



Enabling Grids for E-scienceE

Ganga Tutorial

Adrian Muraru (CERN)

www.eu-egee.org



- **Part I: Ganga introduction**
- **Part II: Ganga hands-on**
- **Part III: More about Ganga**



Enabling Grids for E-science

Part I: Ganga Introduction

Ganga overview

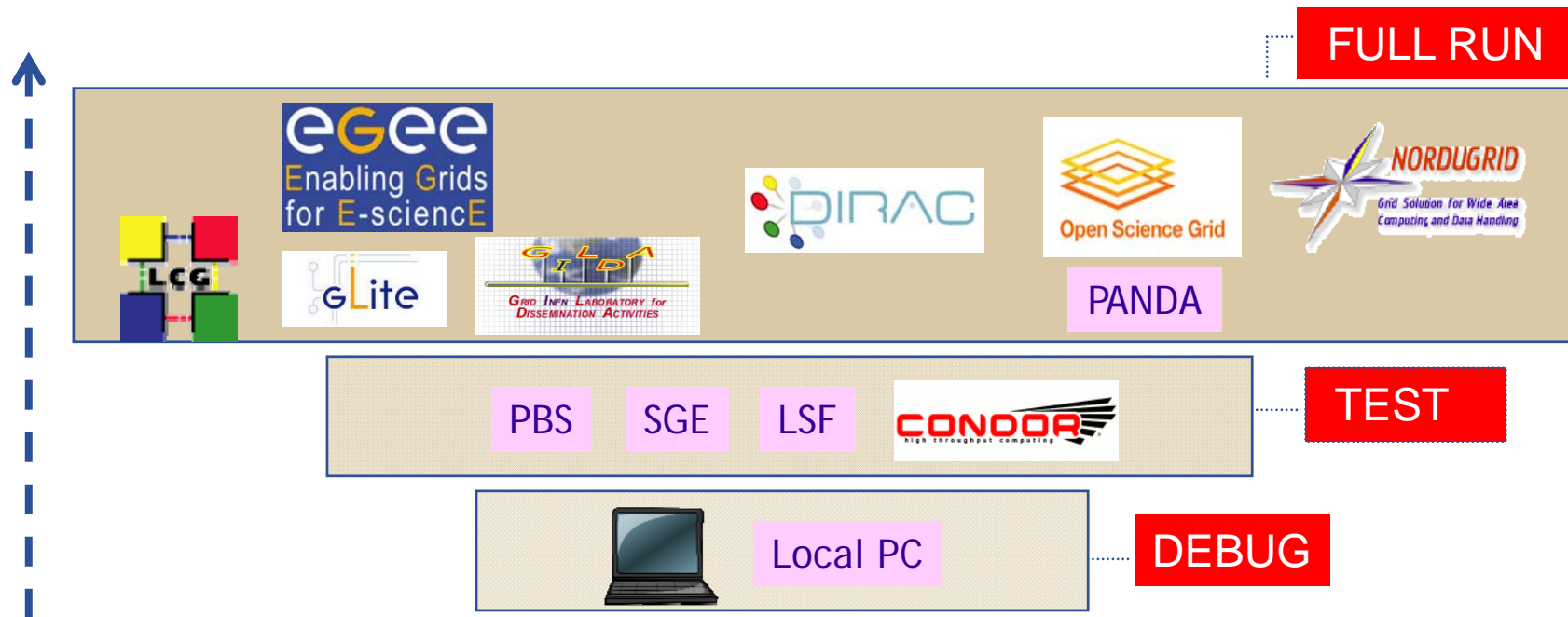
www.eu-egee.org



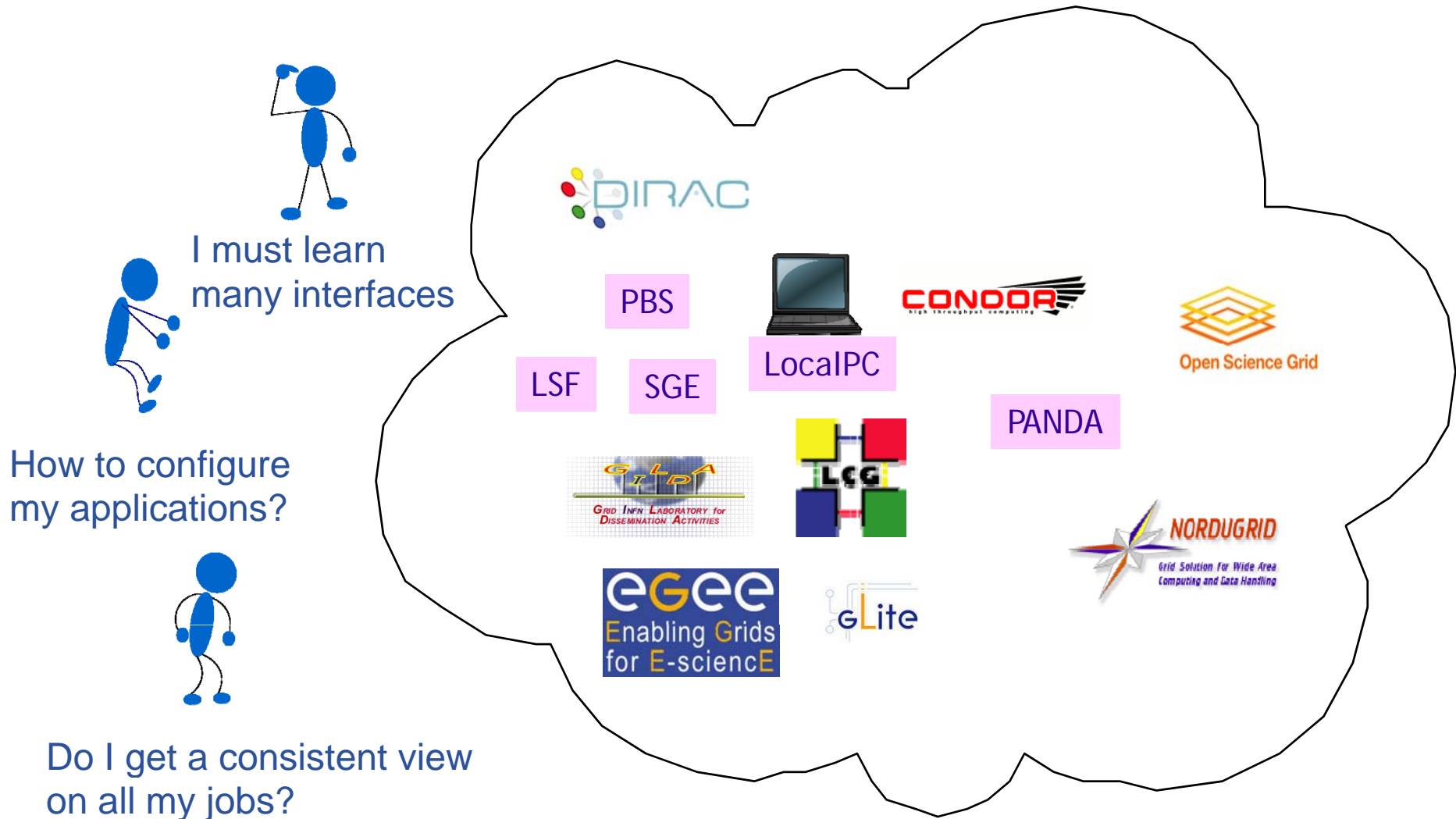
Goals:

- provide a simple and consistent way of preparing, organising and executing jobs on different computing infrastructures
- provide a clean interface which can be used:
 - interactively (CLI / python interpreter)
 - as a Python API for scripting
 - through a GUI
- Make it easy and integrated with application environment
- Allow quick transition between local PC, cluster, Grid...
- Organize work, keep history of jobs,....

