



Enabling Grids for E-scienceE

Ganga Tutorial

Hurng-Chun Lee

www.eu-egee.org



- **Part I: Introduction to Ganga**
 - Ganga overview
 - Using Ganga
- **Part II: Ganga hands-on**
- **Part III: More about Ganga**



Enabling Grids for E-science

Part I: Introduction to Ganga

Ganga overview

www.eu-egee.org

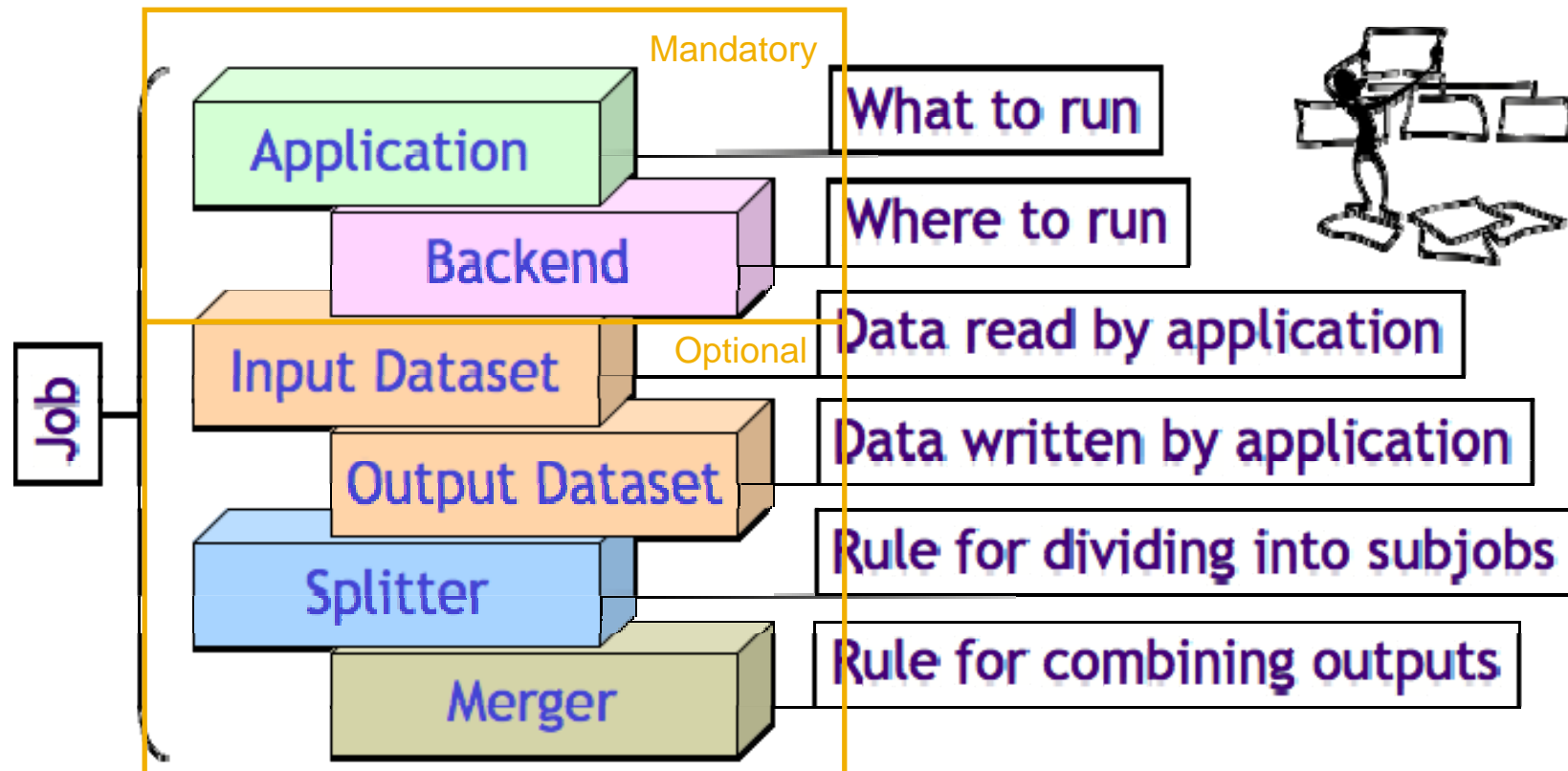


- **Computing environment is heterogeneous**
- **Computing technology is evolving**
- **User requirement is also evolving**
- **Application users prefer to learn as few as possible the tools which are light-weight, handy and well-integrated with each other.**
- **Ganga tries to answer the questions:**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

- **GANGA: Gaudi/Athena and Grid Alliance**
 - jointly developed by ATLAS and LHCb experiments
 - **It's application neutral**
- **A common framework extensible for not only HEP applications**
- **An easy-to-use front-end for job definition and management**
- **A light-weight application component fully implemented in Python**

