February 14, 2006
GSA-WG at GGF16
Athens, Greece

GridWay Scheduling Architecture

Ignacio Martín Llorente
GridWay Project
www.GridWay.org

Distributed Systems Architecture Group
Departamento de Arquitectura de Computadores y Automática
Universidad Complutense de Madrid

Contents

1. Scheduling Requirements for Partner Grid Computing
2. GridWay Functionality
3. GridWay Scheduling Architecture
4. Scheduling Requirements on Core Grid Services
5. Alternative Scheduling Infrastructures Based on GridWay

1. Scheduling Requirements for Partner Grid Computing

Partner Grid Computing

Infrastructure

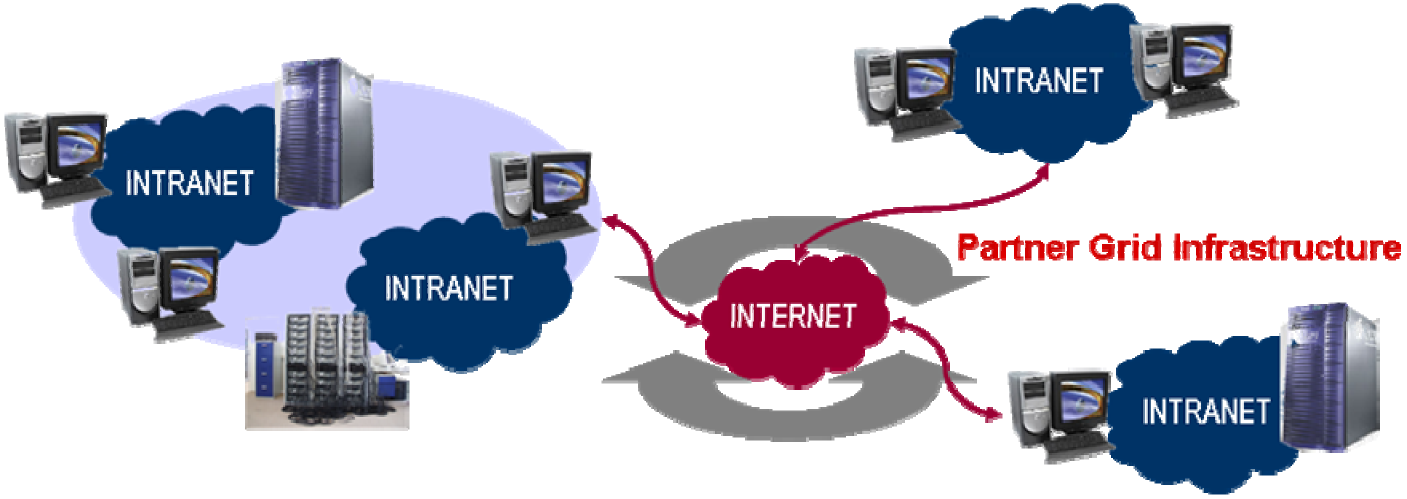
- Resources scattered across several organizations or administrative domains managed by different DRM systems

Objective

- Provide large-scale, secure and reliable sharing of resources among partner organizations and supply-chain participants

Benefits

- Access to a higher computing performance to satisfy peak demands
- Support to face collaborative projects



1. Scheduling Requirements for Partner Grid Computing

Partner Grid Stakeholders

System Managers

- Responsible for infrastructure deployment
- Responsible for implementation of local resource sharing and access policies

