

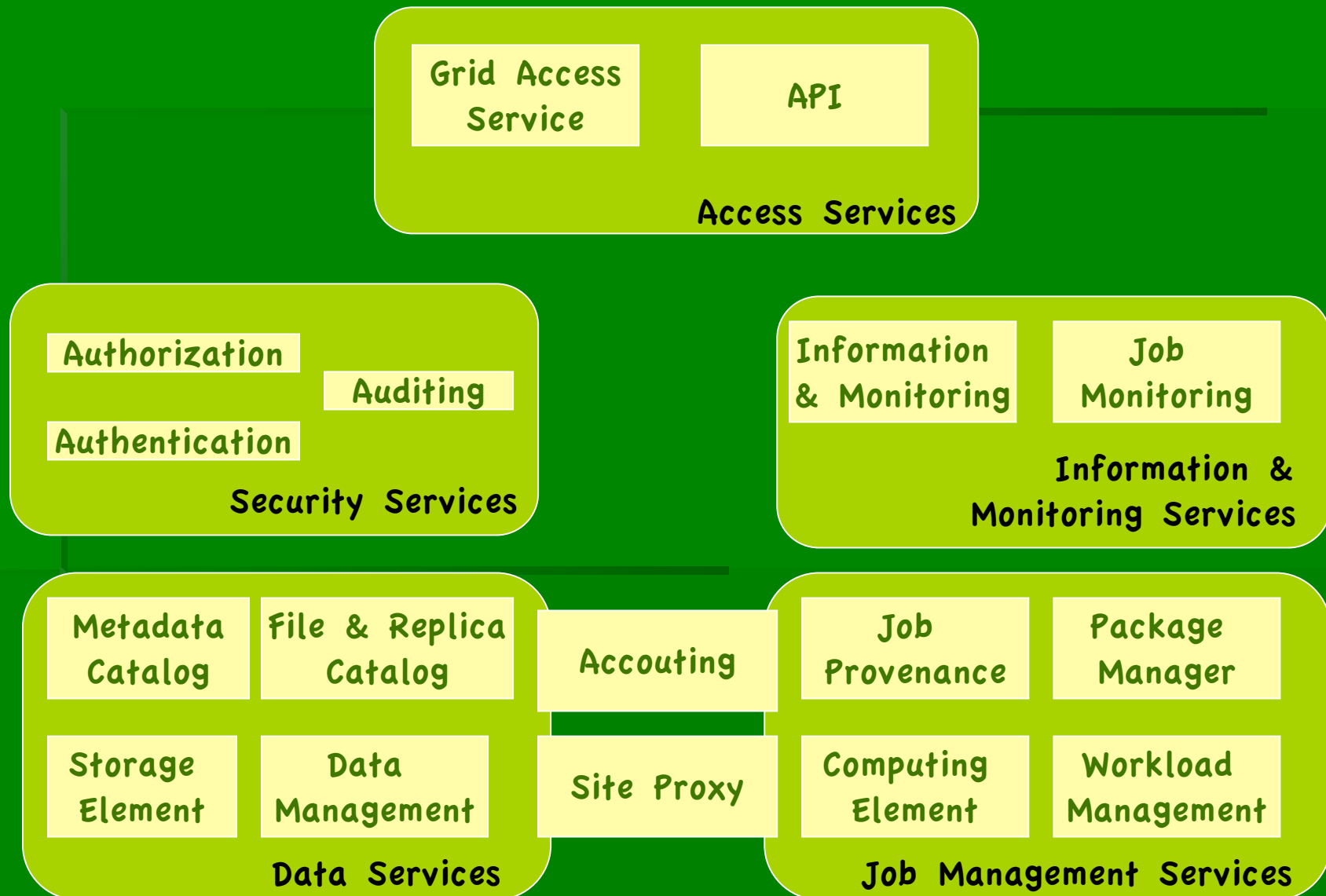
# *Notes about the LCG Grid usage from User point of view*

*Bogdan Lobodzinski  
& MC Production & MC development #1 Groups  
DESY*

## *Agenda:*

- o again general view into the Grid Middleware (gLite Services) structure*
- o Grid initialization*
- o Grid Job: creation, management, etc*
- o R-GMA protocol: job monitoring*
- o Data management*

# *gLite Structure ... for refresh your memory*



## *Few remarks about details subjectively chosen ...*

- Authentication:
  - Who is the User ?
  - Is the User really who represents himself to be ?
- Authorization:
  - Is the User authorized to access given resources ?
  - Is the User authorized to make given operation
- Proxy Certificate requested by the LCG Grid World:
  - Short term, restricted certificate derived from long term X.509 (GSI standard) Proxy Certificate
  - Signed by cert entity
  - Allows a process to act on behalf of a user
  - NOT ENCRYPTED - needs to be in secure place !!!