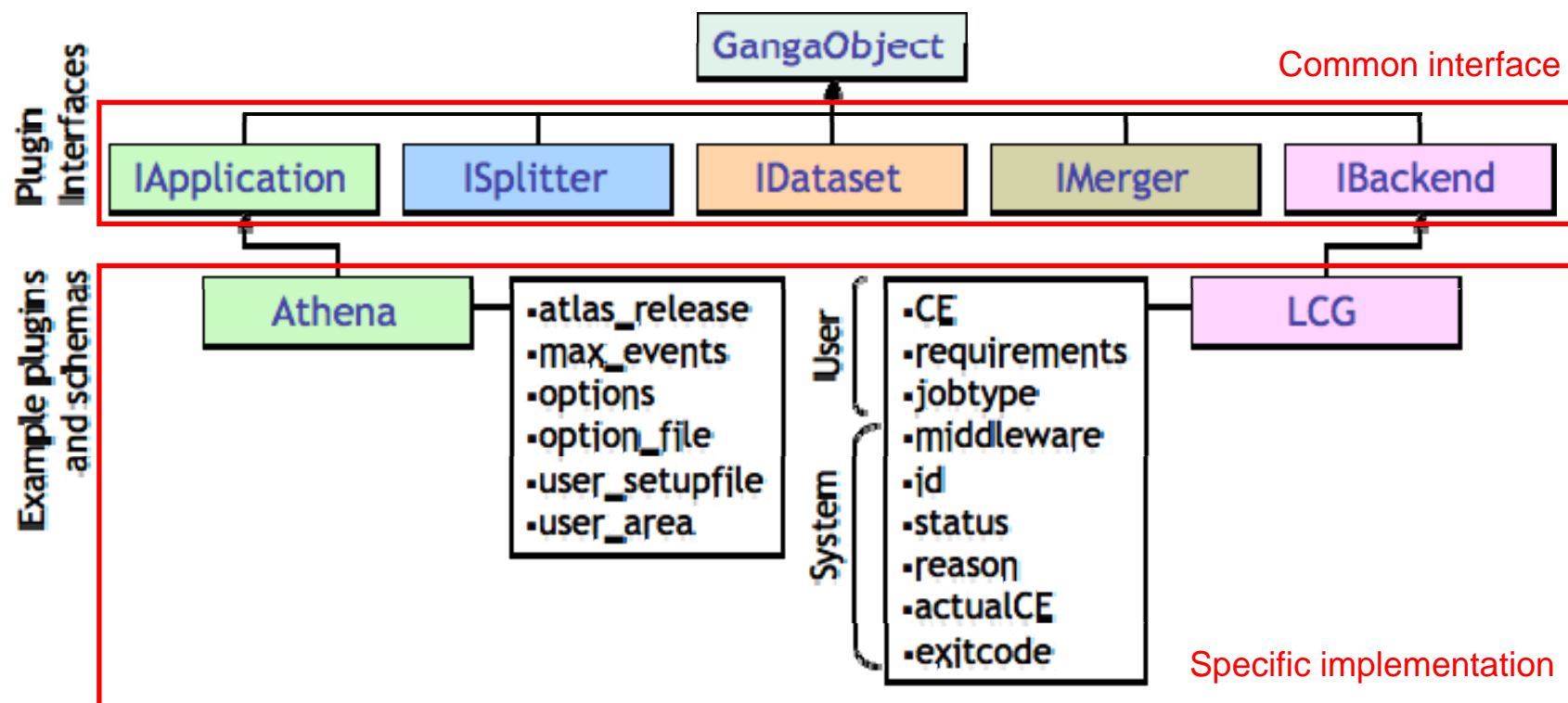
- **F.Brochu (Cambridge), U.Egede (Imperial), J.Elmsheuser (München), K.Harrison (Cambridge), H.C.Lee (ASGC), D.Liko (CERN), A.Maier (CERN), J.T.Moscicki (CERN), A.Muraru (Bucharest), V.Romanovsky (IHEP), A.Soroko (Oxford), C.L.Tan (Birmingham)**
- **Weekly phone meeting**
- **Developers' day before every major release**

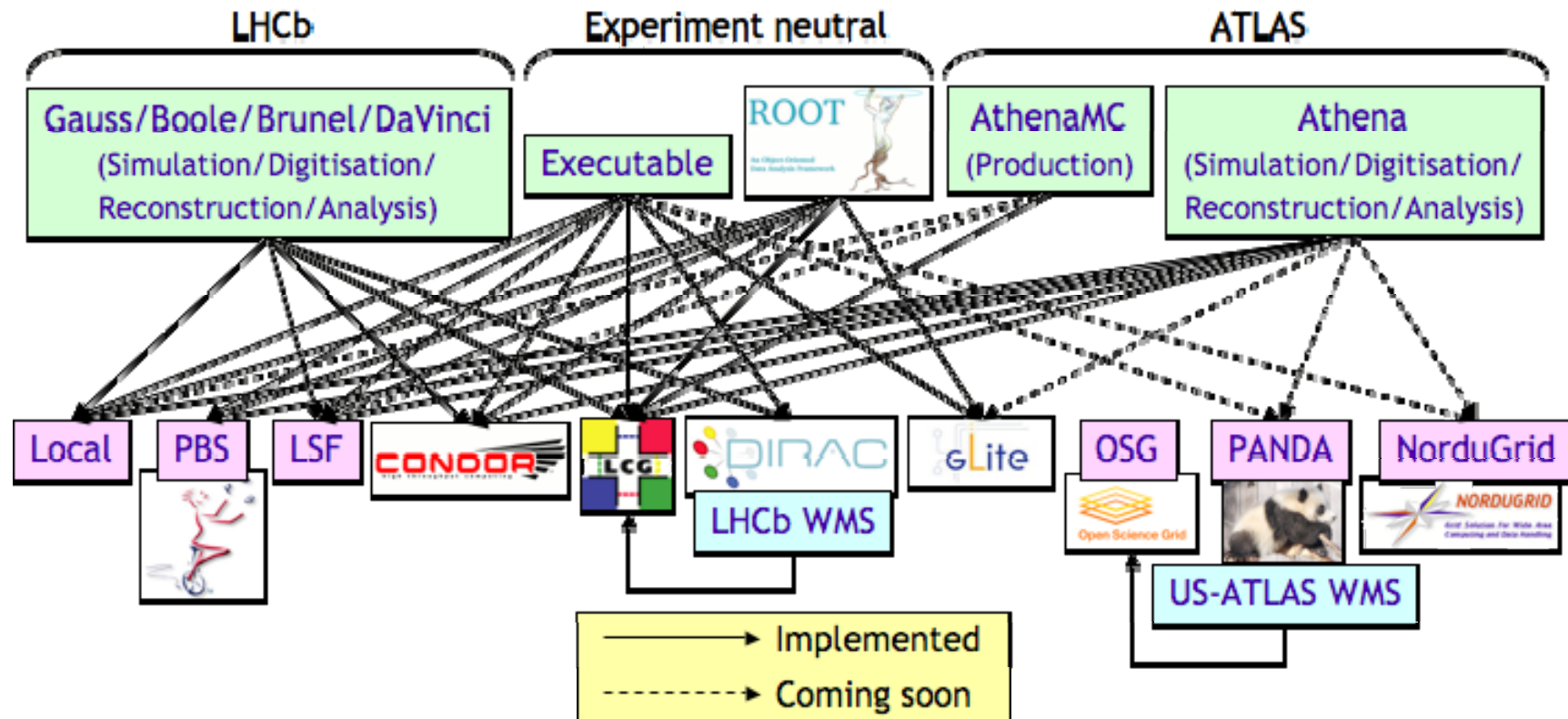
Where the Ganga journey starts ...



