

# Homogenization of access to DCI

**Carlos Blanco, A. S. Cofiño, V. Fernandez-Quiruelas**

Santander Meteorology Group

Department of Applied Mathematics and Computer Sciences

Universidad de Cantabria, Santander, Spain



**Aknowledgments:** This work is funded by the Spanish PLAN NACIONAL de I+D+i 2008-2011 (Ref.# CGL2011-28864, WRF4G) and ERDF

- **Statement of the problem**
- DRM4G
- How does it work ?
- Application example: WRF4G
- Conclusions

# Statement of the problem

## DCI available

- Different types

**CARLOS**



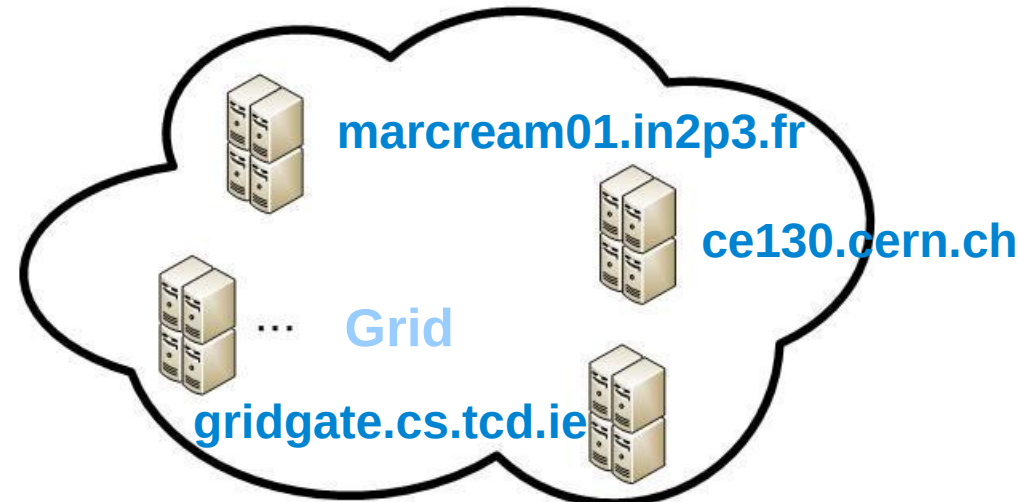
**PC  
Blizzard**



**Workstation Oceano**



**Supercomputer MN3 (BSC)**



# Statement of the problem

## DCI available

- Different types
- Different institutions/organizations

CARLOS



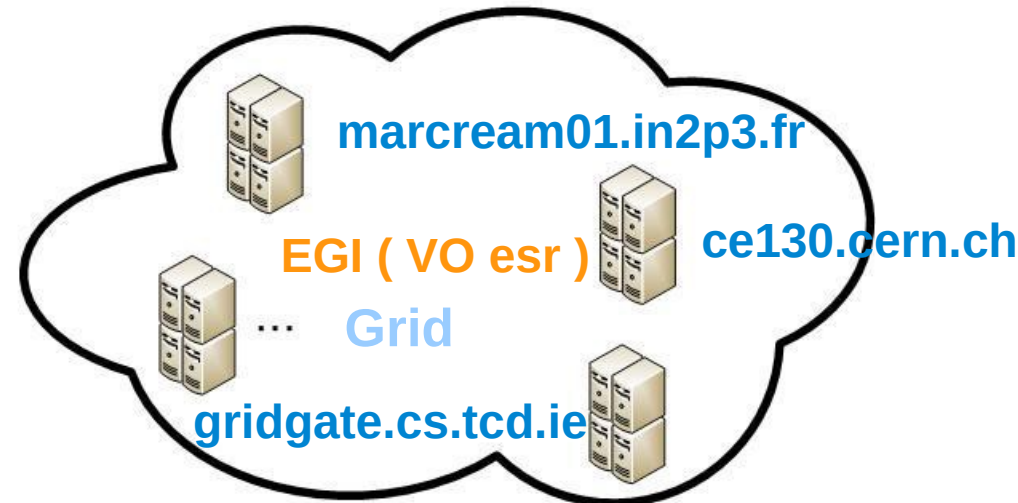
