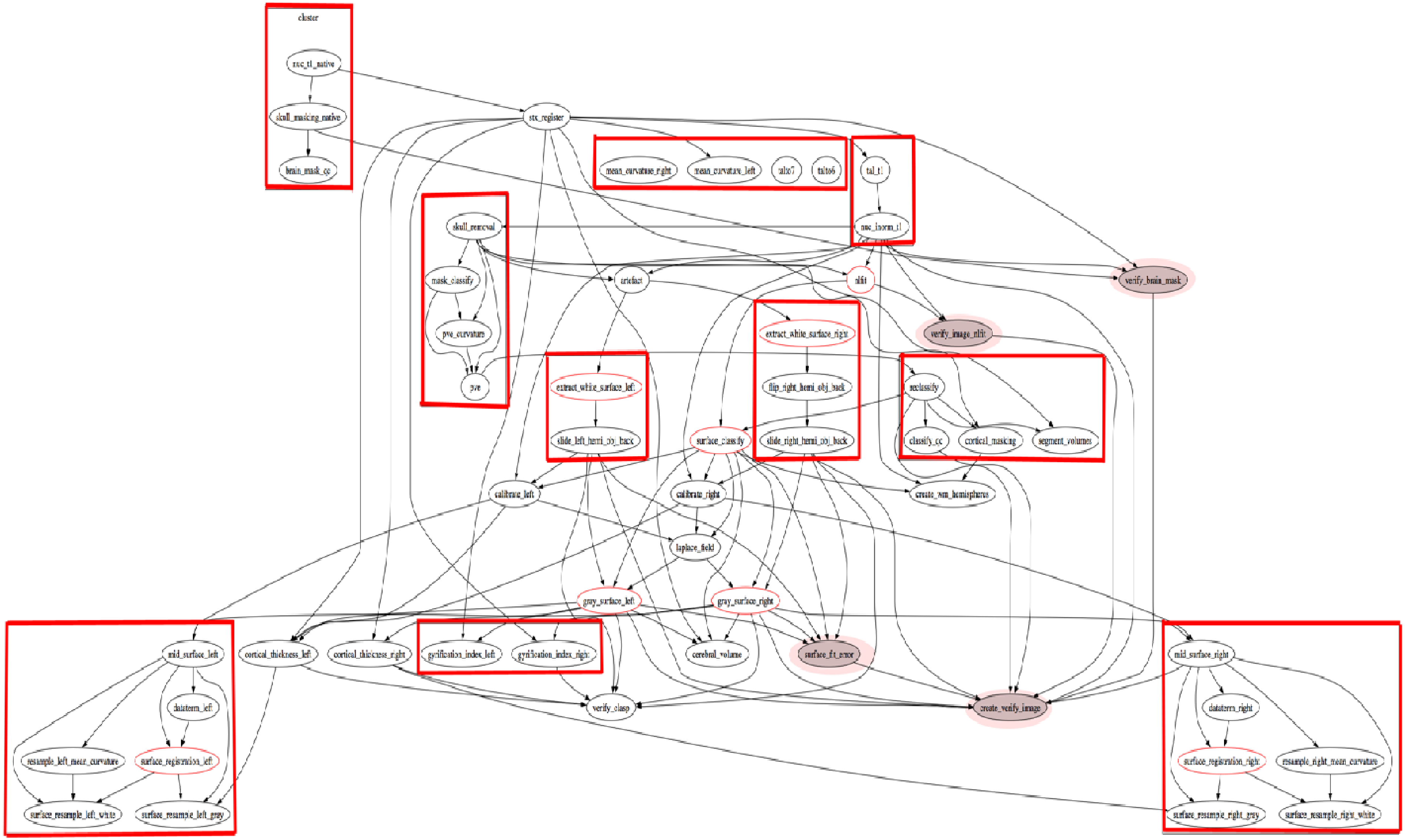


Data Challenge
Facts & Figures

Expected Results

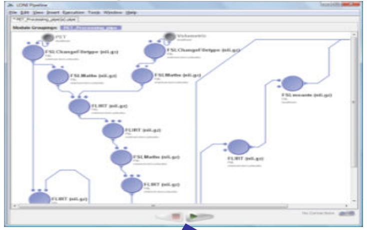
Experiment duration on the Grid		**2 Weeks**
Experiment duration on single computer		> 5 Years
Analyzed data	Patients MR Scans Voxels	715 6'235 2 X 10(exp13)
Hours of total pipeline processing		6'300
Total mining operations		286'810
Operations throughput per hour		853
Max # of processing cores in parallel		184
Number of countries involved		4
Volume of data produced		1 TB