End-users

- Execution of jobs
- Development of distributed applications

Requirement Analysis Sources

Our Expertise in the Deployment of Grid Infrastructures

- http://asds.dacya.ucm.es

User Community Feedback at GridWay Forum

- http://www.gridway.org/forum/

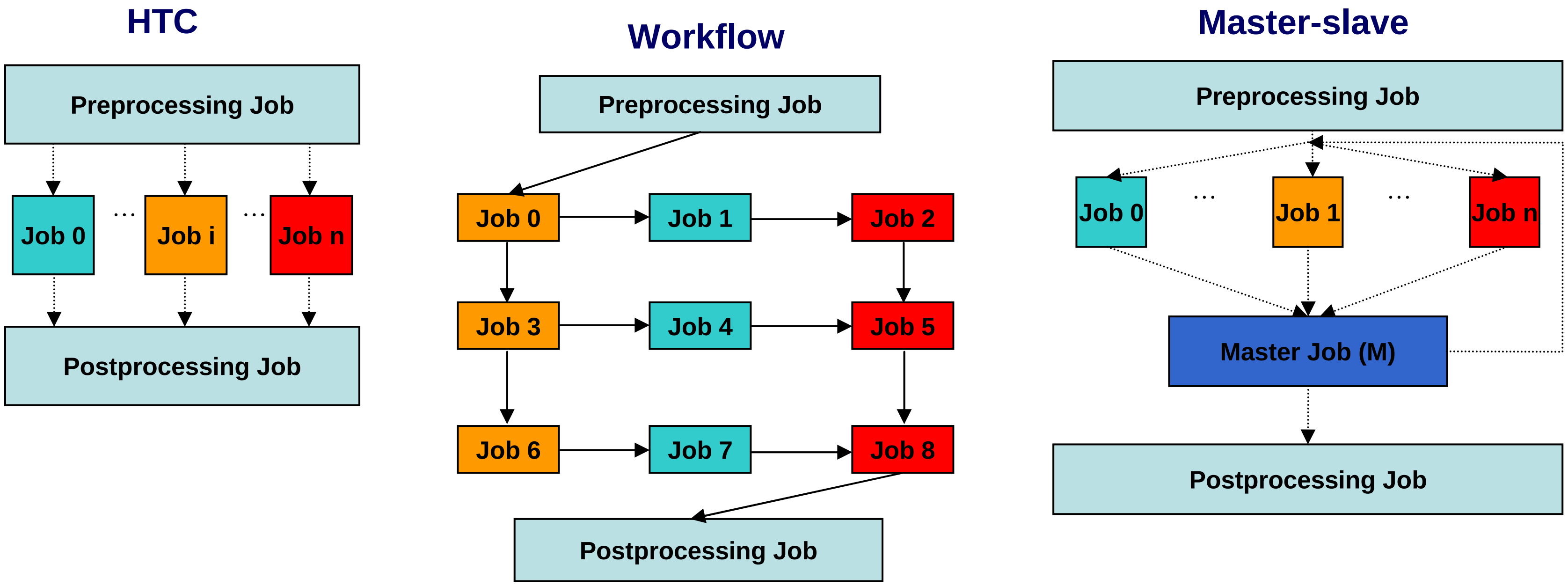
Grid Solutions Based on GridWay

- http://www.gridway.org/solutions.php

End-user Requirements (1/2)

Common Application Profiles

- Submission of single, array or complex jobs consisting of task dependencies, which may require file transferring and/or database access



- Reuse of existing job executables
- The scheduler must not require application deployment on remote hosts

End-user Requirements (2/2)

Reliable and Unattended Execution of Jobs

- Transparently to the end user, the scheduler must be able to manage the different failure situations

Efficient Execution of Jobs

- The jobs must be executed on the “faster” available resources

DRM Command Line Interface

- Similar to that found on Unix and distributed resource management (DRM) systems such as PBS or SGE

Standard Application Programming Interface

- Support of standard API for the development of distributed applications

System Manager Requirements

Distributed Scheduling

- Multiple administration domains prevent the deployment of centralized resource brokers, with total control over client requests and resource status
- Reporting & accounting in each scheduling instance

Easy Deployment

- The scheduler must require the minimum number of new services
- The scheduler must support the existing remote platforms and resource managers (fork, PBS, SGE, LSF, LoadLeveler, Condor…)

Site Autonomy

- Grid resources belong to different administrative domains; so that, once a job is submitted, it can be freely canceled by the resource owner
- The resources can be added or removed continuously

Dynamic Environment Management

- Grid resources may be simultaneously exploited by other grid users, as well as by internal users

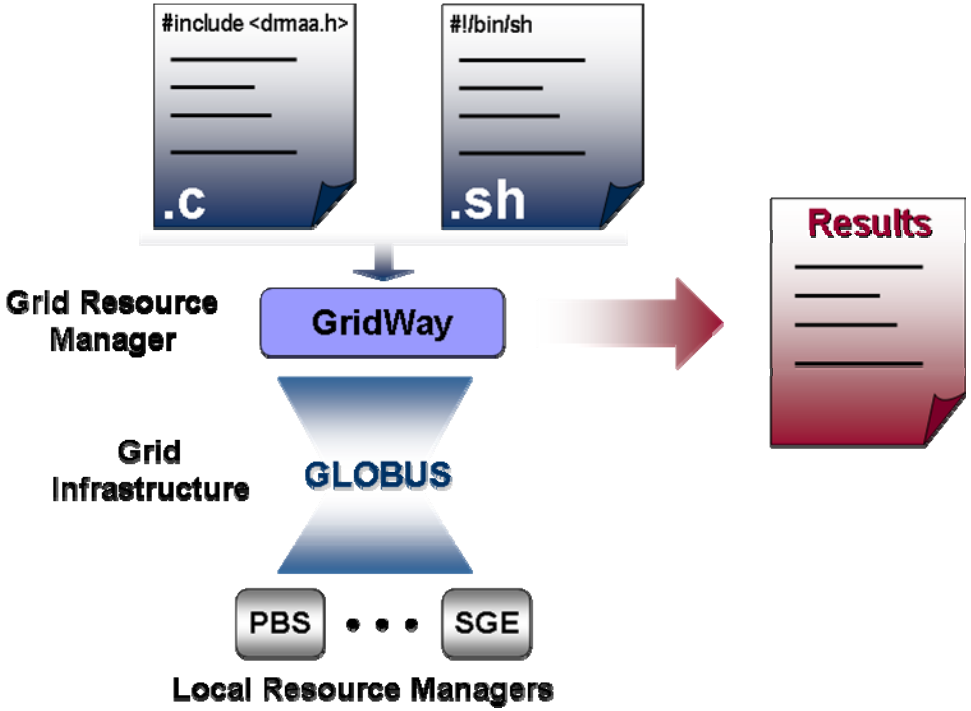
Adaptability and Extension Capability

- The scheduler must provide a modular architecture to allow communication with different resource management, file management and information services
- The scheduler could be extended or used as a building block for more complex architectures that implements service-level agreements (SLAs) or advanced reservation.

