



CENTER FOR COMPUTATION  
& TECHNOLOGY



# SAGA-based Tools and Projects

The SAGA Team

<http://saga.cct.lsu.edu>

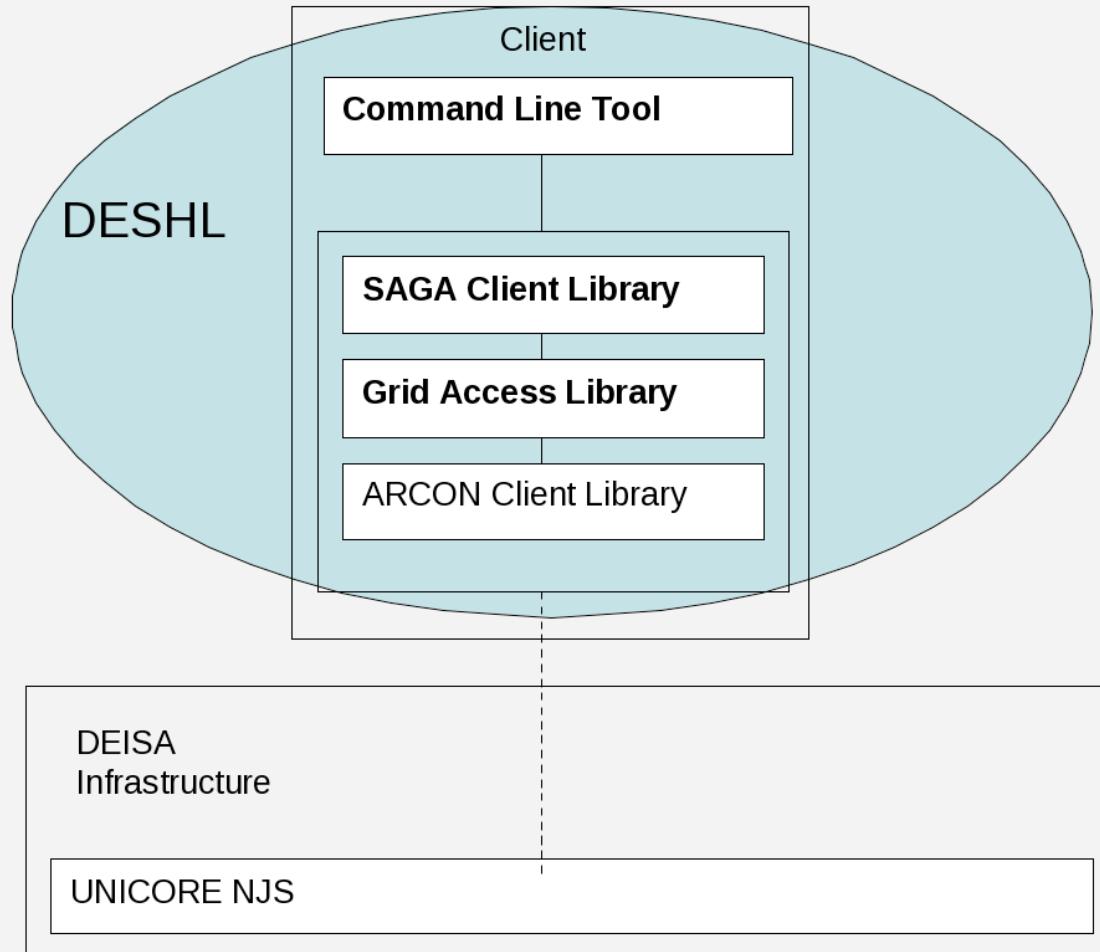


# SAGA-based Tools + Developments

- DESHL
- JSAGA
- GANGA/gLite [Ole]
- XtreemOS
- NAREGI/KEK
- SD [Andre]
- SAGA Gateways

# DESHL v4.0 Architecture

Layered Design sitting on top of existing UNICORE infrastructure



# DESHL Client Overview

- Supplied as a Java command-line application
- Follows Open Grid Service Architecture (OGSA) specification for job batch submission and management
- Uses SAGA (Simple API For Grid Applications) directives for job specification
- Where appropriate, DESHL commands follow the Open Group Technical Standard for Batch Environment Systems
- Layered design to protect against future changes in underlying infrastructure
- Sits on top of existing UNICORE infrastructure



