Distributed Data Analysis with GANGA (Tutorial)

Alexander Zaytsev
Budker Institute of Nuclear Physics
(BudkerINP), Novosibirsk

On the basis of GANGA EGEE Tutorial 4.3
http://twiki.cern.ch/twiki/bin/view/Atlas/GangaTutorial43

RDIG-ATLAS Meeting @ IHEP (20 September 2007)

Overview

- Logging to the LCG UI, install usercert and tuning GRID environment
- Short introduction to Ganga
- Introduction to Ganga CLIP and GUI
- Submitting the simple local jobs
- Setup Athena environment and tune Ganga to submit Athena jobs
- Submitting the test jobs
- Cleaning GRID environment

Setting up GRID Environment

- Log to the ui0001.m45.ihep.su under a test account or lxplus.cern.ch under your CERN account
- Copy your usercert.pem and userkey.pem to the local ~/.globus directory (if necessary)
- Tune LCG tools (if necessary), e.g.:

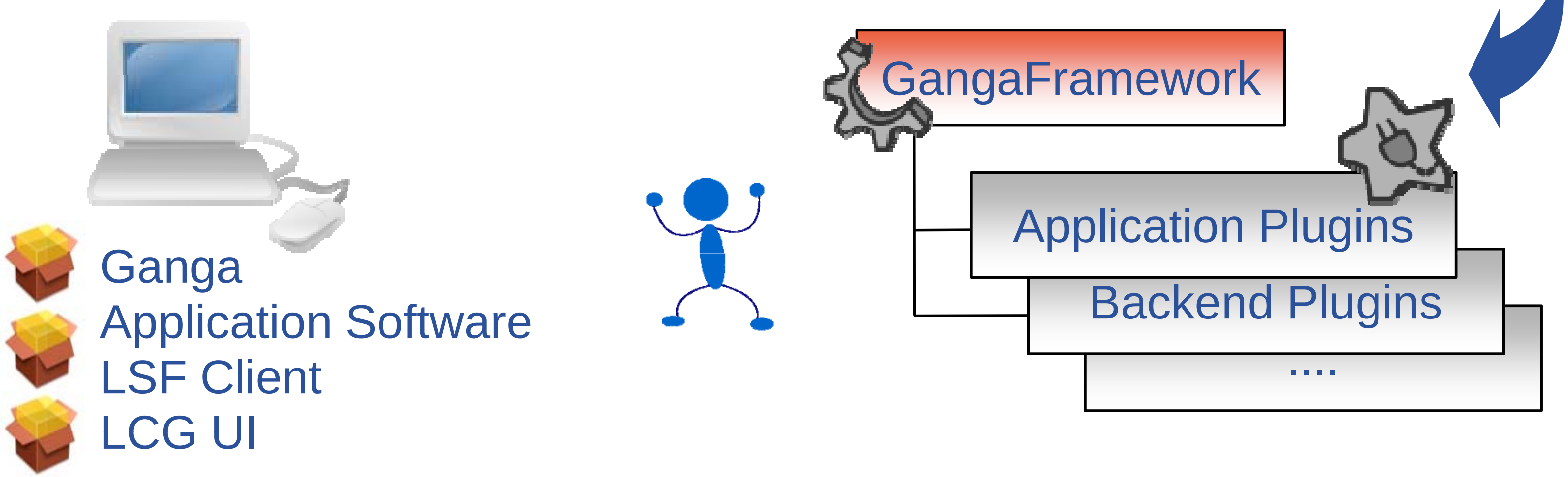
```
bash> which grid-proxy-init
bash> /afs/cern.ch/project/gd/LCG-share/sl3/etc/profile.d/grid_env.sh
bash> export LFC_HOST='prod-lfc-atlas-local.cern.ch‘
bash> export LCG_CATALOG_TYPE=lfc
```