# Grid Initialization

- *LCG Grid UI is already installed, you need to initialize proper environment. Pick up proper script grid-env.sh from <http://grid.desy.de>*
- *Collect all together into single script (init.sh )....*

```
source /afs/desy.de/project/glite/UI/etc/profile.d/grid-env.sh
export LFC_HOST=`$LCG_LOCATION/bin/lcg-infosites --vo hone lfc`
export MYPROXY_SERVER_DN=/O=GermanGrid/OU=DESY/CN=host/grid-pxy.desy.de
export LCG_CATALOG_TYPE=lfc
export X509_USER_PROXY=/tmp/$.k5-ca-proxy.pem
voms-proxy-init -voms hone -valid 128:00 -vomslife 128:00
```

- *. Init.sh*

## *Grid Job: creation, management, etc*

We need to:

- create a program to be run on the Grid (shell script)
- create a job description using the JDL
- select Grid queues
- submit job or jobs to the Grid
- check status of the job or bulk of jobs
- Gather output of the job or cancel the job

Limitations:

- Input & output Sandbox cannot be usually bigger than 10 Mb - detailed value depends on Grid site settings

# Toy Process

cat Launcher\_perl\_rgma\_toy.sh

```
#!/bin/bash -x
# rather advance toy if you need to use LFC catalog:
WDIR=`pwd`

# download private 32-bit version of Perl:
chmod u+x download_perl_hone.pl
./download_perl_hone.pl
PERL_FILE=`cat DOWNLOAD.perl`
echo $PERL_FILE
if [ ! `echo $PERL_FILE` ]
then
    echo "32-bit perl (hone version) is not downloaded, exit !"
    exit 1;
fi

# untar perl 32-bit version:
tar xzf $PERL_FILE

export PATH=$WDIR/perl.hone/bin:$PATH

# setup perl lnraries:
export PERL_INSTALL=$WDIR

env perl -e 'print "arch=".3.2*length(~0).";\n"'

if [ ! $LFC_HOST ]
then
    export LFC_HOST="grid-lfc.desy.de"
fi
export LFC_HOME="/grid/hone"

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$LCG_LOCATION/lib/perl
export PERL5LIB=$PERL5LIB:$WDIR:$WDIR/perl.hone:$LCG_LOCATION/lib/perl/:\
$GLITE_LOCATION/lib/perl5

ulimit -s unlimited;ulimit -l unlimited

export PATH=$PATH:$WDIR
```

Part 1

```
# unpack Jobwrapper_toy ...
tar xzf Jobwrapper_toy.tar.gz

chmod u+x getMd5.pl
chmod u+x Jobwrapper_toy.pl

JOB_MONITOR_FILE=`cat DOWNLOAD.job_monitor`
echo $JOB_MONITOR_FILE
if [ ! `echo $JOB_MONITOR_FILE` ]
then
    echo "RGMA JOB MONITOR client (hone version) is not downloaded, "
    echo "job will be running without RGMA monitor !"

    echo "start ./Jobwrapper_toy.pl ..."
    ./Jobwrapper_toy.pl
else
    # untar job monitor:
    tar xzf $JOB_MONITOR_FILE
    chmod 700 $JOB_MONITOR_FILE

    echo "start ./job-monitor.py Jobwrapper_toy.pl ..."
    ./job-monitor.py Jobwrapper_toy.pl
Fi
```

Part II

# Toy Process

cat Launcher\_perl\_rgma\_toy.sh

```
#!/bin/bash -x
# rather advance toy if you need to use LFC catalog:
WDIR=`pwd`

# download private 32-bit version of Perl:
chmod u+x download_perl_hone.pl
./download_perl_hone.pl
PERL_FILE=`cat DOWNLOAD.perl`
echo $PERL_FILE
if [ ! `echo $PERL_FILE` ]
then
    echo "32-bit perl (hone version) is not downloaded, exit !"
    exit 1;
fi

# untar perl 32-bit version:
tar xzf $PERL_FILE

export PATH=$WDIR/perl.hone/bin:$PATH

# setup perl linraries:
export PERL_INSTALL=$WDIR

env perl -e 'print "arch=".3.2*length(~0).";\n"'

if [ ! $LFC_HOST ]
then
    export LFC_HOST="grid-lfc.desy.de"
fi
export LFC_HOME="/grid/hone"

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$LCG_LOCATION/lib/perl
export PERL5LIB=$PERL5LIB:$WDIR:$WDIR/perl.hone:$LCG_LOCATION/lib/perl/:\
$GLITE_LOCATION/lib/perl5

ulimit -s unlimited;ulimit -l unlimited

export PATH=$PATH:$WDIR
```