CENTER FOR COMPUTATION  
& TECHNOLOGY

# JSAGA



The screenshot shows a Mac OS X desktop environment with a web browser window open to the JSAGA project homepage at <http://grid.in2p3.fr/jsaga/index.html>. The browser's title bar reads "JSAGA -". The page features a large red banner with the "JSAGA" logo and a globe icon. Below the banner, the text "Last Published: 06/26/2008" and logos for ANR, CNRS, and CC-IN2P3 are visible. The main content area has a header "JSAGA" and a paragraph explaining JSAGA as a Java implementation of the Simple API for Grid Applications (SAGA) specification. It highlights the project's ability to hide heterogeneity between middleware and grid infrastructures. A sidebar on the left contains links for "Project Home", "User Documentation", "Developer Documentation", "Project Documentation", "Links", and "Adaptor Modules".

**JSAGA**

Last Published: 06/26/2008

**Project Home**

- Adaptors
- Download
- Presentations
- Related Projects

**User Documentation**

- SAGA Specification
- SAGA Java Binding
- Deviations from SAGA
- JSDL Specification
- FAQ

**Developer Documentation**

- JSAGA Adaptors API
- JSAGA Engine
- How To

**Project Documentation**

- Project Information
- Project Reports

**Links**

- Development Release
- Build Tools
- Common Tools

**Adaptor Modules**

- Default
- Classic
- Globus 2.4
- Globus WS-GRAM
- gLite JDL
- gLite WMS
- SRM
- VOMS
- Unicore 6
- SSH

**Adaptors provided by...**

**JSAGA**

JSAGA is a Java implementation of the Simple API for Grid Applications ([SAGA](#)) specification from the Open Grid Forum ([OGF](#)). SAGA provides a uniform interface to heterogeneous grid middleware for security, data management and execution management.

But JSAGA does not only hide heterogeneity between middleware, it also hides heterogeneity between grid infrastructures. Indeed, heterogeneity between infrastructures leads to additional issues, including selection of the security context, setup of the job's environment, or efficient transport of input/output files to/from worker nodes. Transfer strategy depends on file size, on the possibility of sharing a given file among several jobs, on required data protection level, on protocol access modes and third-party transfer capability, on security context delegation capability, on characteristics of the execution site (supported protocols, network filtering policy, shared file system and preinstalled files availability).

JSAGA addresses these issues to enable efficient jobs submission to several grid infrastructures with a single job description.