- Create a GRID proxy:

```
bash> grid-proxy-init
bash> grid-proxy-info
```

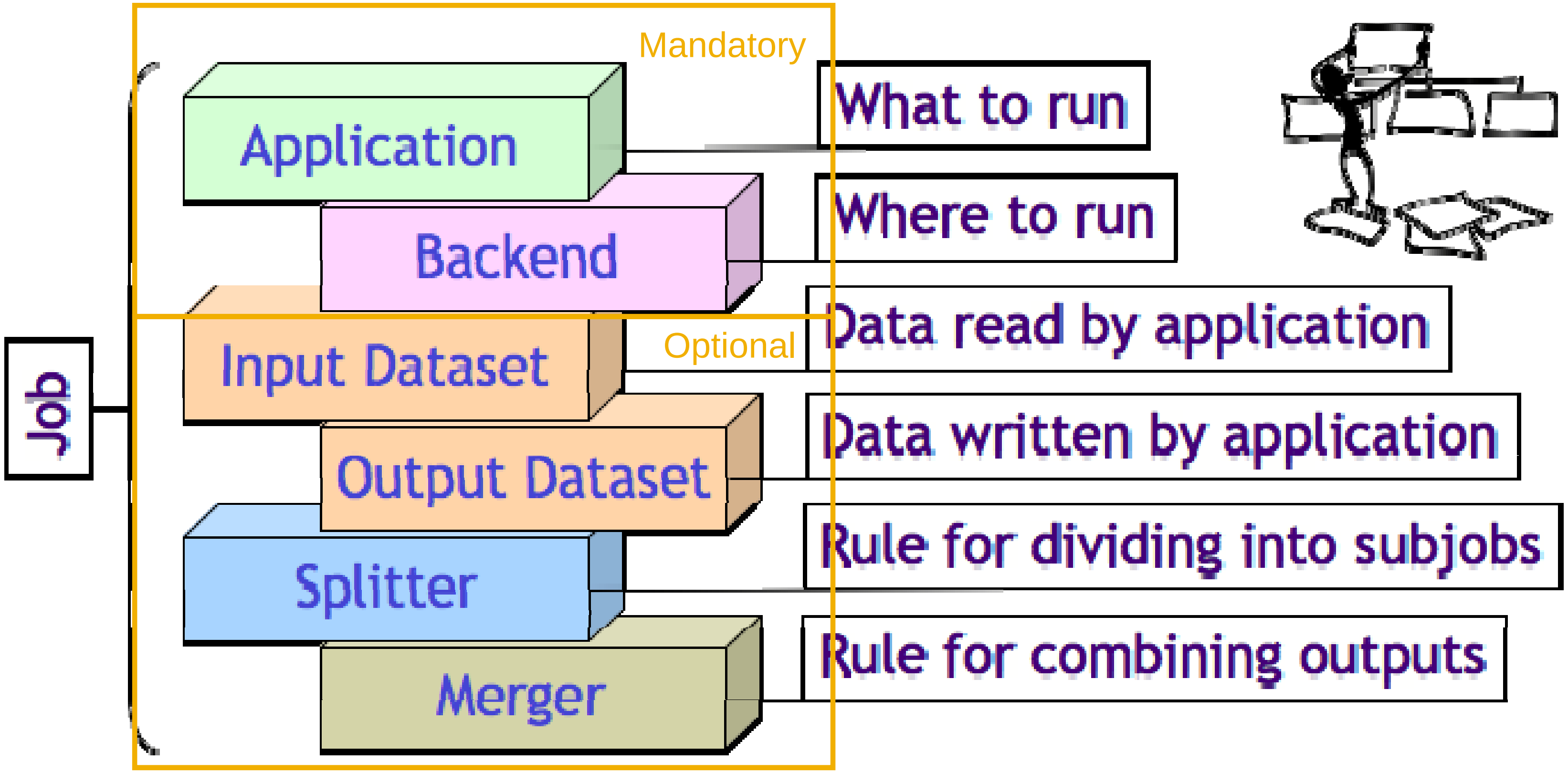
Ganga

- **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/4.2.14*
    - it is an **add-on** to installed software
  - comes with a set of plugins for **Atlas** and **LHCb**
    - **open** - other applications and backend may be easily added
      - *even by users*