📖 Ease user's experience in switching between different technologies

📖 Concentrate developer's effort in specific domain







Enabling Grids for E-science

Part I: Introduction to Ganga

Using Ganga

www.eu-egee.org



Download & Install

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

download installer

```
python ganga-install \
  --prefix=~/.opt/ganga \ installation prefix
  --extern=GangaAtlas,GangaGUI,GangaPlotter \ Installation of external modules
  4.2.8 Ganga version
```

First Launch

```
export PATH $HOME/opt/ganga/install/slc3_gcc323/4.2.8/bin:$PATH
```

start Ganga with inline configurations

```
ganga -o' [LCG]ENABLE_EDG=False' -o' [LCG]ENABLE_GLITE=False'
```

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

Ganga CLIP

<ctrl>-D to exit Ganga CLIP

- `Job().submit()` **submit and run a test job on local machine**
- `Job(backend=LCG()).submit()` **submit and run a test job on LCG**
- `jobs` **browse the created jobs (job history)**
- `j = jobs[1]` **get the first job from the job history**
- `j` **print the details of the job and see what you can set for a job**
- `j.copy().submit()` **make a copy of the job and submit the new job**
- `j.<tab>` **see what you can do with the job**

Syntax

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

Python ConfigParser standard

How to set configurations

Hardcoded configurations

release config

```
export SITE_CONFIG_PATH = /some/physics/subgroup.ini:GangaLHCb/LHCb.ini
ganga --config-path=/some/pyhsycis/subgroup.ini:GangaLHCb/LHCb.ini
```

site config

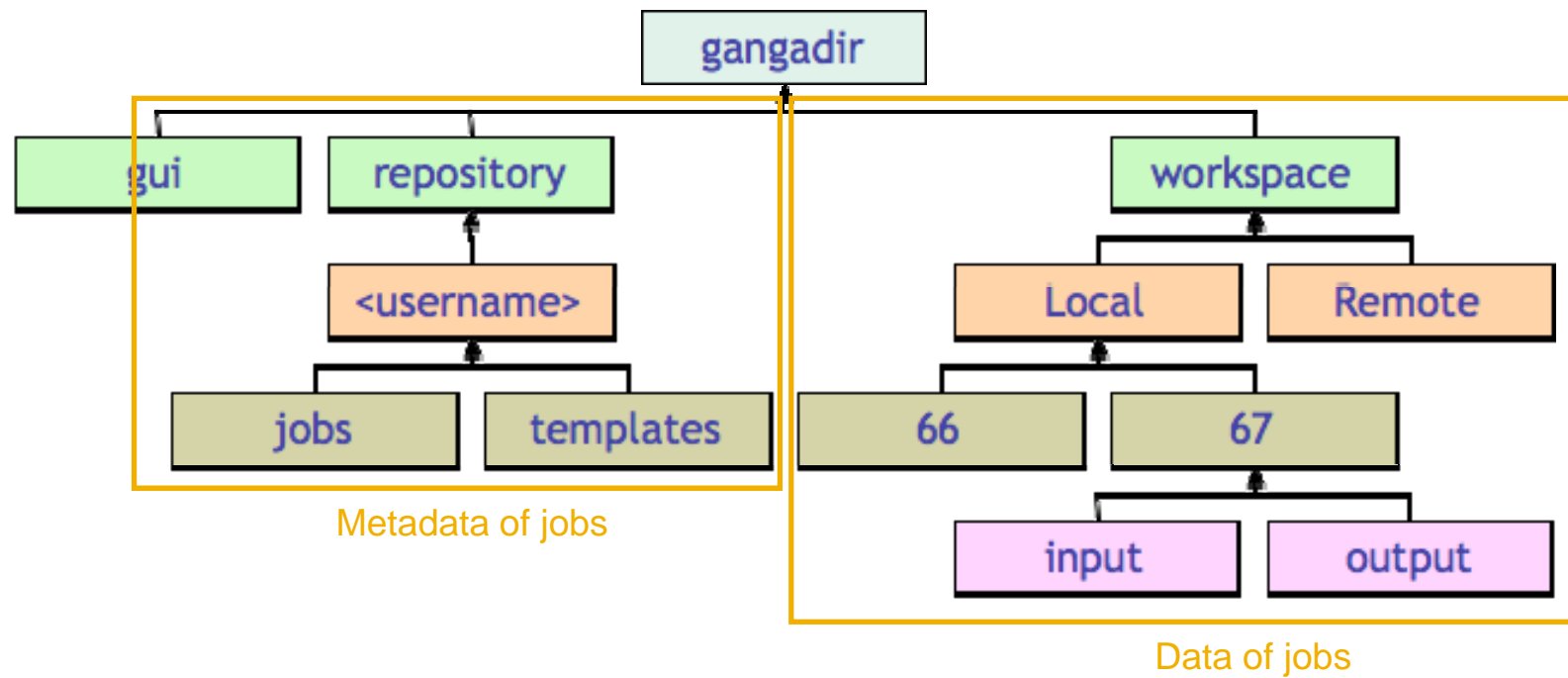
```
~/ .gangarc
ganga -o
```

user config

Sequenc

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

- created at the first launch
- [DefaultJobRepository] local_root = /alternative/gangadir



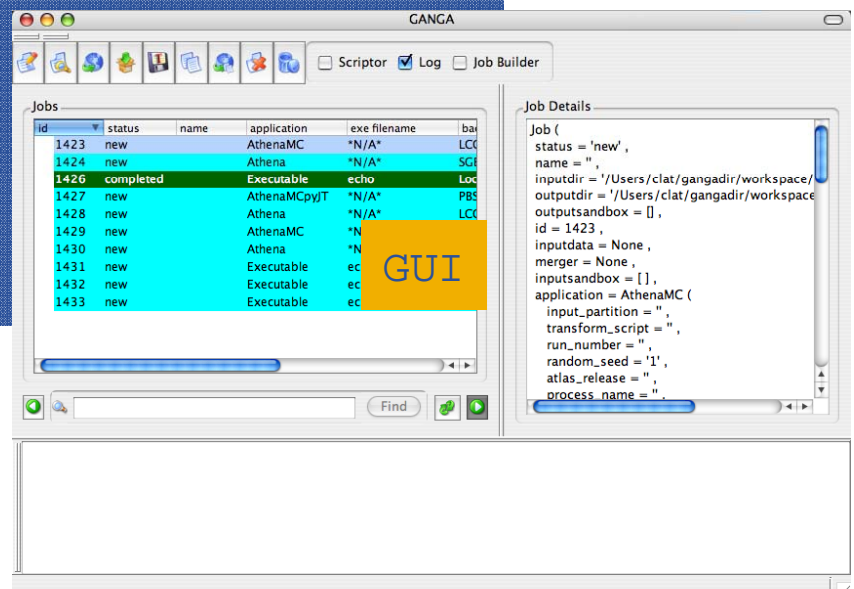
```

*** 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
#   3      template  lcg simple      backend.actualCE
LCG                                     Executable

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