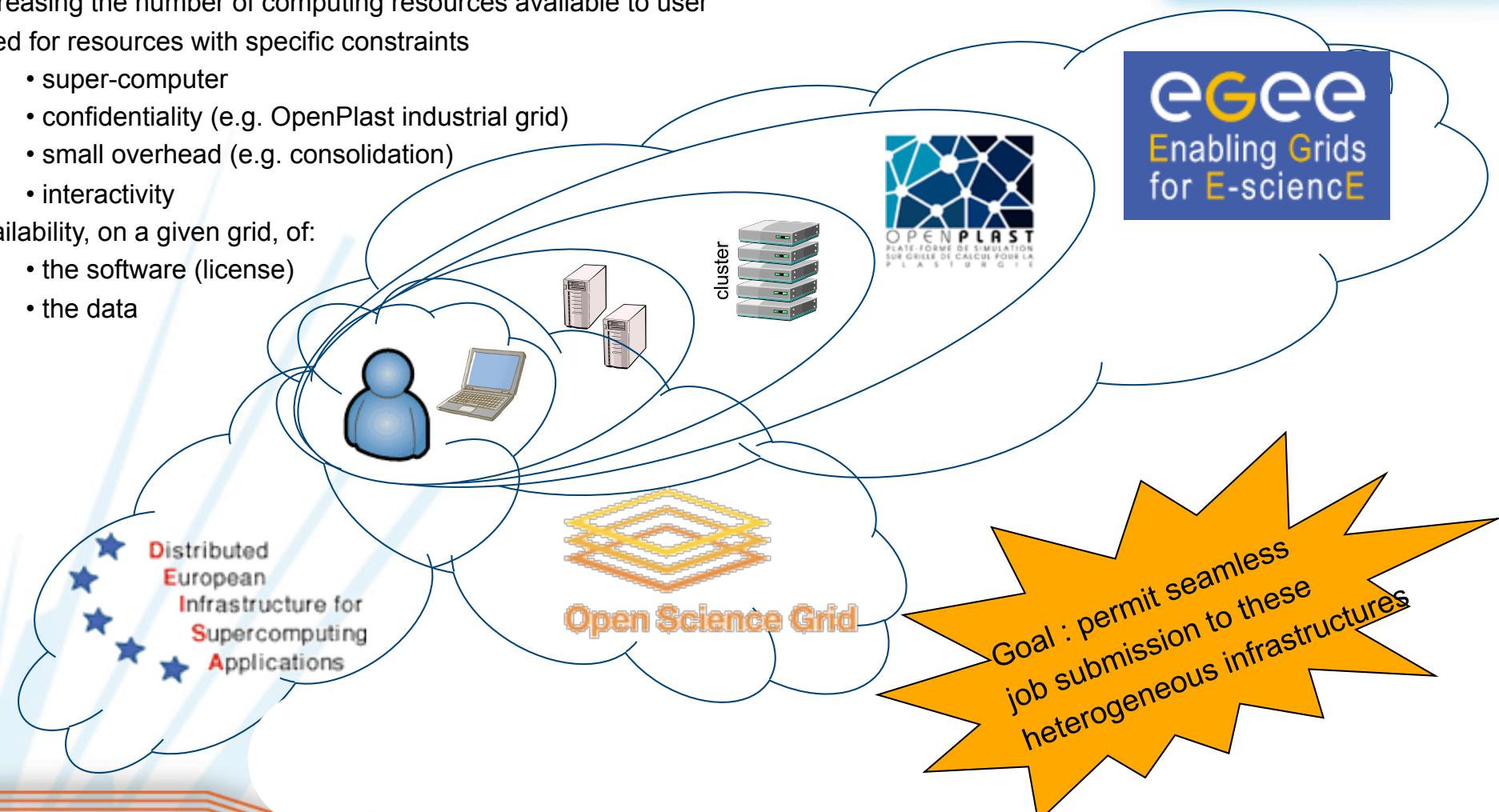


# JSAGA use-cases and goal



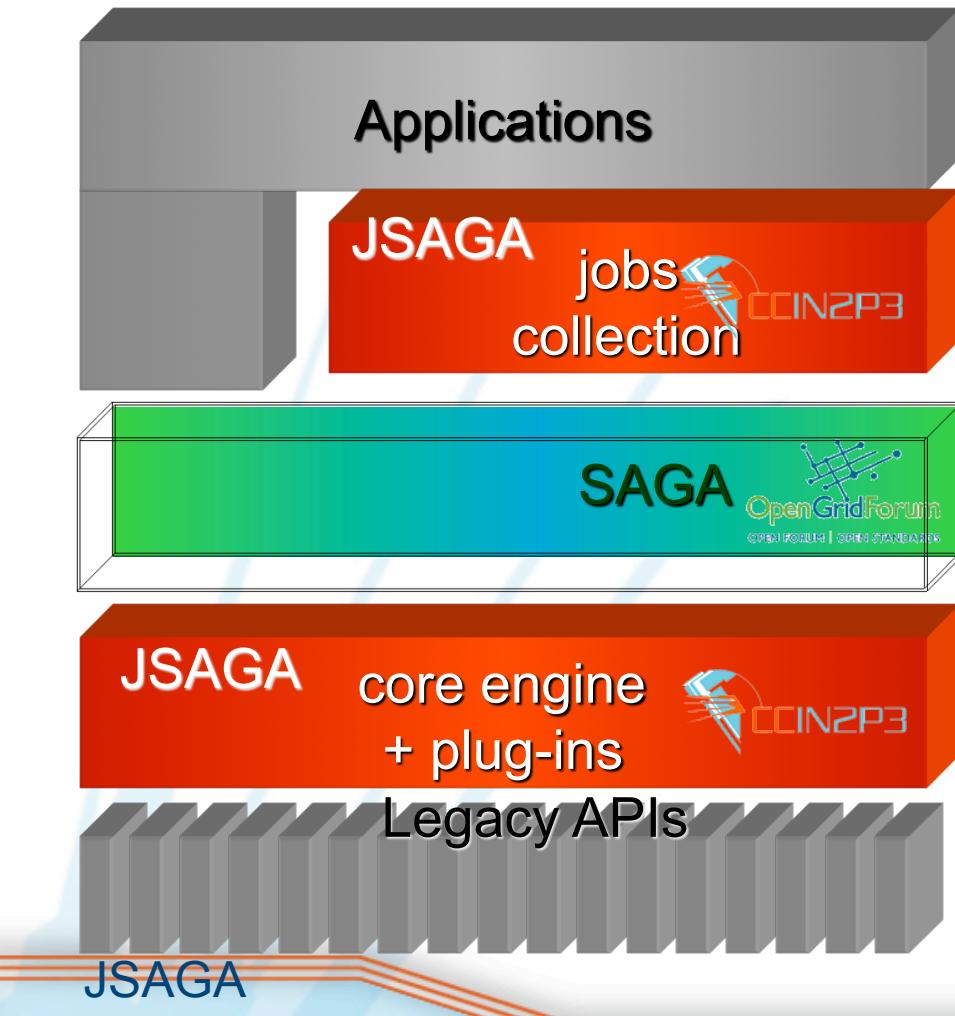
Motivations for using several grid infrastructures