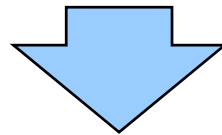
- In practice users deal with multiple computing backends



- FAQ: running applications on multiple computing backends

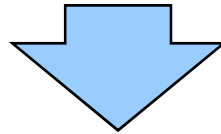


- **User requirements**
 - interact with all backend systems in a very similar way
 - submit, kill, monitor jobs
 - configure the applications easily and transparently across desired backends during different phases of application running (debugging, testing, full-scale run)
 - organize work
 - job history: keep track of what user did
 - save job outputs in a consistent way
 - reuse configuration of previously submitted jobs



Ganga

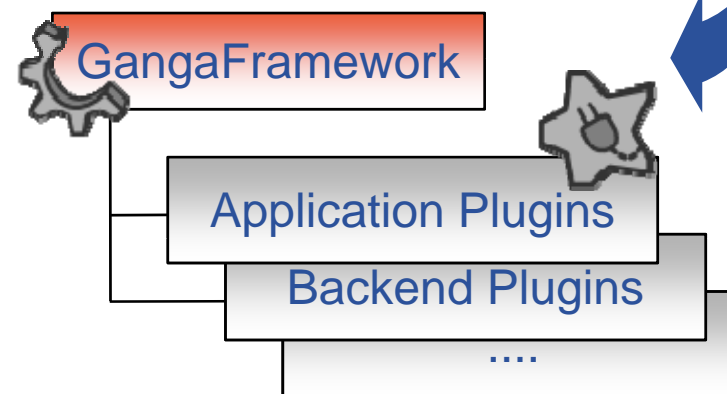
- **Wish list**
 - easy to learn and use
 - powerful: not limiting the features of the backends
 - close to the application (which is typically compiled locally)
 - not imposing single working style: scripts, command line, GUI,...



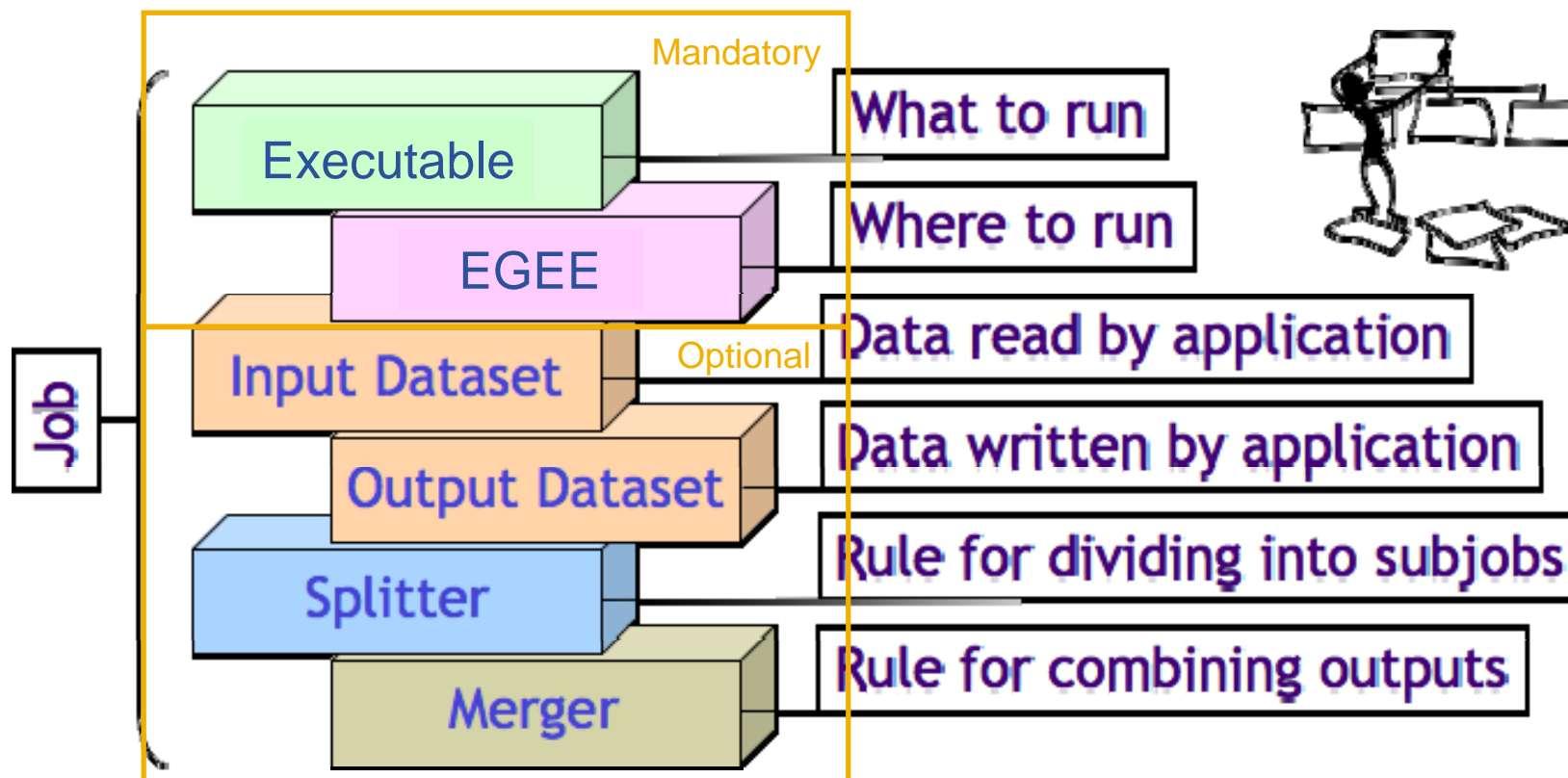
Ganga

- **Realities**
 - Computing environment is heterogeneous
 - Computing technology is evolving
 - User requirement is also evolving
- **Requirement**
 - Application users prefer to learn as few as possible the tools which are light-weight, handy and well-integrated with each other.
- **Ganga tries to answer the questions:**
 - How to minimize developer's effort in gridifying applications?
 - How to minimize user's effort in running applications?

- **Ganga: Job Management Interface**
 - a utility which you download to your computer
 - or it is already installed in your institute in a shared area
 - *for example: /nfs/sw/ganga/install/4.3.2*
 - it is an **add-on** to installed software
 - comes with a set of plugins for some applications
 - **open** - other applications and backend may be easily added
 - *even by users*



Where the Ganga journey starts ...



More than 600 users use Ganga



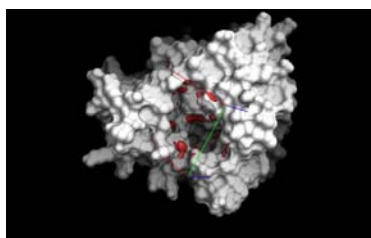
Geant 4



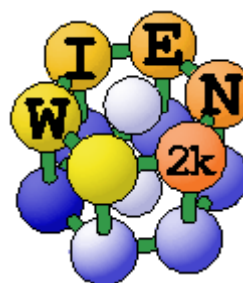
HARP



Garfield



Academia Sinica
Genomics Research Center



- Ganga is supported by HEP



- Support for development work



Science & Technology
Facilities Council



GridPP
UK Computing for Particle Physics



- Core team:

▶ F.Brochu (Cambridge), U.Egede (Imperial), J. Elmsheuser (Munich),
K.Harrison (Cambridge), H.C.Lee (ASGC Taipei), D.Liko (CERN), A.Maier (CERN), J.T.Moscicki
(CERN), A.Muraru (Bucharest), W.Reece (Imperial), A.Soroko (Oxford), CL.Tan (Birmingham)