```

j = Job()
j.application = Athena()
j.application.option_file = 'myOpts.py'
j.application.prepare(athena_compile = False)

j.inputdata = DQ2Dataset()
j.inputdata.dataset = 'interestingDataset.AOD.v12003104'
j.inputdata.type = 'DQ2_Local'

j.outputdata = AthenaOutputDataset()
j.outputdata.outputdata = 'myOutput.root'

j.splitter = AthenaSplitterJob(numsubjobs=2)
j.merger = AthenaOutputMerger()

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

CLIP mode
application

inputdata

Outputdata

Splitter & Merger

Scripting mode

```

ganga athena \
--inDS misalg_csc11.005300.PythiaH130zz41_recon.AOD.v12003104 \
--outputdata AnalysisSkeleton.aan.root \
--split 3 \
--maxevt 100 \
--lcg \
--ce ce102.cern.ch:2119/jobmanager-lcglsf-grid_2nh_atlas \
AnalysisSkeleton_topOptions.py
    
```

Use Ganga's athena script

Input dataset

Output data

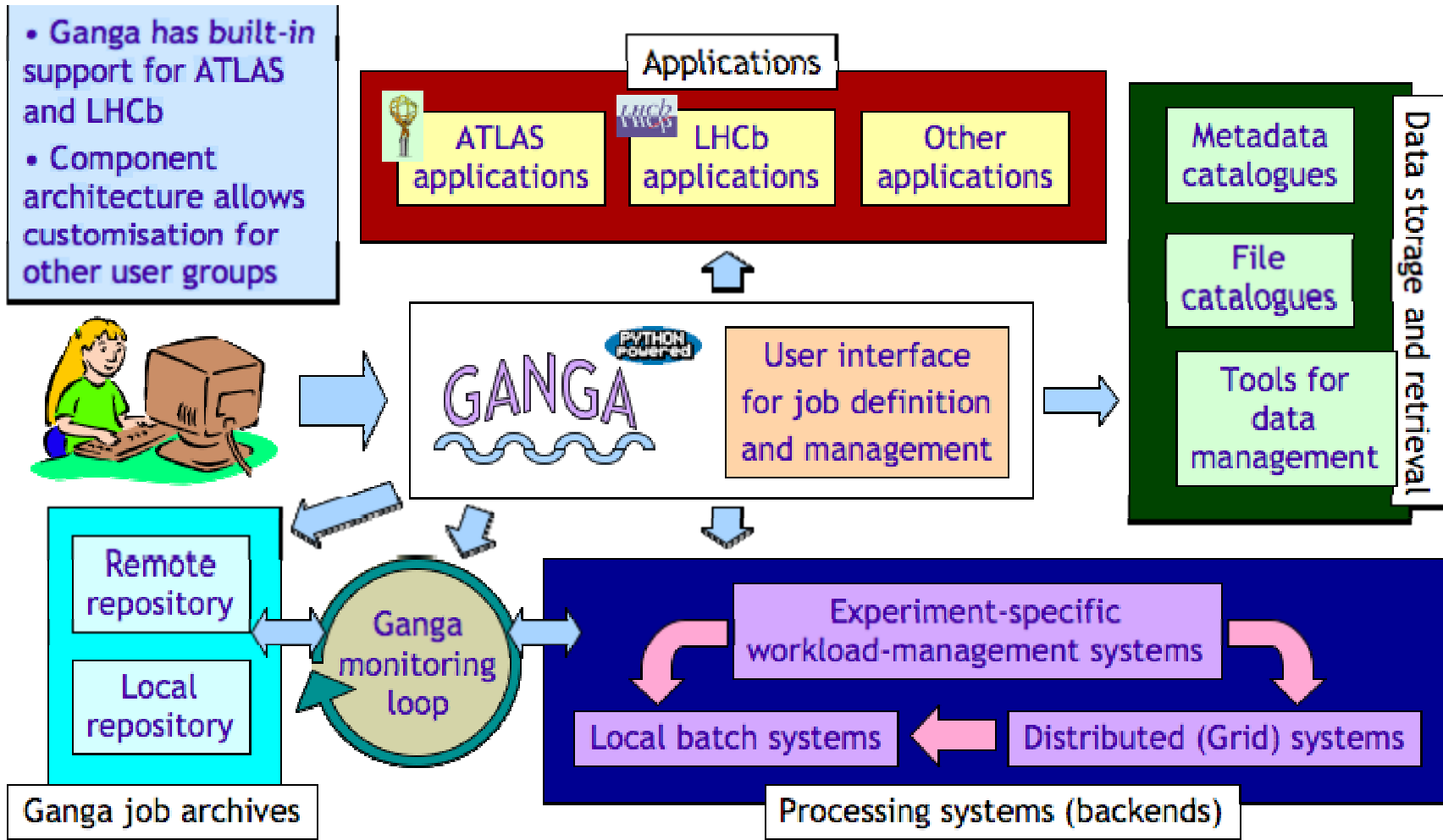
Split job into 3 subjobs

Limit analysis to 100 events per subjob

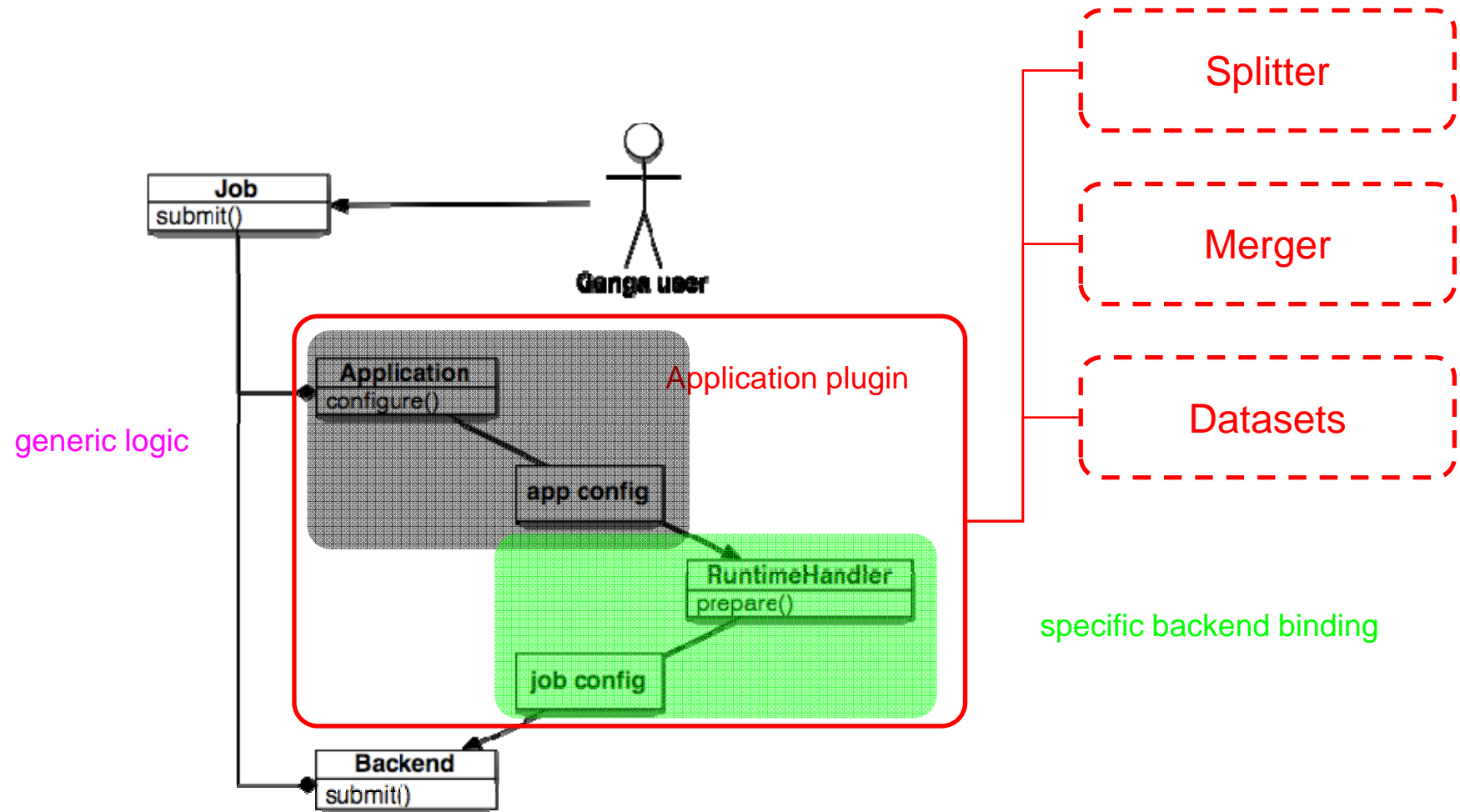
Job options

Submit to LCG

Force use of particular compute element



- Ganga has built-in support for ATLAS and LHCb
- Component architecture allows customisation for other user groups



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