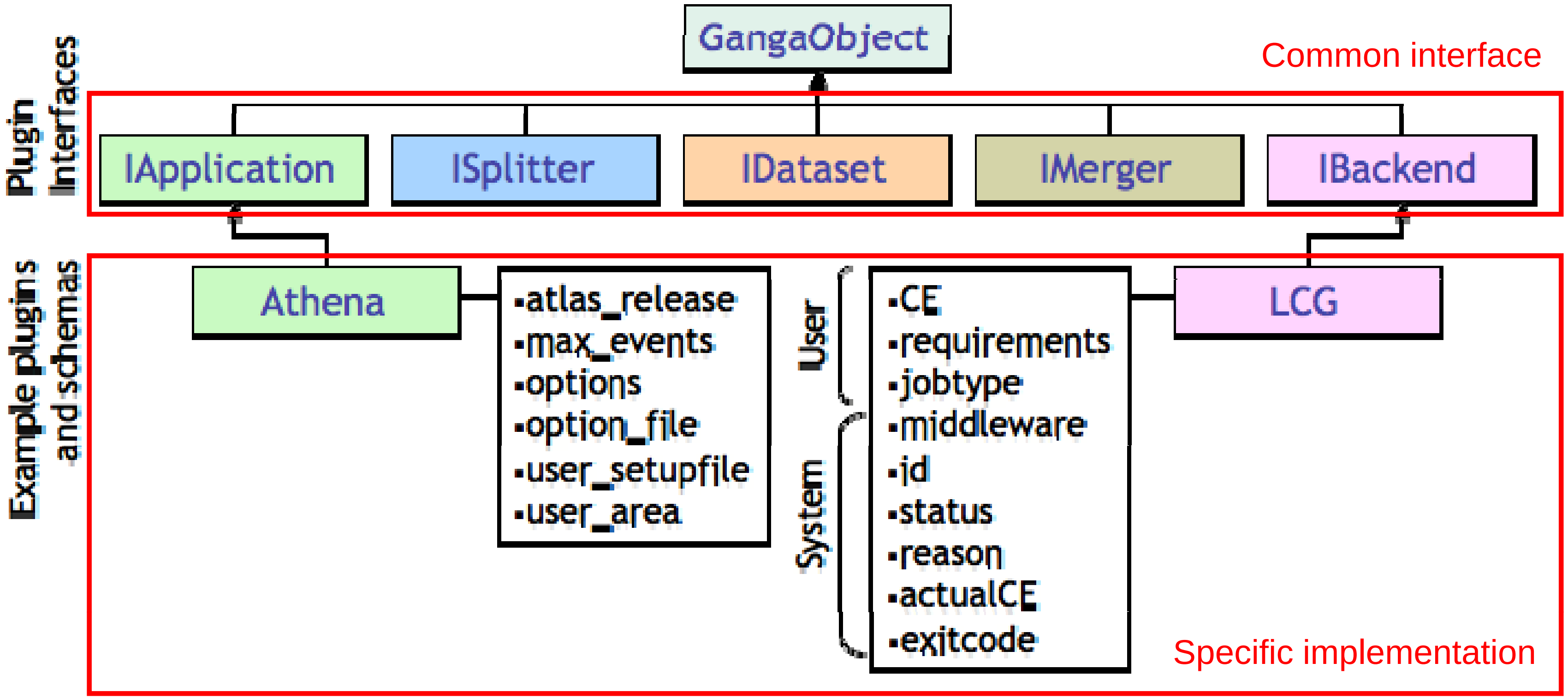
Ganga Job

Where the Ganga journey starts …



Plug-in based design

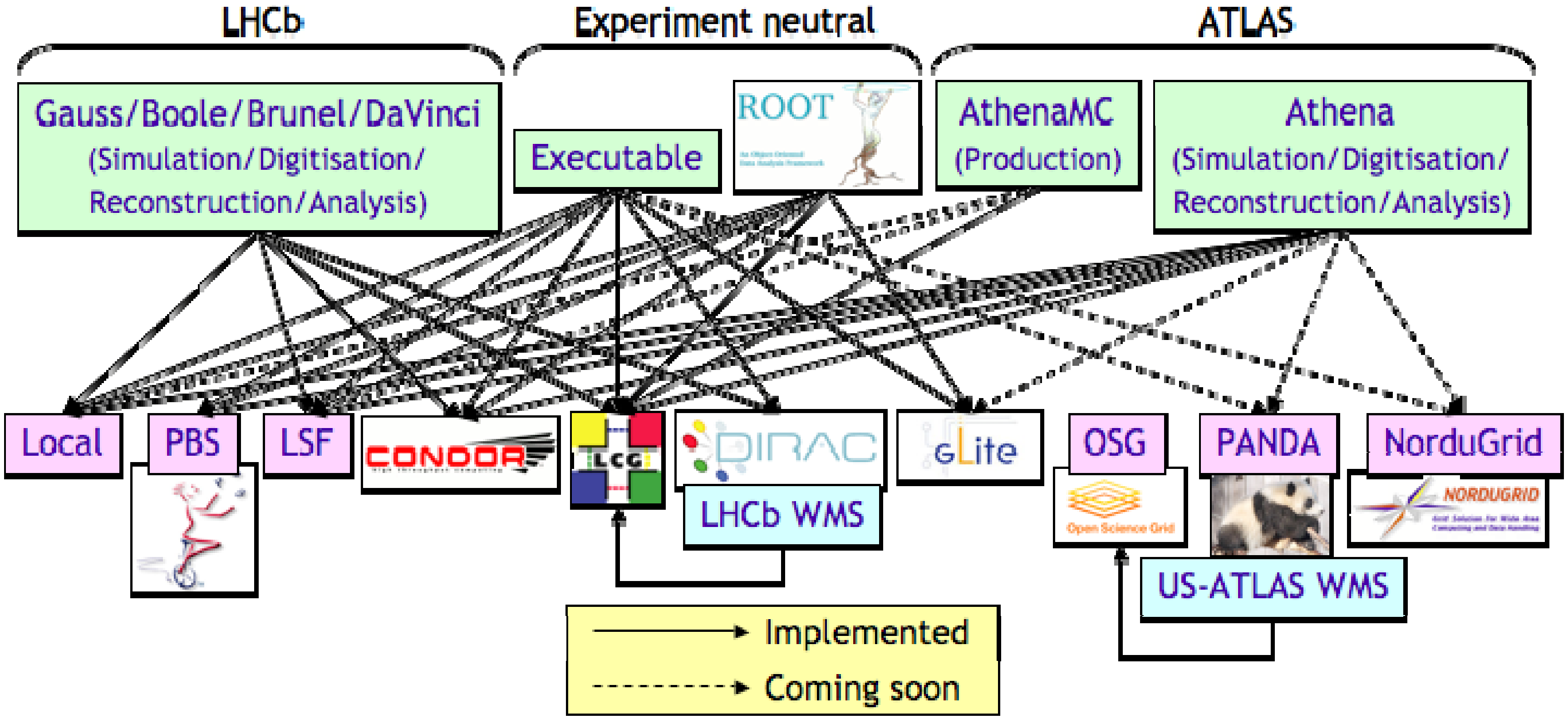
- Ease user's experience in switching between different technologies
- Concentrate developer's effort in specific domain



- atlas_release
- max_events
- options
- option_file
- user_setupfile
- user_area

- CE
- requirements
- jobtype
- middleware
- id
- status
- reason
- actualCE
- exitcode

Applications & backends



Basic Ganga Installation

- From AFS at CERN (lxplus):
  ```
  bash> source /afs/cern.ch/sw/ganga/install/etc/setup-atlas.sh
  ```
  or explicitely
  ```
  bash> source /afs/cern.ch/sw/ganga/install/etc/setup-atlas.sh 4.4.1
  ```
- From AFS at ui0001.m45:
  ```
  bash> export GANGA_CONFIG_PATH=GangaAtlas/Atlas.ini
  bash> export PATH=/afs/cern.ch/sw/ganga/install/4.4.1/bin:$PATH
  ```
  From /tmp test installation at ui0001.m45:
  ```
  bash> export GANGA_CONFIG_PATH=GangaAtlas/Atlas.ini
  bash> export PATH=/tmp/atlasprd/opt/ganga/install/4.4.1/bin:$PATH
  ```
- Try to start Ganga CLIP:
  ```
  bash> ganga
  ```

Download, Install, First launch

Download & Install

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

download installer

```
python ganga-install \
       --prefix=~/opt/ganga \
       --extern=GangaAtlas,GangaGUI,GangaPlotter \
       4.4.1
```

installation prefix

Installation of external modules

Ganga version

First Launch

```
export PATH $HOME/opt/ganga/install/4.4.1/bin:$PATH
```

start Ganga with inline configurations

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

```
*** Welcome to Ganga ***
Version: Ganga-4-4-1
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

Explore basic features of the CLIP

- `Job().submit()` submit and run a test job on local machine
- `Job(backend=LCG()).submit()` submit and run a 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

Configurations

Syntax

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

Python ConfigParser standard

How to set configurations