UNIVERSITY OF
BIRMINGHAM



UNIVERSITY OF
CAMBRIDGE



Imperial College
London





Enabling Grids for E-science

Part I: Introduction to Ganga

Using Ganga

www.eu-egee.org



Download & Install

```
wget http://cern.ch/ganga/download/ganga-install
```

download installer

```
python ganga-install \
  --prefix=/usr/local/ganga/prefix \
  --extern=GangaGUI,GangaPlotter \
  4.3.2
```

installation prefix

Installation of external modules

Ganga version

First Launch

```
export PATH=$HOME/opt/ganga/install/4.3.2/bin:$PATH
```

start Ganga with inline configurations

```
$ ganga
```

```
*** Welcome to Ganga ***
Version: Ganga-4-3-2
Documentation and support: http://cern.ch/ganga
Type help() or help('index') for online help.
```

Ganga CLIP

```
In [1]:
Do you really want to exit ([y]/n)?
```

<ctrl>-D to exit Ganga CLIP

- **submit and run a test job on local machine:** `Job().submit()`
- **submit and run a test job on LCG:** `Job(backend=LCG()).submit()`
- **browse the created jobs (job history):** `jobs`
- **get the first job from the job history:** `j = jobs[0]`
- **print the details of the job and see what you can set for a job:** `j`
- **make a copy of the job and submit the new job:** `j.copy().submit()`
- **see what you can do with the job:** `j.<tab>`
- **get interactive help:** `help`

Syntax

```
[Configuration]
TextShell = IPython
... ..
[LCG]
EDG_ENABLE = True
... ..
```

Python ConfigParser standard

How to set configurations

Hard-coded default configurations

release config

site config

```
export GANGA_CONFIG_PATH=/usr/local/ganga/prefix/install/etc/Gilda.ini:GangaTutorial/Tutorial.ini
ganga --config-path= /usr/local/ganga/prefix/install/etc/Gilda.ini:GangaTutorial/Tutorial.ini
```

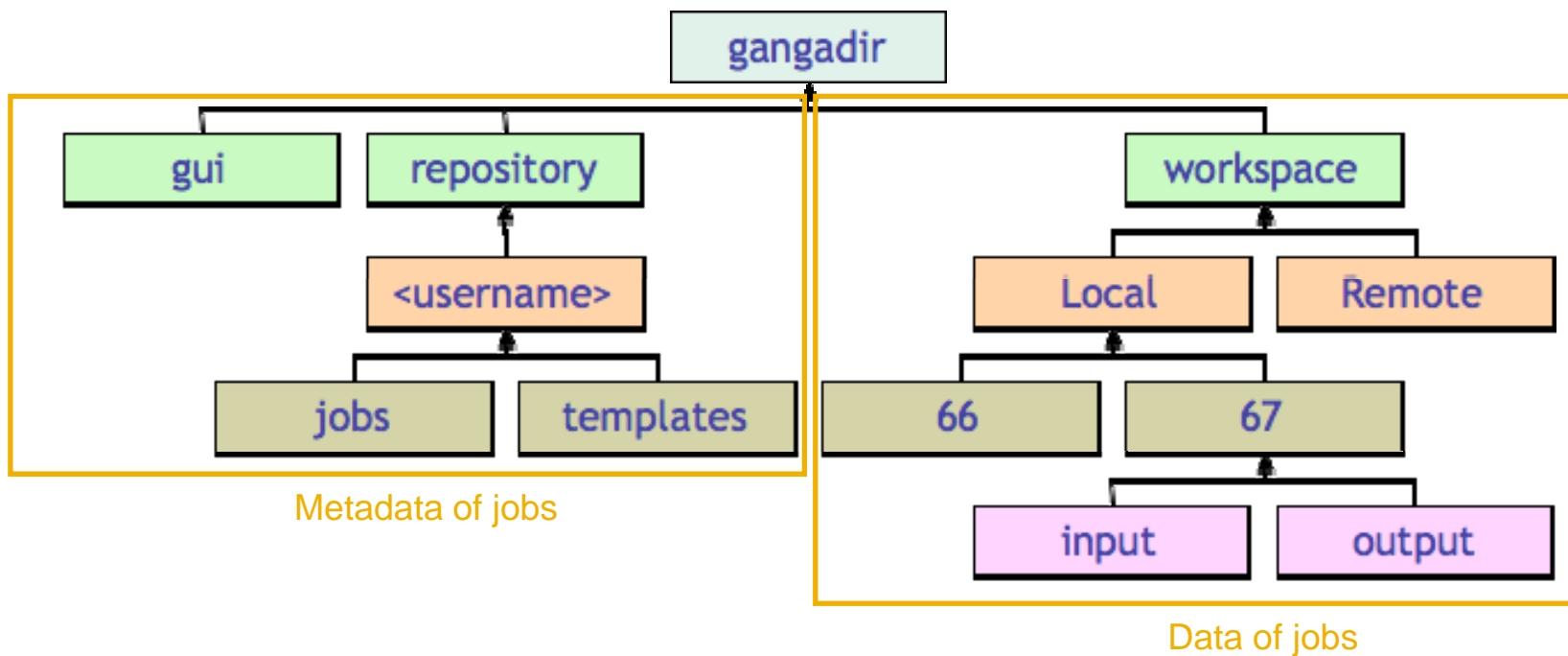
```
~/ .gangarc
ganga -o
```

user config

Override sequence

```
user config > site config > release config
```

- **gangadir** folder is created at the first launch within \$HOME directory
- To locate it in different directory:
 - [DefaultJobRepository] local_root = /alternative/gangadir/repository
 - [FileWorkspace] topdir = /alternative/gangadir/workspace



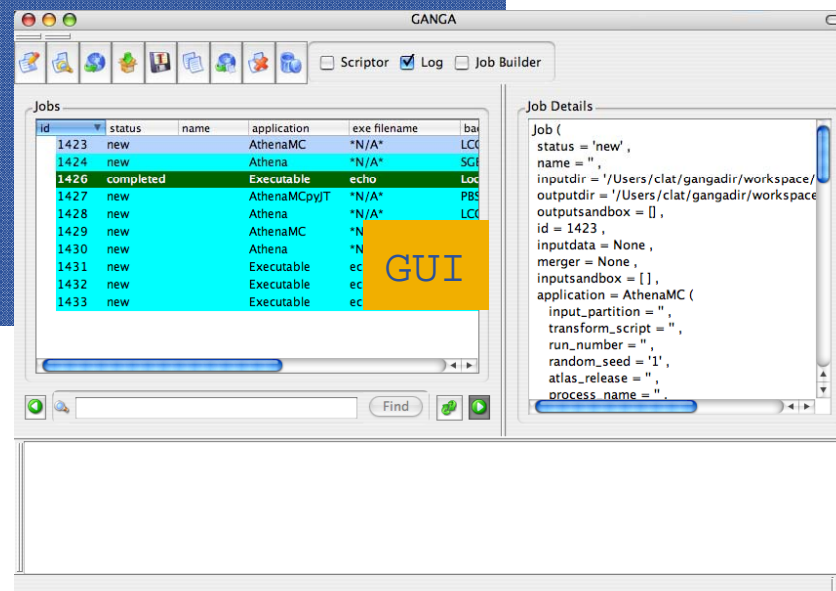
```

*** Welcome to Ganga ***
Version: Ganga-4-2-8
Documentation and support: http://cern.ch/ganga
Type help() or help('index') for online help.
    
```

```

In [1]: jobs
Out[1]: Statistics: 1 jobs
-----
#   id      status      name      subjobs      application
backend.actualCE
#   1      completed
compute.hpc.unimelb.edu.au:2119/jobmanage
    
```

