



# The GridWay Meta-Scheduler

## **Committers**

- Ignacio M. Llorente
- **Ruben S. Montero**
- Eduardo Huedo

## **Contributors**

- Tino Vazquez
- Jose Luis Vazquez
- Javier Fontan
- Jose Herrera



## Goals of the Project

---

- Goals of the Project
- Description of the Project
- GridWay 5.2 Features
- Next Release GridWay 5.2.1



## Goals of the Project

“The GridWay Project is a Research and Development effort that seeks to advance the technology for meta-scheduling on grid

### What is GridWay?

Globus GridWay meta-scheduler is a scheduler virtualization layer on top of Globus services (GRAM, MDS & GridFTP)

#### For the **user**

- A LRM-like environment for submitting, monitoring, and controlling jobs
- A way to submit jobs to the Grid, without having to worry about the details of exactly which local resource will run the job

#### For the **developer**

- An standard-base (OGF, DRMAA C & JAVA) development framework for Grid Applications

#### For the **sys. admin.**

- A policy-driven job scheduler, implementing a variety of access and Grid-aware load balancing policies.
- Accounting

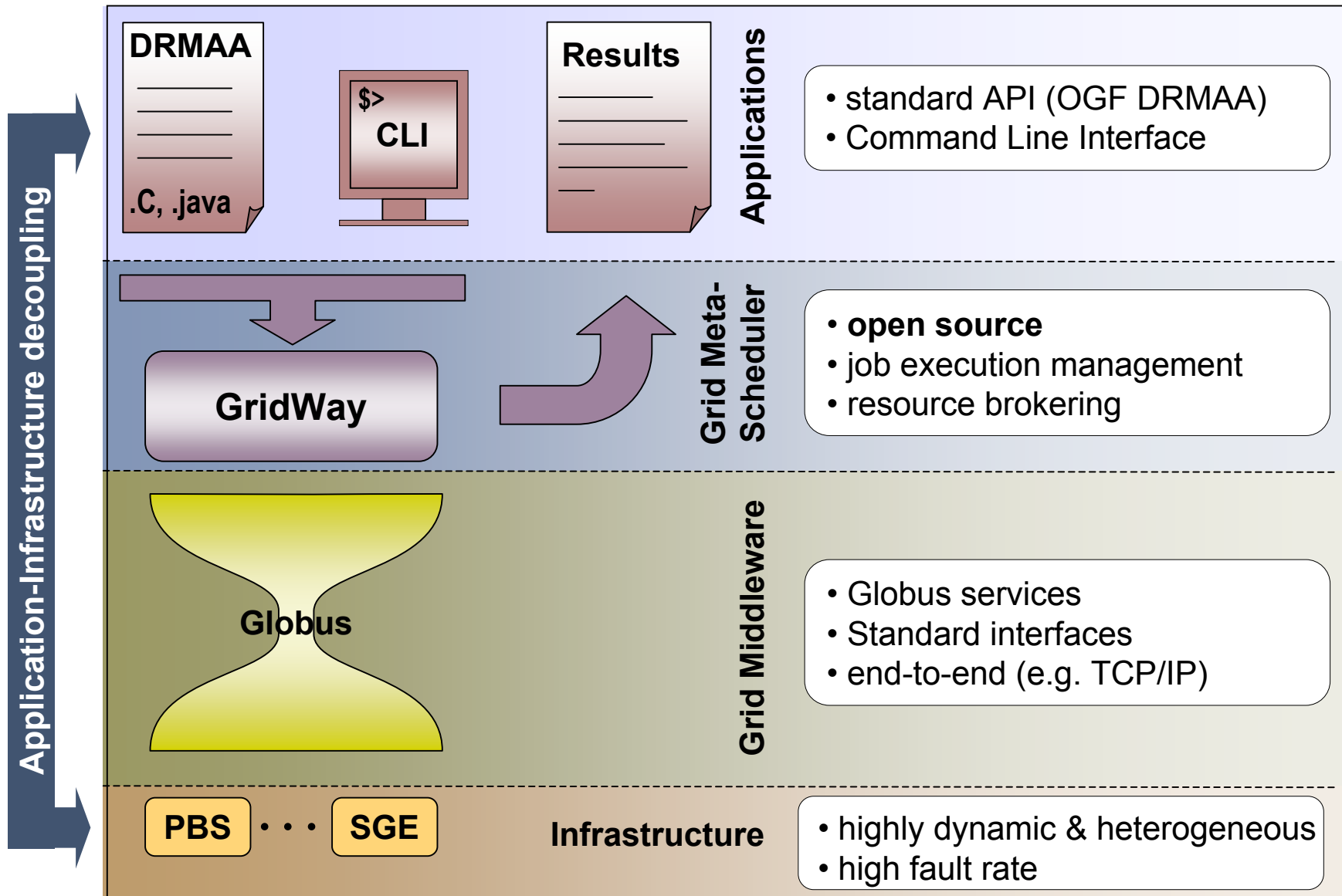
#### For the **Grid architect**

- A modular component to use different infrastructures
- A key component to deploy different Grids (enterprise, partner, utility...)