```
Hardcoded configurations
```

release config

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

site config

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

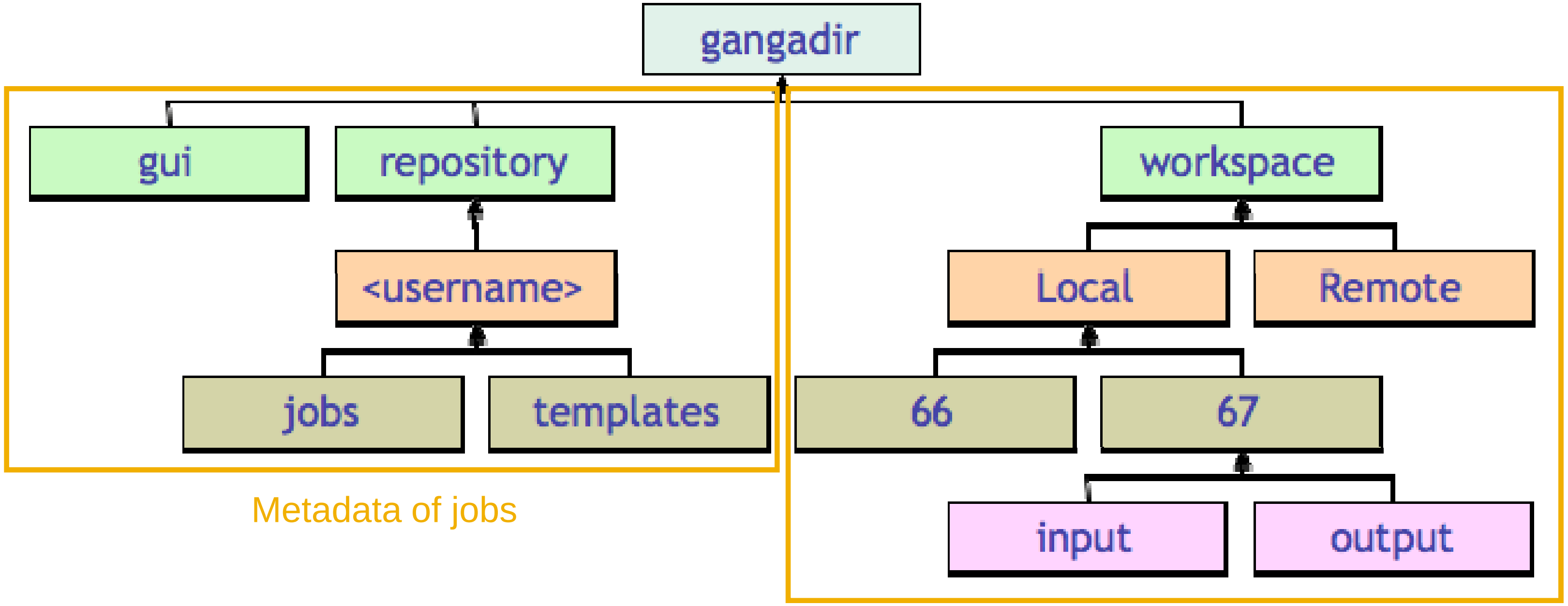
user config

Override sequence

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

The ‘gangadir’

- 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`



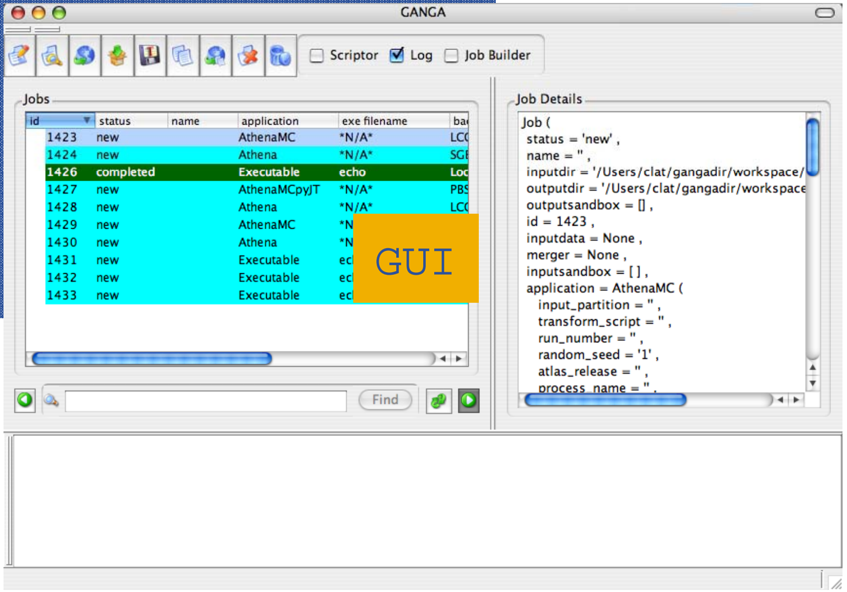
Metadata of jobs

Data of jobs

User interfaces