- increasing the number of computing resources available to user
- need for resources with specific constraints
  - super-computer
  - confidentiality (e.g. OpenPlast industrial grid)
  - small overhead (e.g. consolidation)
  - interactivity
- availability, on a given grid, of:
  - the software (license)
  - the data





# Both implementer and user of SAGA



**JSAGA uses SAGA in a module, which hides heterogeneity of grid infrastructures**

**JSAGA implements SAGA to hide heterogeneity of middlewares**



# Projects using JSAGA



## ■ Elis@

- a web portal for submitting jobs to **industrial and research** grid infrastructures



## ■ SimExplorer

- a set of tools for managing **simulation experiments**
- includes a workflow engine that submit jobs to heterogeneous distributed computing resources



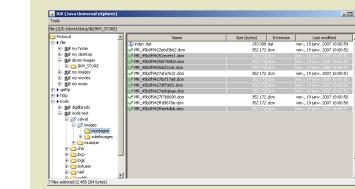
## ■ JJS

- a tool for running efficiently **short-life jobs** on EGEE



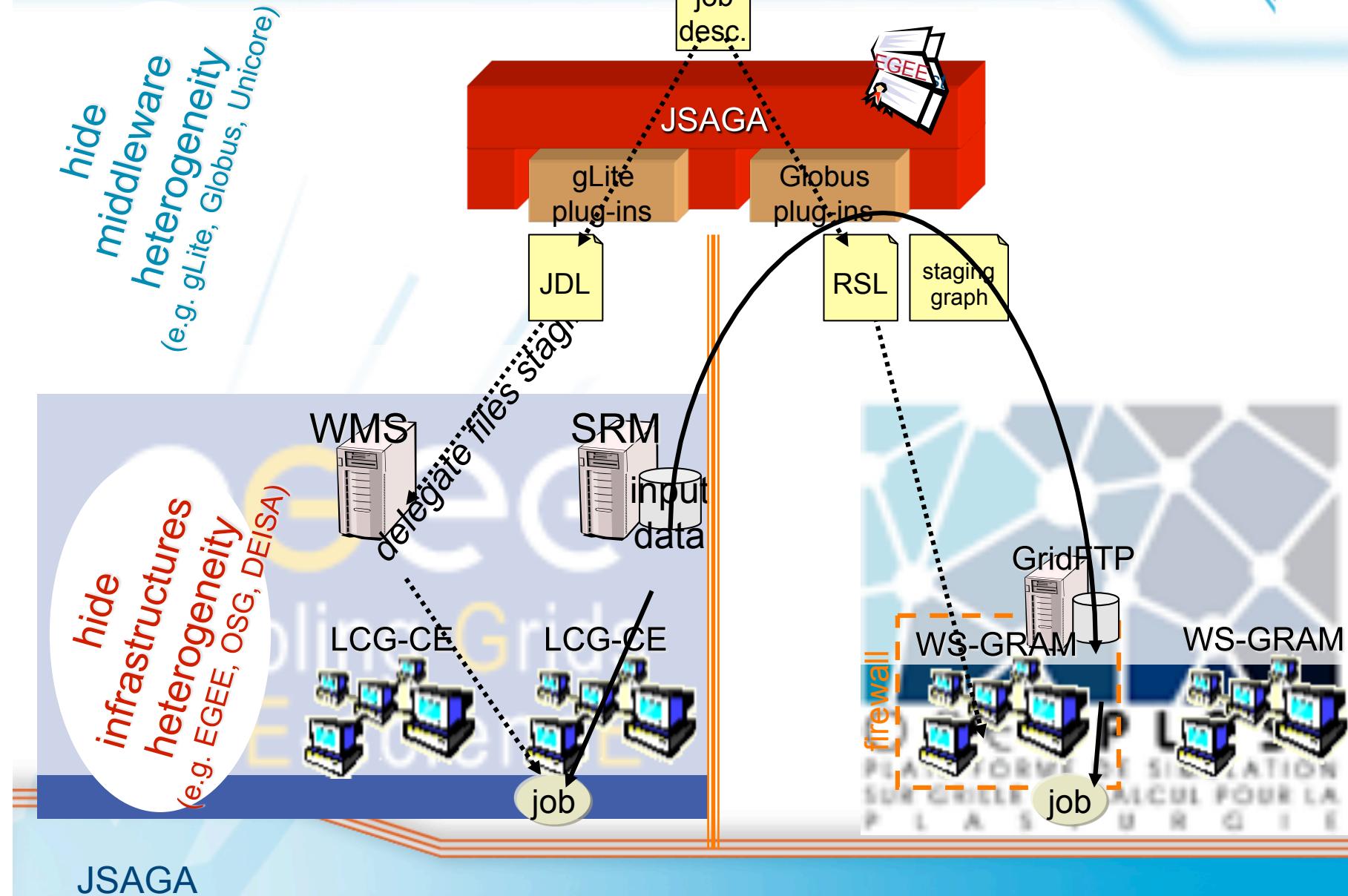
## ■ JUX

- a **multi-protocols** file browser





# m/w vs. infrastructure heterogeneity





CENTER FOR COMPUTATION  
& TECHNOLOGY

# SAGA XtreemOS



- Challenges:
  - Linux applications should run with little (no) modifications
  - Grid applications should run with little (no) modifications
  - XtreemOS functionality must be provided to applications

## Approach:

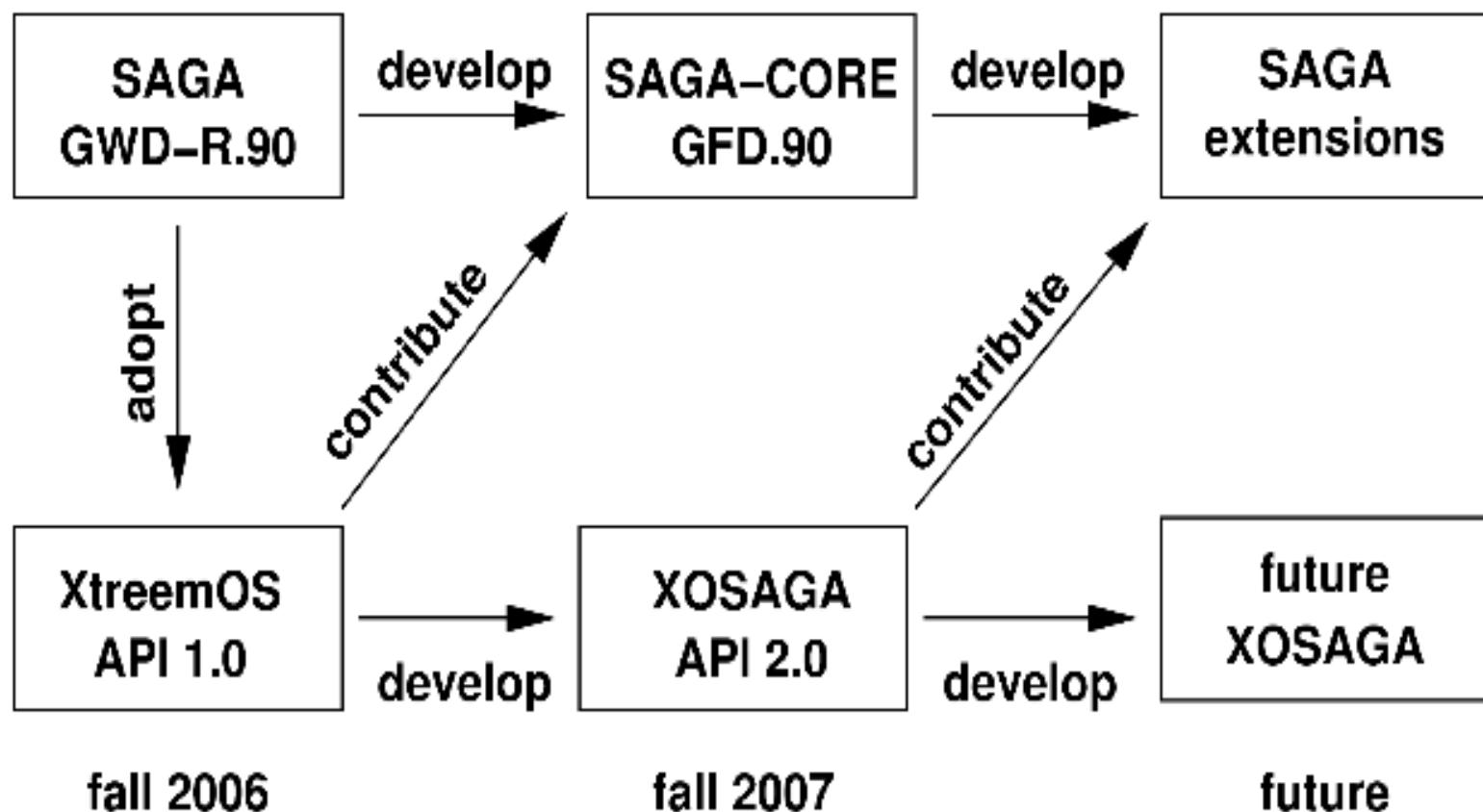
- Use OGF-standardized SAGA API, enabling both Linux apps (via POSIX semantics) and grid apps
- Provide extensions packages to SAGA for XtreemOS functionality and in the process contribute to standards