# Description of the Project

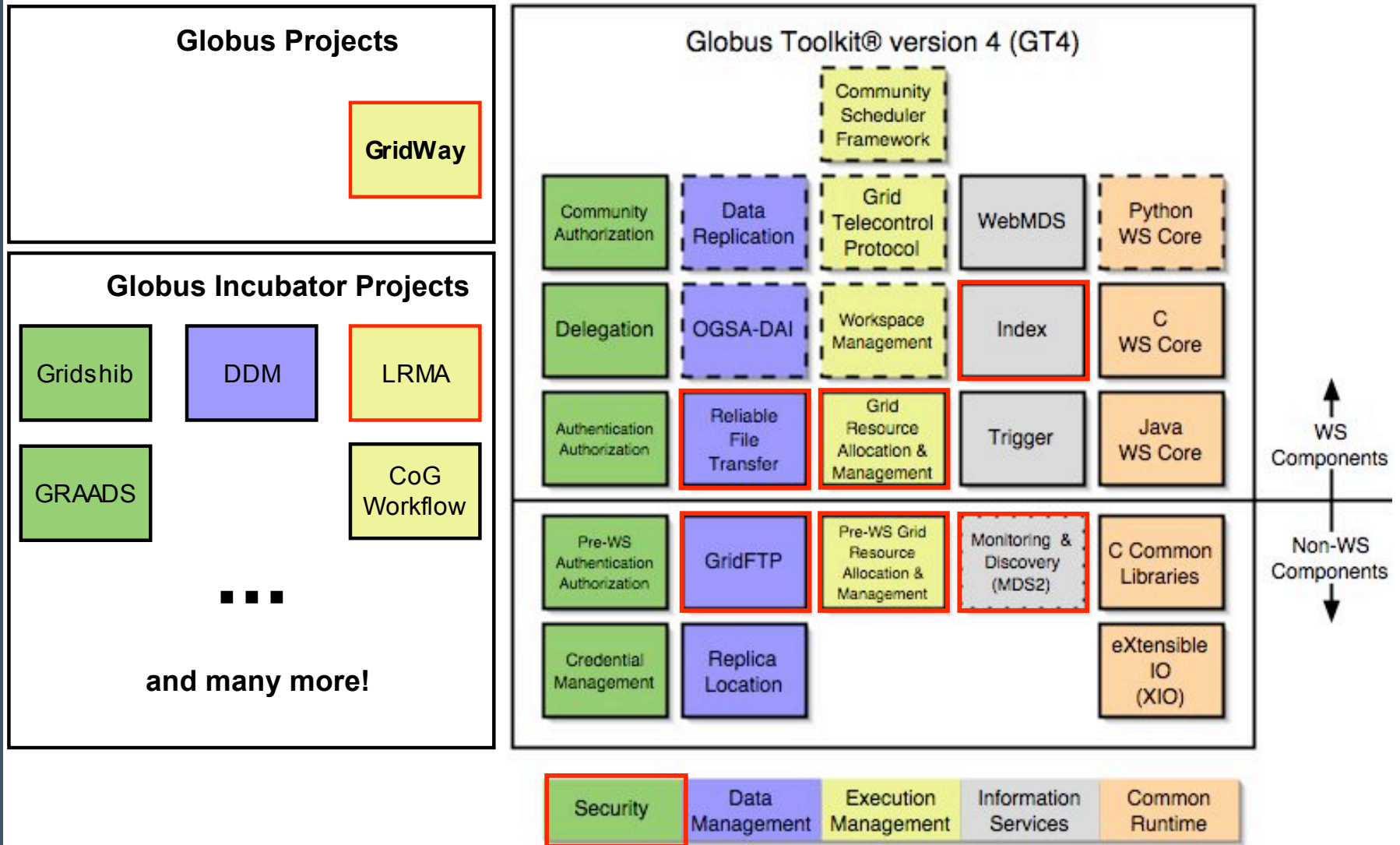
## Architecture of a Computational Grid