CLIP



GUI

```

#!/usr/bin/env ganga
#-*-python-*-
import time
j = Job()
j.backend = LCG()
j.submit()
while not j.status in ['completed','failed']:
    print('job still running')
    time.sleep(30)
    
```



```

./myjob.exec
ganga ./myjob.exec
In [1]:execfile("myjob.exec")
    
```

GPI & Scripting

- <tab> completion
- <page up/down> for cmd history
- system command integration
- Job template
- In[1]: plugins()
- plugins('backends')
- In[2]: help()
- etc.

```
In[1]: j = jobs[1]
In[2]: cat $j.outputdir/stdout
Hello World
```

```
In[1]: t = JobTemplate(name='lcg_simple')
In[2]: t.backend = LCG(middleware='EDG')
In[3]: templates
Out[3]: Statistics: 1 templates
-----
#   id      status      name      subjobs
application      backend
backend.actualCE
#     3      template  lcg_simple
Executable      LCG

In[4]: j = Job(templates[3])
In[5]: j.submit()
```

```
$ ganga athena \
--inDS myInputDataset.txt\
--outputdata myOutput.root \
--split 3 \
--maxevt 100 \
--lsf \
jobOptions.py
```

Scripting mode

quick

```
j = Job()
j.application=Athena()
j.application.prepare()
j.application.option_file='jobOptions.py'
```

CLIP mode
application

```
j.inputdata=DQ2Dataset()
j.inputdata.type='DQ2_LOCAL'
j.inputdata.dataset="myInputDataset.txt"
```

inputdata

```
j.outputdata=DQ2OutputDataset()
j.outputdata.outputdata=['myOutput.root']
```

outputdata

```
j.splitter = AthenaSplitterJob(numsubjobs=3)
j.merger = AthenaOutputMerger()
```

Splitter & Merger

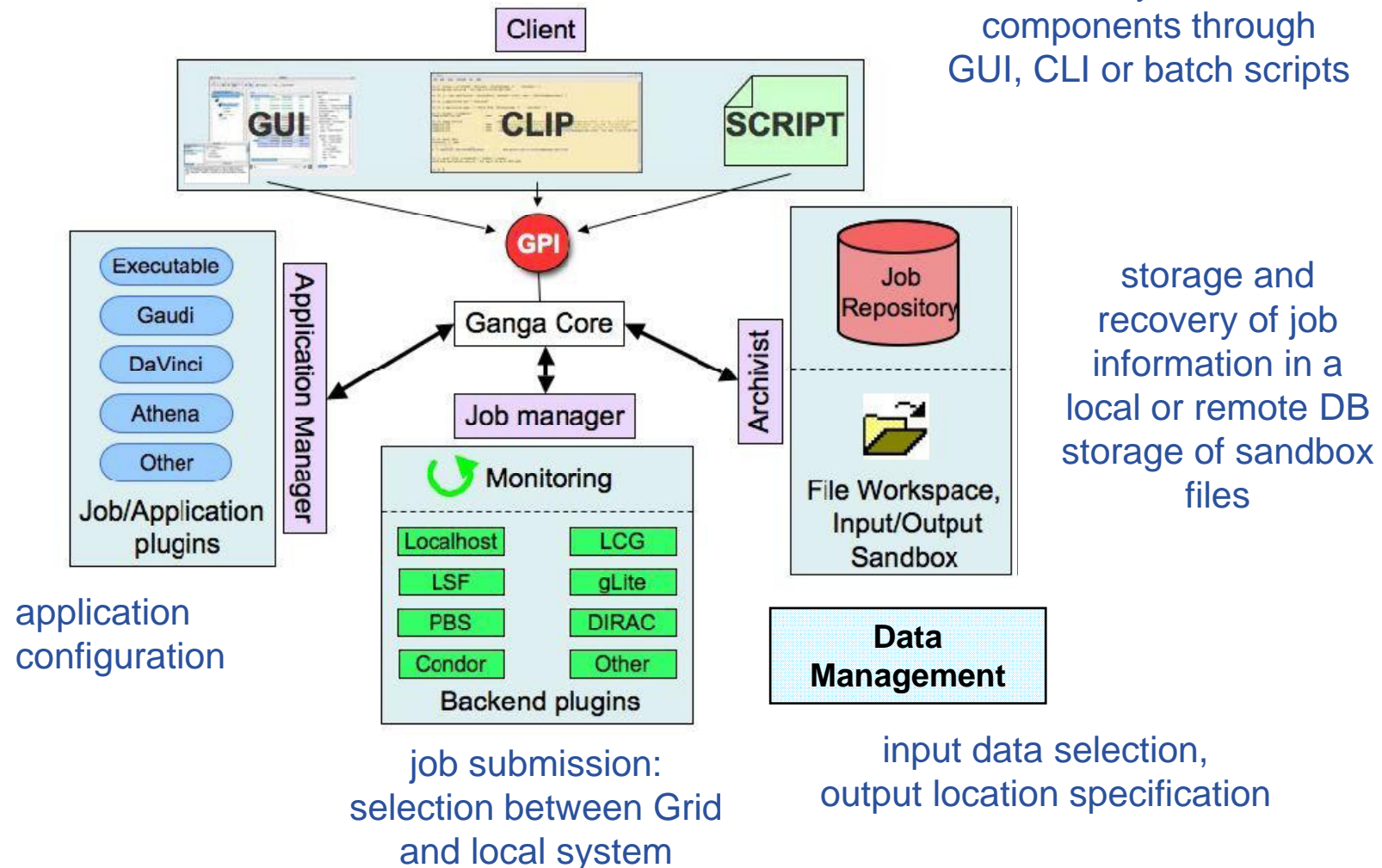
```
j.backend = LSF()
j.submit()
```

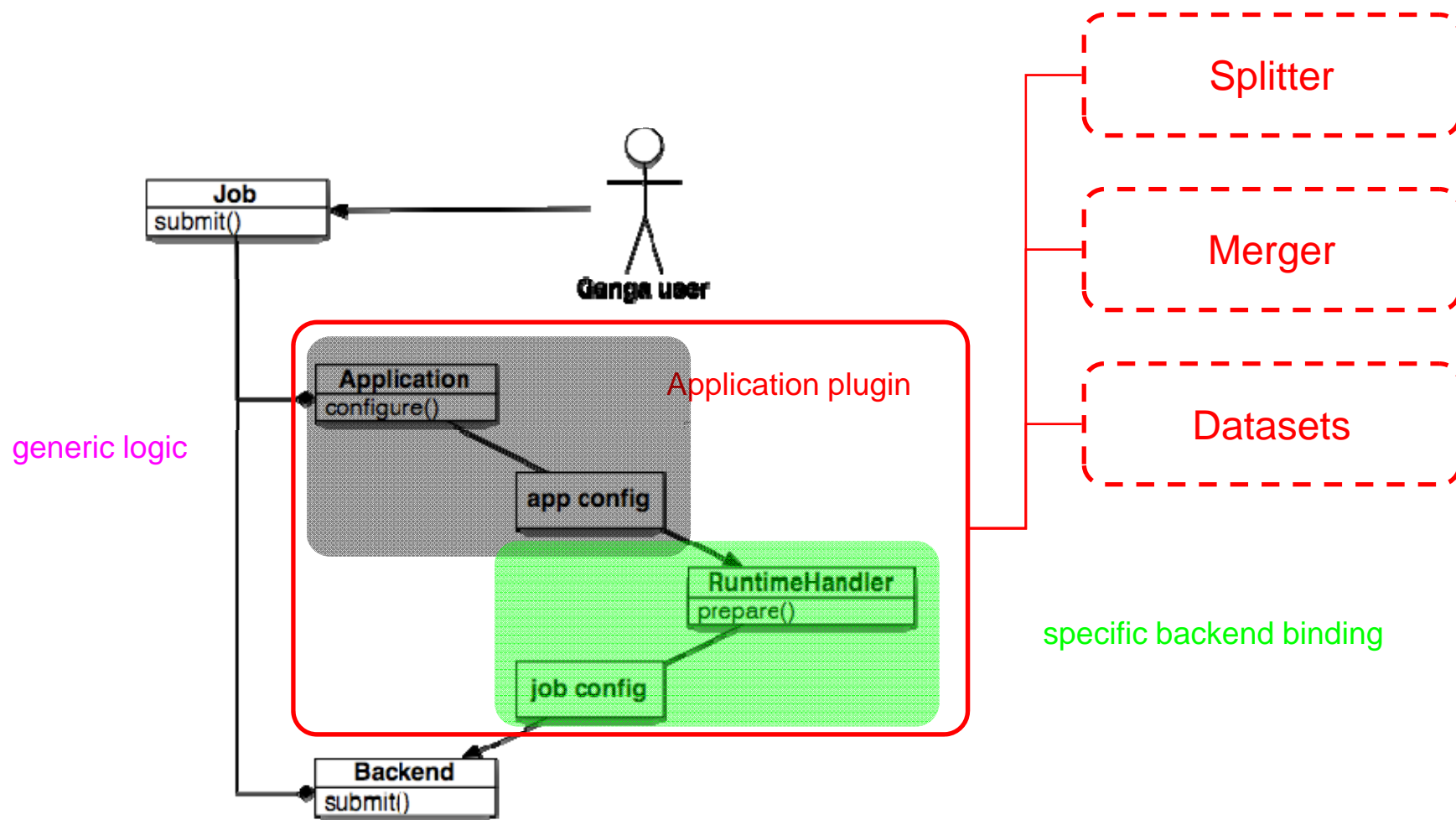
```
j2 = j.copy()
j2.backend=LCG( CE='ce102.cern.ch:2119/jobmanager-lcglsf-grid_2nh_atlas' )
j2.submit()
```