CENTER FOR COMPUTATION  
& TECHNOLOGY

# Co-evolution of SAGA Standard & the XtreemOS API

(slide courtesy Thilo Kielmann)





Enabling Grids for E-sciencE

# An Extension to the SAGA Service Discovery API

*Antony Wilson*

*(Steve Fisher and Paul Livesey)*

*4<sup>th</sup> EGEE Users Forum – Catania 2009*

[www.eu-egee.org](http://www.eu-egee.org)



- **The specification for the Service Discovery has been finalised – GFD.144**
  - <http://www.ogf.org/documents/GFD.144.pdf>
  - Loosely based around the gLite Service Discovery
  - Uses the GLUE (version 1.3) model of a service
    - A Site may host many Services
    - A Service has multiple Service Data entries
    - Each Service Data entry is represented by a key and a value
- **APIs in C, C++, Java and Python complete but not released**
- **Adapter for gLite under development**
  - Based around GLUE 1.3
    - Once GLUE 2 is finalised the adapter will be updated

- **API allows selection based on three filters:**
  - **serviceFilter** – allows filtering on:
    - type, name, uid, site, url, implementor and relatedService
  - **dataFilter** – no predefined values
    - Uses keys from Service Data entries
  - **authzFilter** – authorization, no predefined values, useful values include:
    - vo, dn, group and role (values dependent on adapter)
  - NB if an **authzFilter** is not provided then one is automatically constructed from the users security context
    - The gLite adapter will provide the VOMS proxy credentials as the default value for the security context
- **Each of the filter strings uses SQL92 syntax**
- **The filters act as if they are part of a WHERE clause**

- **Selection returns a list of ServiceDescriptions**
  - Each description contains:
    - type, name, uid, site, url, implementor, list of relatedServices and service data (key value pairs)
- **Example:**

```
discoverer = SDFactory.createDiscoverer()
serviceDescriptions =
    discoverer.listServices("type = 'computing service'", "")
```

loop serviceDescriptions

```
    description.getAttribute("name")
```

returns the value of the name attribute

  - **URL of information system can be passed in with the constructor, or obtained from a conf file**