A Neuroimaging Workflow

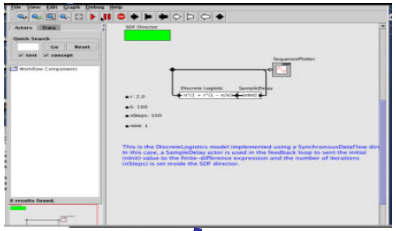


Pipeline Service : Generalisation

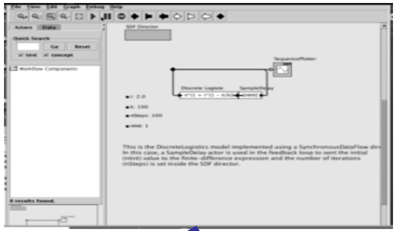
LoNI

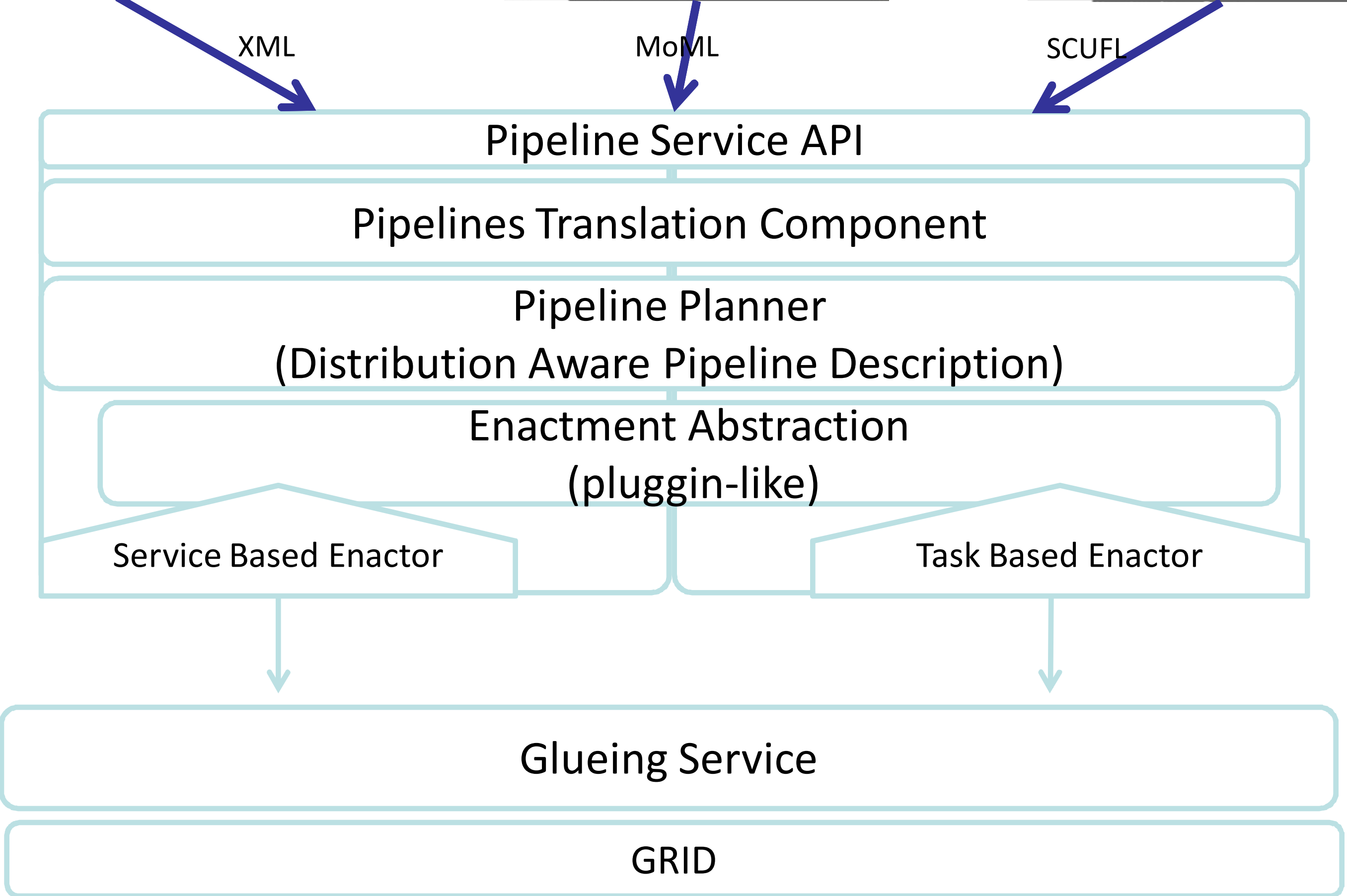


KEPLER



MyFavoriteTool





XML

MoML

SCUFL

Pipeline Service API

Pipelines Translation Component

Pipeline Planner
(Distribution Aware Pipeline Description)

Enactment Abstraction
(pluggin-like)

Service Based Enactor

Task Based Enactor

Glueing Service

GRID

Pipeline Service : Overview

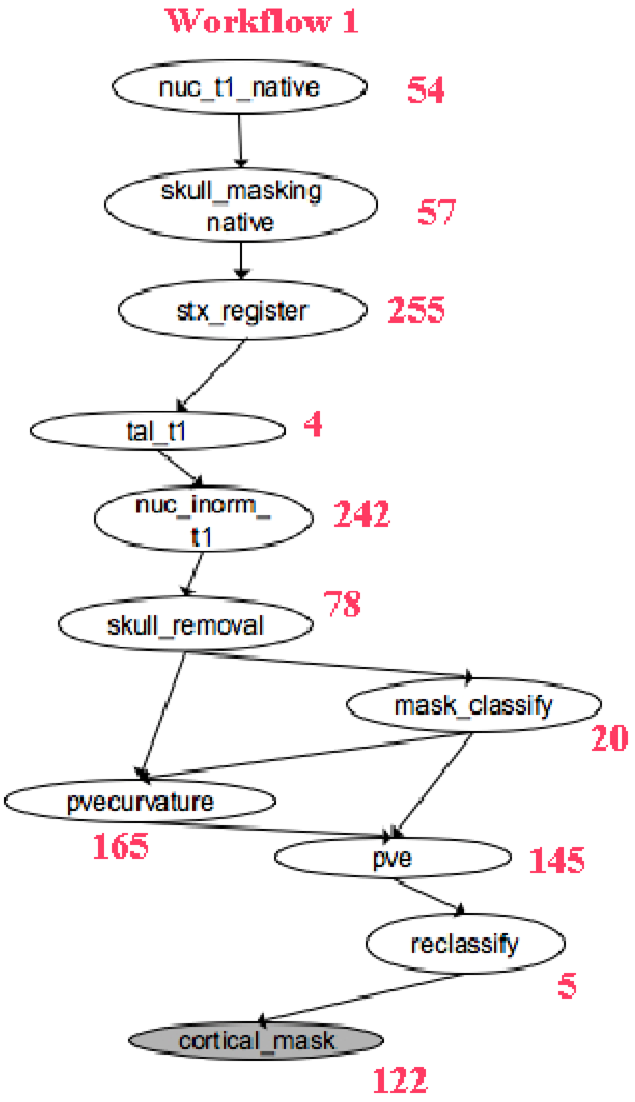
- Designed to provide the required functionality to author, transform and plan workflows

- And orchestrate and facilitate the retrieval of analysis data and intermediary output for Provenance capture.