Part 1

```
# unpack Jobwrapper_toy ...
tar xzf Jobwrapper_toy.tar.gz

chmod u+x getMd5.pl
chmod u+x Jobwrapper_toy.pl

JOB_MONITOR_FILE=`cat DOWNLOAD.job_monitor`
if [ ! `echo $JOB_MONITOR_FILE` ]
then
    echo "RGMA JOB MONITOR client (hone version) is not downloaded, "
    echo "job will be running without RGMA monitor !"
fi

echo "start ./Jobwrapper_toy.pl ..."
./Jobwrapper_toy.pl
else
# untar job monitor:
tar xzf $JOB_MONITOR_FILE
chmod 700 $JOB_MONITOR_FILE

echo "start ./job-monitor.py Jobwrapper_toy.pl ..."
./job-monitor.py Jobwrapper_toy.pl
Fi
```

Part II



# Toy Process

cat Launcher\_perl\_rgma\_toy.sh

```
#!/bin/bash -x
# rather advance toy if you need to use LFC catalog:
WDIR=`pwd`

# download private 32-bit version of Perl:
chmod u+x download_perl_hone.pl
./download_perl_hone.pl
PERL_FILE=`cat DOWNLOAD.perl`
echo $PERL_FILE
if [ ! `echo $PERL_FILE` ]
then
    echo "32-bit perl (hone version) is not downloaded, exit!"
    exit 1;
fi

# untar perl 32-bit version:
tar xzf $PERL_FILE

export PATH=$WDIR/perl.hone/bin:$PATH

# setup perl linraries:
export PERL_INSTALL=$WDIR

env perl -e 'print "arch=".3.2*length(~0).";\n"'

if [ ! $LFC_HOST ]
then
    export LFC_HOST="grid-lfc.desy.de"
fi
export LFC_HOME="/grid/hone"

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$LCG_LOCATION/lib/perl
export PERL5LIB=$PERL5LIB:$WDIR:$WDIR/perl.hone:$LCG_LOCATION/lib/perl/:\
$GLITE_LOCATION/lib/perl5

ulimit -s unlimited;ulimit -l unlimited

export PATH=$PATH:$WDIR
```

Not every LCG Grid Site supports  
proper Perl/Python installation -  
don't trust experts !

```
# unpack Jobwrapper_toy ...
tar xzf Jobwrapper_toy.tar.gz

chmod u+x getMd5.pl
chmod u+x Jobwrapper_toy.pl

JOB_MONITOR_FILE=`cat DOWNLOAD.job_monitor`
echo $JOB_MONITOR_FILE
if [ ! `echo $JOB_MONITOR_FILE` ]
then
    echo "RGMA JOB MONITOR client (hone version) is not downloaded, "
    echo "job will be running without RGMA monitor !"
    echo "start ./Jobwrapper_toy.pl ..."
    ./Jobwrapper_toy.pl
else
    # untar job monitor:
    tar xzf $JOB_MONITOR_FILE
    chmod 700 $JOB_MONITOR_FILE

    echo "start ./job-monitor.py Jobwrapper_toy.pl ..."
    ./job-monitor.py Jobwrapper_toy.pl
Fi
```

Part II

Part 1

# Toy Process

cat Launcher\_perl\_rgma\_toy.sh

```
#!/bin/bash -x
# rather advance toy if you need to use LFC catalog:
WDIR=`pwd`

# download private 32-bit version of Perl:
chmod u+x download_perl_hone.pl
./download_perl_hone.pl
PERL_FILE=`cat DOWNLOAD.perl`
echo $PERL_FILE
if [ ! `echo $PERL_FILE` ]
then
    echo "32-bit perl (hone version) is not downloaded, exit !"
    exit 1;
fi

# untar perl 32-bit version:
tar xzf $PERL_FILE

export PATH=$WDIR/perl.hone/bin:$PATH

# setup perl libraries:
export PERL_INSTALL=$WDIR

env perl -e 'print "arch=".3.2*length(~0).";\n"'

if [ ! $LFC_HOST ]
then
    export LFC_HOST="grid-lfc.desy.de"
fi
export LFC_HOME="/grid/hone"

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$LCG_LOCATION/lib/perl
export PERL5LIB=$PERL5LIB:$WDIR:$WDIR/perl.hone:$LCG_LOCATION/lib/perl/:\
$GLITE_LOCATION/lib/perl5

ulimit -s unlimited;ulimit -l unlimited

export PATH=$PATH:$WDIR
```