# Description of the Project

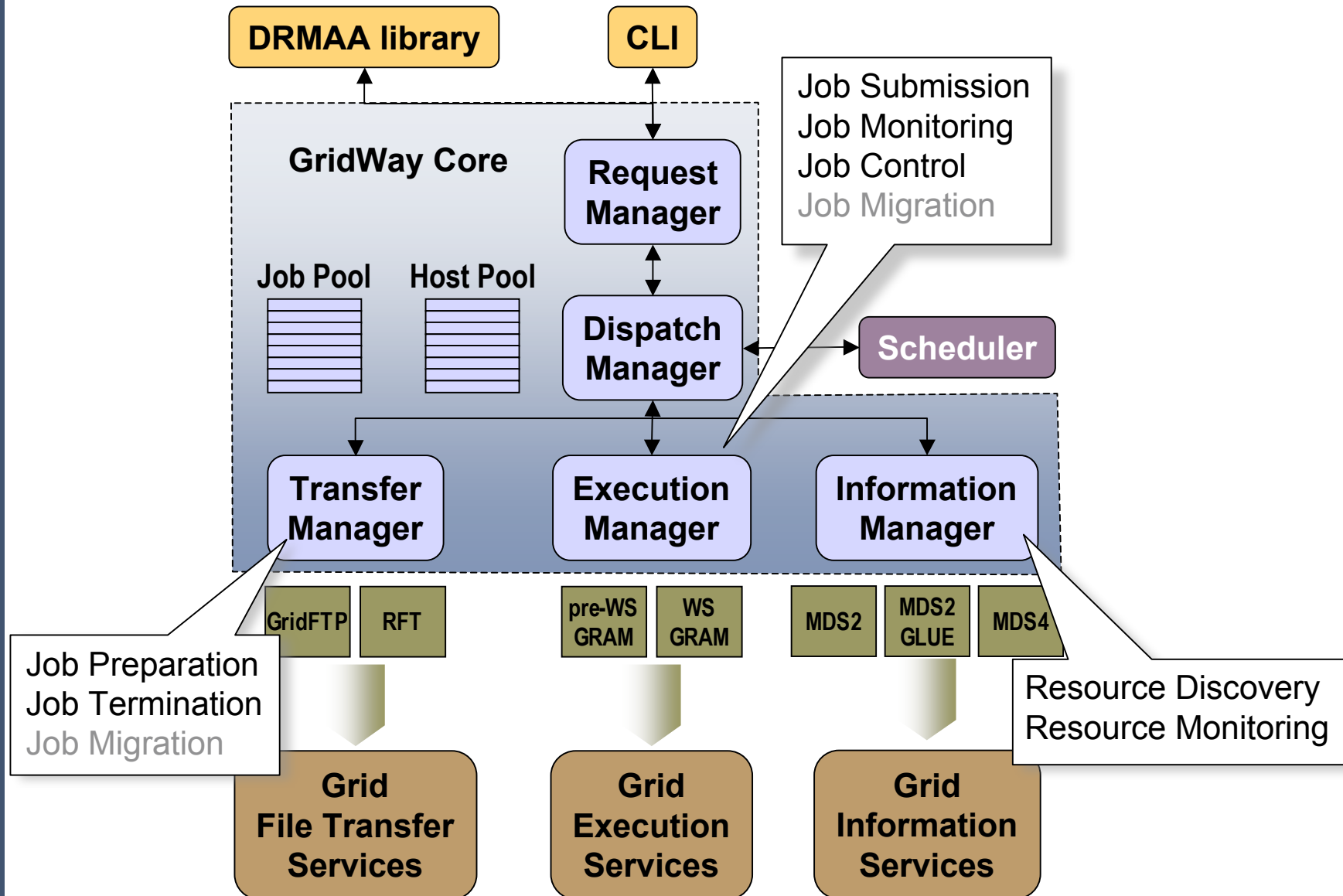
## Relationship to Other Globus Projects





