



## GridWay in IGE

...



**Dr Eduardo Huedo Cuesta**  
**Distributed Systems Architecture Group**  
**Universidad Complutense de Madrid**  
e-mail: [ehuedo@fdi.ucm.es](mailto:ehuedo@fdi.ucm.es) / [eduardo.huedo@ige-project.eu](mailto:eduardo.huedo@ige-project.eu)



**Initiative for Globus in Europe**

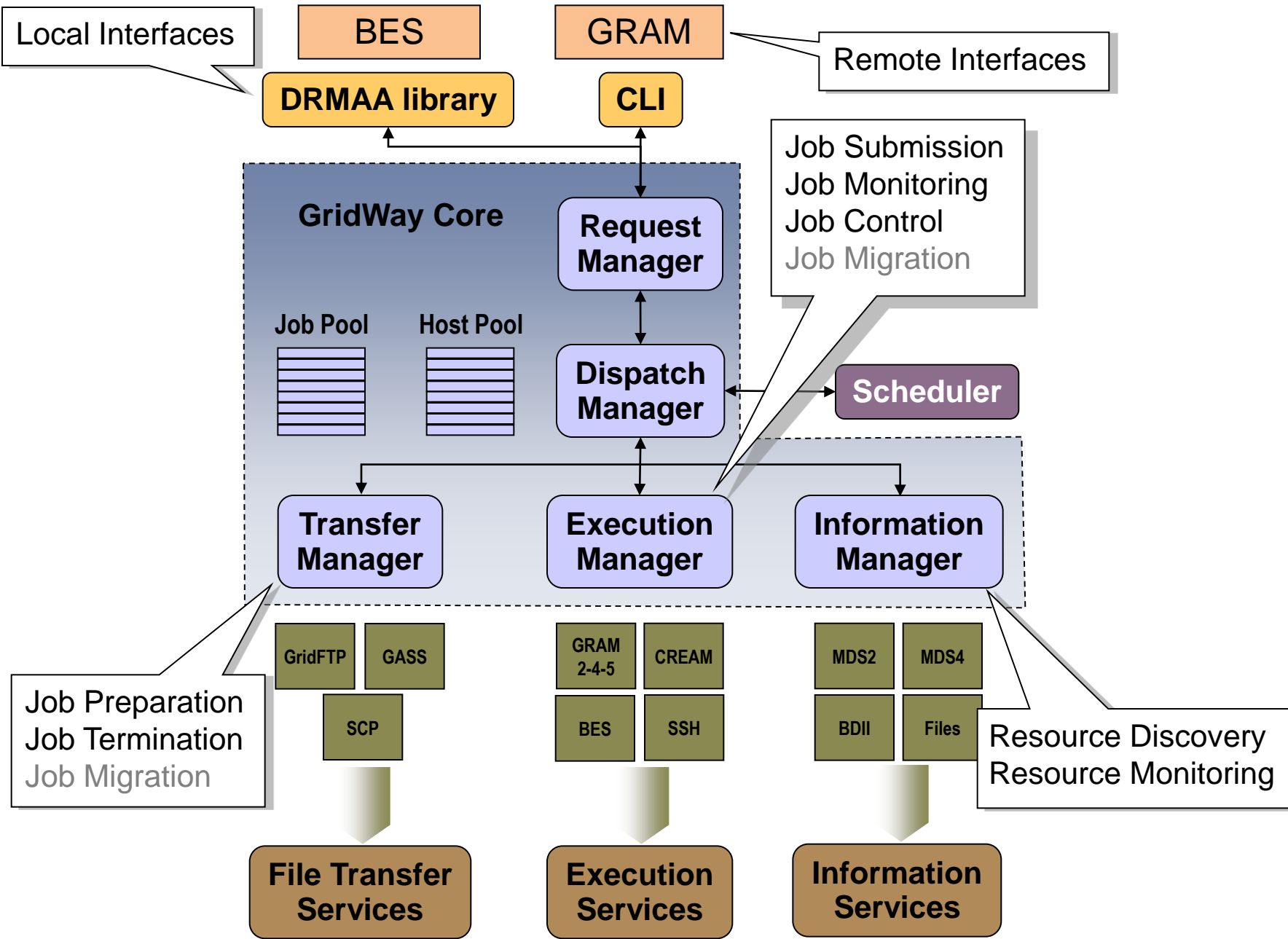


- 1. Introduction**
2. Developments in IGE
3. Simple Usage Example

The GridWay metascheduler enables **large-scale, reliable and efficient sharing of computing resources** over different grid middleware, providing a single point of access for them