Part 1

```
# unpack Jobwrapper_toy ...
tar xzf Jobwrapper_toy.tar.gz

chmod u+x getMd5.pl
chmod u+x Jobwrapper_toy.pl

JOB_MONITOR_FILE=`cat DOWNLOAD.job_monitor`
echo $JOB_MONITOR_FILE
if [ ! `echo $JOB_MONITOR_FILE` ]
then
    echo "RGMA JOB MONITOR client (hone version) is not downloaded, "
    echo "job will be running without RGMA monitor !"

    echo "start ./Jobwrapper_toy.pl ..."
    ./Jobwrapper_toy.pl
else
    # untar job monitor:
    tar xzf $JOB_MONITOR_FILE
    chmod 700 $JOB_MONITOR_FILE

    echo "start ./job-monitor.py Jobwrapper_toy.pl ..."
    ./job-monitor.py Jobwrapper_toy.pl
fi
```

Part II

LFC Perl/Python libraries  
have memory leak  
Always Starts LFC features  
As separate processes

# Toy Process

cat Launcher\_perl\_rgma\_toy.sh

```
#!/bin/bash -x
# rather advance toy if you need to use LFC catalog:
WDIR=`pwd`

# download private 32-bit version of Perl:
chmod u+x download_perl_hone.pl
./download_perl_hone.pl
PERL_FILE=`cat DOWNLOAD.perl`
echo $PERL_FILE
if [ ! `echo $PERL_FILE` ]
then
    echo "32-bit perl (hone version) is not downloaded, exit !"
    exit 1;
fi

# untar perl 32-bit version:
tar xzf $PERL_FILE

export PATH=$WDIR/perl.hone/bin:$PATH

# setup perl libraries:
export PERL_INSTALL=$WDIR

env perl -e 'print "arch=".3.2*length(~0).";\n"'

if [ ! $LFC_HOST ]
then
    export LFC_HOST="grid-lfc.desy.de"
fi
export LFC_HOME="/grid/hone"

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$LCG_LOCATION/lib/perl
export PERL5LIB=$PERL5LIB:$WDIR:$WDIR/perl.hone:$LCG_LOCATION/lib/perl/:\
$GLITE_LOCATION/lib/perl5

ulimit -s unlimited;ulimit -l unlimited

export PATH=$PATH:$WDIR
```

Part 1

```
# unpack Jobwrapper_toy ...
tar xzf Jobwrapper_toy.tar.gz

chmod u+x getMd5.pl
chmod u+x Jobwrapper_toy.pl

JOB_MONITOR_FILE=`cat DOWNLOAD.job_monitor`
echo $JOB_MONITOR_FILE
if [ ! `echo $JOB_MONITOR_FILE` ]
then
    echo "RGMA JOB MONITOR client (hone version) is not downloaded, "
    echo "job will be running without RGMA monitor "
    echo "start ./Jobwrapper_toy.pl ..."
    ./Jobwrapper_toy.pl
else
    # untar job monitor:
    tar xzf $JOB_MONITOR_FILE
    chmod 700 $JOB_MONITOR_FILE

    echo "start ./job-monitor.py Jobwrapper_toy.pl ..."
    ./job-monitor.py Jobwrapper_toy.pl
fi
```

Part II

Dont rely on  
the LCG Information Service  
Try to monitor job using  
more reliable tools

# JDL file

cat Grid\_job\_toy.jdl

```
VirtualOrganisation = "hone";
Executable = "Launcher_perl_rgma_toy.sh";
# Arguments = "$job_conf";
StdOutput = "std.out";
StdError = "std.err";

InputSandbox = {
"/x02/usr/bogdan/MC.presentation_working_set/Launcher_perl_rgma_toy.sh",
"/x02/usr/bogdan/MC.presentation_working_set/download_perl_hone.pl",
"/x02/usr/bogdan/MC.presentation_working_set/Jobwrapper_toy.tar.gz",
"/x02/usr/bogdan/MC.presentation_working_set/steering_mcrequest_5529",
"/x02/usr/bogdan/MC.presentation_working_set/wrapper.conf",
"/x02/usr/bogdan/MC.presentation_working_set/checksums.txt"
};

OutputSandbox = {
"std.out",
"std.err"
};

MyProxyServer = "grid-pxy.desy.de";

Environment = "LCG_GFAL_INFOSYS=grid-bdii.desy.de:2170";

Requirements = ( other.GlueCEPolicyMaxCPUTime >= 1400 ) &&
( other.GlueHostOperatingSystemName == "ScientificSL" ||
  other.GlueHostOperatingSystemName == "ScientificCERNSLC"
) &&
( other.GlueHostOperatingSystemRelease >= 4.0 &&
  other.GlueHostOperatingSystemRelease <= 5.0
);

Rank      = -other.GlueCEStateWaitingJobs
```

## *JDL file (2)*

The JDL attributes can be collected into 2 groups:

- Job Attributes (defines job as it is: input, executable, arguments)
- Resources: used by WMS for finding the proper place for job execution
  - Computing Resources (prefixed with **Other.** )
  - Data & Storage resources (define SE for input & output data, disk space)