```
*** 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                            Executable
compute.hpc.unimelb.edu.au:2119/jobmanage
```

CLIP



GANGA

☐ Scriptor ☑ Log ☐ Job Builder

Jobs

id	status	name	application	exe filename	ba
1423	new		AthenaMC	*N/A*	LC
1424	new		Athena	*N/A*	SG
1426	completed		Executable	echo	Lo
1427	new		AthenaMCpyJT	*N/A*	PBS
1428	new		Athena	*N/A*	LC
1429	new		AthenaMC	*N	
1430	new		Athena	*N	
1431	new		Executable	ec	
1432	new		Executable	ec	
1433	new		Executable	ec	

Job Details

```
Job (
 status = 'new' ,
 name = '' ,
 inputdir = '/Users/clat/gangadir/workspace/
 outputdir = '/Users/clat/gangadir/workspace
 outputsandbox = [] ,
 id = 1423 ,
 inputdata = None ,
 merger = None ,
 inputsandbox = [ ] ,
 application = AthenaMC (
   input_partition = '' ,
   transform_script = '' ,
   run_number = '' ,
   random_seed = '1' ,
   atlas_release = '' ,
   process_name = '' ,
```

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

Some handy functions

- <tab> completion
- <page up/down> for cmd history
- system command integration

```
In[1]: j = jobs[1]

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

- Job template

```
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()
```

- `In[1]: plugins()`
  - `plugins('backends')`
- `In[2]: help()`
- etc.

Step 1: Your first Ganga job - an arbitrary shell script

```
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(‘myscript.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
```