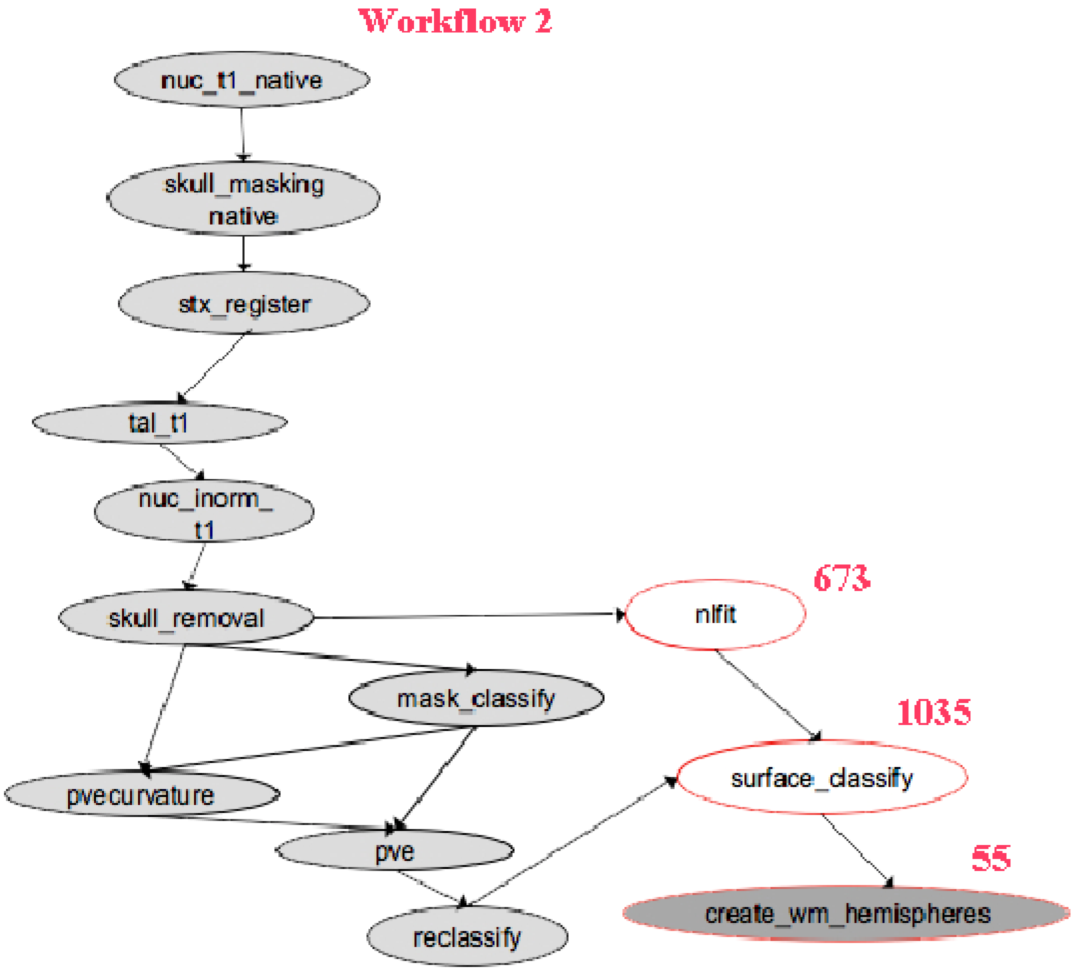
- The Pipeline Service specifies workflows and retrieves the output via the Glueing Service.

Workflow Planning Approaches

- Approaches for workflow planning include:
  - Data-based Methods: Data elimination
  - Task-based Approaches: Task Clustering
  - Experimental evaluations concentrate on automated task clustering.
- Two types of clustering
  - **Automated Horizontal Clustering**
  - *Collapse Factor Based*
  - *Bundle Factor Based*
  - **User defined clustering**



