



## An Overview of the GridWay Metascheduler

PPAM 2007  
Gdansk, Poland  
September 9-12, 2007



**Tino Vázquez**  
**Distributed Systems Architecture Group**  
**Universidad Complutense de Madrid**

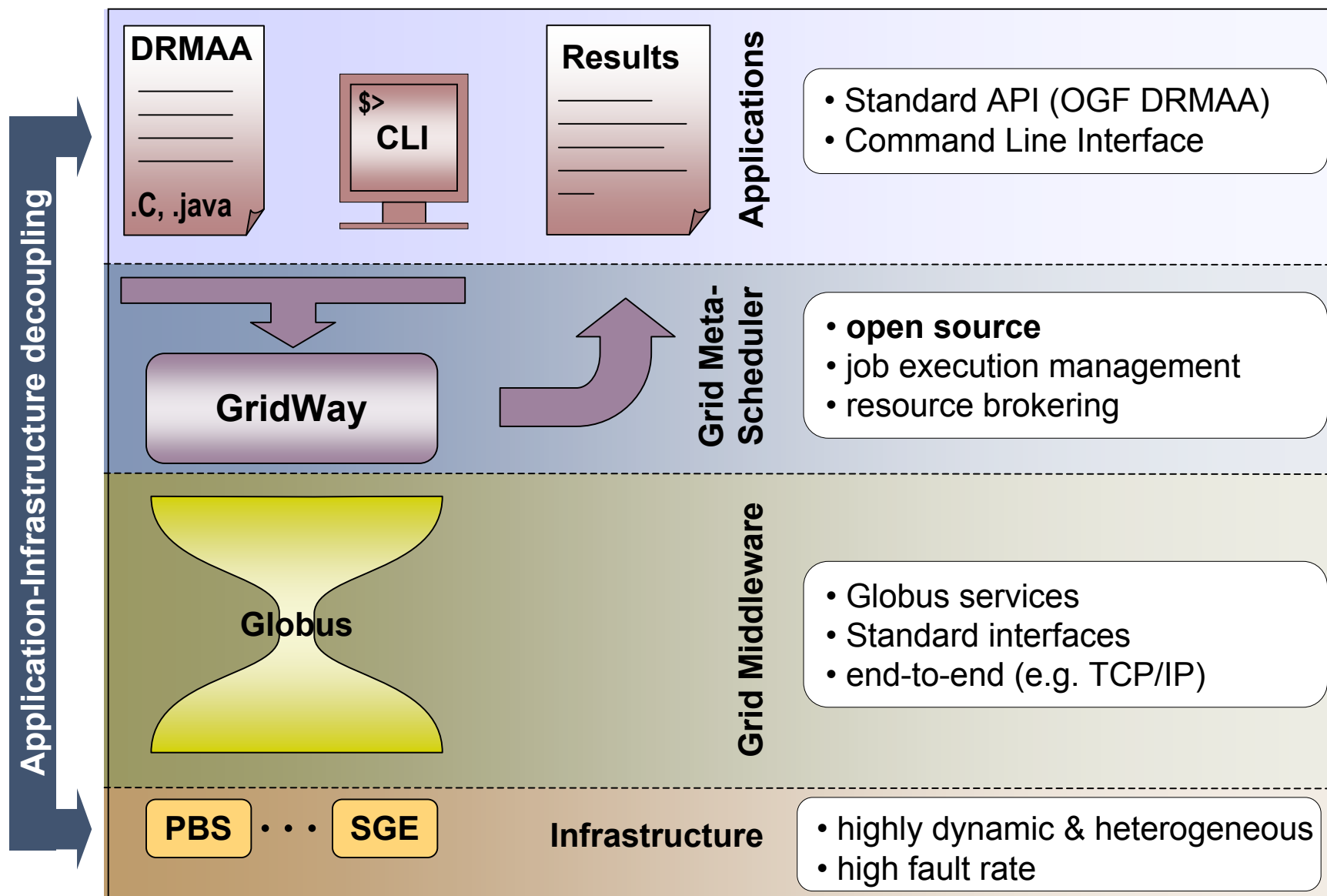
1. **What is GridWay?**
2. A Global Vision
3. Scheduling Policies
4. Scheduling Infrastructures
5. The GridWay Project

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

- For **project and infrastructure directors**
  - GridWay is an open-source community project, adhering to Globus philosophy and guidelines for collaborative development.
- For **system integrators**
  - GridWay is highly modular, allowing adaptation to different grid infrastructures, and supports several OGF standards.
- For **system managers**
  - GridWay gives a scheduling framework similar to that found on local LRM systems, supporting resource accounting and the definition of state-of-the-art scheduling policies.
- For **application developers**
  - GridWay implements the OGF standard DRMAA API (C and JAVA bindings), assuring compatibility of applications with LRM systems that implement the standard, such as SGE, Condor, Torque,...
- For **end users**
  - GridWay provides a LRM-like CLI for submitting, monitoring, synchronizing and controlling jobs, that could be described using the OGF standard JSDL.

1. What is GridWay?
- 2. A Global Vision**
3. Scheduling Policies
4. Scheduling Infrastructures
5. The GridWay Project

## Global Architecture of a Computational Grid



## Benefits

### **Integration of non-interoperable computational platforms (Organization)**

- Establishment of a uniform and flexible infrastructure
- Achievement of greater utilization of resources and higher application throughput

### **Support for the existing platforms and LRM Systems (Sys. Admin.)**

- Allocation of grid resources according to management specified policies
- Analysis of trends in resource usage
- Monitoring of user behavior