What is GridWay?

GridWay is an ***open source meta-scheduling technology*** that performs job execution management and resource brokering on heterogeneous and dynamic Grids based on Globus Toolkit services

The GridWay Concept



Scheduling Capabilities

Dynamic Scheduling

- Grid resources are highly dynamic, both in load and availability. Dynamic scheduling matches job requirements to the static and dynamic load attributes of the resources available in each scheduling

Opportunistic Migration

- Opportunistic migration means automatic resubmission of running jobs to better resources. The scheduler evaluates the performance benefits that can be obtained due to the migration of the job

Performance Slowdown Detection

- The scheduler requests a migration when a intolerable performance loss for a job (performance contract violation) is detected

Support for Self-adaptive Applications

- An application is able to take decisions about resource selection as its execution evolves by modifying its requirement and rank expressions and requesting a migration

Checkpointing Support

- Checkpointing is required to support migration on-request, opportunistic migration and fault tolerance

Scheduling Policies, Reporting and Accounting

- Scheduling policies define the balance of workload among the available resources.
- Scheduling reporting and accounting facilities provide information about overall performance and help troubleshoot configuration problems.

Fault Detection & Recovery Capabilities

Job Cancellation

- A Job could be cancelled for several reasons, for example by the local resource management system when it exceeds the wall time limit or by the system administrator to preserve system performance

Remote System Crash or Outage

- Grid resources could unpredictably fail. These failures comprise hardware, operating system and Grid middleware components

Network Disconnection

- Grid connections could unpredictably fail. Moreover, system administrators are freely to disconnect resources, for example, due to local site maintenance

Client Fault Tolerance

- The system running the scheduler could fail

User Interface Functionality

Broad Application Scope

- The scheduler is not bounded to a specific class of application generated by a given programming environment and does not require application deployment on remote hosts, which extends its application range and allows reusing of existing software

DRM Command Line Interface

- The scheduler command line interface is similar to that found on Unix and DRM systems such as PBS or SGE. It allows users to submit, kill, migrate, monitor and synchronize jobs

DRMAA Application Programming Interface

- The scheduler provides full support for **DRMAA** (GGF standard) to develop distributed applications (**C** and **JAVA** bindings)

Installation & Configuration Issues

Modular Architecture

- The scheduling modular architecture, based on MADs (Middleware Access Drivers) allows communication with the resource management, file management and information services available in a given infrastructure

Requirements on Core Grid Services

- The scheduler is installed on the client system and does not require the installation or deployment of new services in the remote resource, apart from Globus services

Supported Remote Services

- The scheduler includes MADs to interface to Globus pre-WS and WS GRAM, GridFTP, RFT, MDS 2.0 and MDS 4.0 services

Supported Client Platforms

- The scheduler has been tested on Linux and Solaris

Decentralized Architecture

- The scheduler decentralized architecture gives the possibility of boosting scalability