## *JDL file (3)*

### Job Attributes:

- JobType - normal (sequential job), DAQ, Interactive (doesn't work :( ), MPICH, Checkpointable
- VirtualOrganisation - *mandatory*
- Executable - **mandatory**
- Arguments - **optional**
- StdInput, StdOutput, StdError - **optional**
- Environment - **optional**
- InputSandbox - **list of files staged automatically to the WN**
- OutputSandbox - **optional, list of files downloaded via Glite download command**

## *JDL file (4)*

### Resource Attributes:

- Requirements - optional, job's requirements for proper job execution
- Rank - *optional, selects the LCG queues according rule defined by Rank method*  
*default:* other.GlueCEStateEstimatedResponseTime  
*Or*  
other.GlueCEStateFreeCPUs  
*depends what is inside The UI configuration file*
- InputData- optional, list of LFN or GUID
- DataAccessProtocol - mandatory if InputData defined
- OutputSE - optional

<http://glite.web.cern.ch/glite/documentation>

# Select Grid Queues

With ready jdl file we can start to use the Grid:

```
$ glite-wms-job-list-match -a Grid_job_toy.jdl
```

```
Connecting to the service https://grid-wms5.desy.de:7443/glite_wms_wmproxy_server
```

```
=====
```

## COMPUTING ELEMENT IDs LIST

The following CE(s) matching your job requirements have been found:

\*CEId\*

- cclcgceli03.in2p3.fr:2119/jobmanager-bqs-long
- cclcgceli04.in2p3.fr:2119/jobmanager-bqs-long
- ce-iep-grid.saske.sk:2119/jobmanager-lcgpbs-hone
- ce.cyf-kr.edu.pl:2119/jobmanager-pbs-hone
- ce01.lcg.cscs.ch:2119/jobmanager-lcgpbs-others
- ce01.tier2.hep.manchester.ac.uk:2119/jobmanager-lcgpbs-hone
- ceglite.itep.ru:2119/jobmanager-lcgpbs-hone
- ceitep.itep.ru:2119/jobmanager-lcgpbs-hone

...



# submit job or jobs to the Grid

```
$ glite-wms-job-submit -a -o out Grid_job_toy.jdl
```

```
Connecting to the service https://grid-wms5.desy.de:7443/glite_wms_wmproxy_server
```

```
===== glite-wms-job-submit Success =====
```

```
The job has been successfully submitted to the WMPProxy  
Your job identifier is:
```

```
https://grid-lb1.desy.de:9000/0SegdV_WrHLOv2J0bBFDLQ
```

```
The job identifier has been saved in the following file:  
/x02/usr/bogdan/MC.presentation_working_set/out
```

```
=====
```

## Single Submission

```
$ ls -ltr bulk
```

```
total 8
```

```
-rw-r--r-- 1 bogdan h1 1121 Jul 8 21:22 Grid_job_toy_1.jdl
```

```
-rw-r--r-- 1 bogdan h1 1121 Jul 8 21:22 Grid_job_toy_2.jdl
```

```
$ glite-wms-job-submit -a -o out.bulk --collection bulk
```

```
Connecting to the service https://grid-wms5.desy.de:7443/glite_wms_wmproxy_server
```

```
===== glite-wms-job-submit Success =====
```

```
The job has been successfully submitted to the WMPProxy  
Your job identifier is:
```

```
https://grid-lb1.desy.de:9000/aWt-j4pQ9PhGYdZFS15www
```

```
The job identifier has been saved in the following file:  
/x02/usr/bogdan/MC.presentation_working_set/out.bulk
```

```
=====
```

## Bulk Submission

# check status of the job or bulk of jobs

```
$ glite-wms-job-status -i out
```

```
*****
```

```
BOOKKEEPING INFORMATION:
```

```
Status info for the Job : https://grid-lb1.desy.de:9000/0SegdV_WrHLOv2J0bBFDLQ
```

```
Current Status:   Running
```

```
Status Reason:    Job successfully submitted to Globus
```

```
Destination:      grid-ce3.desy.de:2119/jobmanager-lcgpbs-default
```

```
Submitted:        Tue Jul  8 21:20:11 2008 CEST
```

```
*****
```