Total Runtime: 1147 secs



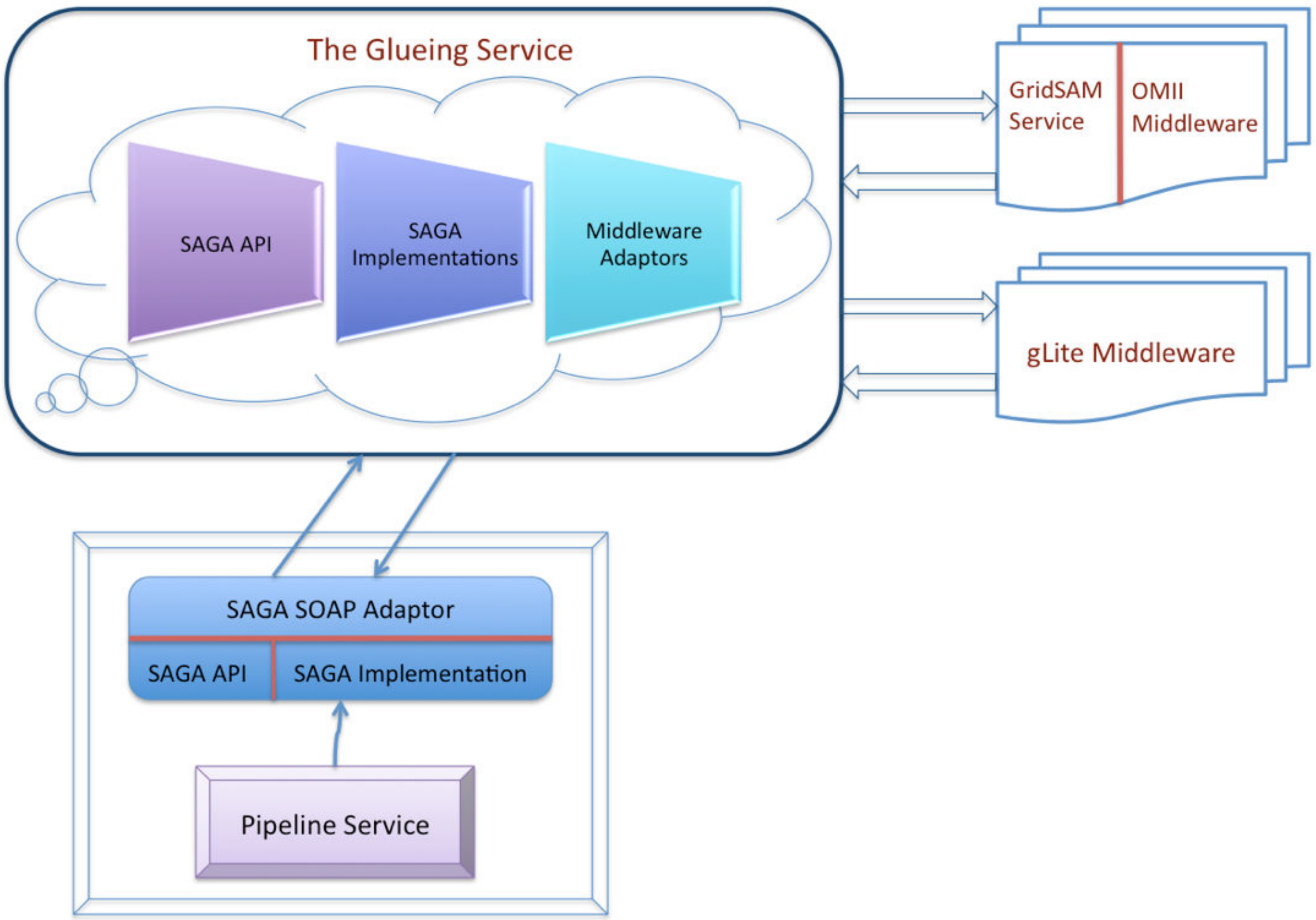
Total Runtime(without optimisation): 2788 secs

Total Runtime (with optimisation): 1763 secs

Reduction in Runtime: 36.7%

Runtime in secs

Enactment via Glueing Service

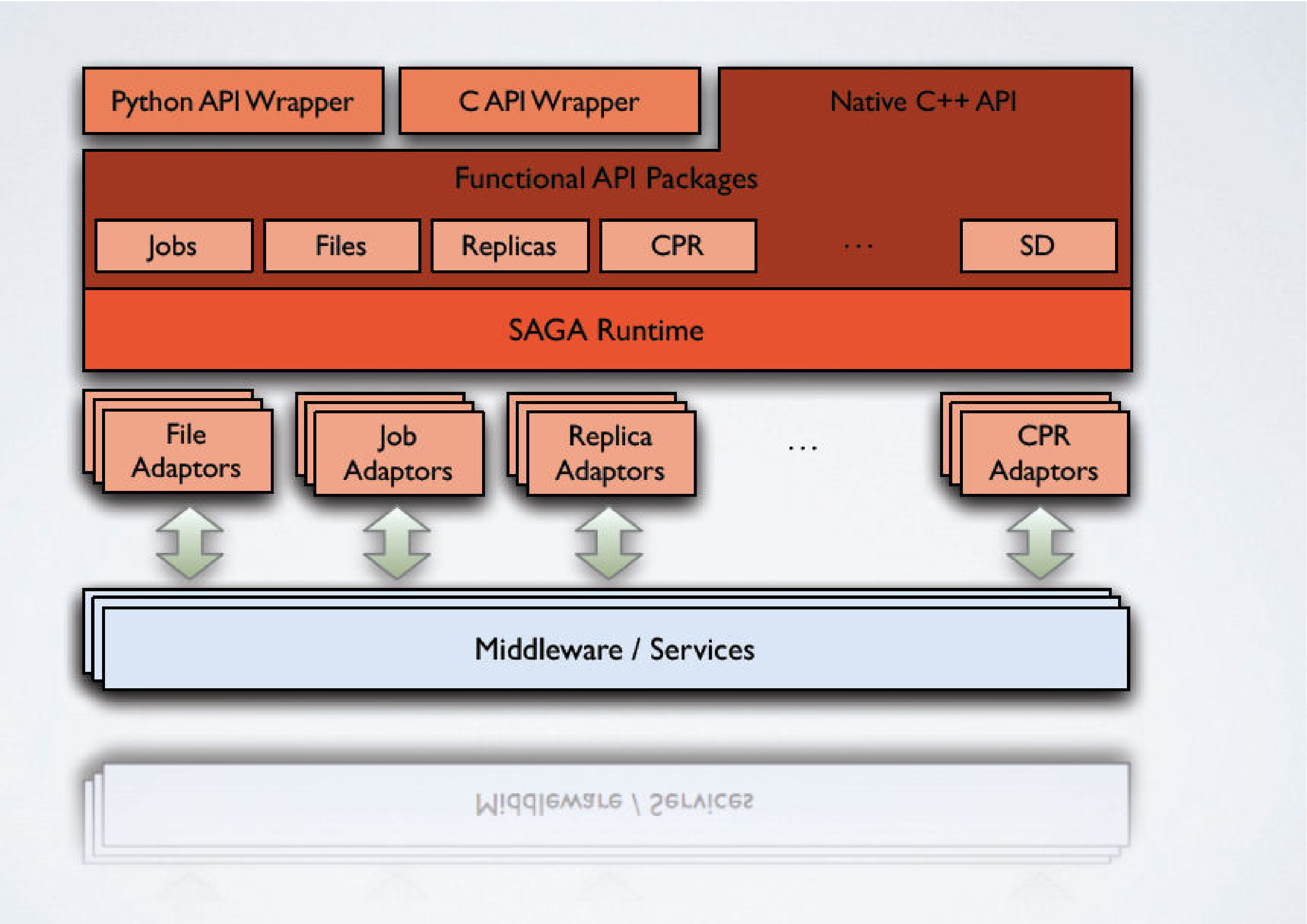


- Uses SAGA to communicate with an underlying infrastructure.
- Able to cater for multiple infrastructures → interoperability.
- Enables flow of data and control to and from the infrastructure (here gLite) for Provenance.