GridWay provides a **LRM-like CLI** for submitting, monitoring, synchronizing and controlling jobs

GridWay implements the **OGF standard DRMAA API**, assuring compatibility of applications with LRM systems that implement the standard



1. Introduction
- 2. Developments in IGE**
3. Simple Usage Example

## CREAM, BES and EMI-ES Execution Drivers

---

Job submission, monitoring and control to computing resources managed by:

- **CREAM** (Computing Resource Execution And Management)
- **BES** (Basic Execution Service)
- **EMI-ES** (EMI Execution Service)

GridWay core automatically translates job templates to:

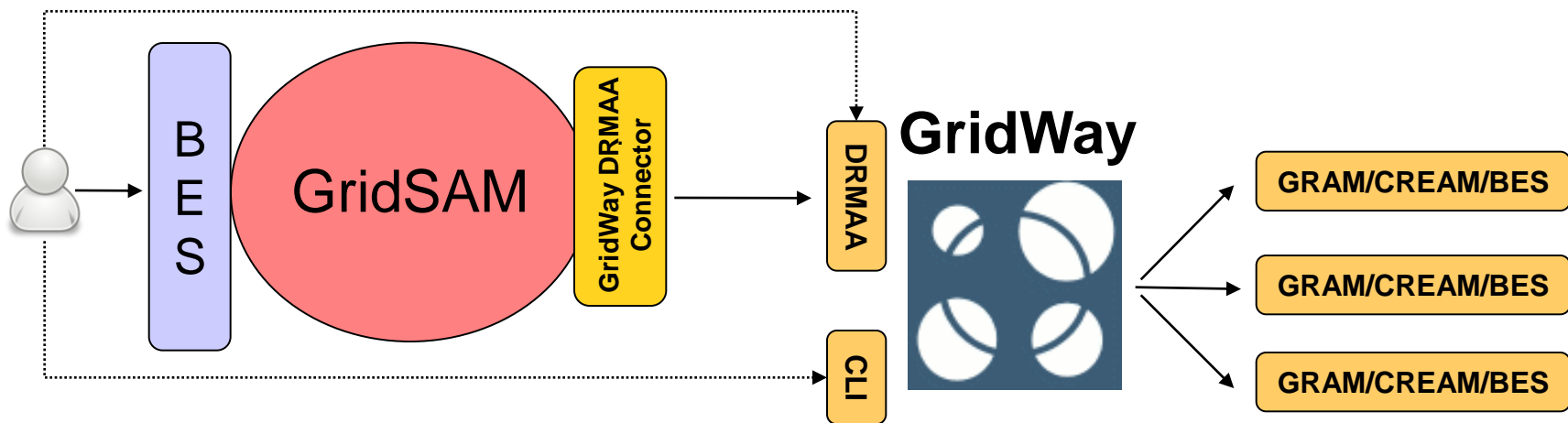
- **JDL** (Job Description Language) for CREAM
- **JSDL** (Job Submission Description Language) for BES
- **ADL** (Activity Description Language) for EMI-ES

## BES Interface for GridWay

Remote access to GridWay's metascheduling capabilities through a standard interface, allowing users to access services provided by different grid middlewares

### BES interface provided by **GridSAM**

- A GridWay connector for GridSAM allows interoperability between both systems.
- Communication is based on Java DRMAA API
- Jobs described in JSDL and managed with standard BES clients



## Other Developments in IGE

---

DRMAA2 C binding (by Uppsala University)

New installation procedure

- Core and drivers are built and installed independently
- Native and FHS-compliant installation

Logging with syslog

Adaptation of the scheduler to submit jobs to resources based only on their rank, not on their free slots

Set resource requirements in job templates (CPU time, memory...)

Randomized job state polling to avoid saturation due to multiple simultaneous requests

Disposing jobs when they have finished the execution

Usage statistics reporting to EGCS

Intensive bug fixing





● Host   
 ● Group of hosts   
 ● Host with errors

Showing hosts in GridWay system.