More samples on
http://twiki.cern.ch/twiki/bin/view/ArdaGrid/EGEETutorialPackage

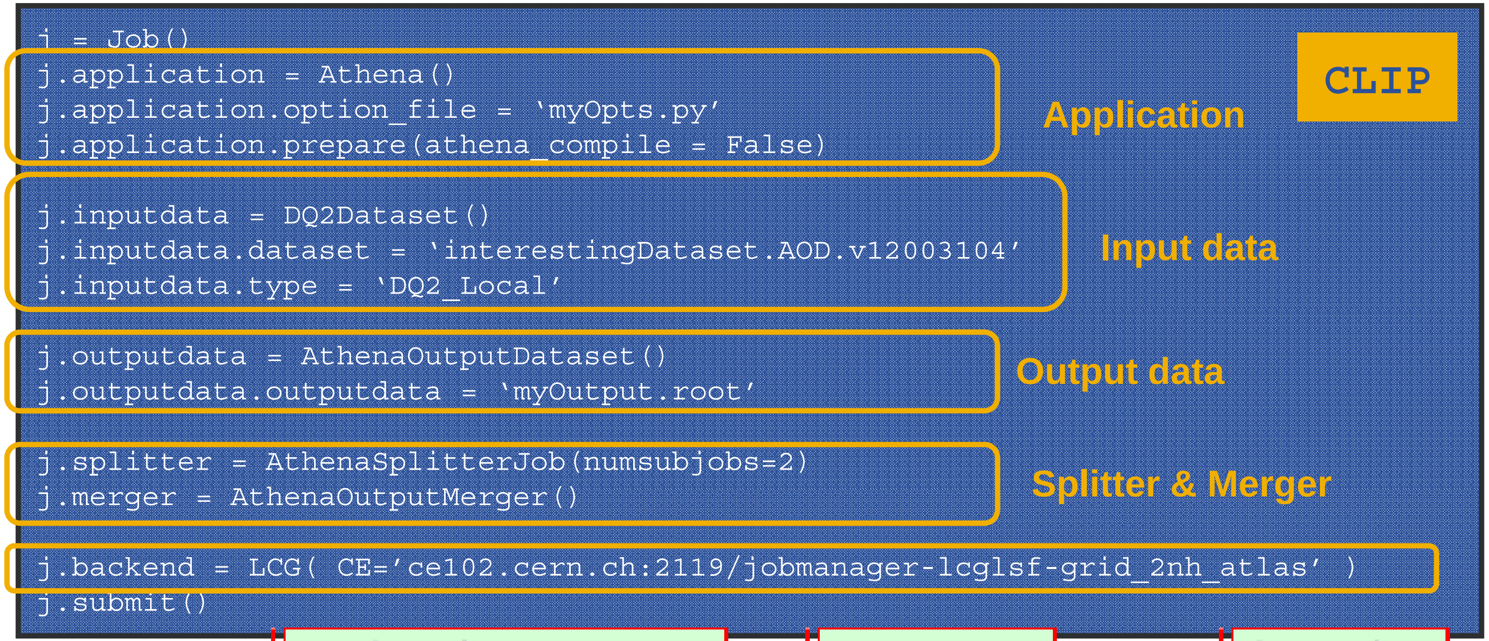
Tuning Ganga to Submit ATLAS Jobs

- Edit $HOME/.gangarc as follows:
  - In the section labelled [Configuration] add the line:
    RUNTIME_PATH = GangaAtlas
  - In the section labelled [LCG] add the line:
    VirtualOrganisation = atlas
  - In the section labeled [Athena] add the lines:
    # local path to base paths of dist-kits (lxplus example)
    ATLAS_SOFTWARE = /afs/cern.ch/project/gd/apps/atlas/slc3/software
- Setup Athena environment (if necessary):
  http://twiki.cern.ch/twiki/bin/view/Atlas/WorkBookRunAthenaHelloWorld
- Tune up DQ2 environment:

```
bash> source /afs/usatlas.bnl.gov/Grid/Don-Quijote/dq2_user_client/setup.[c,z]sh.CERN
bash> dq2_ls -f users.JoeUser.ganga.105.20070321
bash> dq2_get -r users.JoeUser.ganga.105.20070321
```

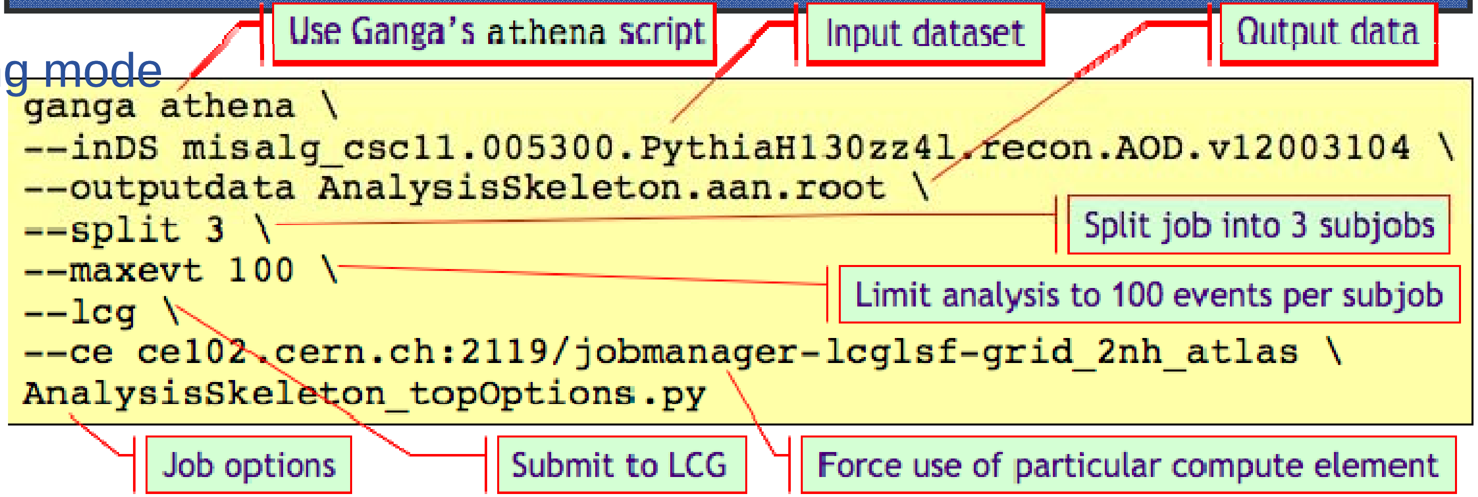
- Many job examples on:
  http://twiki.cern.ch/twiki/bin/view/Atlas/GangaTutorial43

Real Application: the ATLAS data analysis application



```
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()
```