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 using the commands:

```
Bash> ganga --config-path=GangaTutorial/Tutorial.ini \  
          --option=' [LCG]VirtualOrganisation=gilda'
```

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

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

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

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

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

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

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

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

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

In [8]: `jobs`

In [9]: `j.peek()`

In [10]: `cat $j.outputdir/stdout`

```
./myscript.sh ganga
```

Step 2: your first Ganga job on the Grid

```
In [11]:j = j.copy()
```

```
In [12]:j.backend = LCG()
```

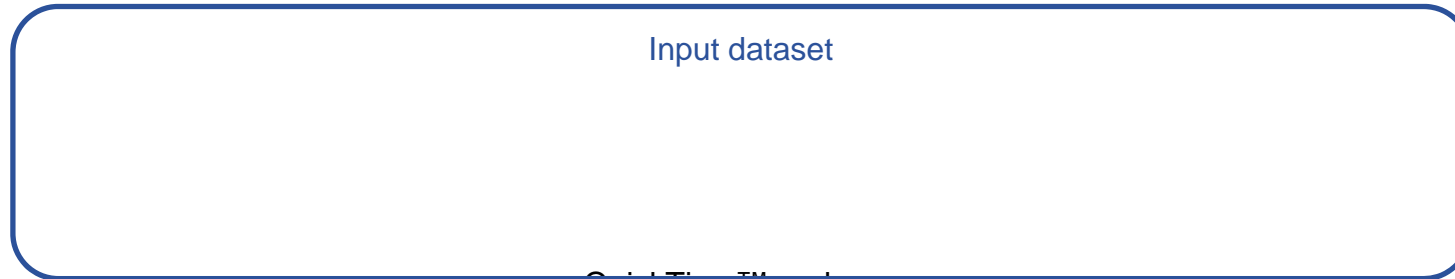
```
In [13]:j.application.args = ['grid']
```

```
In [14]:j.submit()
```

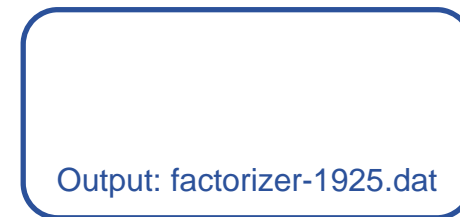
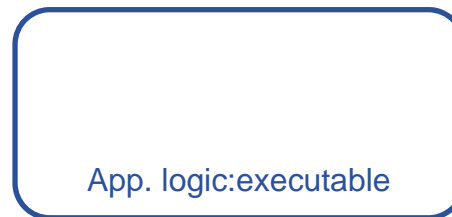
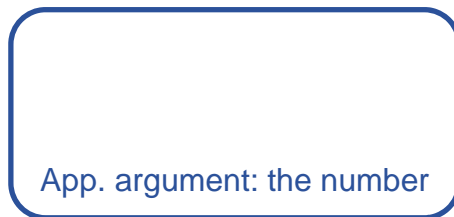
```
In [15]:j
```

```
In [16]:cat $j.backend.loginfo(verbosity=1)
```

```
In [17]:jobs
```



QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.