Security

- Globus Toolkit GSI infrastructure
- A global name space is not assumed

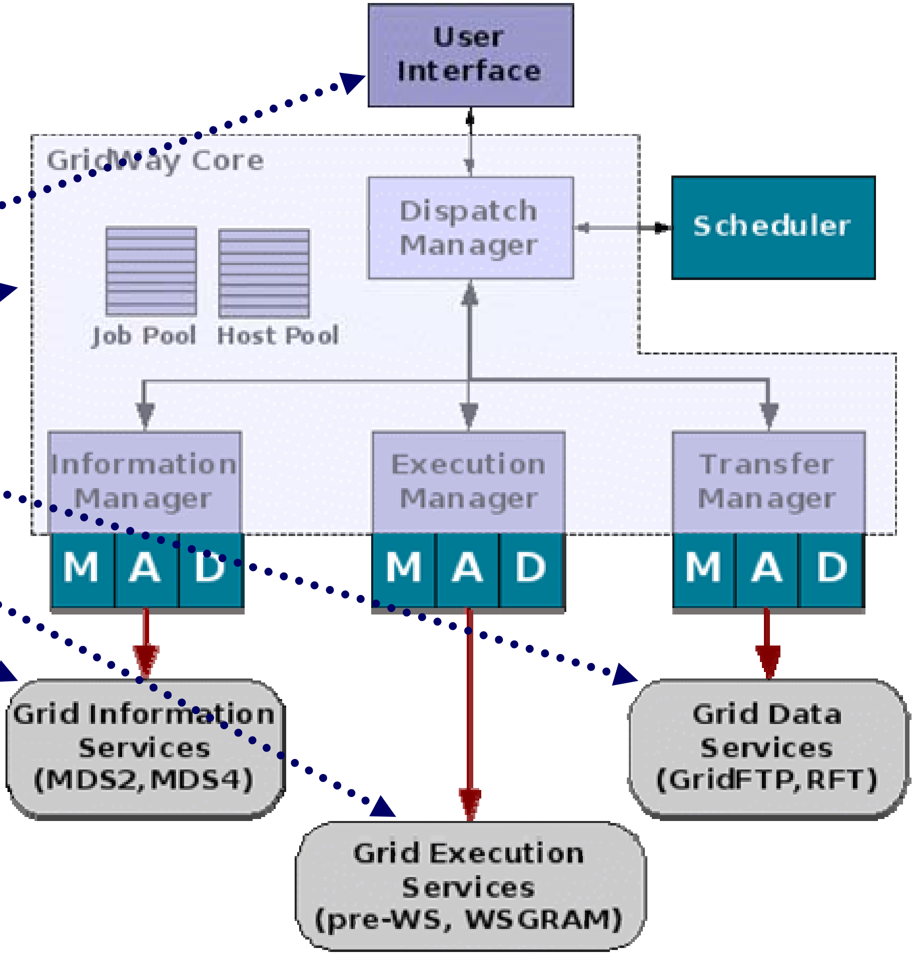
Requirements vs. Functionality Table

			GridWay Functionality																		
			Scheduling						Fault Recovery				User			Installation/Configuration					
			Dynamic Scheduling	Opportunistic Migration	Slowdown Detection	Self-adaptive Applications	Checkpointing Support	Policies, Reporting and Accounting	Job Cancellation	Remote System Crash or Outage	Network Disconnection	Client Fault Tolerance	Broad Application Scope	DRM Command Line Interface	DRMAA Application Program. Interface	Modular Architecture	Requirements on Core Grid Services	Supported Remote Services	Supported Client Platforms	Decentralized Architecture	Security
Stakeholders Requirements	End-user	Common Application Profile											■	■	■						
		Reliable/Unattended Execution of jobs					■		■	■	■	■									
		Effcient Execution of jobs	■	■	■	■	■														
		DRM Command Line Interface												■							
		Standard Application Program. Interface													■						
	System Manager	Distributed Scheduling						■												■	
		Easy Deployment															■	■	■		■
		Site Autonomy							■	■	■										
		Dynamic Environment Management	■	■	■	■	■														
		Adaptability and Extension														■					

