



Enabling Grids for E-scienceE

Ganga User Interface

EGEE Review

Jakub Moscicki

CERN / IT

www.eu-egee.org



Material for the demo

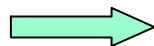
The word 'GANGA' is written in a purple, outlined font. Below it is a blue wavy line representing water.

EGEE Review

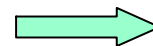
- Based on K. Harrison at the Computing in High-Energy and Nuclear Physics conference Mumbai, India, 13-17 February 2006
- New material from the GANGA team
- New material from Hurng-Chun Lee (ASGC)
- New material from Andrea Manara (ITU)
- Other material from IT/PSS/ED (ARDA and EIS teams)

- **Ganga is an easy-to-use front end for job definition and management**
 - Access to local and remote (Grid) resources through a uniform interface
 - The Grid is *one* of the environment for the scientists
 - If the Grid is not integrated in the everyday environment, its impact is reduced
 - Developed in the context of ATLAS and LHCb
 - built-in support for applications based on Gaudi/Athena framework
 - Potentially interesting for other applications (also non HEP)
 - Component architecture readily allows it

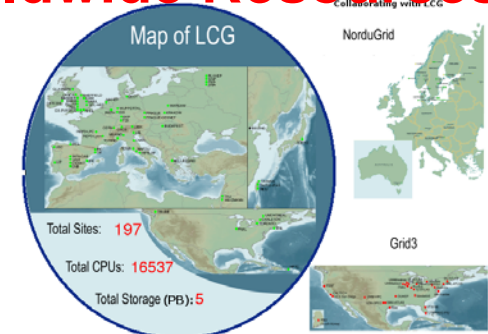
HEP Computing



Uniform User Interface



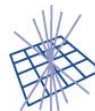
Worldwide Resources



- Ganga is an ATLAS/LHCb joint project



- Support for development work from UK (PPARC/GridPP) and EU (EGEE/NA4 HEP ARDA)



- Core team:

- U.Egede (Imperial), K.Harrison (Cambridge),
D.Liko (CERN), A.Maier (CERN), J.T.Moscicki (CERN),
A.Soroko (Oxford), CL.Tan (Birmingham)