**Single Submission**

```
$ glite-wms-job-status -i out.bulk
```

```
*****
```

```
BOOKKEEPING INFORMATION:
```

```
Status info for the Job : https://grid-lb1.desy.de:9000/aWt-j4pQ9PhGYdZFS15www
```

```
Current Status:   Waiting
```

```
Submitted:        Tue Jul  8 21:23:16 2008 CEST
```

```
*****
```

```
- Nodes information for:
```

```
Status info for the Job : https://grid-lb1.desy.de:9000/3-cC5gWGBnwQgcfzzok5Rw
```

```
Current Status:   Scheduled
```

```
Status Reason:    Job successfully submitted to Globus
```

```
Destination:      ce2.ppgrid1.rhul.ac.uk:2119/jobmanager-pbs-hone
```

```
Submitted:        Tue Jul  8 21:23:16 2008 CEST
```

```
*****
```

```
Status info for the Job : https://grid-lb1.desy.de:9000/l50WeiS_1ZOxW9S-1swpuA
```

```
Current Status:   Scheduled
```

```
Status Reason:    Job successfully submitted to Globus
```

```
Destination:      dgc-grid-40.brunel.ac.uk:2119/jobmanager-lcgpbs-hone
```

```
Submitted:        Tue Jul  8 21:23:16 2008 CEST
```

```
*****
```

**Bulk Submission**

# Gather output of the job

```
$ glite-wms-job-output https://grid-lb1.desy.de:9000/l50WeiS_1ZOxW9S-1swpuA
```

```
Connecting to the service https://grid-wms5.desy.de:7443/glite_wms_wmproxy_server
```

```
=====
```

```
JOB GET OUTPUT OUTCOME
```

```
Output sandbox files for the job:
```

```
https://grid-lb1.desy.de:9000/l50WeiS_1ZOxW9S-1swpuA
```

```
have been successfully retrieved and stored in the directory:
```

```
/tmp/bogdan_l50WeiS_1ZOxW9S-1swpuA
```

```
=====
```

# or cancel the job

```
$ glite-wms-job-cancel https://grid-lb1.desy.de:9000/3-cC5gWGBnwQgcfzzok5Rw
```

```
Are you sure you want to remove specified job(s) [y/n]y : y
```

```
Connecting to the service https://grid-wms5.desy.de:7443/glite_wms_wmproxy_server
```

```
===== glite-wms-job-cancel Success =====
```

```
The cancellation request has been successfully submitted for the following job(s):
```

```
- https://grid-lb1.desy.de:9000/3-cC5gWGBnwQgcfzzok5Rw
```

```
=====
```

# *R-GMA protocol: job monitoring*

## Relational Grid Monitoring

### Architecture

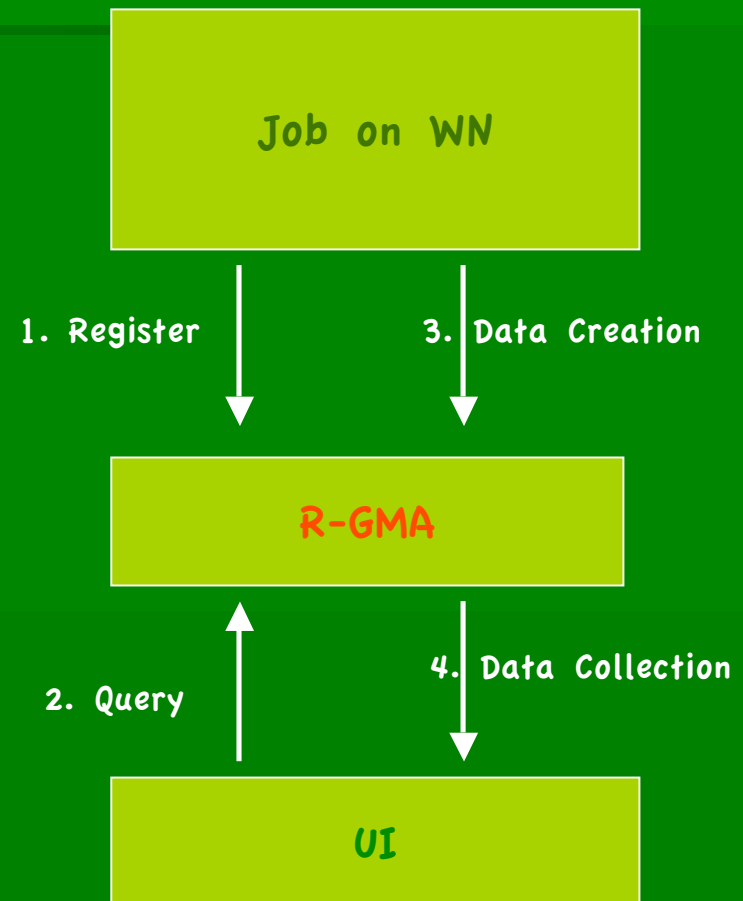
<http://www.r-gma.org>

- EGEE Standard (supported by GGF)
- manage data using SQL queries
- allows to trace job state during job execution
- allows to transfer arbitrary (but not too big) information from WN to UI

Therefore R-GMA allows to localize forgotten somewhere on the LCG Grid running jobs - visible as a **forever-running** creatures ...