Glueing Service

- Provides file management; workflow submission & monitoring; and provenance retrieval functionality in a generic manner.
- Builds upon SAGA to provide a middleware agnostic way for services and users to interact with the Grid.
- The Glueing Service provides a SOAP wrapper over the OGF SAGA.
- In order to use the Glueing Service in a SAGA compliant manner we have developed the UWESOAP Adaptor.

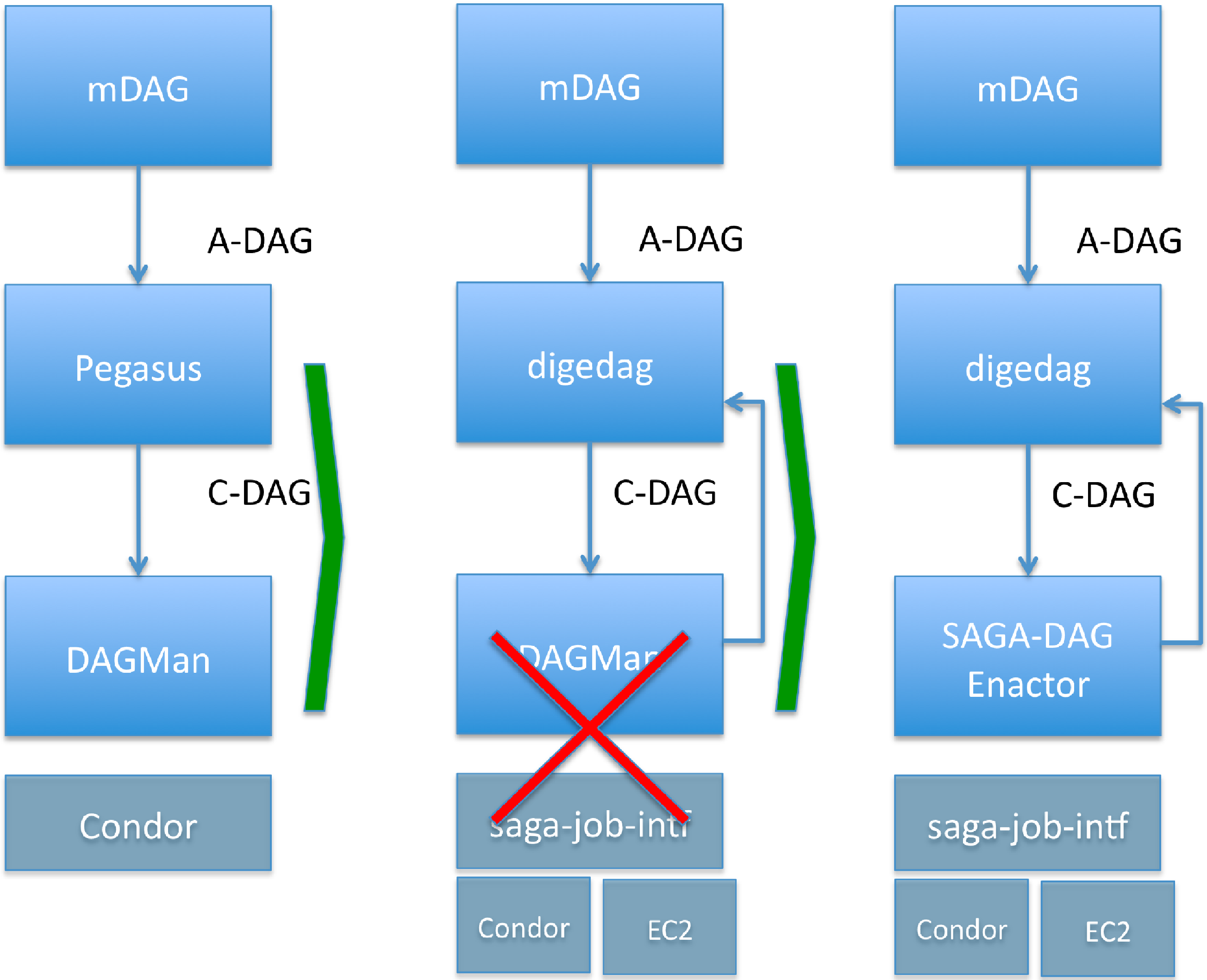
SAGA: In a thousand words..



digedag

- digedag - prototype implementation of a SAGA-based workflow package, with:
  - an API for programatically expressing workflows
  - a parser for (abstract or concrete) workflow descriptions
  - an (in-time workflow) planner
  - a workflow enactor (using the SAGA engine)
    - this will eventually be separated from digedag, but will continue to use SAGA
- Can accept mDAG output, or Pegasus output
- Can move back and forth between abstract and concrete DAG