- **In this application, we will try out the following Ganga blocks**
 - Application
 - Dataset
 - Splitter

Step 3: run the PrimeFactorizer application

```
In [1]:j = Job()
In [2]:j.application = PrimeFactorizer()
In [3]:j.application.number = 113466789002166778767
In [4]:j.application

In [5]:j.inputdata = PrimeTableDataset()
In [6]:j.inputdata.table_id_upper = 15
In [7]:j.inputdata

In [8]:j.splitter = PrimeFactorizerSplitter()
In [9]:j.splitter.numsubjobs = 5
In [10]:j.splitter

In [11]:j.submit()
In [12]:len(j.subjobs)
In [13]:j.subjobs[0]
```

You knew already how to factor out
prime numbers from any integers
using the Grid



Enabling Grids for E-scienceE

Part III: More about Ganga

www.eu-egee.org



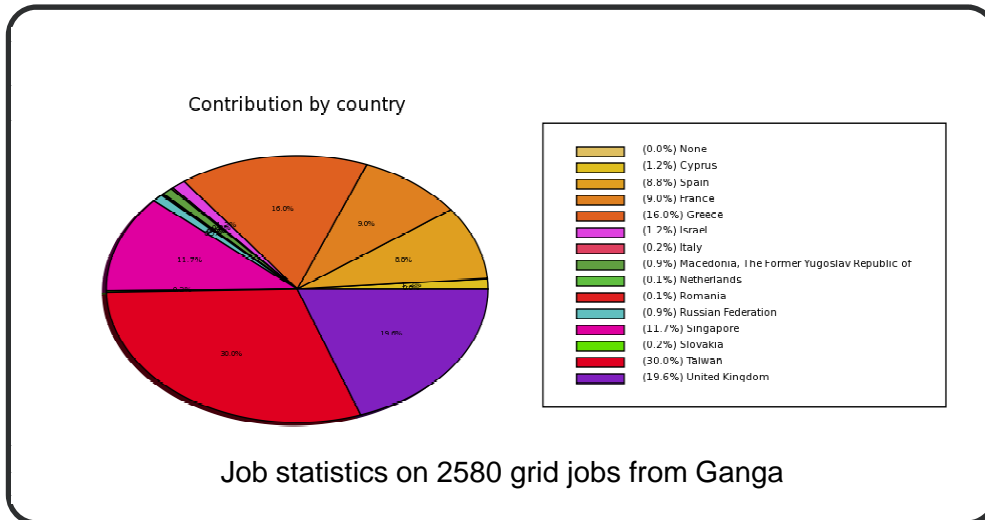
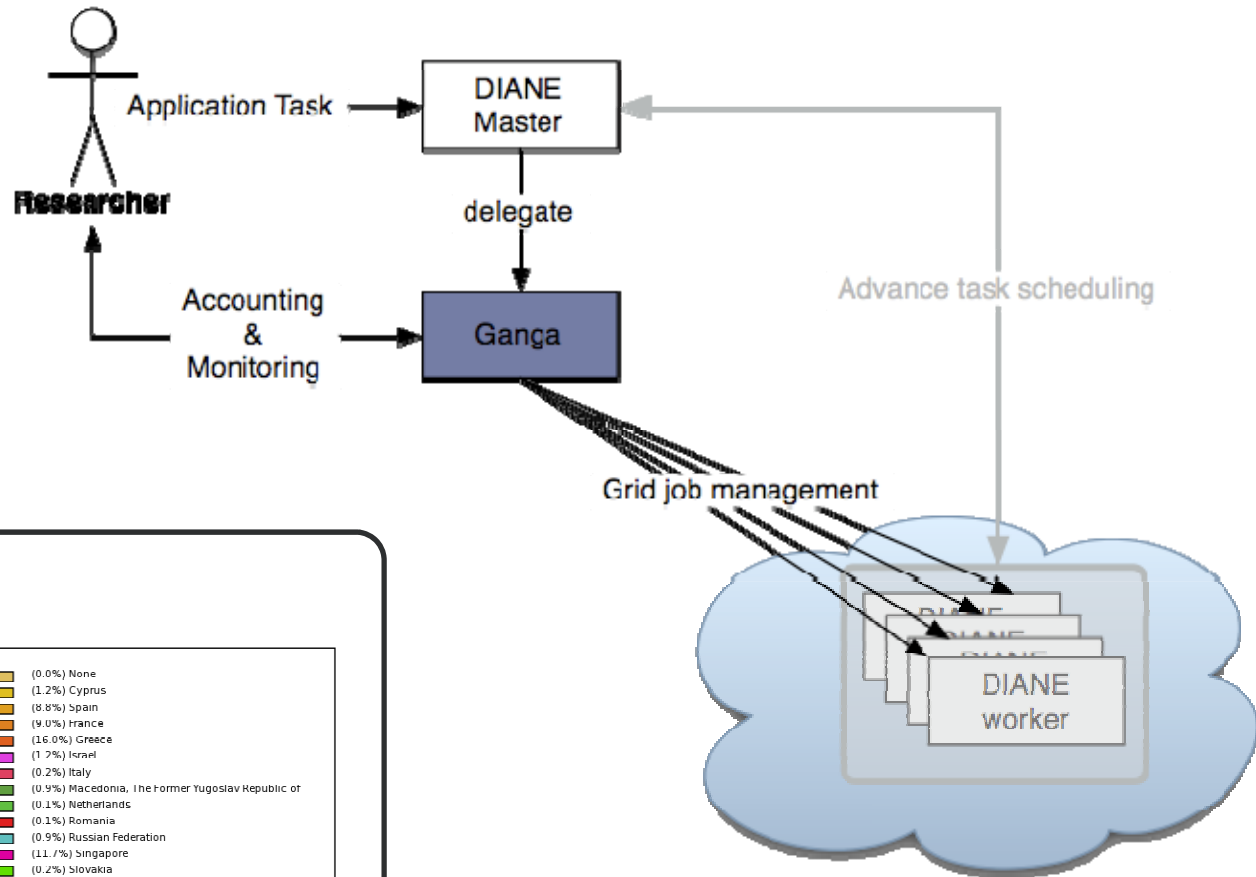
BBC NEWS

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



- Main Users



- Other activities

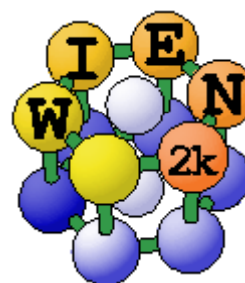
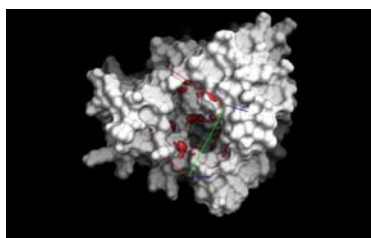
Geant 4