### **Familiar CLI and standard APIs (End Users & Developers)**

- High Throughput Computing Applications
- Workflows

## Features

### Workload Management

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

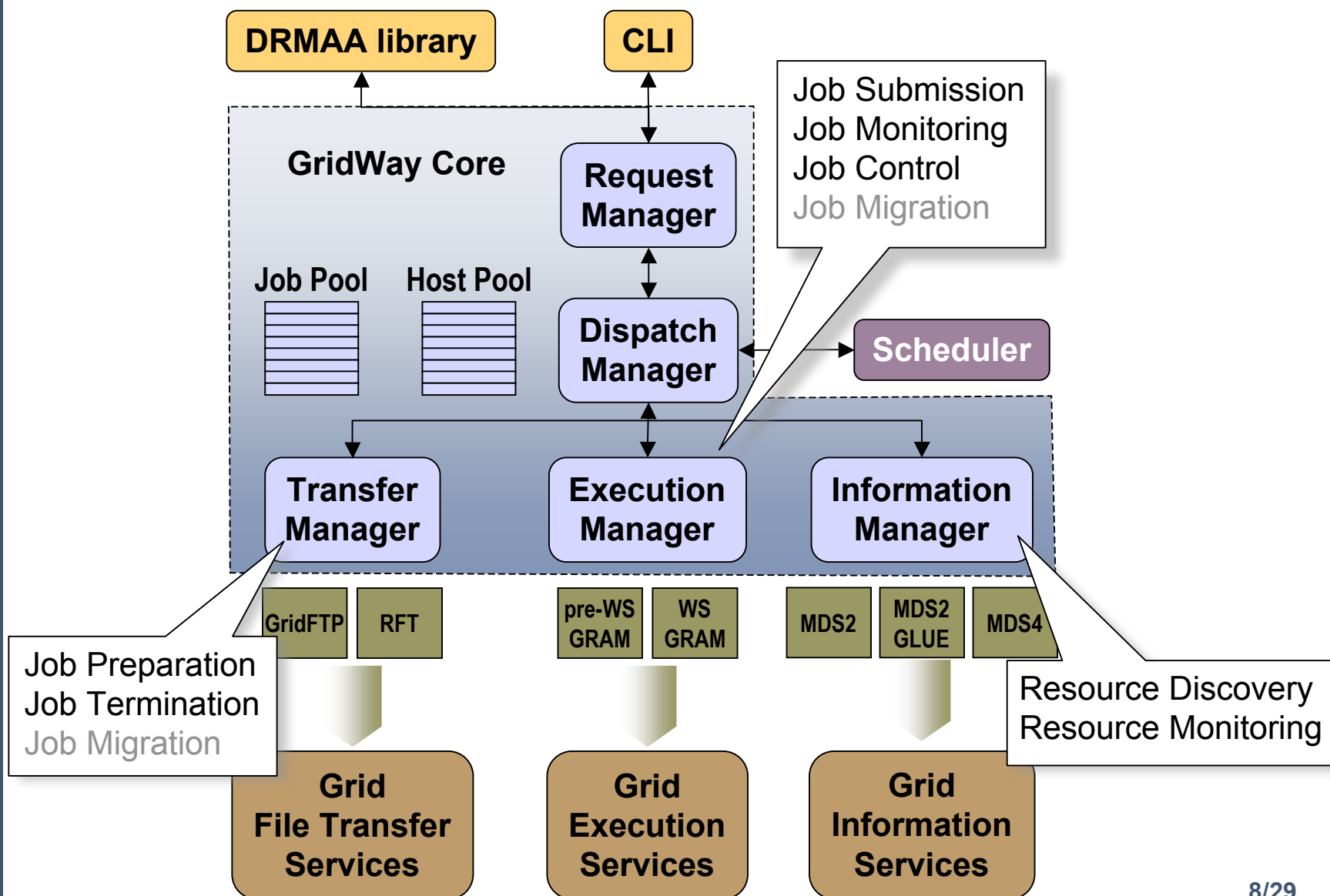
### User Interface

- OGF standards: JSDL & DRMAA (C and JAVA)
- Analysis of trends in resource usage
- Command line interface, similar to that found on local LRM Systems

### Integration

- Straightforward deployment as new services are not required
- Interoperability between different infrastructures

## GridWay Internals





1. What is GridWay?
2. A Global Vision
- 3. Scheduling Policies**
4. Scheduling Infrastructures
5. The GridWay Project

## Resource Policies

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

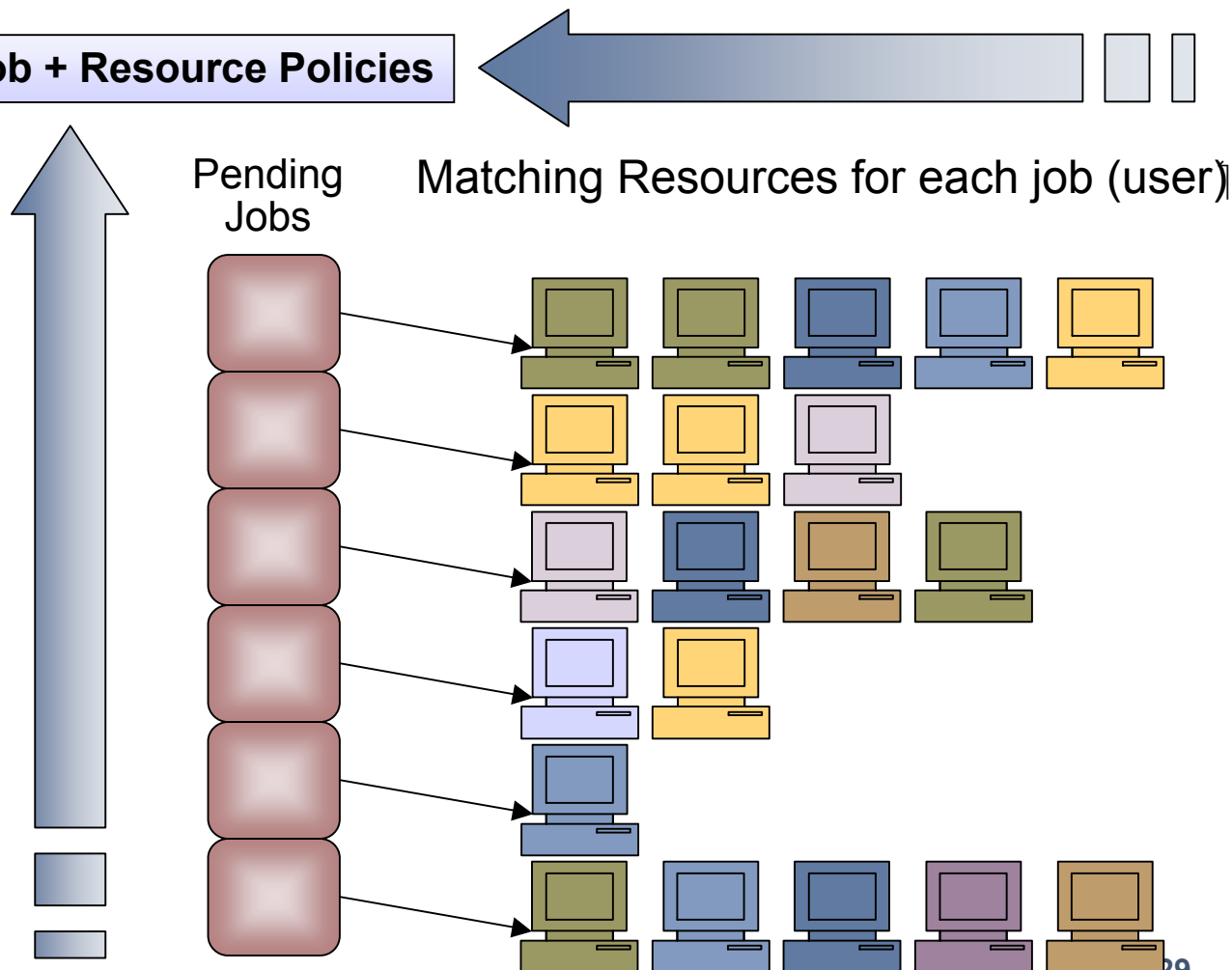
**Grid Scheduling = Job + Resource Policies**