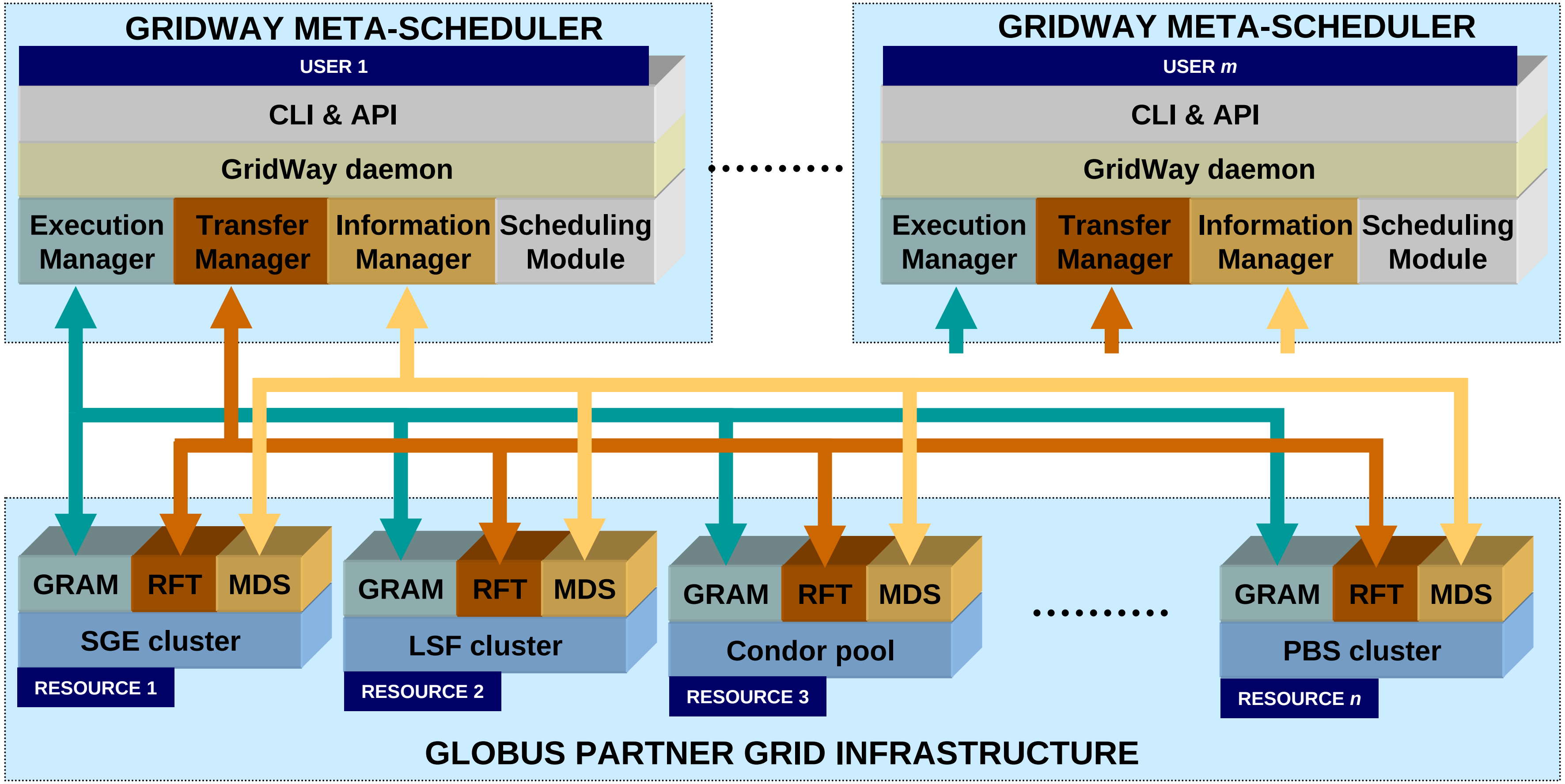
GridWay Architecture

Acting Entities in the GSA (GGF GSA-RG)

- Scheduling Access Client
- Grid Scheduling Instance
- Data Management Services
- Job Management Services
- Monitoring Services



GridWay Integration



Current Requirements (GGF GSA Requirement Document)

	GridWay Requirements	GridWay Component	Current Implementation (GT4.7)
Information Services	Static and dynamic load attributes of network and processing resources	Information Manager	MDS 2.0 GRIS and MDS 4.0 Index Service
Job Description	Job requirements & rank, and configuration about scheduling failure recovery, files, database access, streams…	Dispatch Manager	End-user or DRMAA (C and JAVA)
Resource Discovery	Index service	Information Manager	MDS 2.0 GIIS and MDS 4.0 Index Service
Job Management	Uniform service interface for remote job submission and control	Execution Manager	Pre-WS and WS GRAM
Monitoring	Uniform service interface for remote job monitoring	Execution Manager	Pre-WS and WS GRAM
Interaction with Data Management	Uniform service interface to file-based storage systems	Transfer Manager	GridFTP and RFT

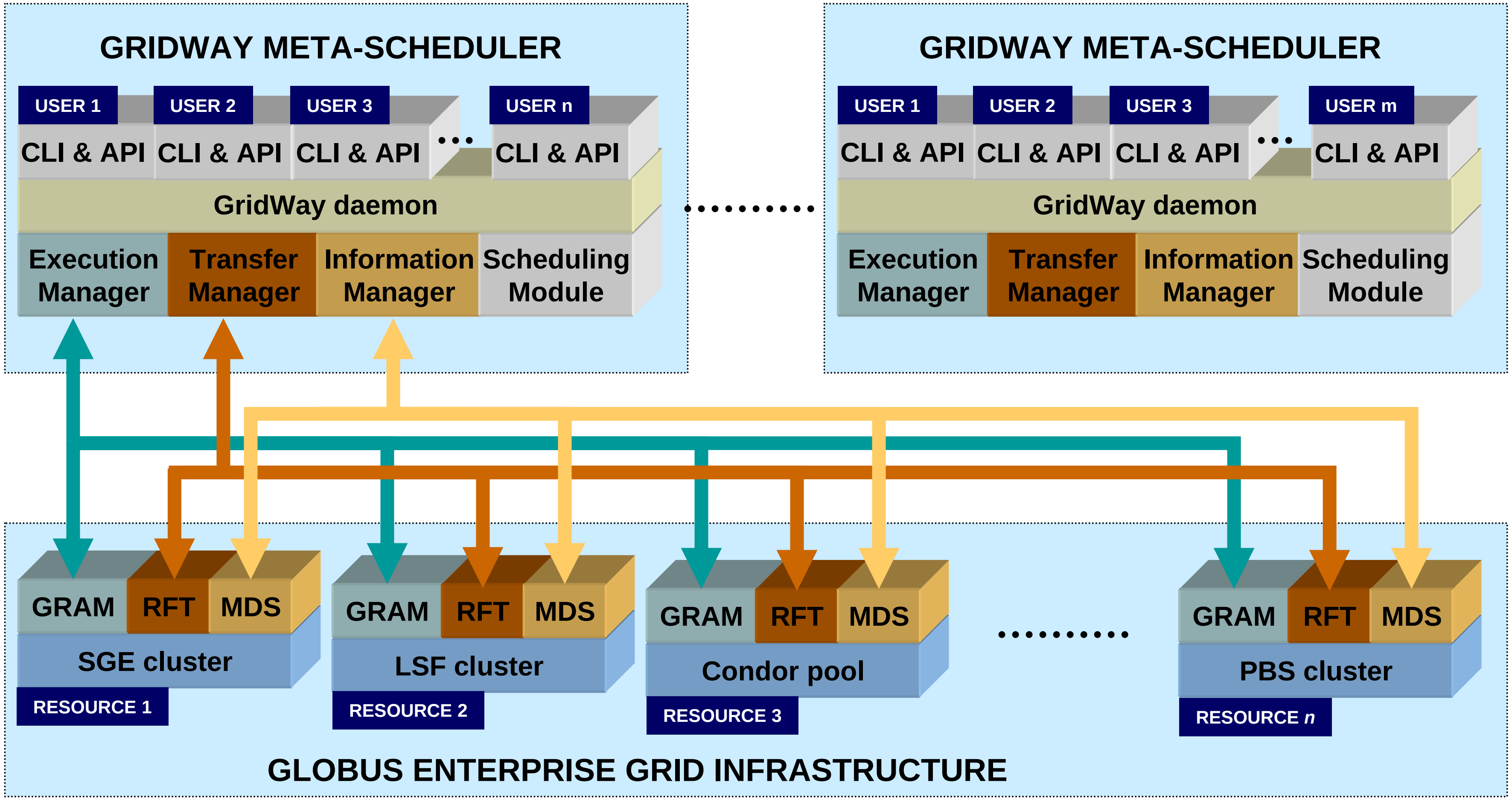
- Other services, such as reservation and negotiation, accounting and billing, network management and performance prediction, would considerably improve GridWay functionality