DAG-based Workflow Applications: Extensibility Approach



Application Development Phase

Generation & Exec. Planning Phase

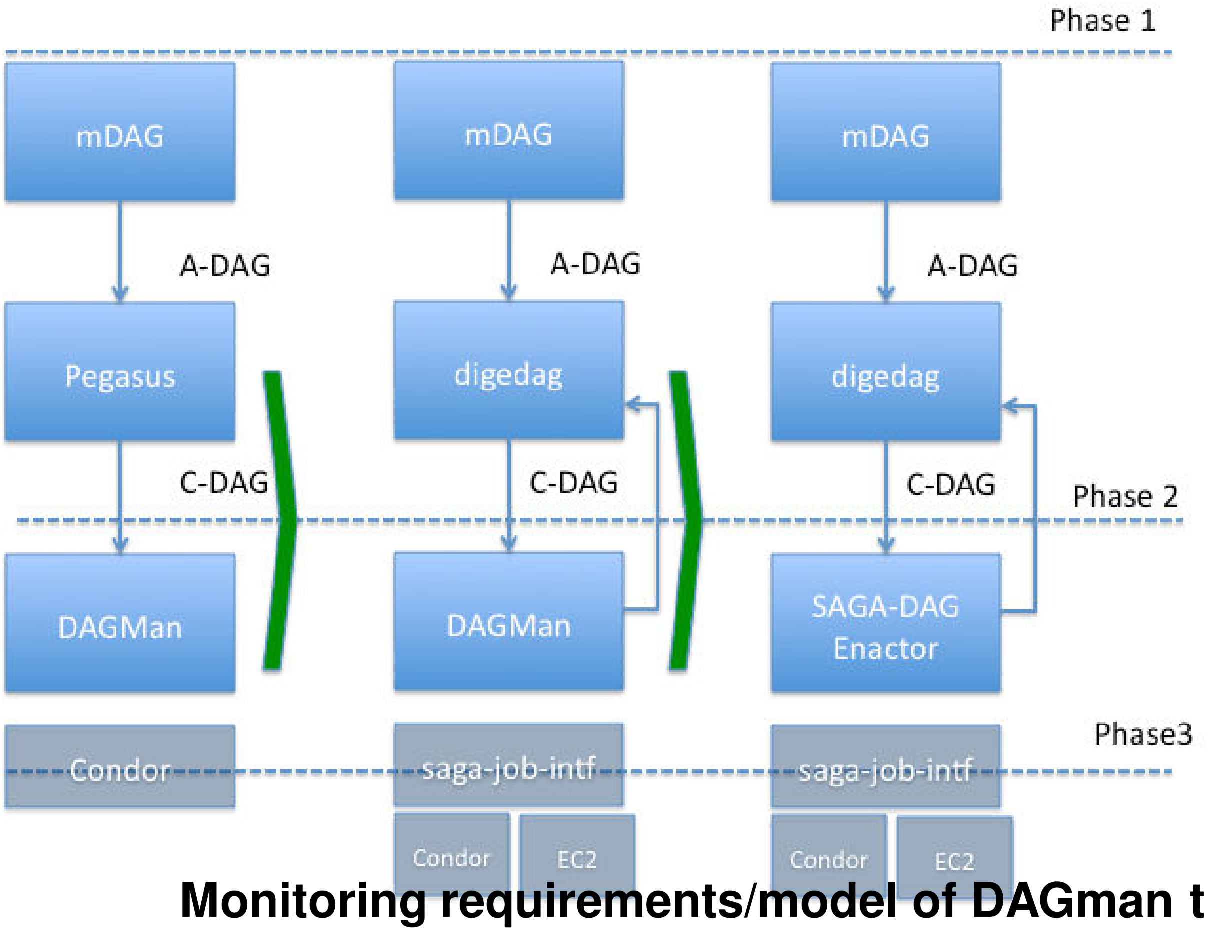
Execution Phase

Digedag: SAGA Workflow Package

- Development Phase: Creation & management of nodes and edges of a DAG and parts of the DAG
- Planning Phase: Digedag planner is fired when creating and executing C-DAG – thus responding to dynamic changes instantly
  - When adding/removing nodes/edges
  - Node/edge firing succeeds/fails, or edge transfer fails/succeeds
- Mixed Planning and Execution Phase
  - Having the full A-DAG, current C-DAG and *live Information*
- Execution Phase: SAGA-based Enactor designed to support explicit dynamic execution
  - SAGA-based DAG enactor, which changes the Concrete-DAG on the fly, thus remapping workflow elements (DAG nodes).