## Job Policies

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

Pending Jobs

Matching Resources for each job (user)



1. What is GridWay?
2. A Global Vision
3. Scheduling Policies
- 4. Scheduling Infrastructures**
5. The GridWay Project

**Centralized  
Coupled**

- Network Links
- Administration
- Homogeneity

**Decentralized  
Decoupled**

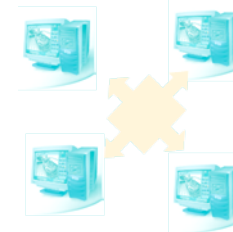
**SMP** (Symmetric  
Multi-processors)

**MPP** (Massive  
Parallel Processors)

**Clusters**

**Network Systems**  
Intranet/Internet

**Grid  
Infrastructures**



**High Performance Computing**

**High Throughput Computing**

## Enterprise Grid Infrastructures

### Characteristics

- “Small” scale infrastructures (campus/enterprise) with one meta-scheduler instance providing access to resources within the same administration domain that may be running different DRMS and be geographically distributed

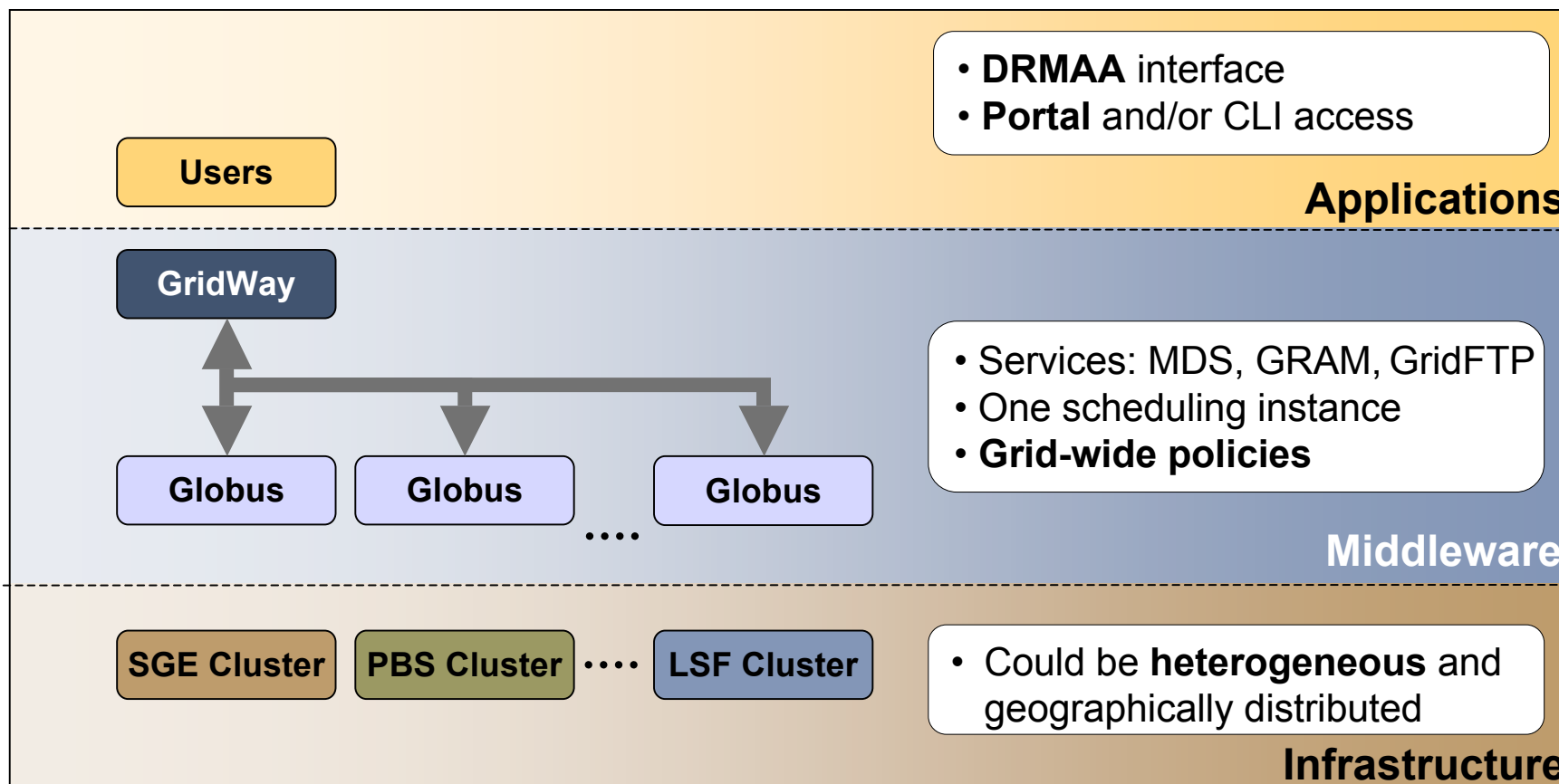
### Goal & Benefits

- Integrate multiple systems, that could be heterogeneous, in an *uniform/centralized* infrastructure
- Decoupling of applications and resources
- Improve return of IT investment
- Performance/Usage maximization

### Scheduling

- Centralized meta-scheduler that allows the enforcement of **Grid-wide policies** (e.g. resource usage) and provides **centralized accounting**