Unfortunately...

R-GMA is not too stable tool



# R-GMA protocol: job monitoring (2)

```
$ glite-wms-job-submit -r grid-ce3.desy.de:2119/jobmanager-lcgpbs-default -a Grid_job_toy.jdl
```

```
Connecting to the service https://grid-wms1.desy.de:7443/glite_wms_wmproxy_server
```

```
===== glite-wms-job-submit Success =====
```

```
The job has been successfully submitted to the WMPProxy
```

```
Your job identifier is:
```

**Job Submission**

```
https://grid-lb0.desy.de:9000/3DDPXYQDsVdXX9BfKHS9pw
```

```
=====
```

```
$ RGMA.monitoring.sh https://grid-lb0.desy.de:9000/3DDPXYQDsVdXX9BfKHS9pw
```

```
Tue Jul 8 22:13:04 CEST 2008
```

```
Set query type to latest
```

```
No results returned
```

```
WARNING: No producers found to answer query; no relevant producers of any type exist.
```

```
Tue Jul 8 22:15:00 CEST 2008
```

```
Set query type to latest
```

## R-GMA Monitoring

```
+-----+-----+-----+-----+
| JobId           | BeatType | N | MeasurementDate | MeasurementTime |
+-----+-----+-----+-----+
| https://grid-lb0.desy.de:9000/3DDPXYQDsVdXX9BfKHS9pw | A | -1 | 2008-07-08 | 20:14:55 |
+-----+-----+-----+-----+
1 rows
```

```
Tue Jul 8 22:15:20 CEST 2008
```

```
Set query type to latest
```

```
+-----+-----+-----+-----+
| JobId           | BeatType | N | MeasurementDate | MeasurementTime |
+-----+-----+-----+-----+
| https://grid-lb0.desy.de:9000/3DDPXYQDsVdXX9BfKHS9pw | A | -1 | 2008-07-08 | 20:14:55 |
+-----+-----+-----+-----+
1 rows
```

```
...
```

```
+-----+-----+-----+-----+
| JobId           | BeatType | N | MeasurementDate | MeasurementTime |
+-----+-----+-----+-----+
| https://grid-lb0.desy.de:9000/3DDPXYQDsVdXX9BfKHS9pw | X | 0 | 2008-07-08 | 20:16:36 |
+-----+-----+-----+-----+
1 rows
```

## *R-GMA protocol: job monitoring (3)*

```
$ cat RGMA.monitoring.sh
#!/bin/sh

while [ true ]; do
  date
  #rgma -c "set query latest" -c "SELECT * from JobHeartBeat1 where JobId = 'local_job'"
  rgma -c "set query latest" -c "SELECT * from JobHeartBeat1"
  echo
  sleep 15
done
```

- for R-GMA access you need a valid Grid proxy
- run rgma command - interactive and batch mode is available

# Data Management

- much more complicated than file management on local host
- gLite supports 2 different catalog types:
  - LFC (LCG Catalog ) - we will continue about this catalog only
  - RLS (Replica Location Server)

## Basic Definitions:

- Logical File Name (LFN):

alias created by the user for specific item of data,

example: `lfn:/grid/hone/bogdan/my_test_file_on_the_Grid`

- Globally Unique Identifier (GUID):