Scheduling Infrastructures Based on Multi-user GridWay

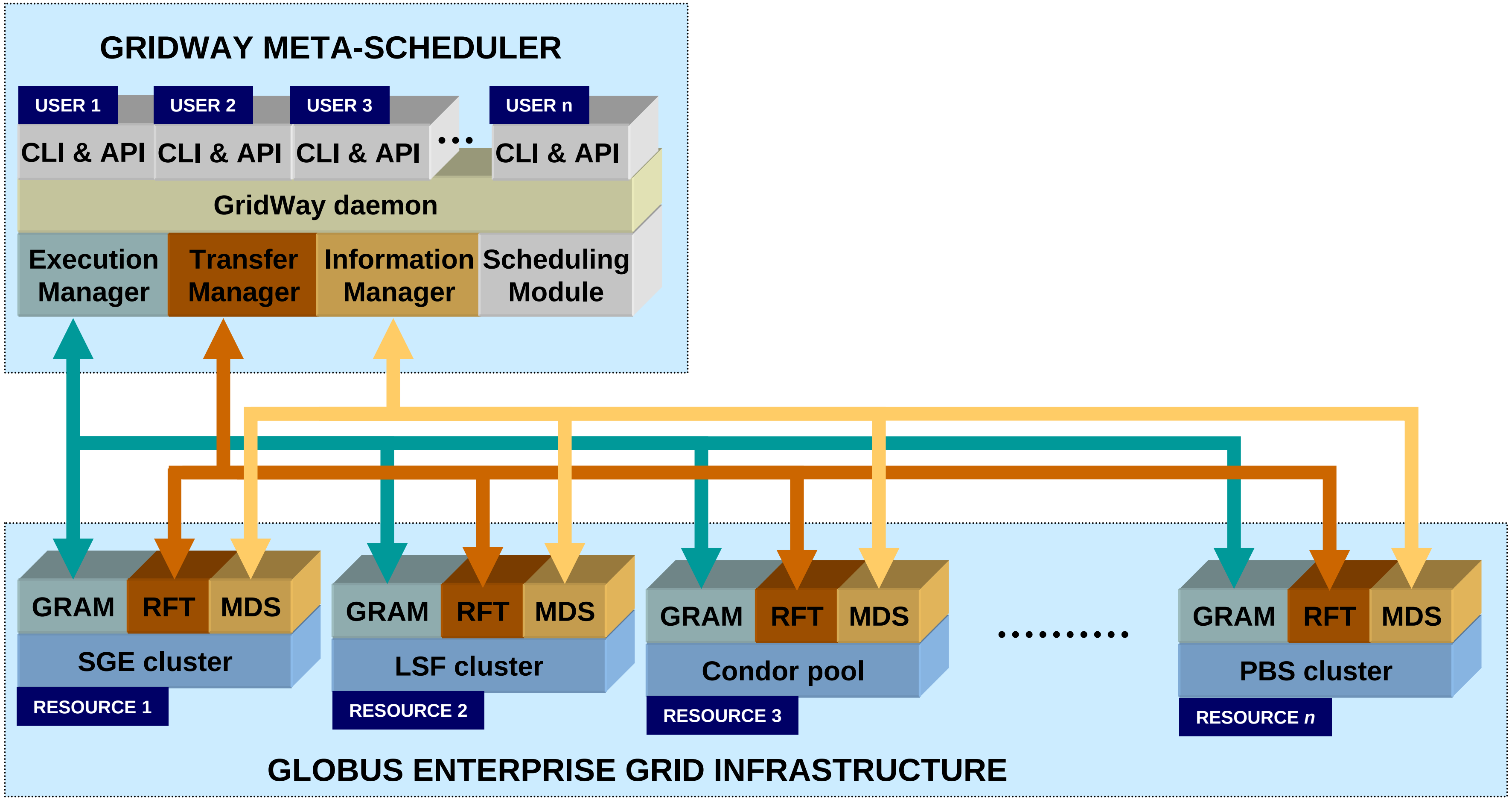
Multi-user GridWay (to be available in 5.0 release, scheduled for April 2006)

- Decentralized VO scheduling for partner grids
- Centralized scheduling for enterprise grids
- Decentralized Scheduling for Federated Grids (Partner/Outsourced)