## Enterprise Grids: Deployment with GridWay

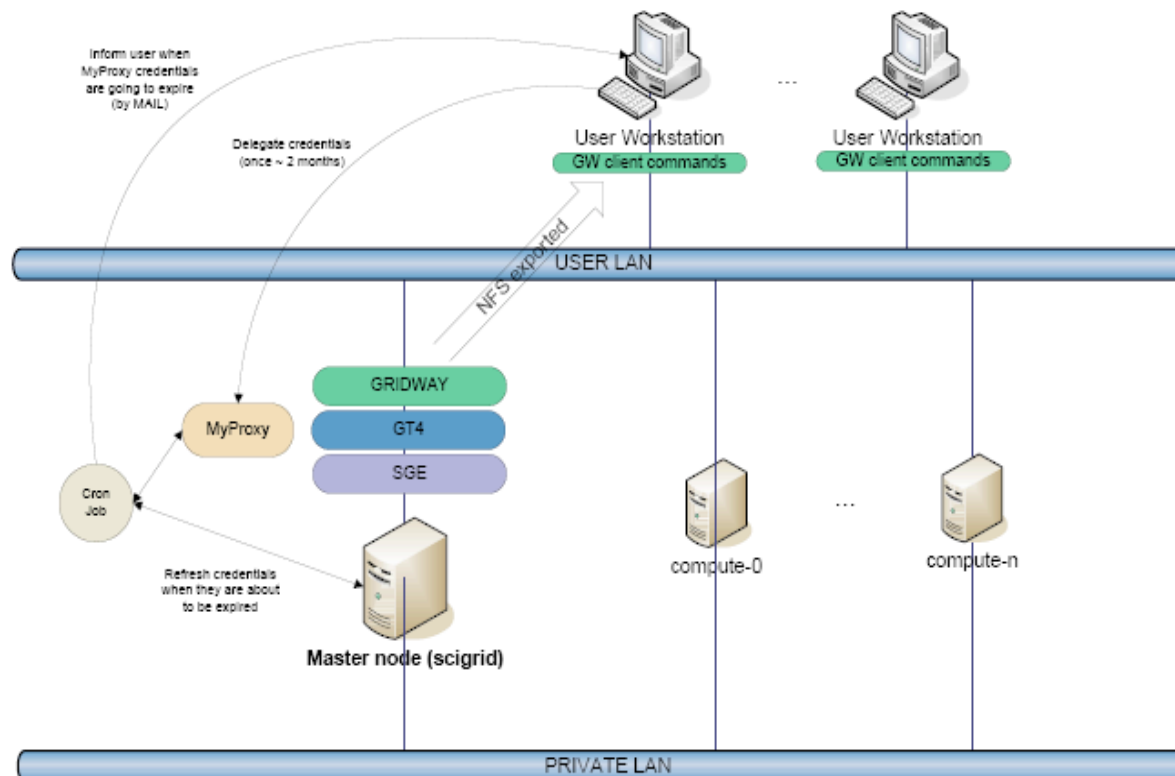


## Enterprise Grids: Examples

### European Space Astronomy Center



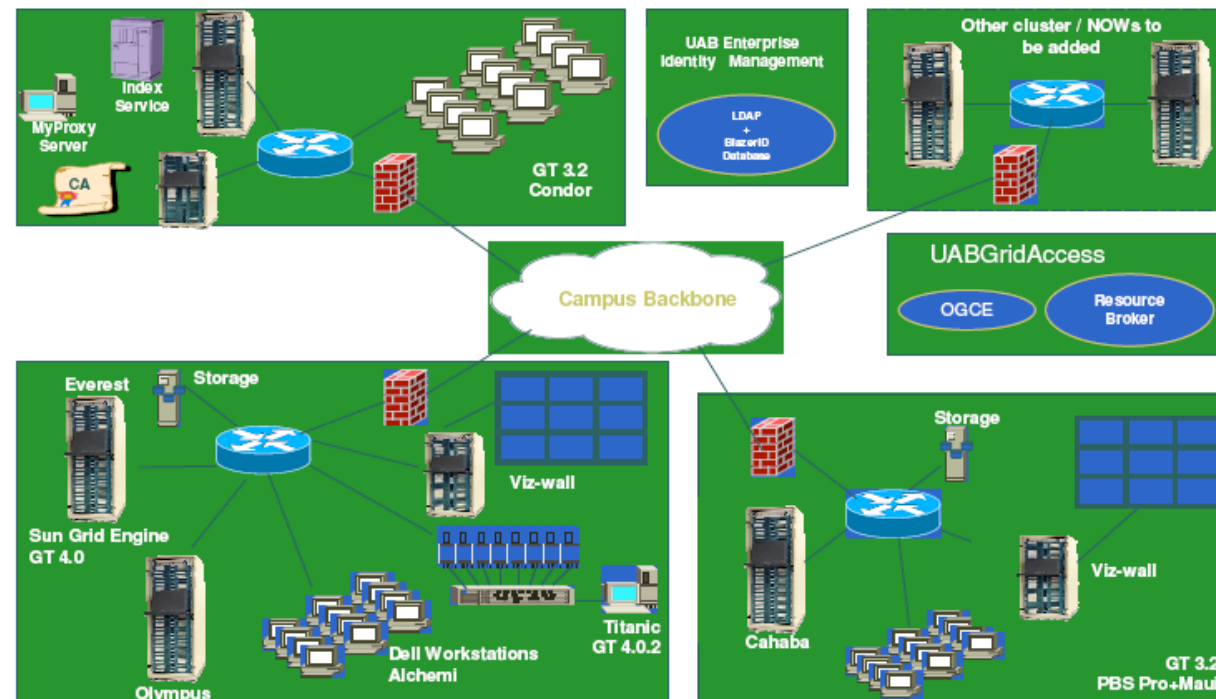
- Data Analysis from space missions (DRMAA)
- Site-level meta-scheduler
- Several clusters



## Enterprise Grids: Examples

### UABGrid, University of Alabama at Birmingham

- Bioinformatics applications
- Campus-level meta-scheduler
- 3 resources (PBS, SGE and Condor)





## Partner Grid Infrastructures

### Characteristics

- “Large” scale infrastructures with one or several meta-scheduler instances providing access to resources that belong to different administrative domains (different organizations or partners)