flexible

User has access to functionality of GANGA components through GUI, CLI or batch scripts





Detailed guide available at:

<https://twiki.cern.ch/twiki/bin/view/ArdaGrid/HowToPlugInNewApplicationType>

A good example: `python/GangaTutorial/Lib/*.py`

- **What is Ganga?**
 - An easy-to-use front-end for job definition and management
 - A generic framework extensible for specific applications
 - A light-weight application component fully implemented in Python
- **What Ganga can help in application development?**
 - Ganga provides a set of ready-to-use APIs for high-level job management
 - It simplifies developers' effort in developing applications
 - End-users can easily extend Ganga for their own purpose



Enabling Grids for E-scienceE

Part II: Ganga hands-on

www.eu-egee.org



<https://twiki.cern.ch/twiki/bin/view/ArdaGrid/EGEETutorialPackage>

- **Skip the installation step**
- **Start your Ganga CLIP session:**

```
shell> ganga
```

In [1]: `!pico myscript.sh`

```
#!/bin/sh
echo "Hello ${1} !"
echo $HOSTNAME
cat /proc/cpuinfo | grep 'model name'
cat /proc/meminfo | grep 'MemTotal'
echo "Run on `date`"
```

In [2]: `!chmod +x myscript.sh`

In [2]: `j = Job()`

In [3]: `j.application = Executable()`

In [4]: `j.application.exe = File('myscript.sh')`

In [5]: `j.application.args = ['Budapest']`

In [6]: `j.backend = Interactive()`

In [7]: `j.submit()`

In [8]: `jobs`

Step 2: Your first Ganga job - an arbitrary shell script

```
In [9]: j = j.copy()
In [10]: j.backend = Local()
In [11]: j.submit()

In [12]: jobs

In [13]: j.peek()
In [14]: cat $j.outputdir/stdout
```

```
./myscript.sh Budapest
```

Step 3: your first Ganga job on the Grid

```
In [15]:j = j.copy()
```

```
In [16]:j.backend = LCG()
```

```
In [17]:j.application.args = ['Grid']
```

```
In [18]:j.submit()
```

```
In [19]:j
```

```
In [20]:cat $j.backend.loginfo(verbosity=1)
```

```
In [21]:jobs
```

- **Please follow the instructions from:**

https://twiki.cern.ch/twiki/bin/view/ArdaGrid/EGEETutorialPackage#Exercise_Prime_number_factorizat