HARP



Garfield



med
austron



Academia Sinica
Genomics Research Center

JOB SUMMARY

jobsummary

any user

any site

any site

any ce

ganga

any dataset

any application

any rb

analysis

any grid

unk pend run term

done canc abort g-unk

succ fail a-unk

done:success

2007-01-30 15:48:27 to

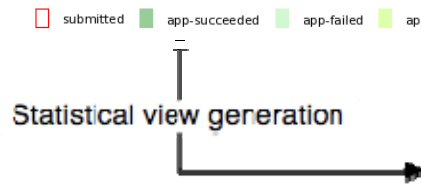
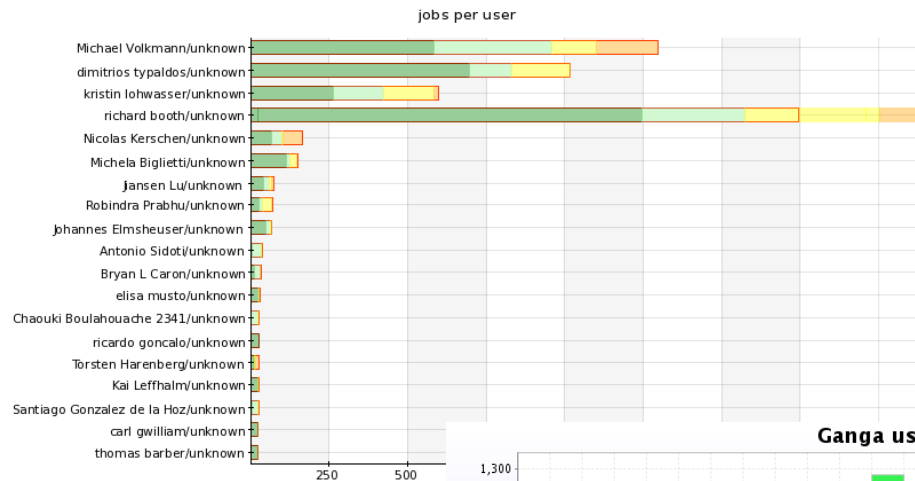
2007-02-21 15:48:27