# Description of the Project

## Architecture of GridWay





# Gridway 5.2 Features

---

## Workload Management

---

- Advanced (Grid-specific) **scheduling policies**
- Fault detection & recovery
- Accounting
- Array jobs, DAG workflows and **MPI** jobs

## User Interface

---

- OGF standards: **JSDL (POSIX Profile)** & DRMAA (C and JAVA)
- Analysis of trends in resource usage
- Command line interface, similar to that found on local LRM Systems
- Easier installation through the **auto-tools** framework

## Deployment

---

- Straightforward deployment as new services are not required
- Interoperability between different infrastructures
- Flexible component integration to deploy several Grid architectures



# Gridway 5.2 Features

## Scheduling Policies

### Resource Policies

- Rank Expressions
- Fixed Priority
- User Usage History
- Failure Rate

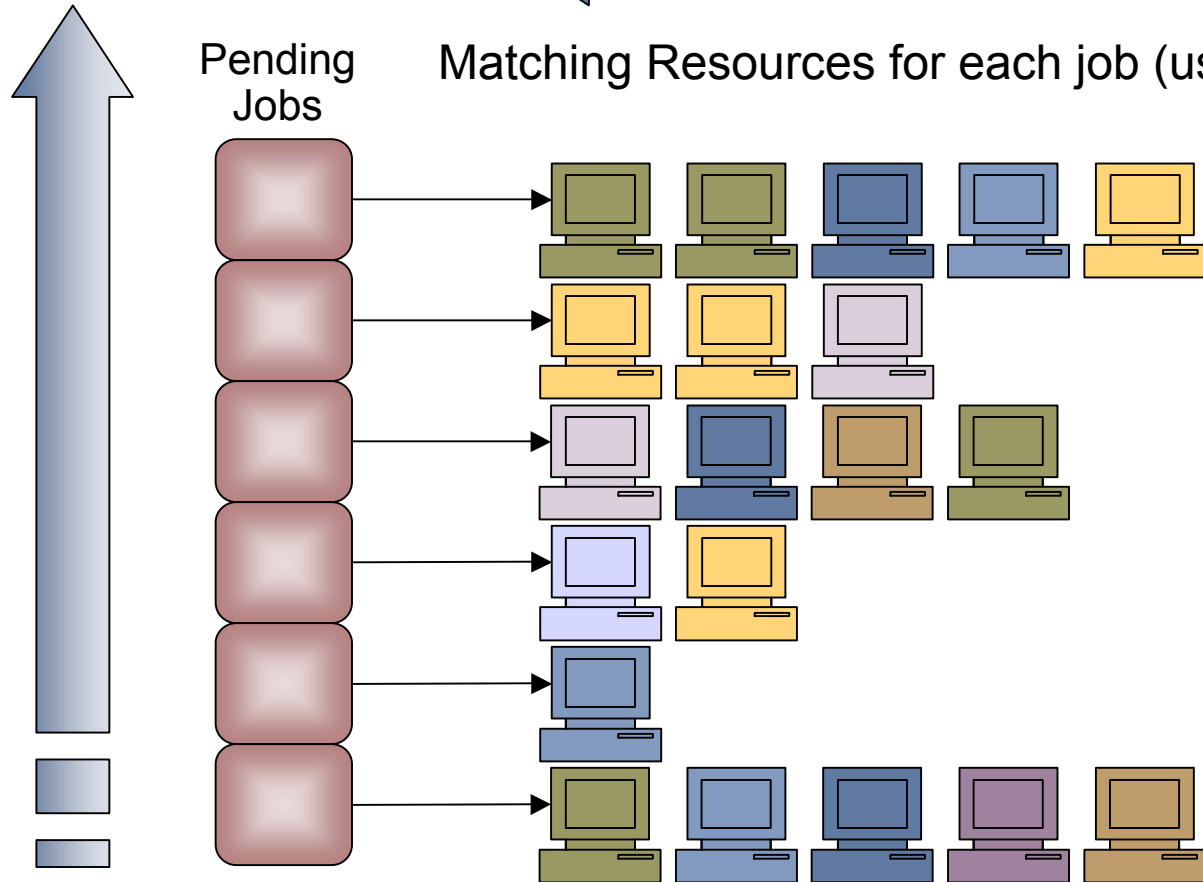
**Grid Scheduling = Job + Resource Policies**