Enabling Grids for E-scienceE

Part III: More about Ganga

www.eu-egee.org





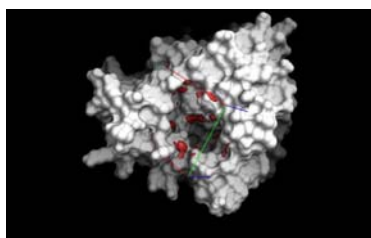
Geant 4



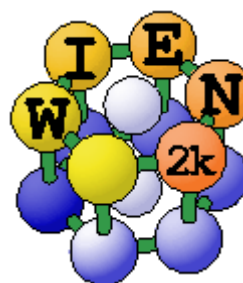
HARP



Garfield

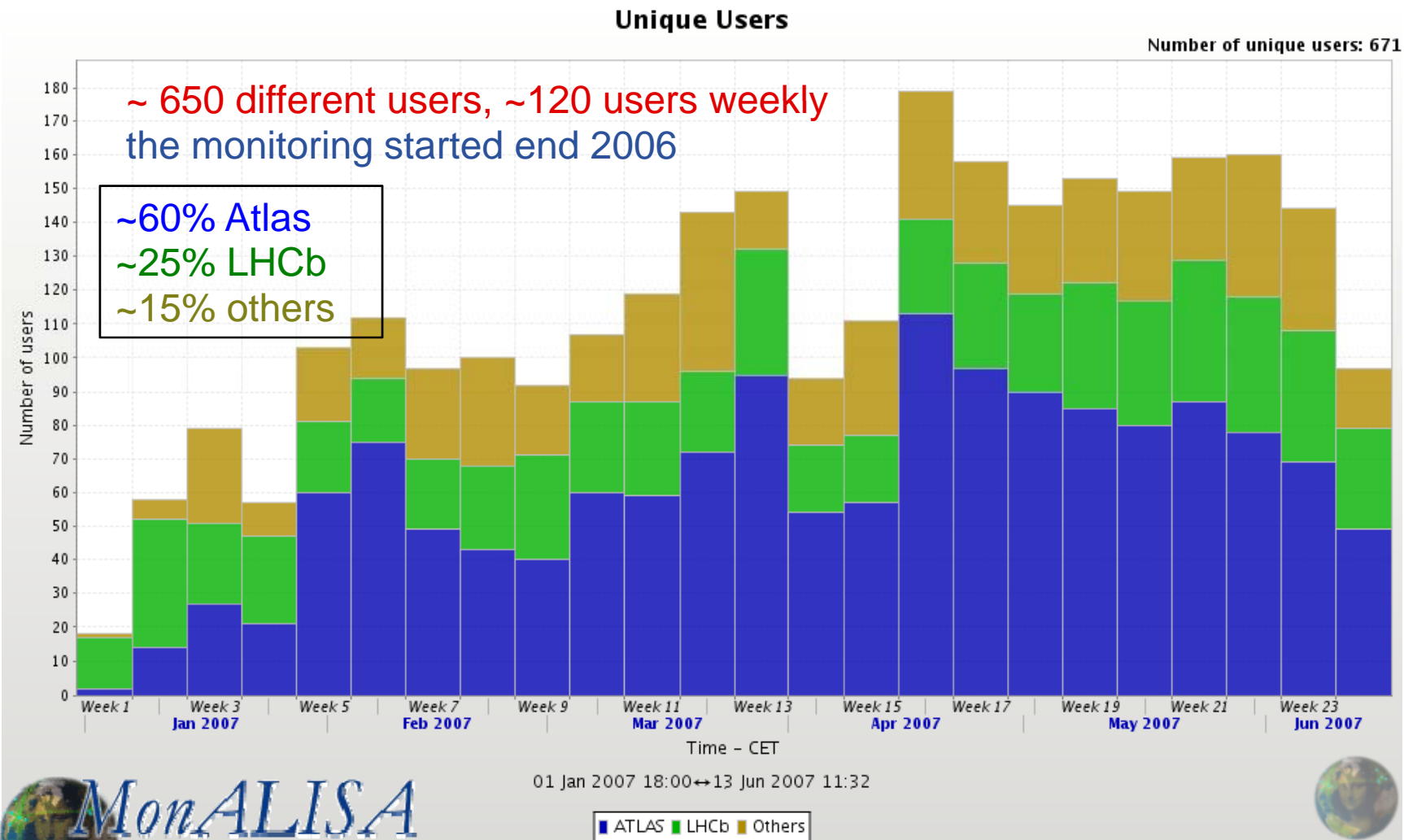


Academia Sinica
Genomics Research Center

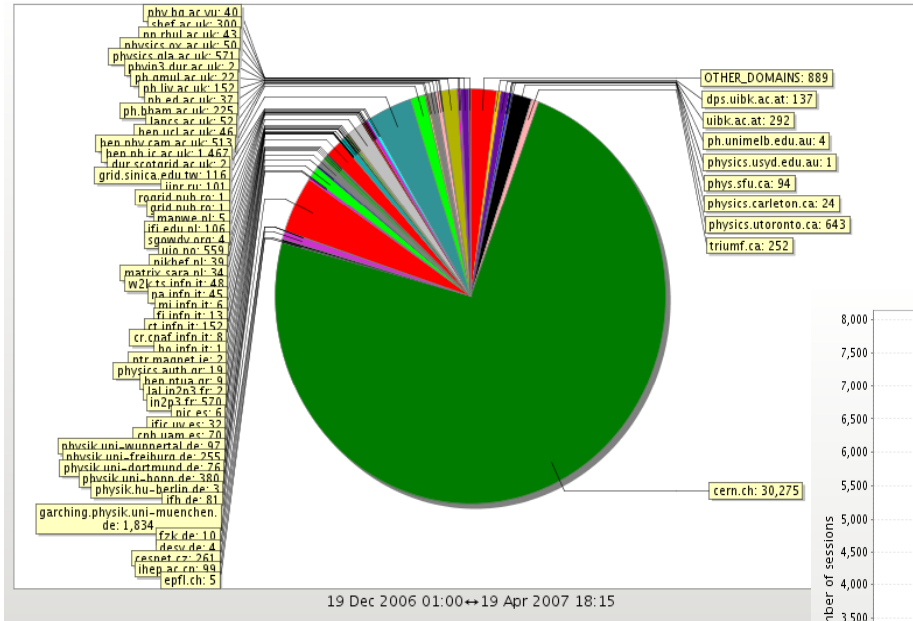


med
austron



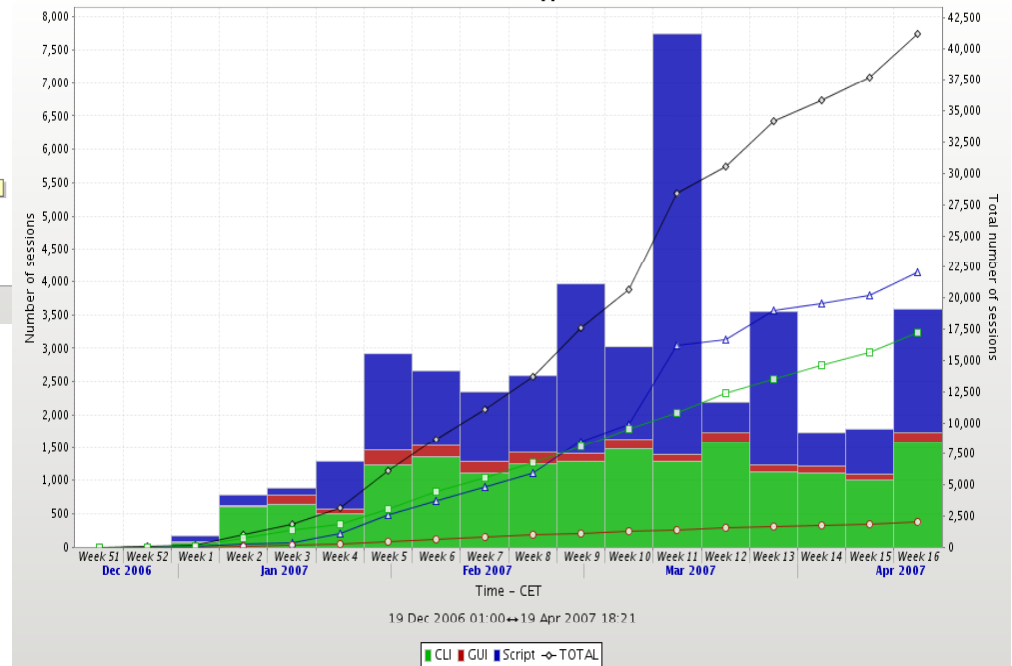


Total number of sessions per domain



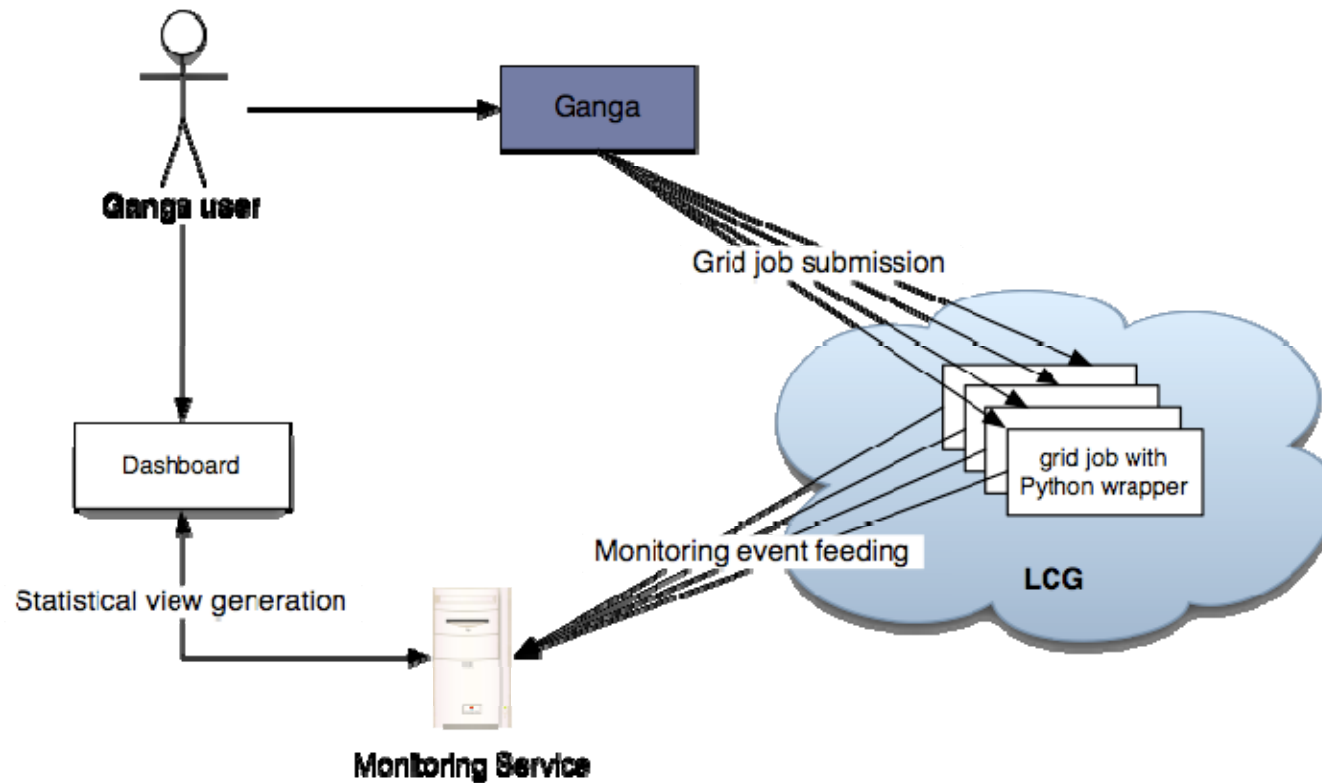