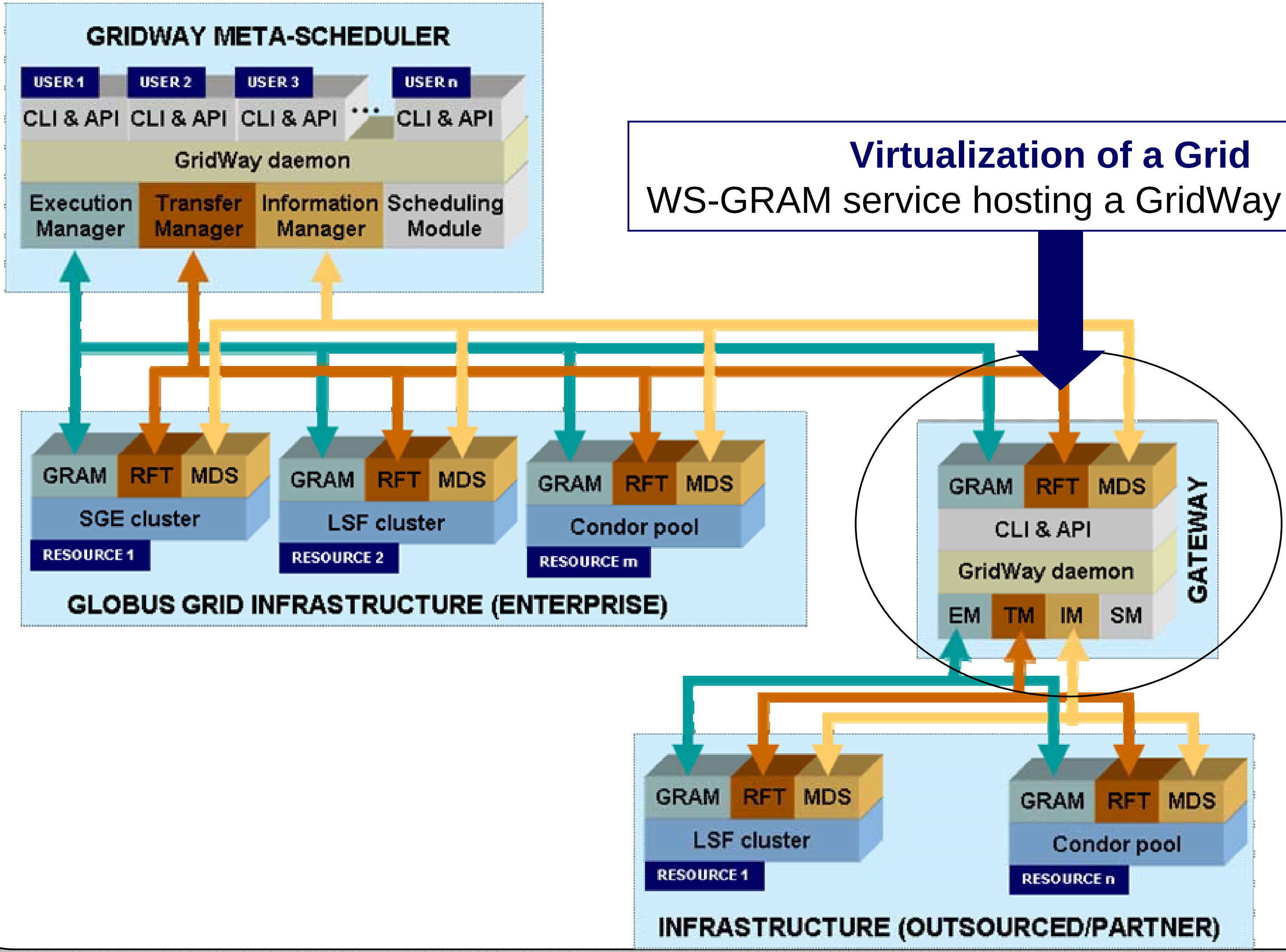
Decentralized VO Scheduling for Partner Grids



Centralized Scheduling for Enterprise Grids



Decentralized Scheduling for Federated Grids (Partner/Outsourced)