### Job Policies

- Fixed Priority
- Urgent Jobs
- User Share
- Deadline
- Waiting Time

Pending Jobs

Matching Resources for each job (user)





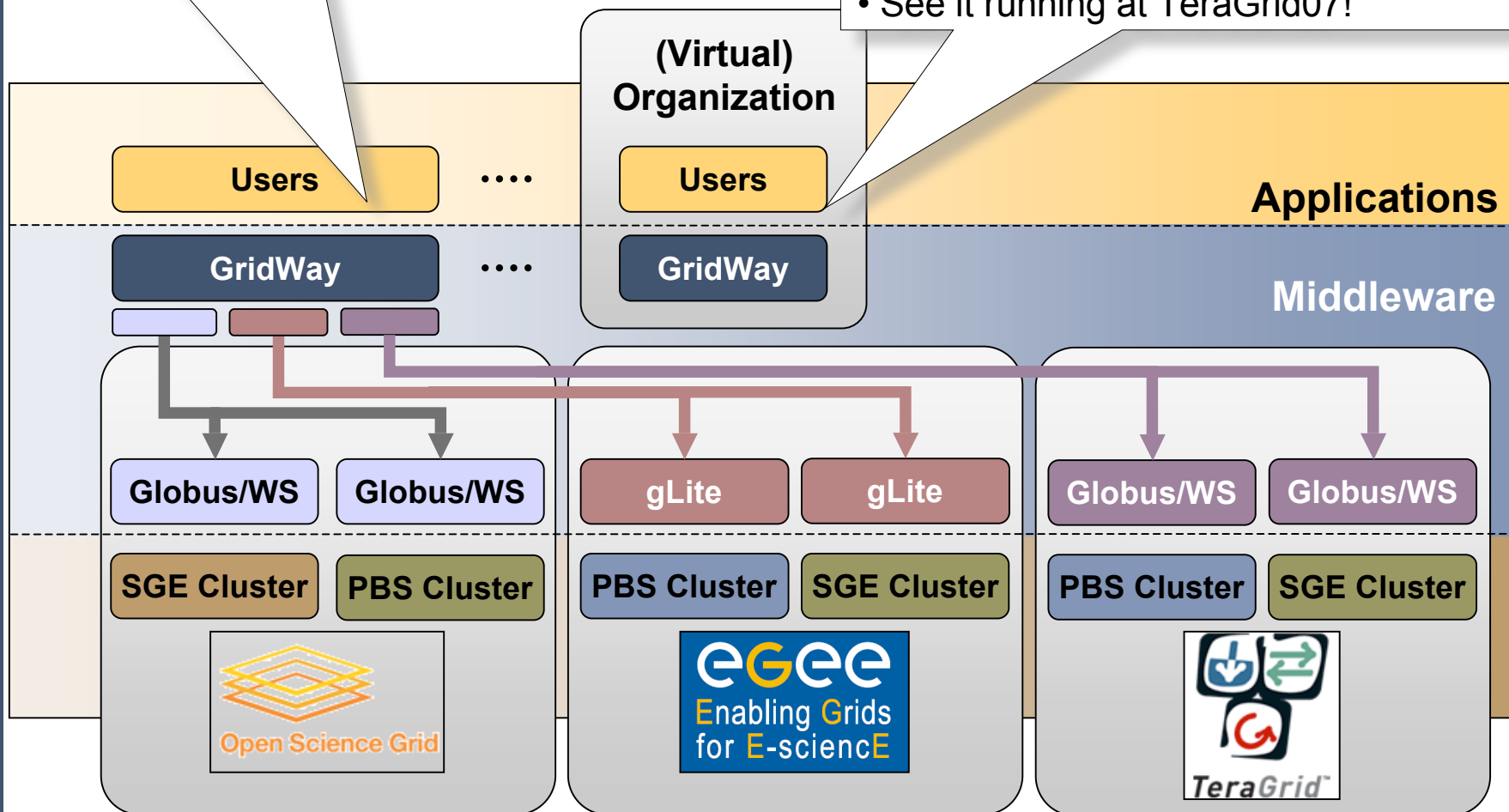


# Next Release GridWay 5.2.1

*Interoperability: OSG, EGEE, TG & NorduGrid*

- (V) Organization meta-schedulers
- Science gateways
- Organization-wide Grid-aware policies

- Different Middleware stacks
- Different Data/Execution architectures
- Different Information models
- Integration through adapters
- See it running at TeraGrid07!