over 50 local sites

Session Type

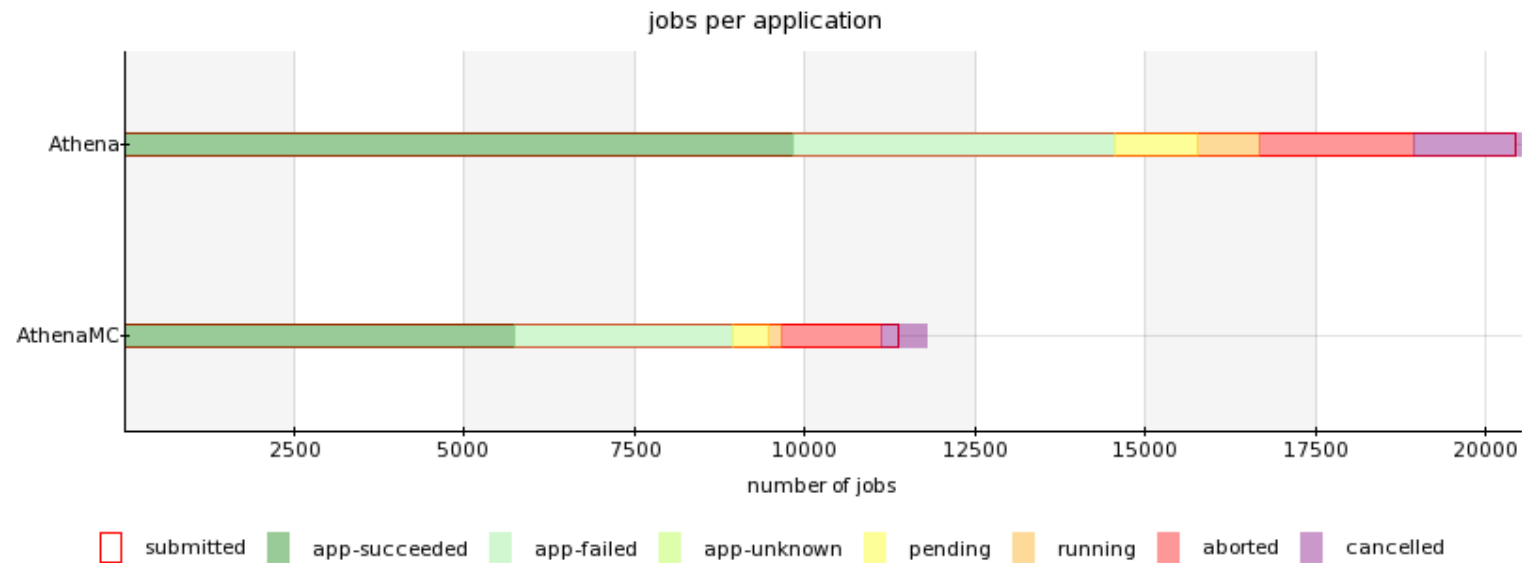


CLIP and scripts most popular





- analysis and simulations jobs run in ATLAS
 - one month (mid-March to mid-April 2007)
 - 20K analysis jobs, 10K simulation jobs
 - only LCG jobs, others not shown
 - data collected by a **ARDA Dashboard** sensor integrated with Ganga



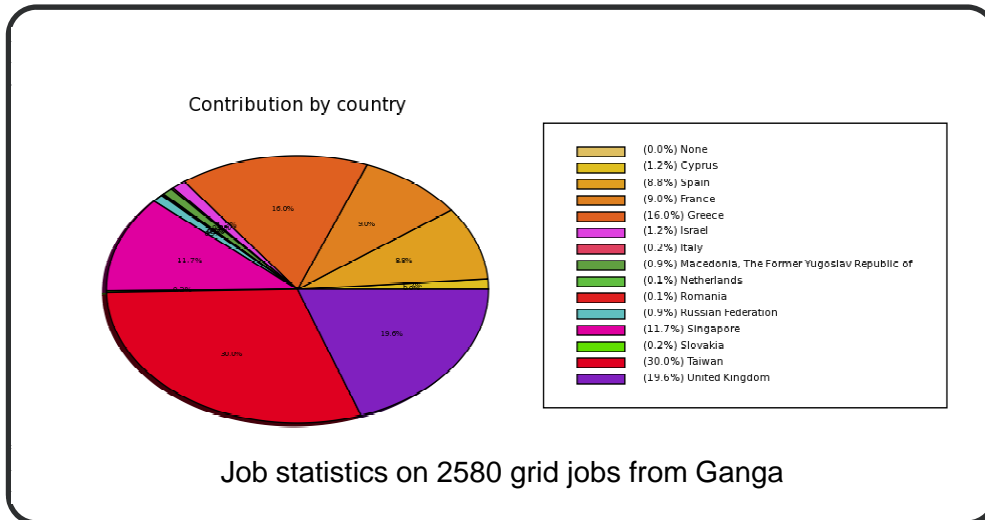
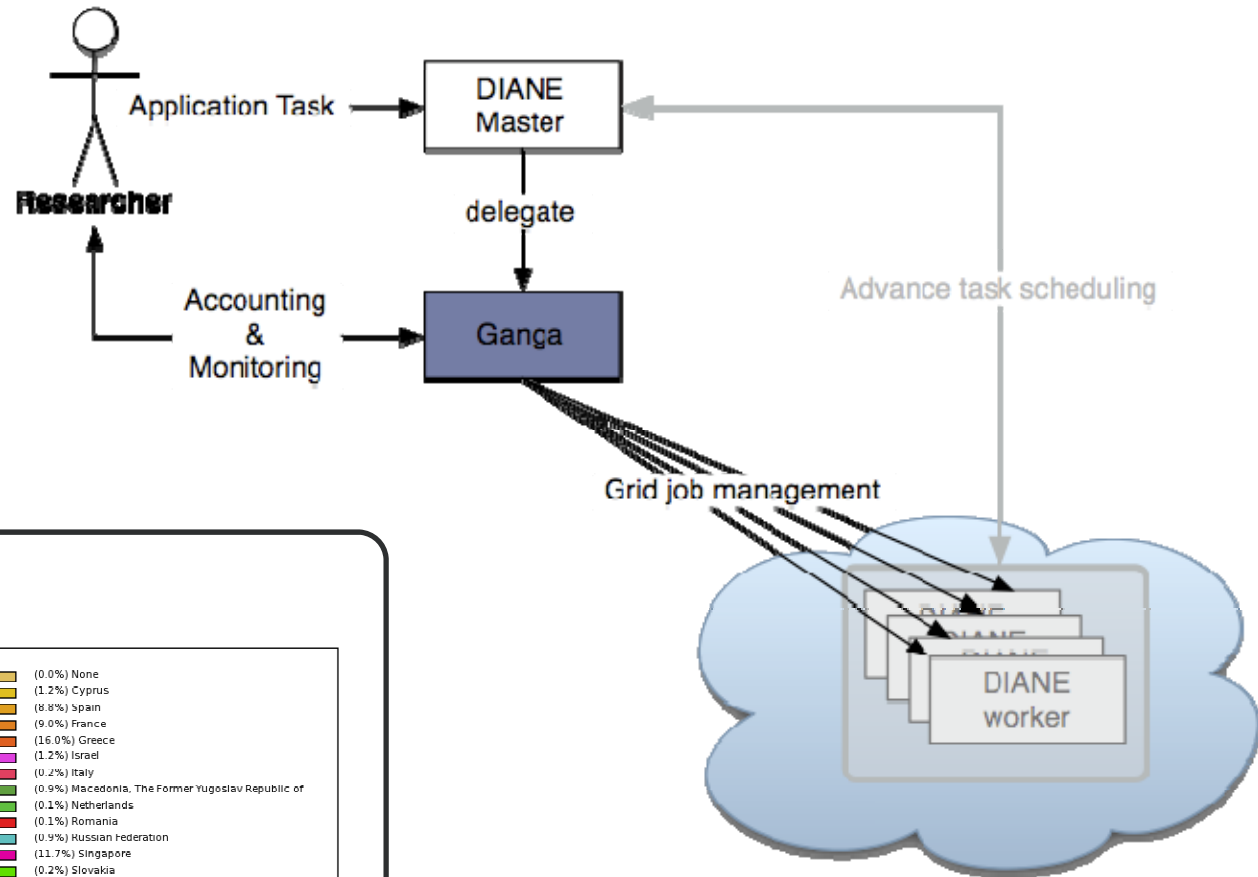
BBC NEWS