DAG-based Applications

Extensibility and Higher-level API



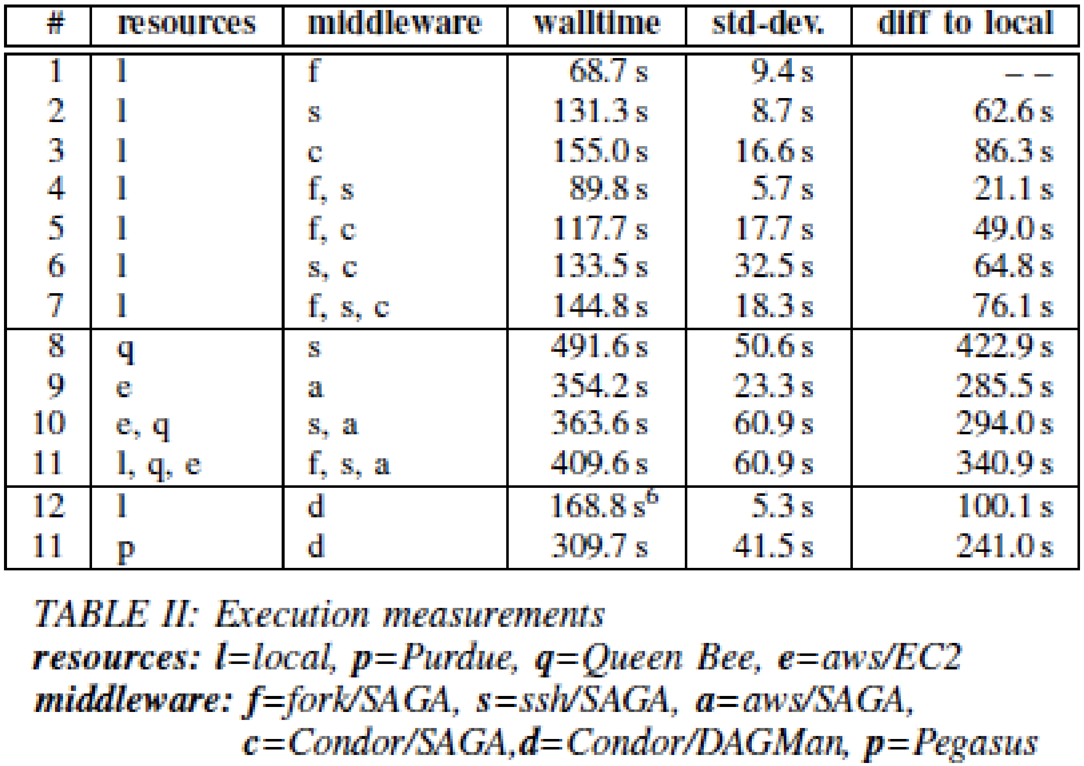
Application Development Phase

Generation & Exec. Planning Phase

Execution Phase

Monitoring requirements/model of DAGman tied with Condor

SAGA-based DAG Execution Preserving Performance



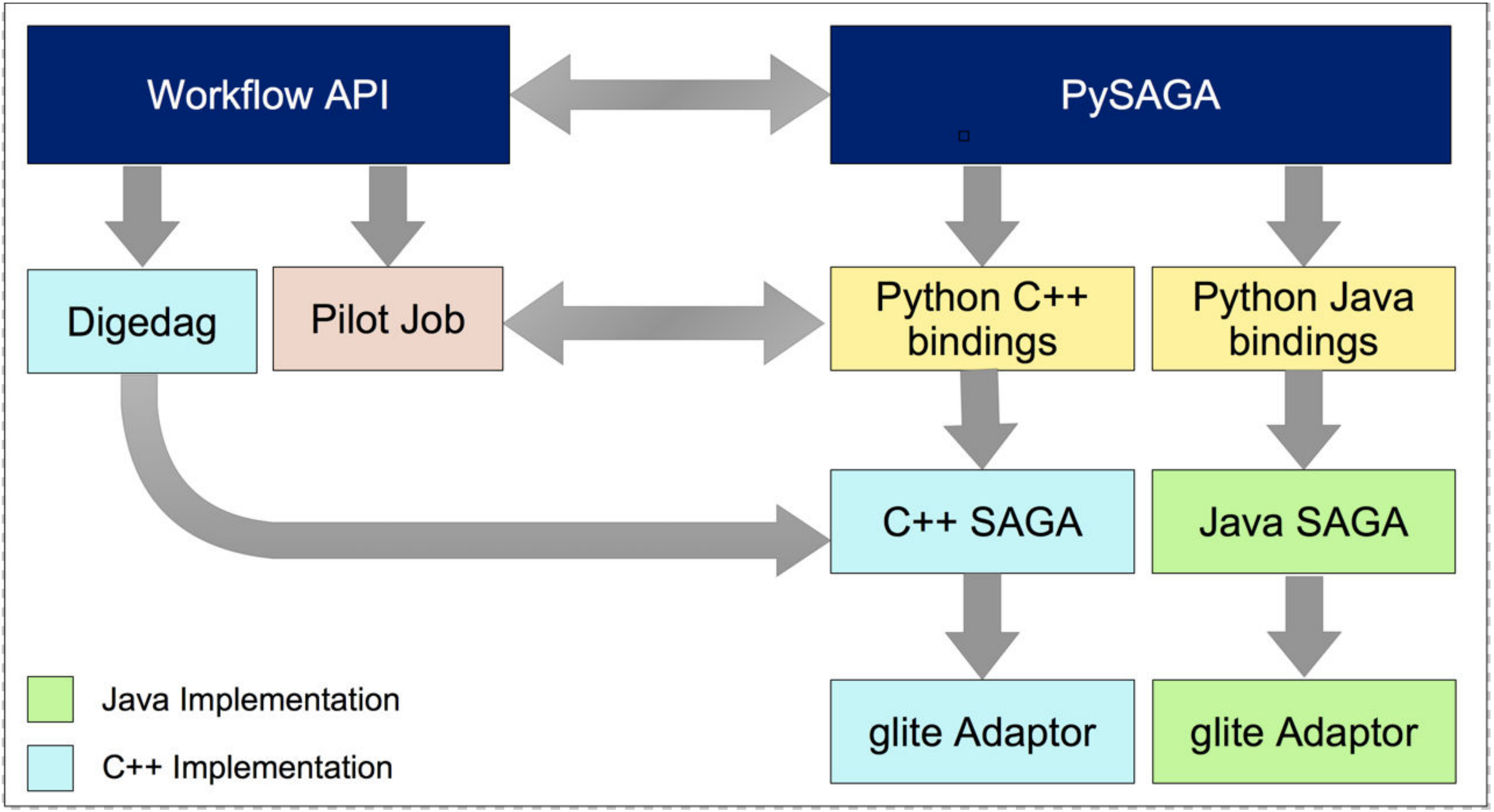
#	resources	middleware	walltime	std-dev.	diff to local
1	l	f	68.7 s	9.4 s	– –
2	l	s	131.3 s	8.7 s	62.6 s
3	l	c	155.0 s	16.6 s	86.3 s
4	l	f, s	89.8 s	5.7 s	21.1 s
5	l	f, c	117.7 s	17.7 s	49.0 s
6	l	s, c	133.5 s	32.5 s	64.8 s
7	l	f, s, c	144.8 s	18.3 s	76.1 s
8	q	s	491.6 s	50.6 s	422.9 s
9	e	a	354.2 s	23.3 s	285.5 s
10	e, q	s, a	363.6 s	60.9 s	294.0 s
11	l, q, e	f, s, a	409.6 s	60.9 s	340.9 s
12	l	d	168.8 s[6]	5.3 s	100.1 s
11	p	d	309.7 s	41.5 s	241.0 s