PC  
Blizzard



UC  
Workstation Oceano



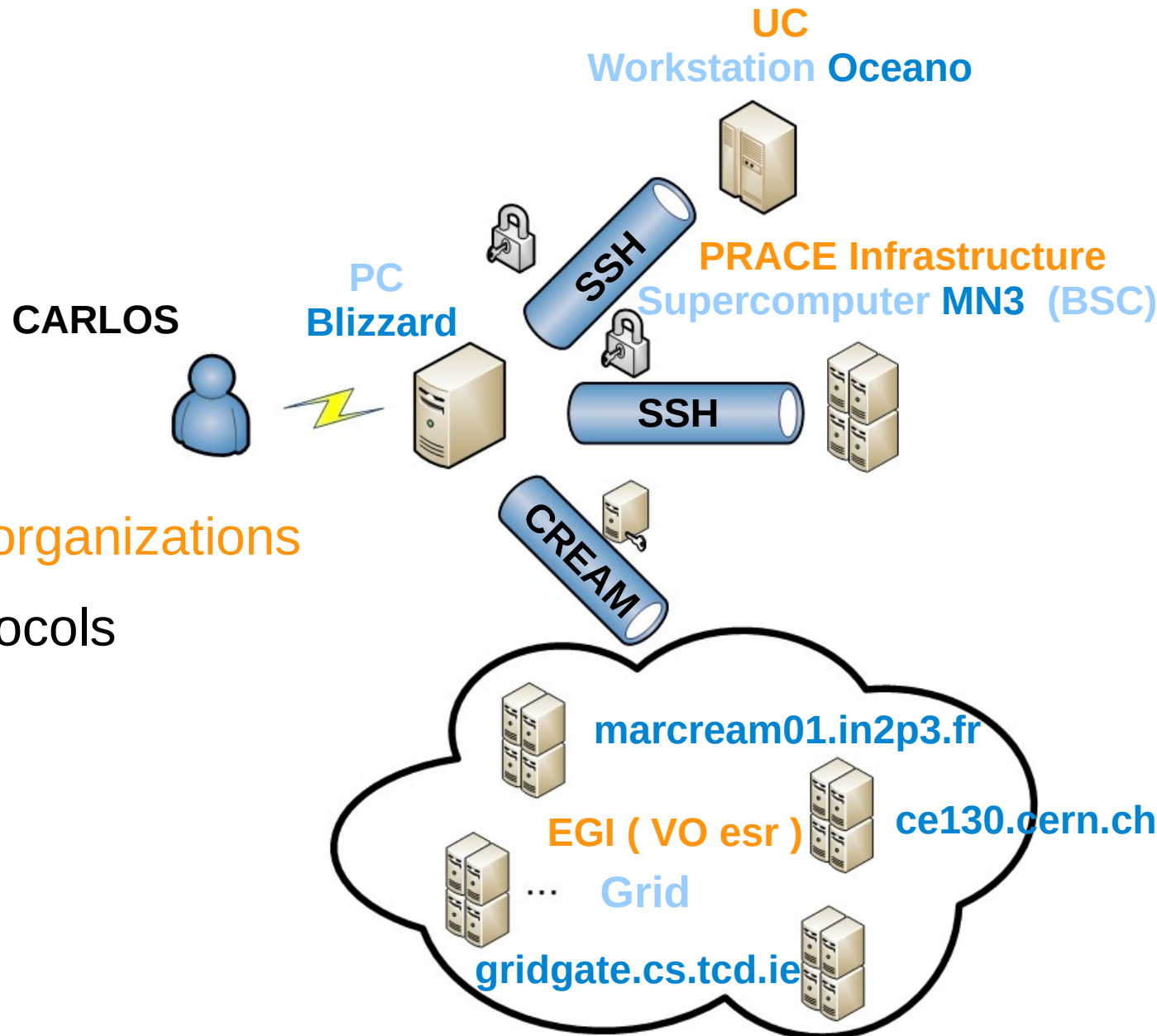
PRACE Infrastructure  
Supercomputer MN3 (BSC)



# Statement of the problem

## DCI available

- Different types
- Different institutions/organizations
- Different access protocols



# Requirements

- Uniform access to the available resources
- Simple interface
- Easy configuration
- Easy installation (batteries included)
- Robust and scalable

- Statement of the problem
- **DRM4G**
- How does it work ?
- Application example: WRF4G
- Conclusions

## Main features

- Based on GridWay meta-scheduler
- CLI interface
- Tools to manage private/public keys and grid certificates
- Download and run
- Requirements: python  $\geq 2.4$  and  $< 3.0$
- Scalable  $\sim 100,000$  jobs