sort by user

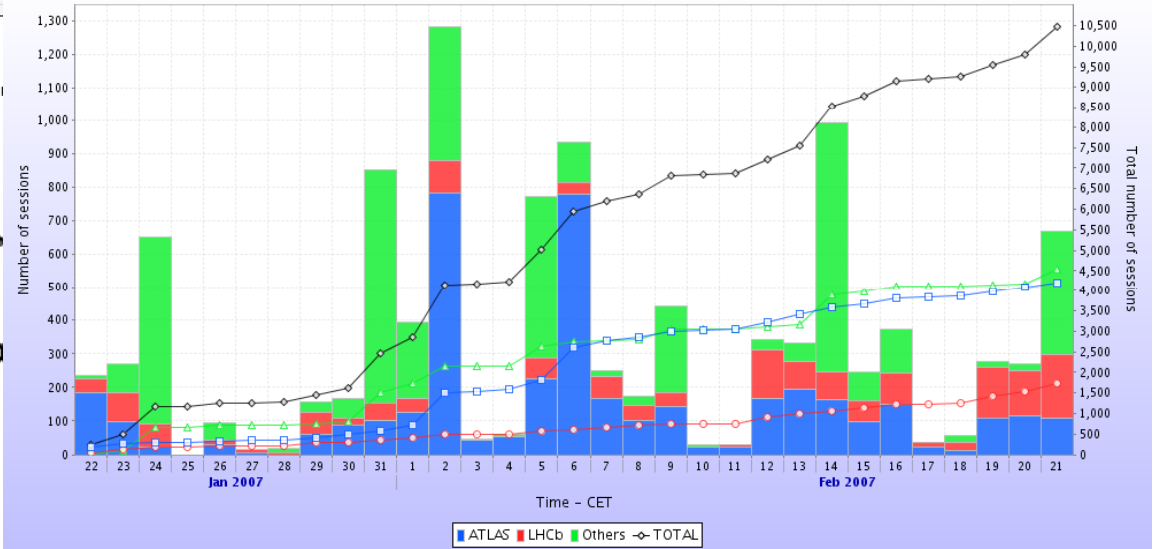
bars in the plot

submit

submission tool



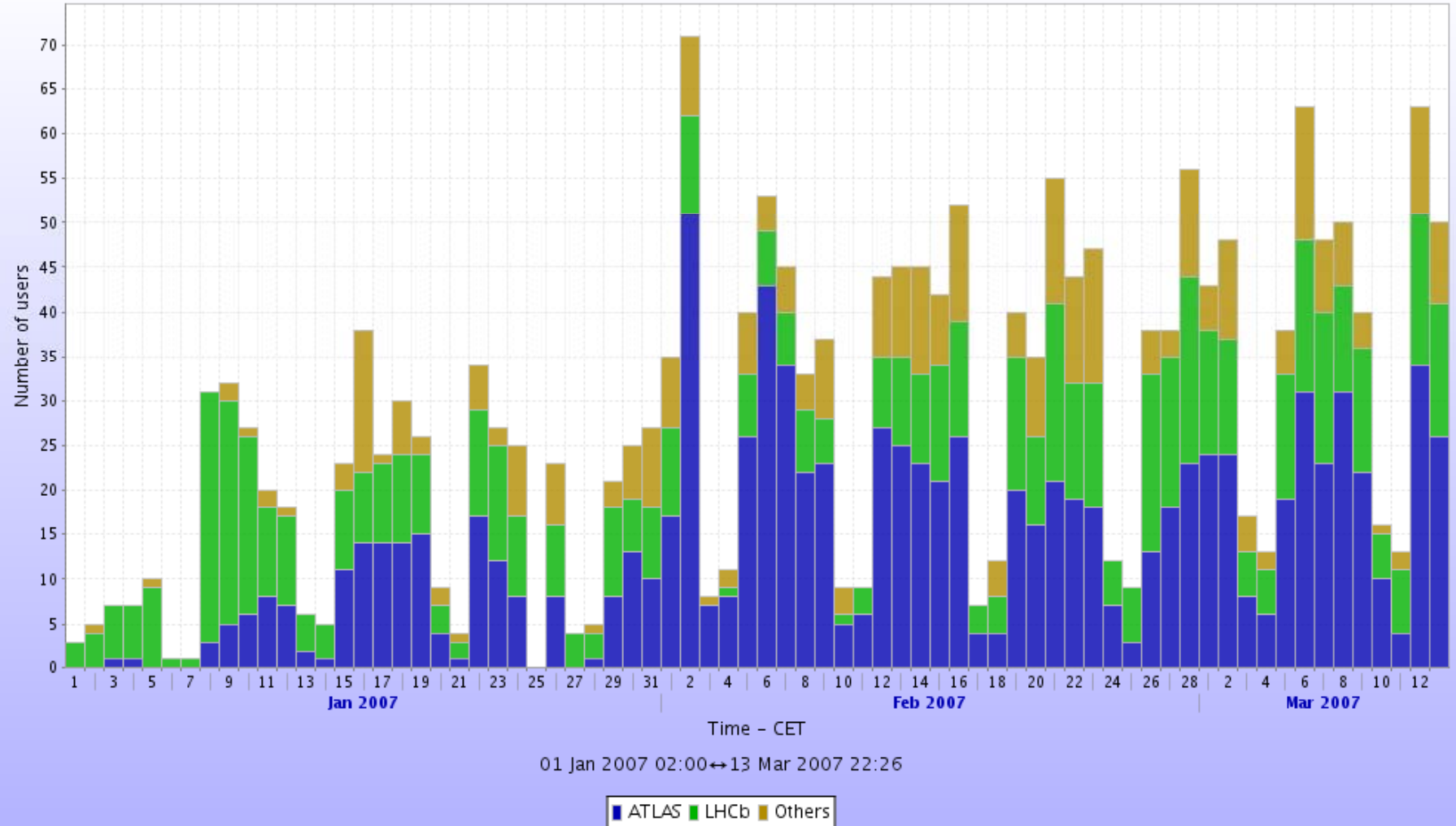
Ganga usage in different experiments



More than 300 GANGA Users

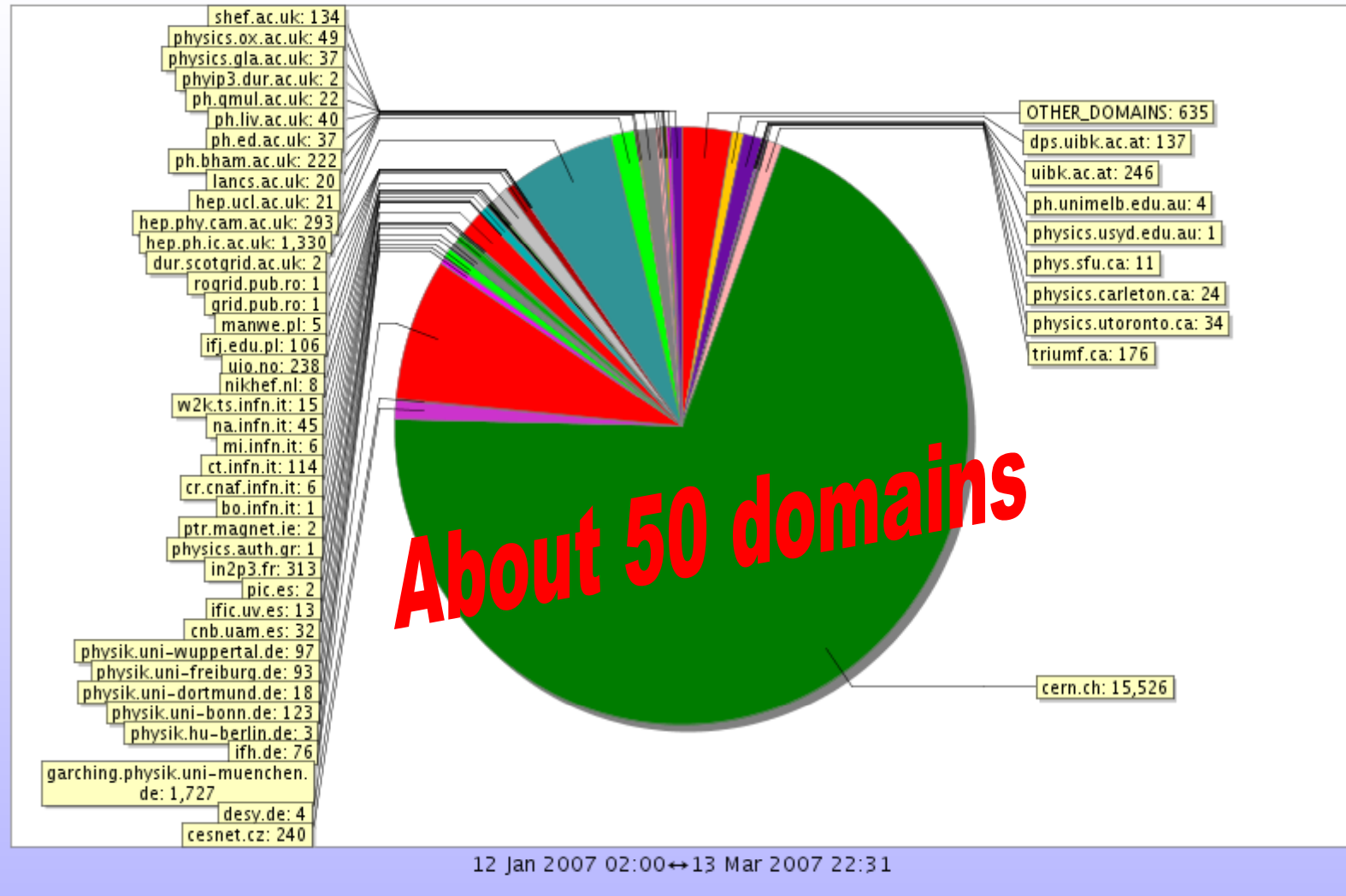
Unique Users

Number of unique users: 317



Where are those Ganga users?

Total number of sessions per domain



- **Ganga Home:** <http://cern.ch/ganga>
- **Official Ganga User's Guide:**
<http://ganga.web.cern.ch/ganga/user/html/GangaIntroduction/>
- **Tutorial for ATLAS data analysis using Ganga:**
<https://twiki.cern.ch/twiki/bin/view/Atlas/DistributedAnalysisUsingGanga>
- **Looking for helps:**
 - ATLAS user support: hn-atlas-GANGAUserDeveloper@cern.ch
 - direct support from developers: project-ganga-developers@cern.ch

- **Your applications developed in testbed environments can be smoothly migrated to production environments**
- **Your jobs are managed in a systematic way**
- **Your grid jobs benefit from a hidden job wrapper instrumented for**
 - advanced input/output control
 - runtime progress monitoring
- **New technologies will be transparent for you without changing your way of running applications**