Scripting mode

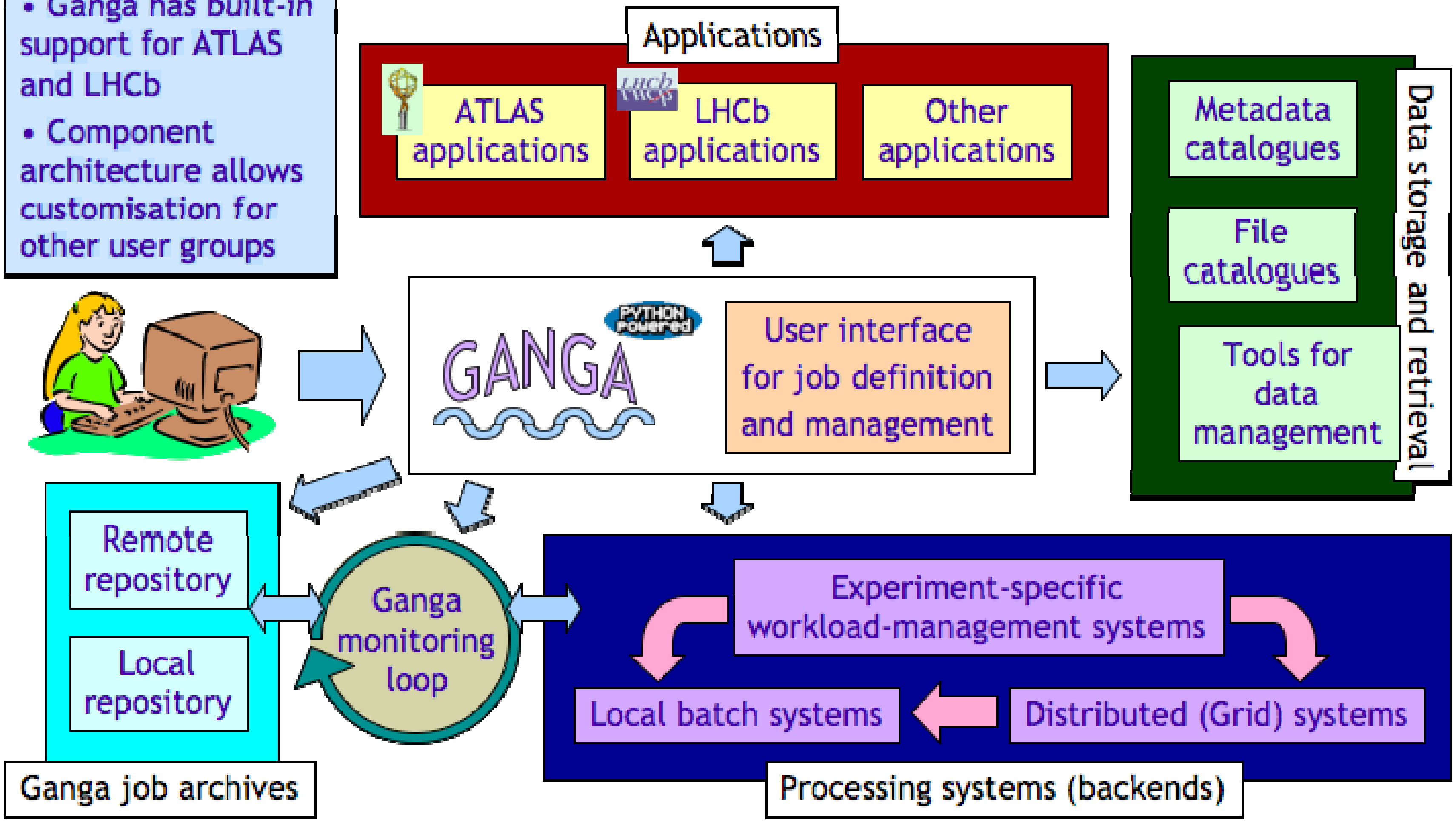


```
ganga athena \
--inDS misalg_csc11.005300.PythiaH130zz4l.recon.AOD.v12003104 \
--outputdata AnalysisSkeleton.aan.root \
--split 3 \
--maxevt 100 \
--lcg \
--ce ce102.cern.ch:2119/jobmanager-lcglsf-grid_2nh_atlas \
AnalysisSkeleton_topOptions.py
```

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

Behind the scene ...

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



Submitting ATLAS LCG Jobs

- Submitting Local Athena Analysis Job: http://twiki.cern.ch/twiki/bin/view/Atlas/GangaTutorial43#3_2_Local_Athena_Analysis_Job_us
- Submitting DQ2Dataset input and ATLASOutputDataset Job http://twiki.cern.ch/twiki/bin/view/Atlas/GangaTutorial43#20TWiki.htm#3_3_LCG_grid_job
- Monitoring job execution
- Retrieving output

Cleaning up GRID Environment

- Destroy your GRID proxy:

  ```
  bash> grid-proxy-destroy
  ```

- Remove your usercert.pem and userkey.pem out of local ~/.globus directory

The End