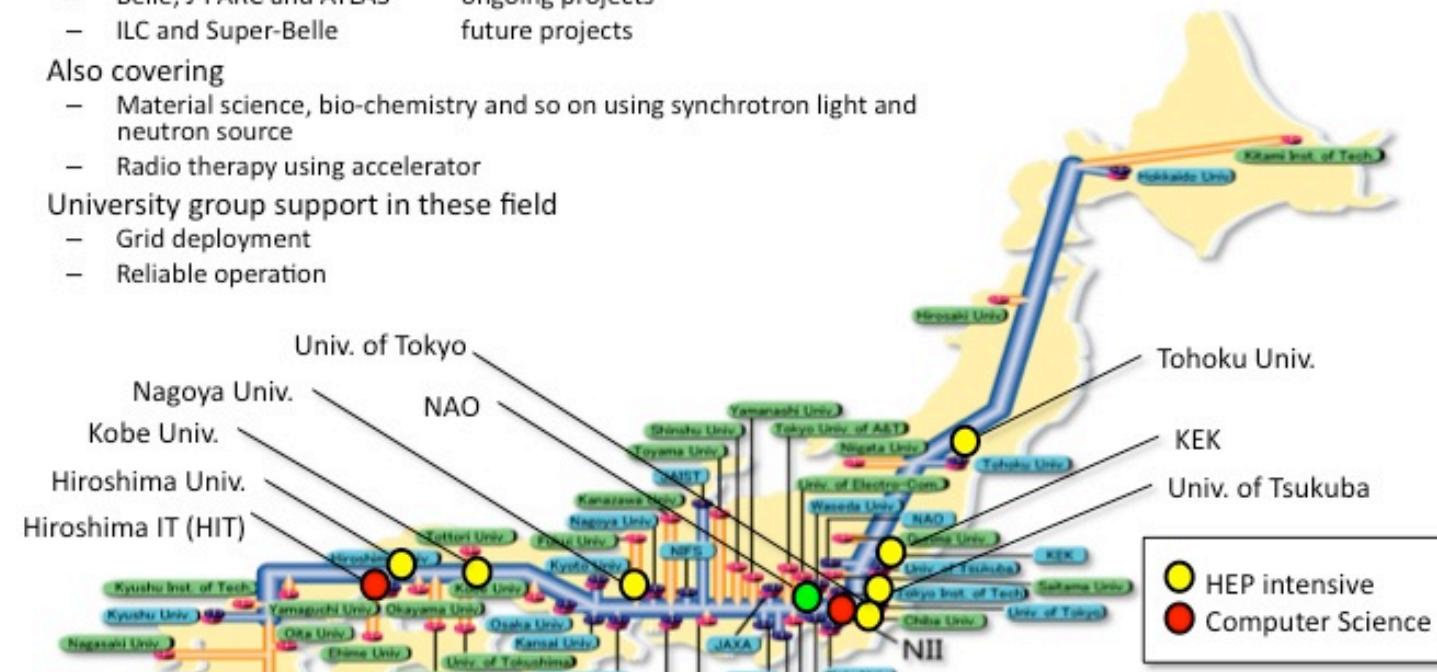
- **Problem:** The service discovery API only gives basic information as it cannot represent the GLUE model in three tables
- **Purpose:** To navigate the information model starting from a selected service
- API will be independent of the underlying information system
- Different information systems supported by means of adapters
  - We will provide a gLite adapter
- Navigation will be from entity to entity as expressed in the GLUE entity relationship models



SAGA @ KEK

- KEK is the central laboratory for HEP in Japan
  - Major HEP projects:
    - Belle, J-PARC and ATLAS      ongoing projects
    - ILC and Super-Belle      future projects
  - Also covering
    - Material science, bio-chemistry and so on using synchrotron light and neutron source
    - Radio therapy using accelerator
  - University group support in these field
    - Grid deployment
    - Reliable operation

SINET3: Production R&E network  
10-40 Gbps national backbone  
10 Gbps for NYC and LA



Coordination and implementation of infrastructures for particularly Japanese HEP institutes are KEK's responsibility.

# Grid Deployment at KEK

- NAREGI middleware is being deployed as the general purpose e-science infrastructure in Japan
- Seamless user environment between the local resource and multiple grid environment should be provided.
  - Otherwise user have to create as many applications as middleware.

Matrix between experiment and middleware

	gLite	NAREGI	Gfarm	SRB	iRODS
Belle	Using	Planning	Using	Using	
Atlas	Using				
Radio therapy	Using	Developing	Planning		
ILC	Using	Planning	Planning		
J-PARC	Planning	Planning	Planning		Testing
Super-Belle	To be decided by 2010				