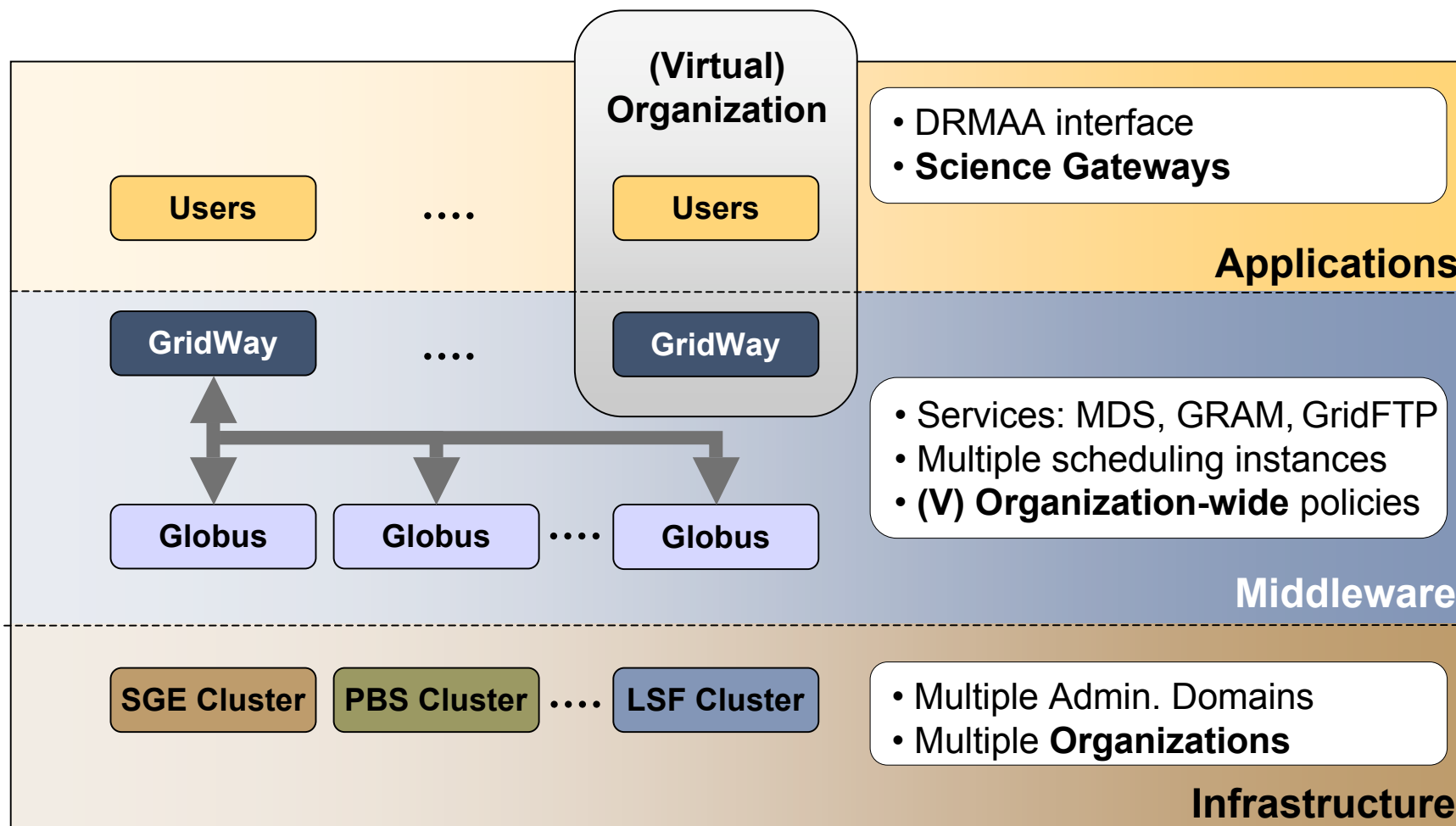
### Goal & Benefits

- Large-scale, secure and reliable sharing of resources between partners or supply-chain participants
- Support collaborative projects
- Access to higher computing power to satisfy peak demands

### Scheduling

- Decentralized scheduling system that allows the enforcement of **organization-wide** policies

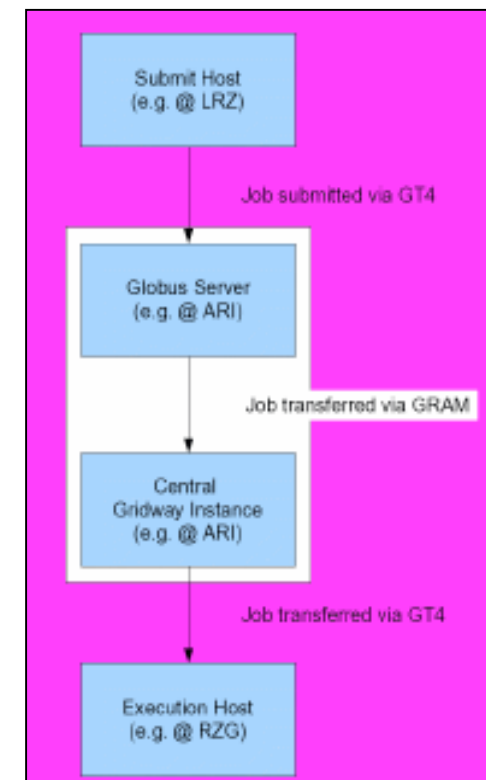
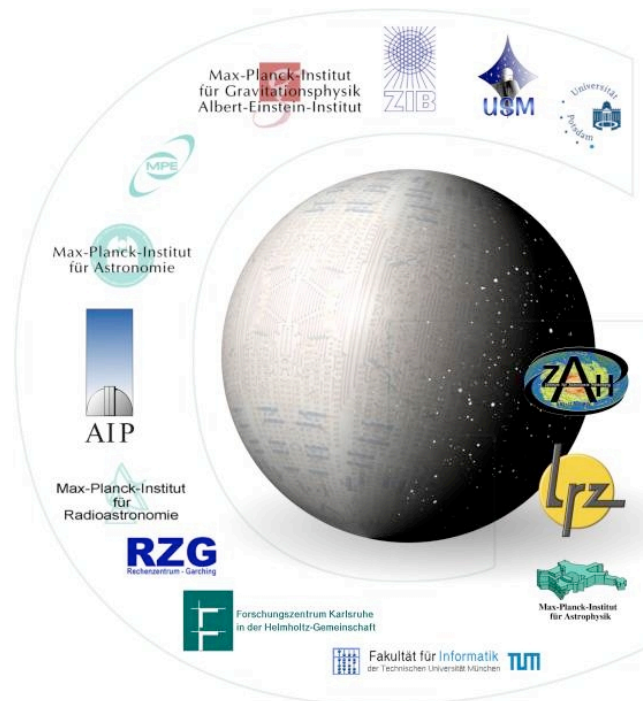
## Partner Grids: Deployment with GridWay