TABLE II: Execution measurements
resources: ***l***=*local,* ***p***=*Purdue,* ***q***=*Queen Bee,* ***e***=*aws/EC2*
middleware: ***f***=*fork/SAGA,* ***s***=*ssh/SAGA,* ***a***=*aws/SAGA,*
c=*Condor/SAGA,*&#8203;***d***=*Condor/DAGMan,* ***p***=*Pegasus*

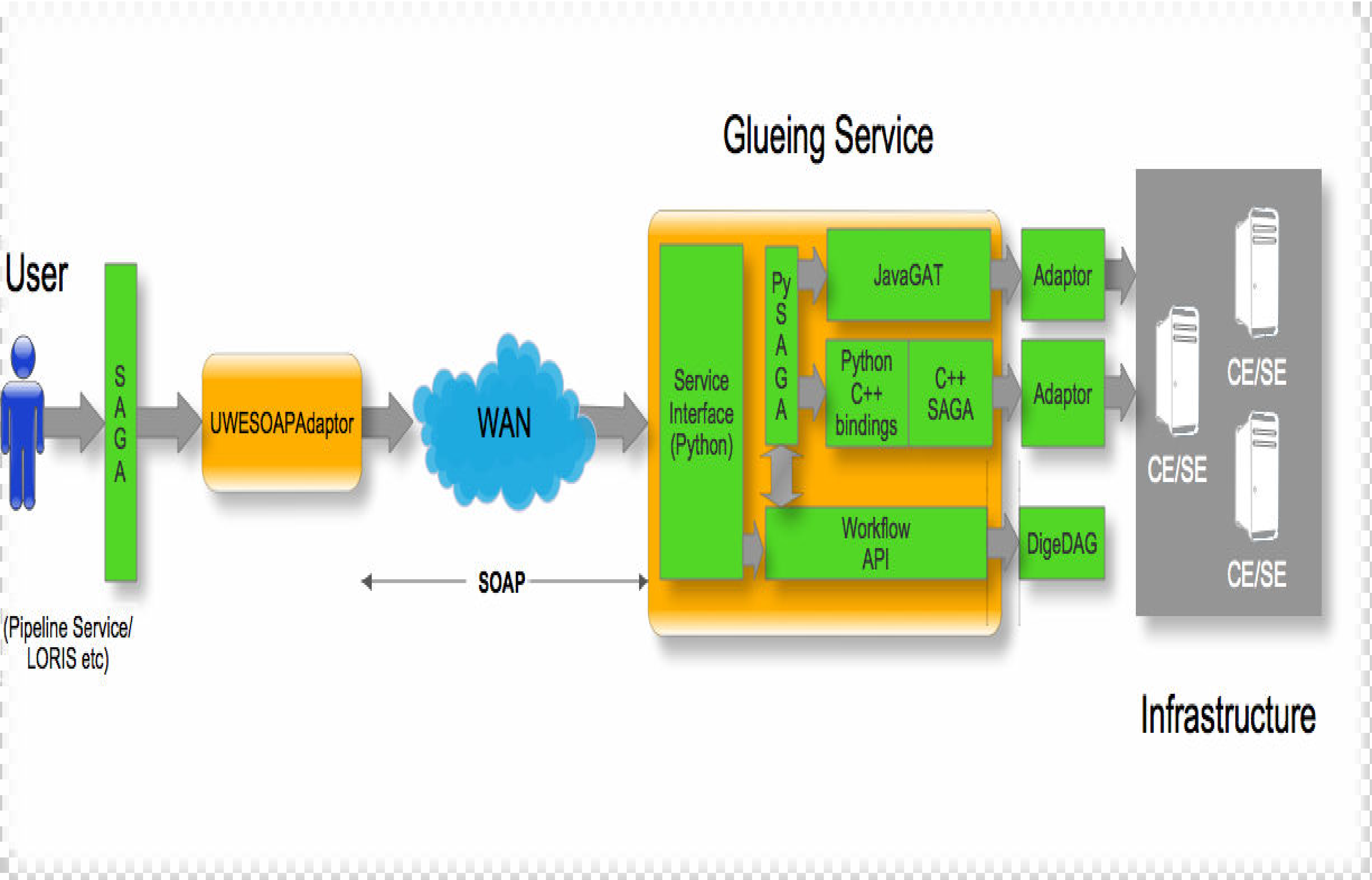
Glueing Service : Current Status

- V1.0 Available, integrated with LORIS & operable with gLite
- Secure authentication with the infrastructure is implemented.
- The glueing service software
  - Can be compiled from source
  - Can be deployed using binaries
  - Can be tested using preconfigured VM
- UWE SOAP Adaptor
  - Supports job submission, monitoring and file transfers
  - Supports file reading, writing, listing
  - Translates SAGA API calls written by an end user to SOAP calls
  - Supports SOAP attachments using Java activation framework

Future Work


Workflow API
PySAGA
Digedag
Pilot Job
Python C++ bindings
Python Java bindings
C++ SAGA
Java SAGA
glite Adaptor
glite Adaptor
Java Implementation
C++ Implementation

Glueing Service (Future)


Glueing Service
User
SAGA
UWESOAPAdaptor
WAN
Service Interface (Python)
PySAGA
JavaGAT
Python C++ bindings
C++ SAGA
Workflow API
Adaptor
Adaptor
DigeDAG
CE/SE
CE/SE
CE/SE
SOAP
(Pipeline Service/ LORIS etc)
Infrastructure