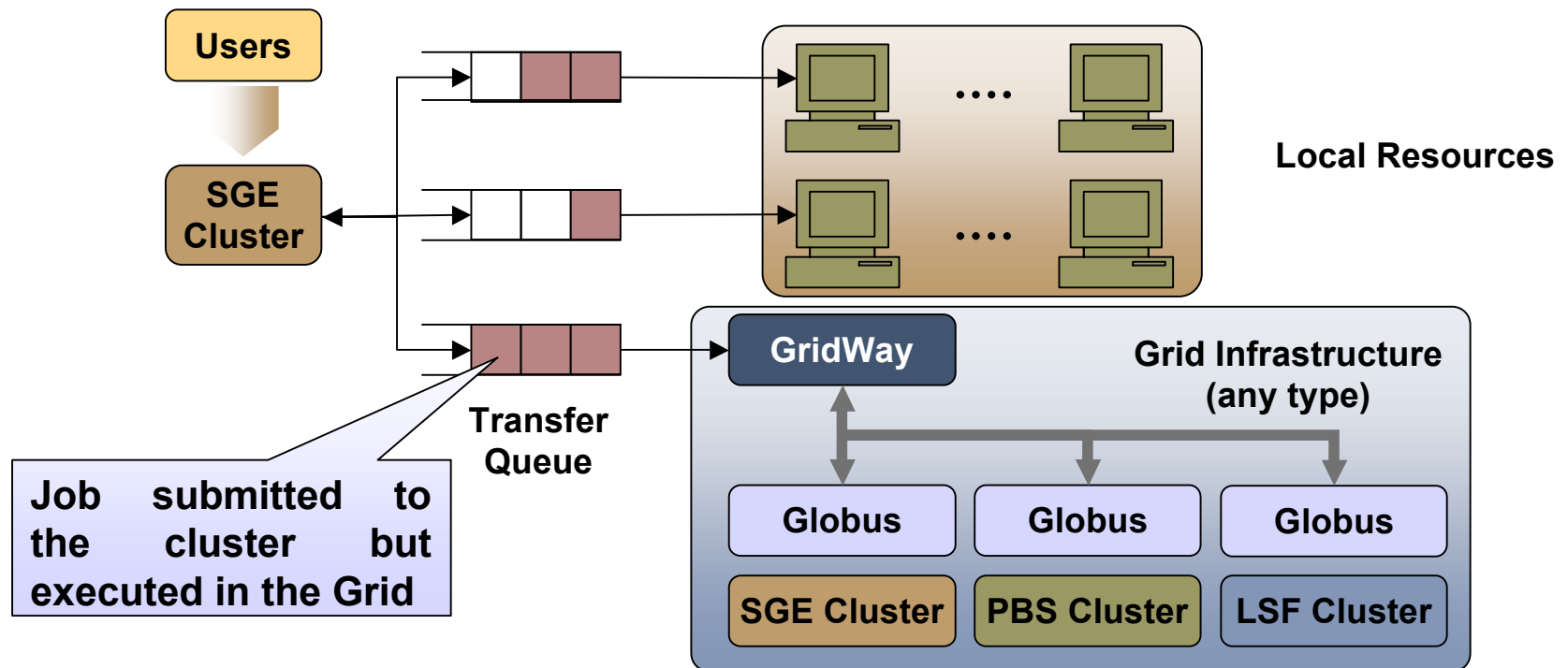


# Next Release GridWay 5.2.1

*Usability: Transfer Queues*

## Seamless integration of a Grid

- Communicate LRM systems with meta-schedulers (the other way)
- Users keep using the same interface, even applications (e.g. DRMAA)





**Thank you for your attention!**

**Want to see GridWay in action...  
look for the GridWay logo at the  
Exhibition Hall!**

[www.gridway.org](http://www.gridway.org)