Virtualization of a Grid

WS-GRAM service hosting a GridWay scheduler

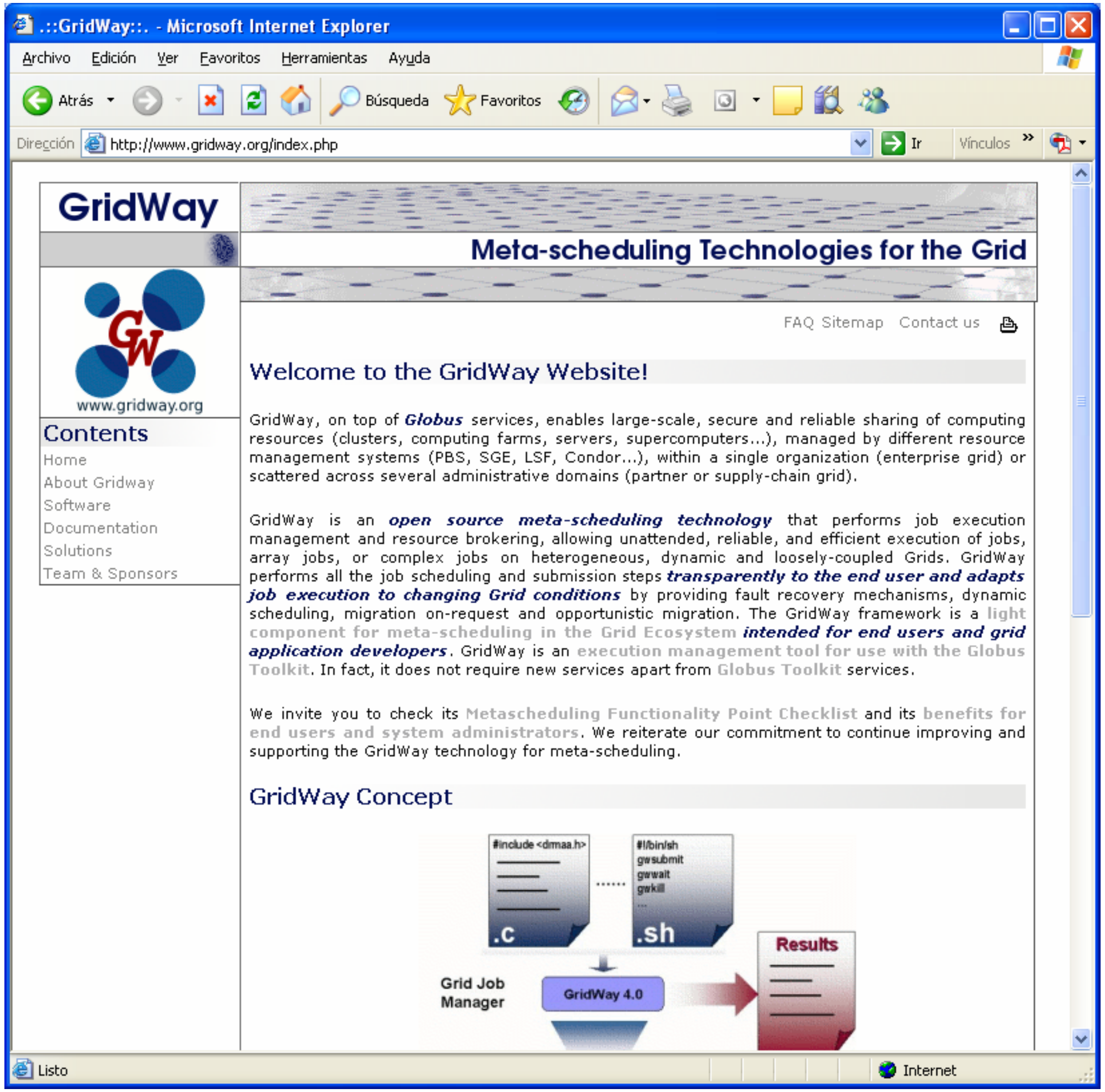
Grid Infrastructures Using GridWay

- **IRISGrid:** The Spanish National Grid Infrastructure
- **CABGrid:** A Virtual Laboratory for Computational Astrobiology
- **C2VO:** Grid infrastructure development for the implementation of a Computational Chemistry Virtual Organization
- Grid Activities at **ESAC (European Space Agency)**
- **CRO-GRID** Infrastructure
- **Sun Solution Center World Grid**
- **…**

More information at http://www.gridway.org/solutions.php

GridWay Scheduling Architecture

Information and download at http://www.GridWay.org
Open source license



GridWay

Meta-scheduling Technologies for the Grid

www.gridway.org

Contents

- Home
- About Gridway
- Software
- Documentation
- Solutions
- Team & Sponsors

FAQ Sitemap Contact us

Welcome to the GridWay Website!

GridWay, on top of ***Globus*** services, enables large-scale, secure and reliable sharing of computing resources (clusters, computing farms, servers, supercomputers...), managed by different resource management systems (PBS, SGE, LSF, Condor...), within a single organization (enterprise grid) or scattered across several administrative domains (partner or supply-chain grid).

GridWay is an ***open source meta-scheduling technology*** that performs job execution management and resource brokering, allowing unattended, reliable, and efficient execution of jobs, array jobs, or complex jobs on heterogeneous, dynamic and loosely-coupled Grids. GridWay performs all the job scheduling and submission steps ***transparently to the end user and adapts job execution to changing Grid conditions*** by providing fault recovery mechanisms, dynamic scheduling, migration on-request and opportunistic migration. The GridWay framework is a light component for meta-scheduling in the Grid Ecosystem ***intended for end users and grid application developers***. GridWay is an execution management tool for use with the Globus Toolkit. In fact, it does not require new services apart from Globus Toolkit services.

We invite you to check its Metascheduling Functionality Point Checklist and its benefits for end users and system administrators. We reiterate our commitment to continue improving and supporting the GridWay technology for meta-scheduling.

GridWay Concept

More Information and Tutorials

Grid Ecosystem at **Globus** site

Tutorial at **IBM** site

Installation on Solaris at **Sun Microsystems** site

DRMAA support and scheduling use case at **GGF** site

Thank you
for your attention!

More information…
http://www.GridWay.org
llorente@dacya.ucm.es