# DCI access

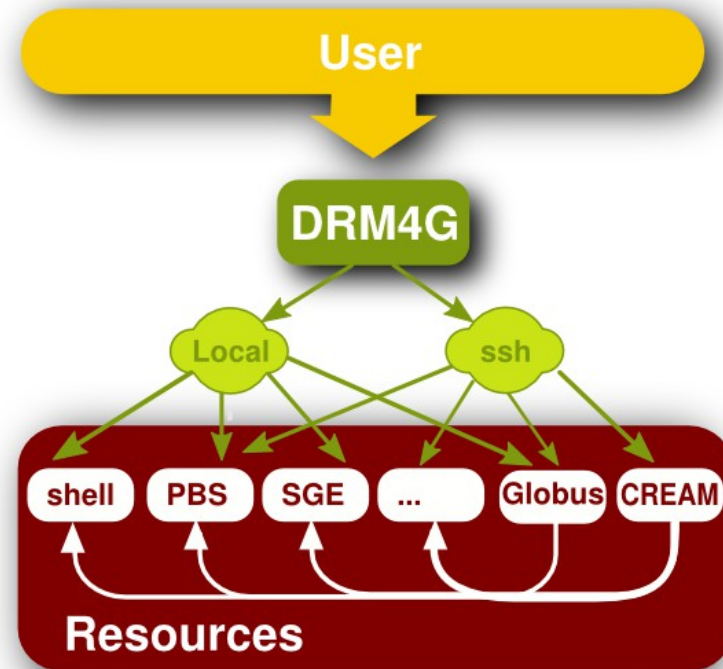
**DRM4G** allows the user to **merge different computing resources** concurrently in a transparent way:

**Local** resources

**Remote** resources (ssh)

**Grid** infrastructures (Globus , CREAM)

- Directly in a shell session
- Interacting with LRMS
  - PBS
  - SGE
  - SLURM
  - LSF
  - ...



- Statement of the problem
- DRM4G
- **How does it work ?**
- Application example : WRF4G
- Conclusions

```
$ wget http://meteo.unican.es/drm4g-1.1.2-x86_64.tar.gz
```

```
$ tar xzf drm4g-1.1.2-x86_64.tar.gz
```

```
$ export PATH=$PWD/drm4g/bin:$PATH
```

```
$ drm4g daemon start
```

```
Creating a DRM4G local configuration in '/home/user/.drm4g'
```

```
Creating '/home/user/.drm4g/etc' directory
```

```
Creating '/home/user/.drm4g/var' directory
```

```
Starting DRM4G daemon ..... OK
```

```
$ drm4g daemon status
```

```
DRM4G daemon is running
```

```
$ drm4g resource edit
```

```
The resources.conf is edited and loaded dynamically
```

# Resource Configuration I

## resources.conf

### [DEFAULT]

```
enable          = True
communicator    = local
frontend        = localhost
lrms            = fork
```

### [blizzard]

```
ncores         = 1
```

### [mn3]

```
communicator    = ssh
username        = user
frontend        = mn3.bsc.es
auth            = ssh-agent
lrms            = lsf
queue           = mic
ncores          = 400
```



# Resource Configuration II

## resources.conf

```
[oceano]
communicator = ssh
username     = user
frontend    = oceano.unican.es
auth        = private_key | ~/.ssh/id_rsa
ncores      = 60

[esr]
communicator = ssh
username     = user
frontend    = ui.macc.unican.es
auth        = private_key | ~/.ssh/id_rsa
lrms        = cream
vo          = esr
```



```
$ drm4g resource esr copy-key --identity-file id_rsa.pub
```

```
Introduce your password to log into ui.macc.unican.es :
```

```
id_rsa.pub added to authorized_keys file
```

```
$ drm4g resource esr create-proxy
```

```
Insert your GRID pass:
```

```
Insert MyProxy password:
```

```
Your identity: /DC=es/DC=irisgrid/O=unican/CN=josecarlos.blanco
```

```
Creating proxy ..... Done
```

```
Proxy Verify OK
```

```
Your proxy is valid until: Wed May 7 10:22:42 2014
```

```
A proxy valid for 168 hours (7.0 days) for user carlos  
now exists on px.grid.sara.nl.
```

```
$ drm4g resource list
```

Name	State
blizzard	enabled
esr	enabled
oceano	enabled
mn3	enabled

```
$ drm4g resource check-frontends
```

```
--> Resource 'esr' ...  
    The front-end ui01.macc.unican.es is accessible  
--> Resource 'mn3' ...  
    The front-end mn3.bsc.es is accessible  
--> Resource 'oceano' ...  
    The front-end oceano.unican.es is accessible
```