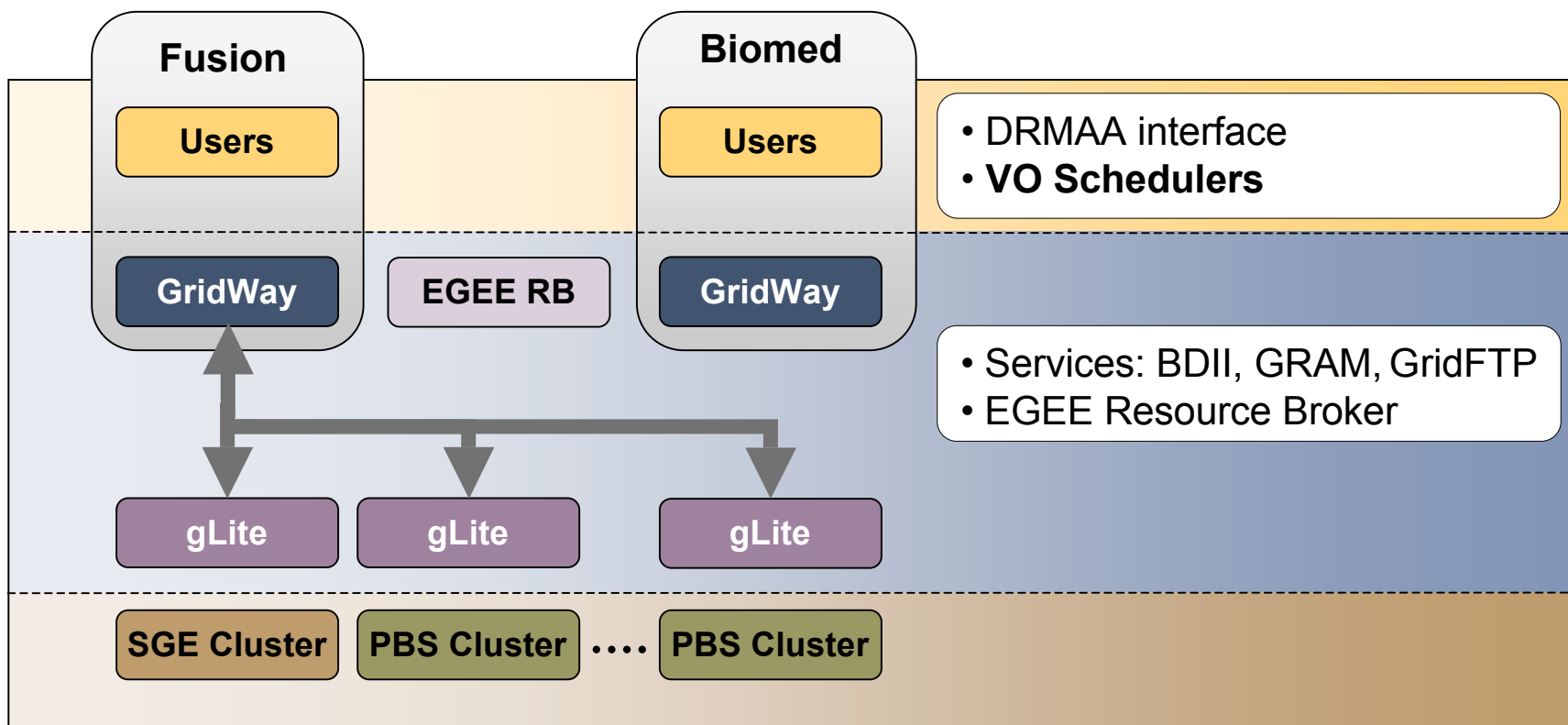
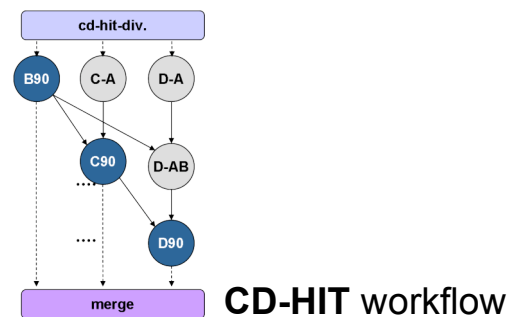
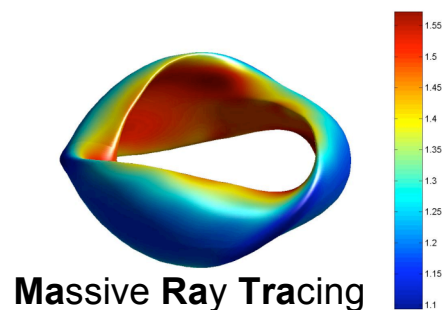
## Partner Grids: Examples

### AstroGrid-D, German Astronomy Community Grid

- Collaborative management of supercomputing resources & astronomy-specific resources
- Grid-level meta-scheduler (GRAM interface)
- 22 resources @ 5 sites, 800 CPUs