a non-human-readable unique name for specific file,

example: `guid:c38f7b89-dbda-4392-aaa7c5e19f96`

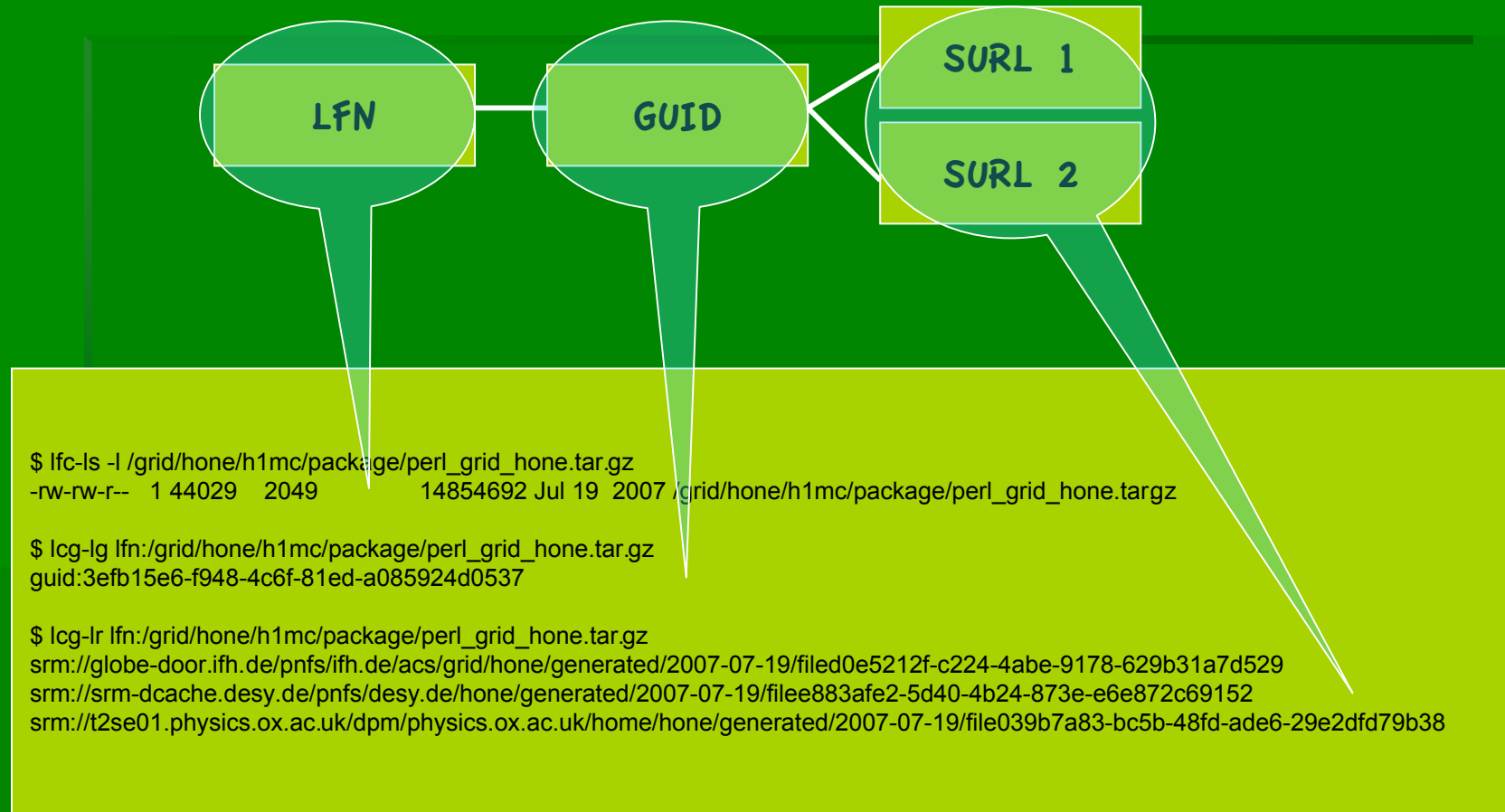
- Site URL (SURL) Or Physical File Name (PFN) or Site FN:

location of specific file on a given SE, example:

`srm://globe-door.ifh.de/pnfs/ifh.de/acs/grid/hone/generated/2008-02-22/fileb450f25-aaa-47b9-9883-6036b624f3b3`

# Data Management (2)

Real example:





# Data Management (3)

LFC - virtual directory tree, each entry can be stored on different SE

How to use LFC:

- setup LFC environment:

```
export LFC_HOST=grid-lfc.desy.de
```

```
export LCG_CATALOG_TYPE=lfc
```

- start to use LFC commands (interaction with the catalog only !):

`lfc-*` - for details, see <http://grid.desy.de/users/> - nice examples

LCG - utils: couples catalog operations with file management

- available set of commands `lcg-*` -

for details, see <http://grid.desy.de/users/> - nice examples

## *Data Management (4): some chosen examples*

*(not available in <http://grid.desy.de>)*

- you need to remove all replicas in one go:

```
lcg-del -a -t 240 --vo hone lfn:/grid/hone/bogdan/file_name
```

- you need to remove specific replica/file from selected SE:

- make a list of all replicas and select proper file

```
lcg-lr --vo hone -v -t 240 lfn:/grid/hone/bogdan/file_name
```

- remove selected replica from the Grid (other remains):

```
lcg-del --vo hone -v -t 240
```

```
srm://t2se01.physics.ox.ac.uk/dpm/physics.ox.ac.uk/home/hone/generated/20070719/fil  
e018d7a23-ba3b-48fd-ede4-24f2dfe79u58
```

- ~~decouple catalog entry and the real file on given SE (after this action  
file is not available via lfc commands but still exists on the SE):~~

```
guid=`lcg-lg --vo hone -v -t 240 lfn:/grid/hone/bogdan/file_name`
```

```
srm=`lcg-lr --vo hone -v -t 240 lfn:/grid/hone/bogdan/file_name`
```

```
lcg-uf --vo hone -v -t 240 $guid $srm
```

-

## *Summary ...*

Please refer to this presentation as notes.  
Thank you for your attention ...