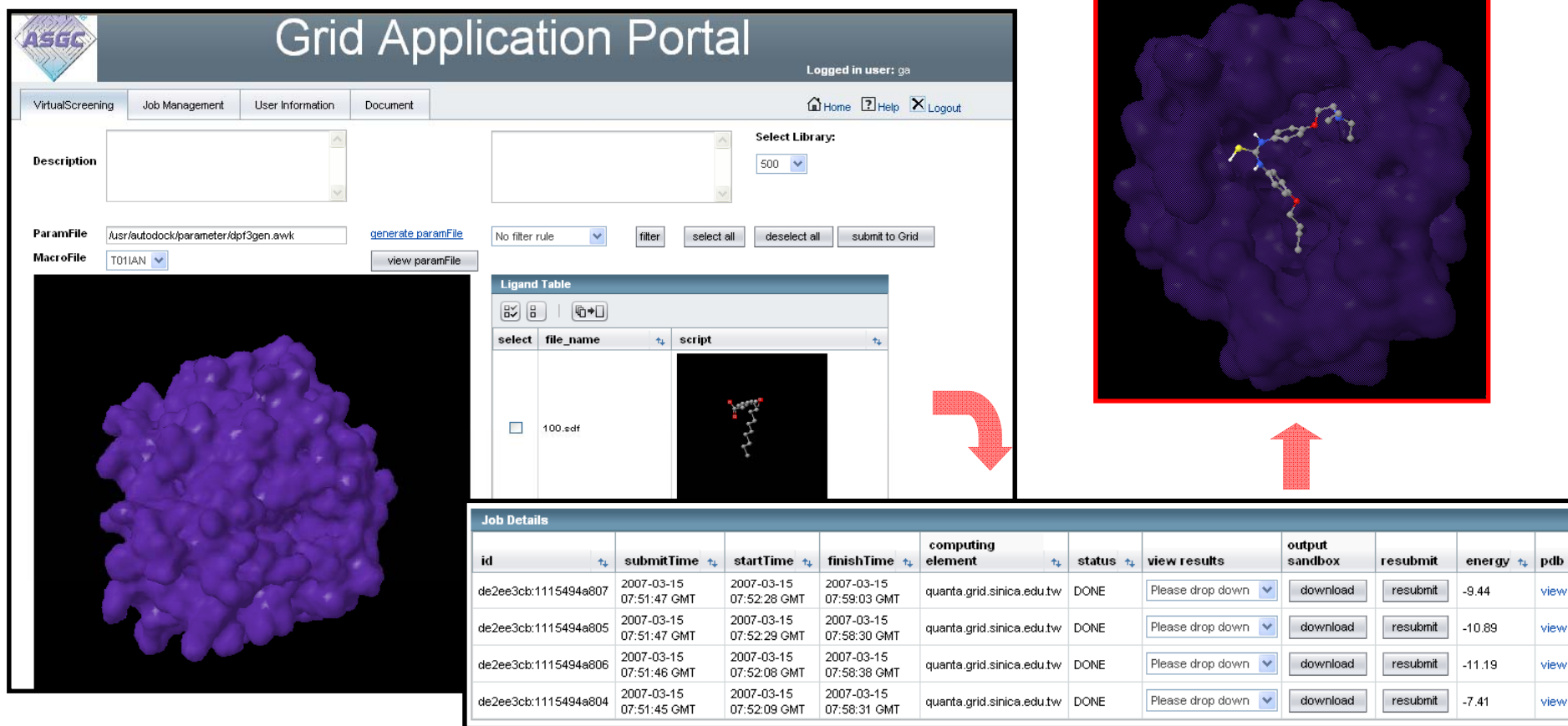
Grid searches for avian flu cure

A cure for bird flu is being sought by computers that usually search for the fundamental elements of matter.



Geant 4





Grid Application Portal

Logged in user: ga

VirtualScreening | Job Management | User Information | Document

Home | Help | Logout

Description: [Text Area]

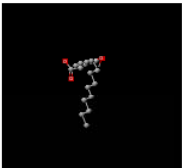
ParamFile: /usr/autodock/parameter/dpf3gen.awk [generate paramFile](#)

MacroFile: T01IAN [view paramFile](#)

Select Library: 500

No filter rule | filter | select all | deselect all | submit to Grid

Ligand Table

select	file_name	script
<input type="checkbox"/>	100.sdf	

Job Details

id	submitTime	startTime	finishTime	computing element	status	view results	output sandbox	resubmit	energy	pdb
de2ee3cb:1115494a807	2007-03-15 07:51:47 GMT	2007-03-15 07:52:28 GMT	2007-03-15 07:59:03 GMT	quanta.grid.sinica.edu.tw	DONE	Please drop down	download	resubmit	-9.44	view
de2ee3cb:1115494a805	2007-03-15 07:51:47 GMT	2007-03-15 07:52:29 GMT	2007-03-15 07:58:30 GMT	quanta.grid.sinica.edu.tw	DONE	Please drop down	download	resubmit	-10.89	view
de2ee3cb:1115494a806	2007-03-15 07:51:46 GMT	2007-03-15 07:52:08 GMT	2007-03-15 07:58:38 GMT	quanta.grid.sinica.edu.tw	DONE	Please drop down	download	resubmit	-11.19	view
de2ee3cb:1115494a804	2007-03-15 07:51:45 GMT	2007-03-15 07:52:09 GMT	2007-03-15 07:58:31 GMT	quanta.grid.sinica.edu.tw	DONE	Please drop down	download	resubmit	-7.41	view

- Interface created by biologists (Model-View-Controller design pattern)
 - The **Model** makes use of **Ganga** as a submission tool and **DIANE** to better handle docking jobs on the Grid
 - The **Controller** organizes a set of actions to perform the virtual screening pipeline; The **View** represents biological aspects

- **Ganga Home:** <http://cern.ch/ganga>
- **Official Ganga User's Guide:** <http://cern.ch/ganga/user/html/GangaIntroduction/>
- **GangaTutorial GPI Reference Manual :**
<http://ganga.web.cern.ch/ganga/release/4.3.2/reports/html/Manuals/GangaTutorialManual.html>
- **Looking for help:** project-ganga-developers@cern.ch