# Submit, cancel and monitor jobs

```
$ cat wrf4g.jt
```

```
EXECUTABLE      = /bin/bash ./wrf4g.sh
```

```
INPUT_FILES     = WRF4-1.5.1-x86_64.tar.gz, wrf4g.sh
```

```
REQUIREMENTS  = ARCH = 'x86_64'
```

```
$ drm4g job submit wrf4g.jt
```

```
Job id: 0
```

```
$ drm4g job status list
```

JID	EM	START	END	EXIT	NAME	HOST
0	actv	18:05:33	--:--:--	--	wrf4g.jt	esr_sbgce2.in2p3.fr

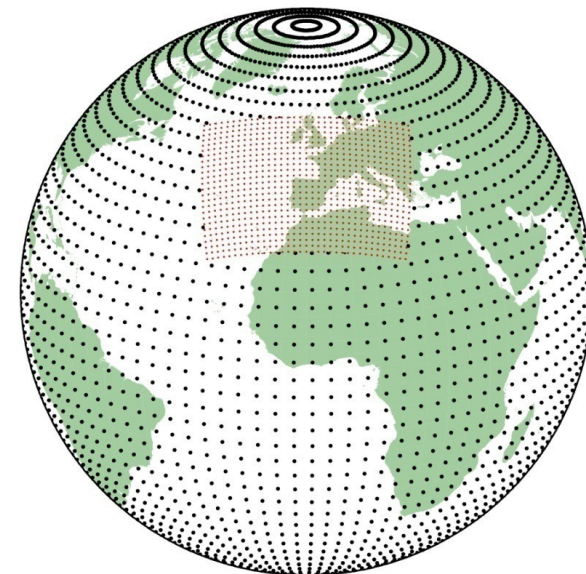
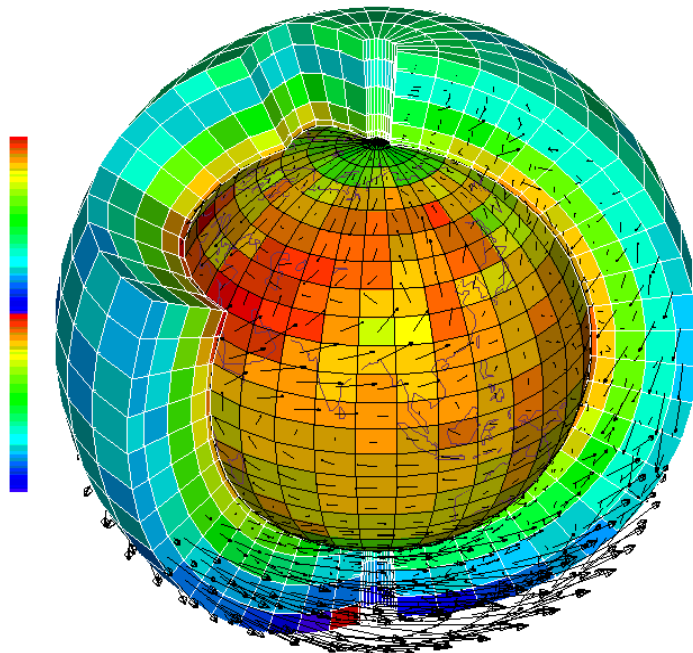
```
$ drm4g job cancel --jid 0
```



- Statement of the problem
- DRM4G
- How does it work ?
- **Application example: WRF4G**
- Conclusions

**WRF4G** is a framework developed by the Santander Meteorology Group, provides:

- Flexible WRF experiment **design**, **execution** and **monitoring**, and ...
- ... the ability to run these experiments on different computing resources at the same time in a **transparent** way (DRM4G).



- Statement of the problem
- DRM4G
- How does it work ?
- Application example: WRF4G
- **Conclusions**

As we have seen DRM4G is ....

- Easy
  - To install
  - To configure
- Useful to other frameworks
  - Providing a single access point
  - Python API
- Scalability (~100,000 jobs)
  - Stable
  - Reliable

# Thank you!

**Contact:** blancojc@unican.es

**More info:** <https://www.meteo.unican.es>



**Aknowledgments:** This work is funded by the Spanish PLAN NACIONAL de I+D+i 2008-2011 (Ref.# CGL2011-28864, WRF4G) and ERDF