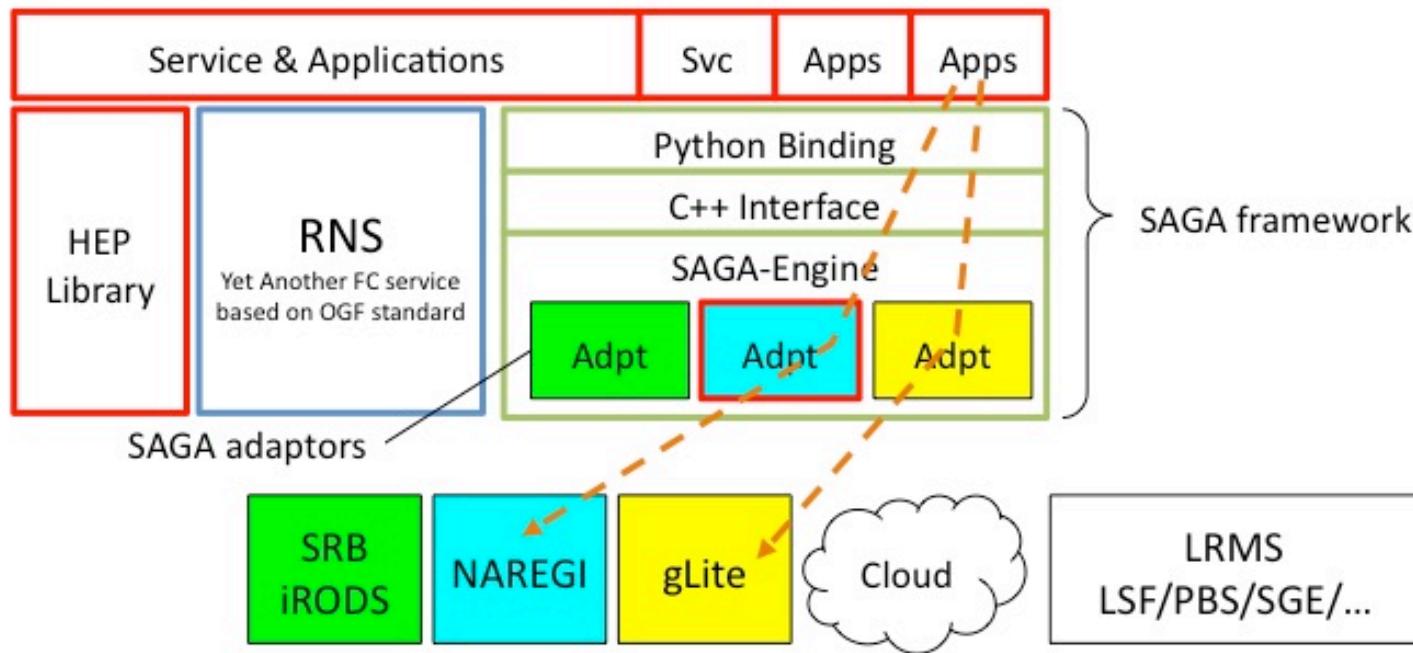
Osaka Univ.  
Tsukuba Univ.

KEK

SAGA

# RENKEI Project Aims

Middleware-independent service & application

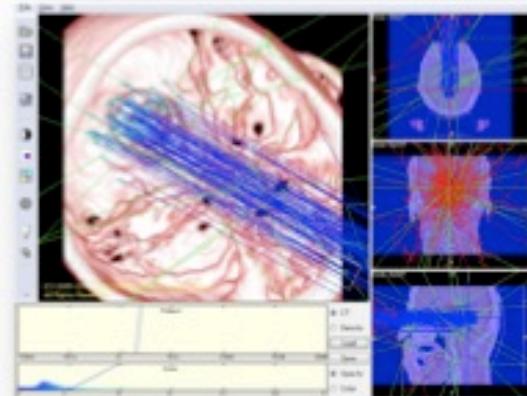
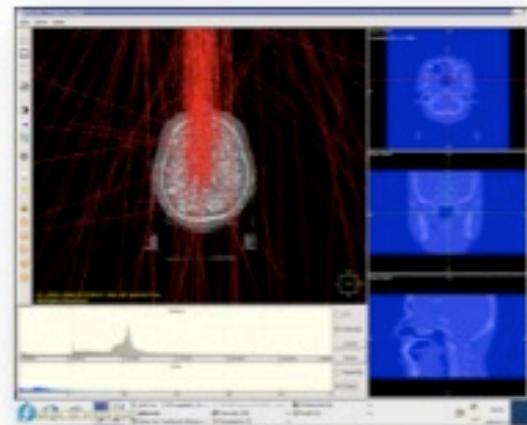


This activity is funded by MEXT as a part of RENKEI project which develops seamless linkage of resources in the Grids and the local one for e-Science.

# Practical Examples in Testbed

- Grid environment
  - MW: NAREGI v1.1 released in October 2008
  - VO scale: KEK, NAO, HIT, and NII
- SAGA adaptors:
  - NAREGI adaptor for job completed in April 2009
  - Torque adaptor completed in June 2009
- Demonstration in testbed
  - Particle therapy simulation based on Geant4 as the 1<sup>st</sup> practical example
  - Resource scale
    - 3 sites: KEK, NAO, HIT
    - CPU: 10 cores
    - OS: CentOS 5.2 x86\_64
    - Memory: 2 GB each

Well done!



More application-wise development in 2009