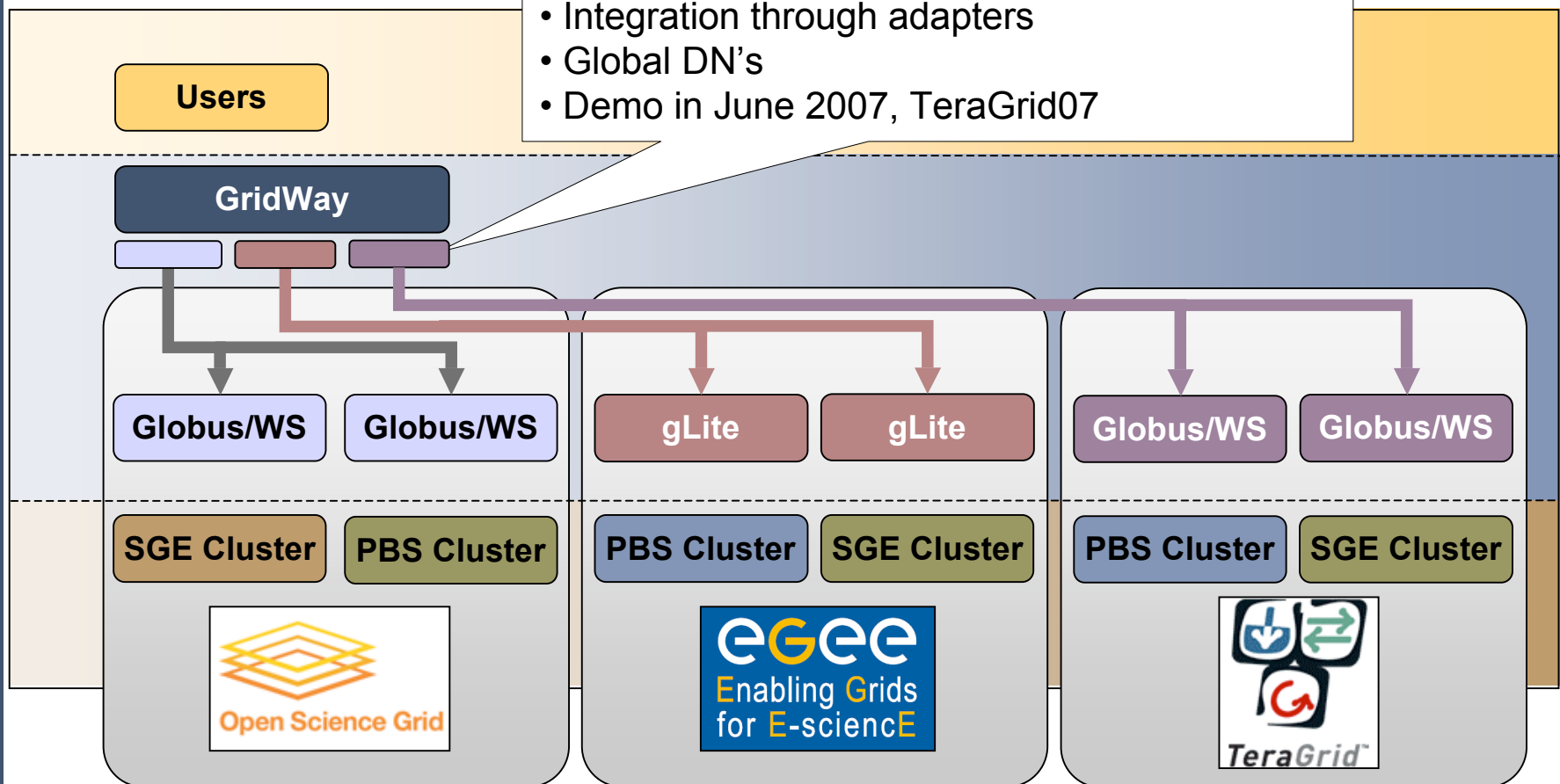


## Partner Grids: Examples



## A Tool for Interoperability

- Different Middlewares (e.g. WS and pre-WS)
- Different Data/Execution architectures
- Different Information models
- Integration through adapters
- Global DN's
- Demo in June 2007, TeraGrid07



## Utility Grid Infrastructures

### Characteristics

- Multiple meta-scheduler layers in a hierarchical structure
- Resource provision in a utility fashion (provider/consumer)

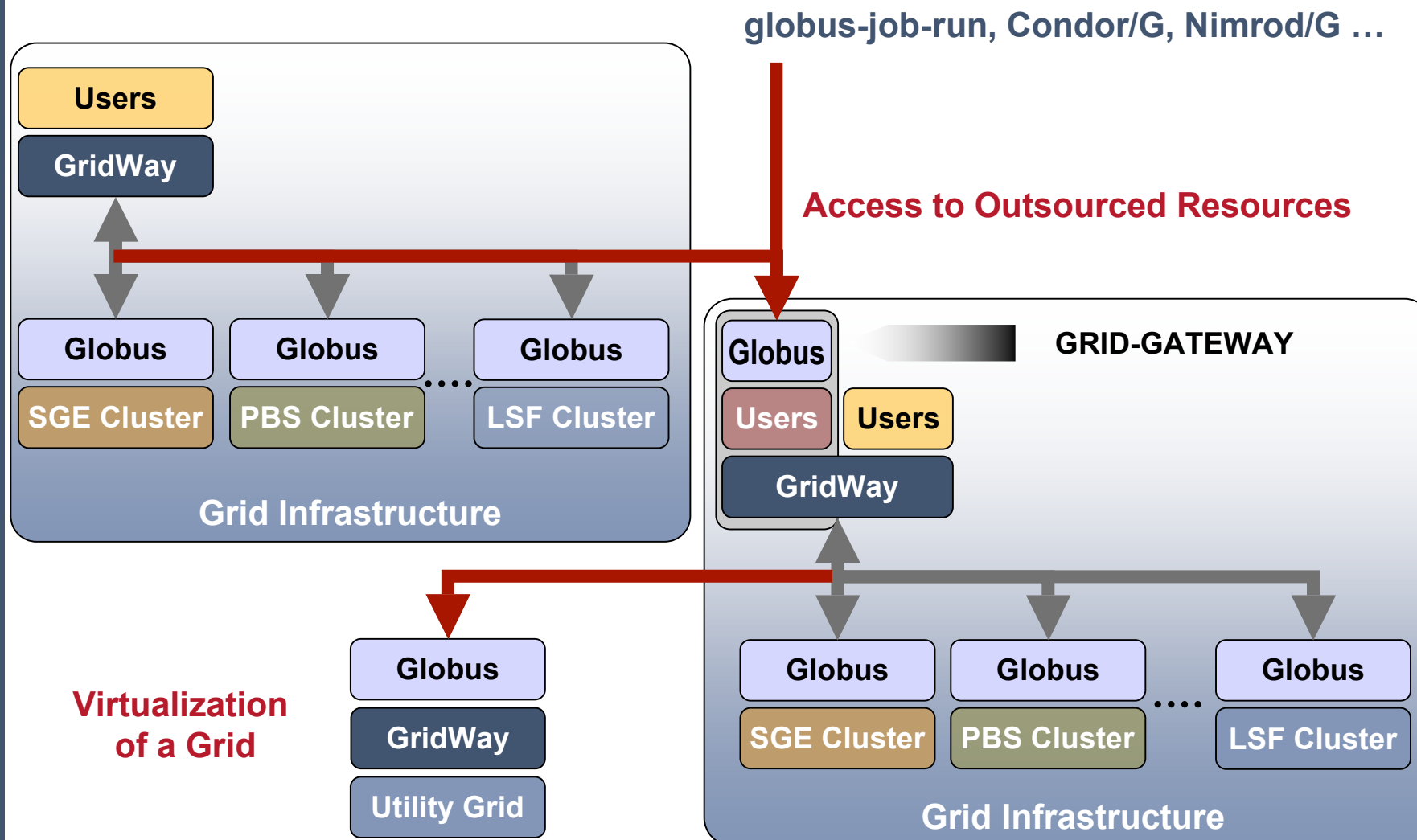
### Goal & Benefits

- Supply resources on-demand, making resource provision more adaptive
- Access to *unlimited* computational capacity
- Transform IT costs from fixed to variable
- Seamless integration of different Grids (The Grid)