Filters  
 Host:  User:  Job ID:  Virtual organization:

Hosts		Queues		OS		CPU		Memory (MB)		Disk (MB)		Running jobs		Nodes		LRMS	
ID	Prio.	Name	Name	Version	Arch.	MHZ	% Free	Free	Total	Free	Total	Jobs	Used	Free	Total	Name	Type
0	1	gt5-ige.drg.	Linux	2.6.32.27-0.	x86_64	2533	400	2007	2007	73688	73688	0	0	2	2	jobmanag	sge
1	1	udo-gt01.gr	Linux	2.6.18-238.	x86_64	1995	600	2007	2007	949718	949718	0	0	44	44	jobmanag	pbs
2	1	ve.nikhef.nl	Linux	2.6.18-194.	x86_64	2993	800	2011	2011	3670	3670	0	0	8	8	jobmanag	fork
3	1	gt1.epcc.ed	Linux	2.6.18-194.	i386	1600	100	1024	1024	40960	40960	0	0	1	1	jobmanag	fork
4	1	gt01.ige.psr	Linux	2.6.18-194.	x86_64	2328	400	512	512	97280	97280	0	0	1	1	jobmanag	pbs

Number of hosts: 7.

1. Introduction
2. Developments in IGE
- 3. Simple Usage Example**

Create a proxy certificate.

Create a simple Job Template and save it as **jt**:

```
$ cat jt
EXECUTABLE = /bin/ls
```

Use *gws submit* command to submit the job:

```
$ gws submit -t example/jt
```

Use *gwhost* command to see available resources:

```
$ gwhost
```

HID	PRI	OS	ARCH	MHZ	%CPU	MEM(F/T)	DISK(F/T)	N(U/F/T)	LRMS	HOSTNAME
0	1	Linux2.6.32.27	x86_64	2533	400	2007/2007	73688/73688	0/2/2	jobmanager-sgc	gt5-ige.drg.lrz.de
1	1	Linux2.6.18-23	x86_64	1995	600	2007/2007	1580G/1580G	0/44/44	jobmanager-pbs	udo-gt01.grid.tu-do
2	0	Linux2.6.18-19	x86_64	2993	800	2011/2011	3670/3670	0/8/8	jobmanager-fork	ve.nikhef.nl
3	1	Linux2.6.18-19	x86_64	1600	100	1024/1024	40960/40960	0/1/1	jobmanager-fork	gt1.epcc.ed.ac.uk
4	0	Linux2.6.18-19	x86_64	2328	400	512/512	97280/97280	0/1/1	jobmanager-pbs	gt01.ige.psnc.pl

Follow the evolution of the job with *gwps* command:

```
$ gwps
USER  JID DM  EM  START  END  EXEC  XFER  EXIT NAME HOST
user:0 0  done ---- 10:38:24 10:39:04 0:00:21 0:00:08 0  jt  gt5-
ige.drg.lrz.de/jobman
user:0 1  done ---- 10:46:05 10:46:39 0:00:11 0:00:08 0  jt  udo-gt01.grid.tu-
dortmund
user:0 2  wrap actv 10:48:39 --:--:-- 0:00:39 0:00:03 --  jt  gt5-
ige.drg.lrz.de/jobman
```

**HINT:** Use *gwps -c <seconds>* for continuous output.

Once finished... time to check results:

```
$ ls -l example/
total 8
-rw-r--r-- 1 user user 21 2011-05-04 16:47 jt
-rw-r--r-- 1 user user  0 2011-05-05 10:39 stderr.0
-rw-r--r-- 1 user user 72 2011-05-05 10:39 stdout.0

$ cat example/stdout.0
job.env
stderr.execution
stderr.wrapper
stdout.execution
stdout.wrapper
```

# Useful Links

**The GridWay Project:** <http://www.gridway.org>

**GridWay Development:** <http://dev.gridway.org>

**Documentation:** <http://gridway.org/doku.php?id=documentation>

**Support:** <http://gridway.org/doku.php?id=support>

- User discussion: [gridway-user@globus.org](mailto:gridway-user@globus.org)
- Announcements: [gridway-announce@globus.org](mailto:gridway-announce@globus.org)

**The IGE Project:** <http://www.ige-project.eu>

**GridWay Showcase:** <http://www.ige-project.eu/ige-showcases/gridway>

**Thank you  
for your attention!**