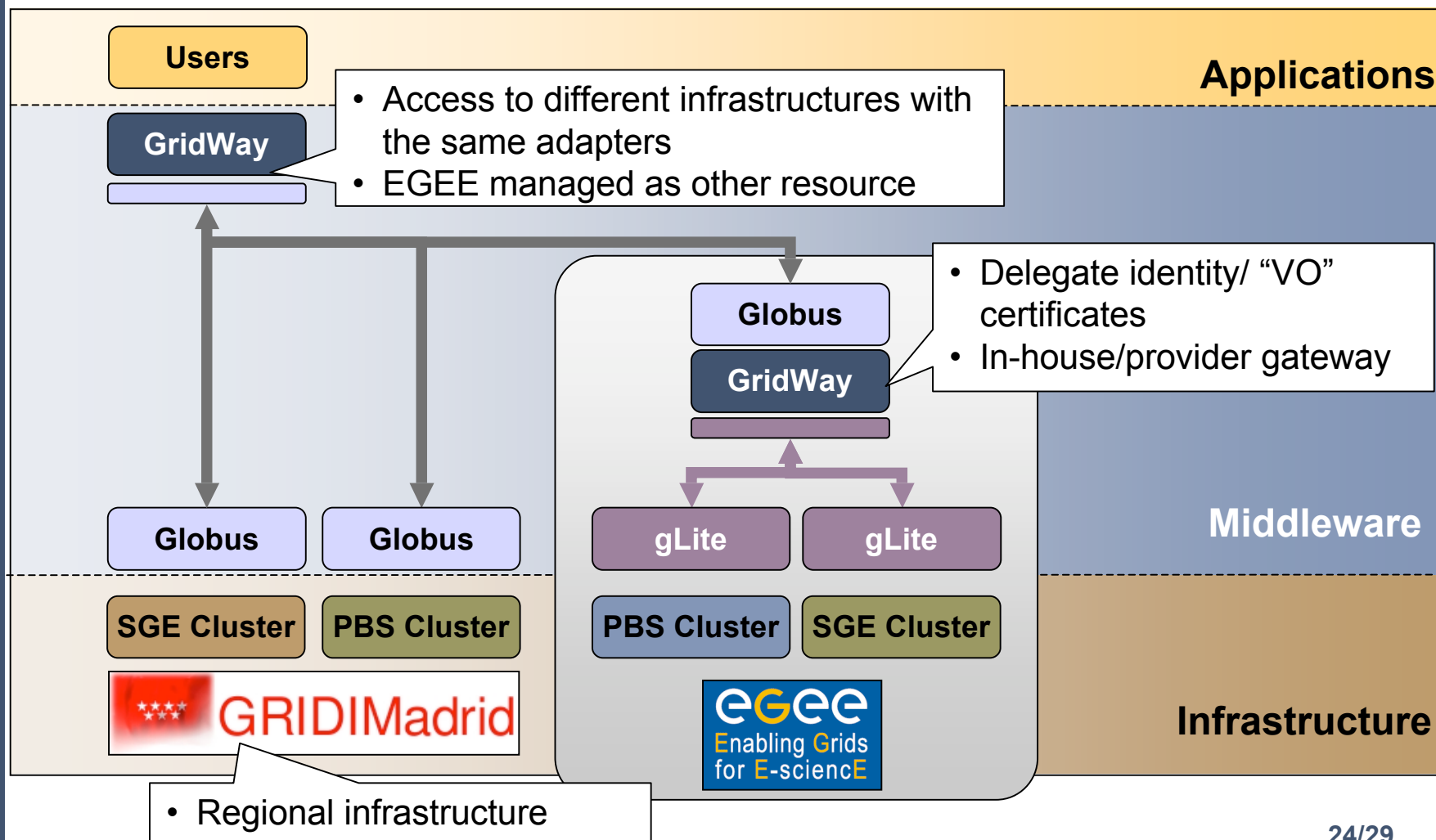
### Scheduling

- Each Grid is handled as any other resource
- Characterization of a Grid as a single resource
- Use standard interfaces to virtualize a Grid infrastructure

## Deploying Utility Grid Infrastructures with GridWay



## Utility Grids: Example





1. What is GridWay?
2. A Global Vision
3. Scheduling Policies
4. Scheduling Infrastructures
- 5. The GridWay Project**

## Some Projects and Infrastructures

- . IRISGrid
- . Politecnico di Torino
- . CABGrid (Centro de Astrobiología)
- . C2VO (Universidad de Castilla La Mancha)
- . Grid en ESAC (Agencia Espacial Europea)
- . CRO-GRID (Croacia)
- . Sun Microsystems Solution Center World Grid
- . Infraestructura EGEE
- . Proyecto BeinGRID
- . GridX1 (Canadian Grid for HEP applications)
- . Universidade do Porto
- . Madras Institute of Technology
- . National Center for High-Performance Computing



## Some Application Porting Areas

- . Life-Sciences
- . Aerospace
- . Fusion Physics
- . Computational Chemistry

## History of the Project

- Started in **2002**, first releases were only distributed on request in binary format
- First open source release (v4.0) in **January 2005** (Apache license v2.0)
- Adhering to Globus philosophy and guidelines for **collaborative development**
- In June 2007 GridWay became part of the **Globus Toolkit**
- Since January 2005, more than **1000 downloads from 80 different countries**, 25% are private companies and 75% are universities and research centers.
- Best-effort support provided (contract support is also available)
  - **Based on a strong open source community**

## Development Process

- **Community** – Open Source Project. Globus Development Philosophy
- **Development Infrastructure** (thanks to Globus Project!)
  - Mailing Lists
  - Bugzilla
  - CVS
- **You are very welcome to contribute:**
  - Reporting Bugs ([gridway-user@globus.org](mailto:gridway-user@globus.org))
  - Making feature requests for the next GridWay release ([gridway-user@globus.org](mailto:gridway-user@globus.org))
  - Contributing your own developments (bug fixes, new features, documentation)
- Detailed **Roadmap**:
  - **GridWay Campaigns** at [bugzilla.mcs.anl.gov/globus/query.cgi](http://bugzilla.mcs.anl.gov/globus/query.cgi)
  - [www-unix.mcs.anl.gov/~bacon/cgi-bin/big-roadmap.cgi#Gridway](http://www-unix.mcs.anl.gov/~bacon/cgi-bin/big-roadmap.cgi#Gridway)

**Thank you  
for your attention!**