UNIVERSITY OF
BIRMINGHAM



UNIVERSITY OF
CAMBRIDGE

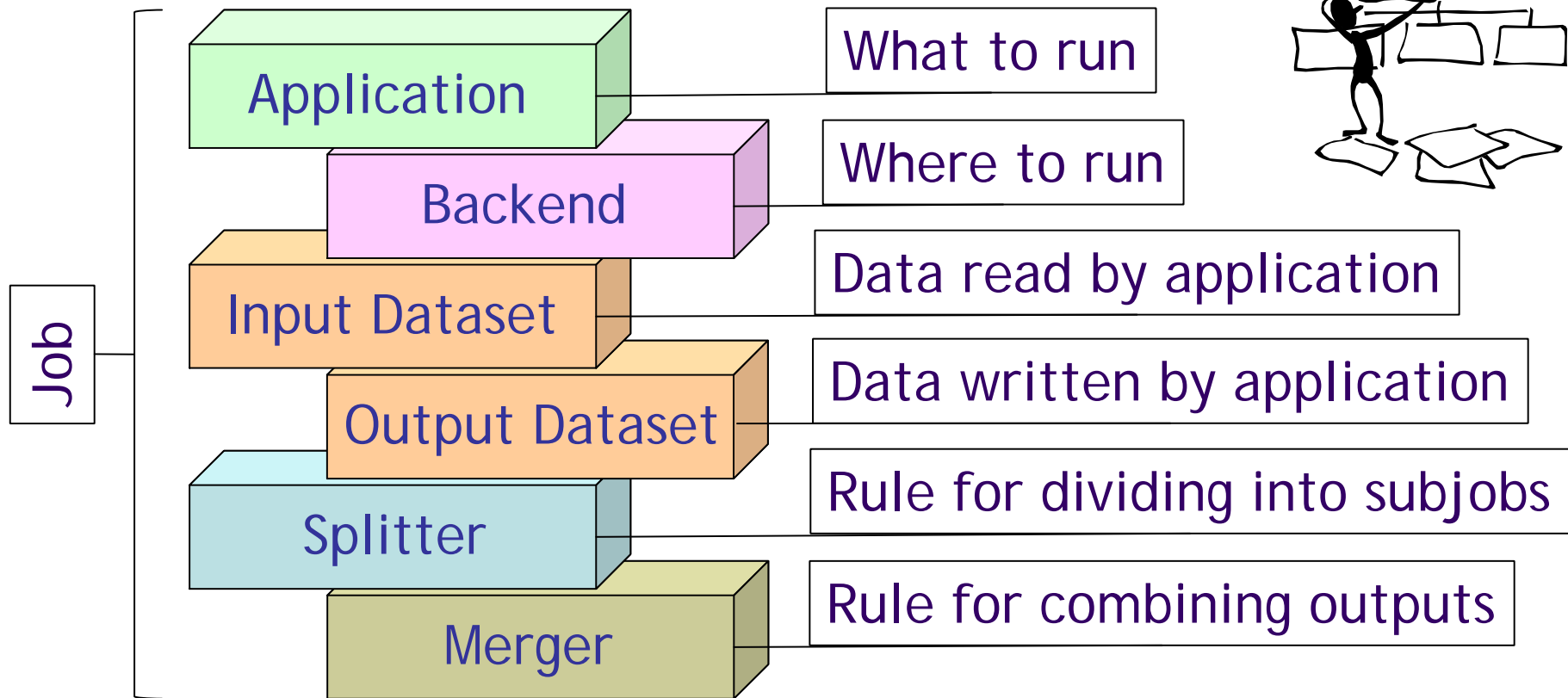


Imperial College
London

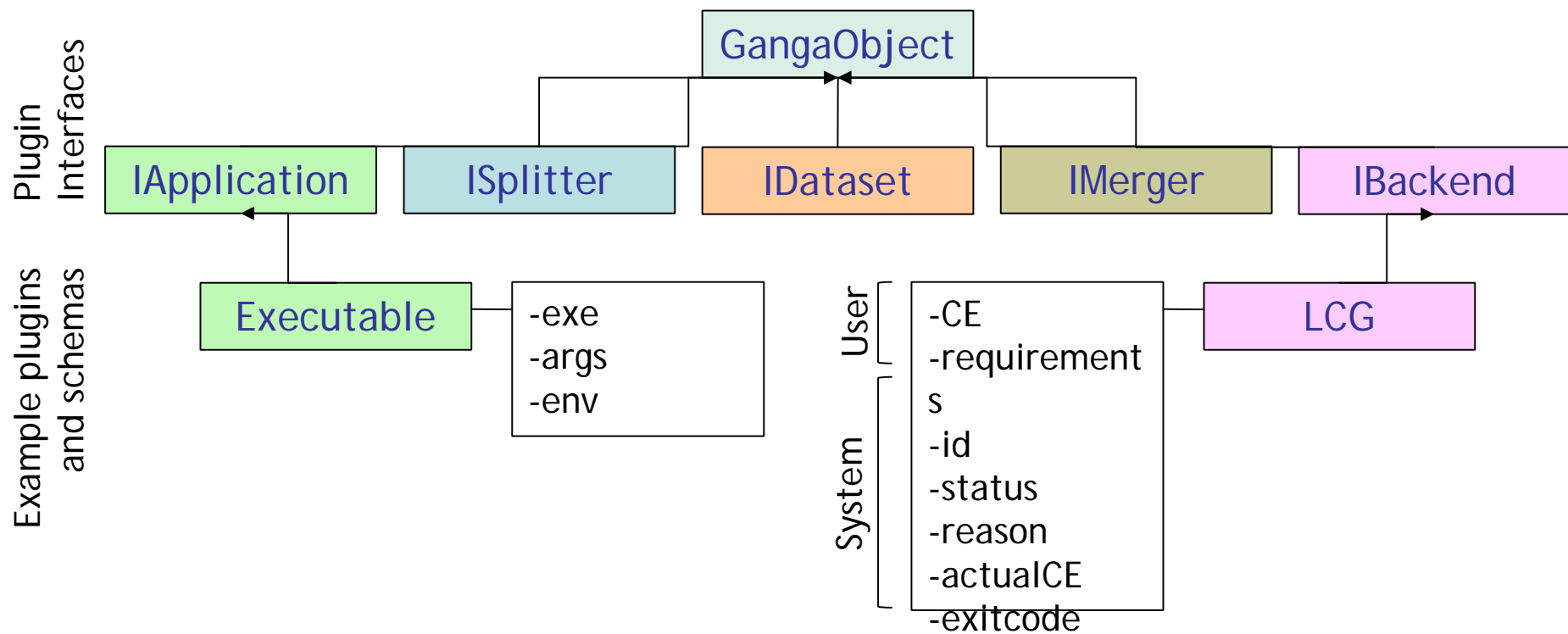


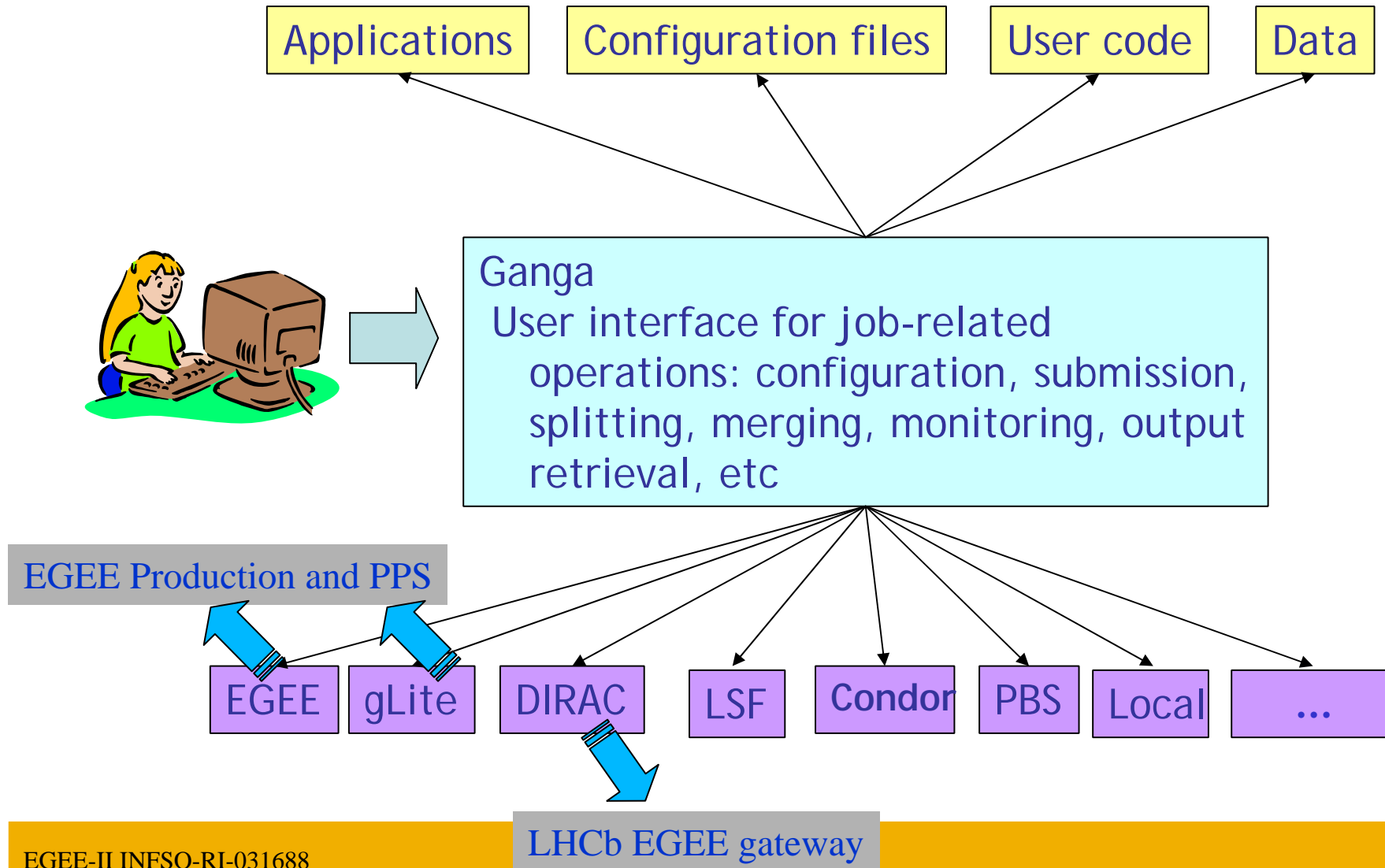
- Contributions from many others, from summer students to senior researchers from HEP experiments

- A job in Ganga is constructed from a set of building blocks, not all required for every job



- Ganga provides a framework for handling different types of Application, Backend, Dataset, Splitter and Merger, implemented as plugin classes
- Each plugin class has its own schema





- **Ganga supports different scenarios because different communities have different ways of working**



- **Command-Line Interface in Python (CLIP)**

- Interactive job definition and submission from enhanced Python shell (IPython)

- Possibility to organise jobs in logical folders
- Possibility to create job templates

- Python scripts

- Export Python code in scripts
- Automate repetitive tasks
- Examples → starting point for new users

- Graphical User Interface (GUI)

- Ease the tasks for newcomers
- Graphical access to information (job statuses etc..)

```

In [1]: status = os.system("bin/echo /bin/hostname -f - /bin/date")
         wget http://www.gnu.org - Fri Sep 8 11:19:31 BST 2005

In [2]: j = Job(application = Executable(), backend = LCG, name = "HelloFromBackend")

In [3]: j.application.exe = "/bin/echo"

In [4]: j.application.args = ["Hello from /bin/hostname -f - /bin/date"]

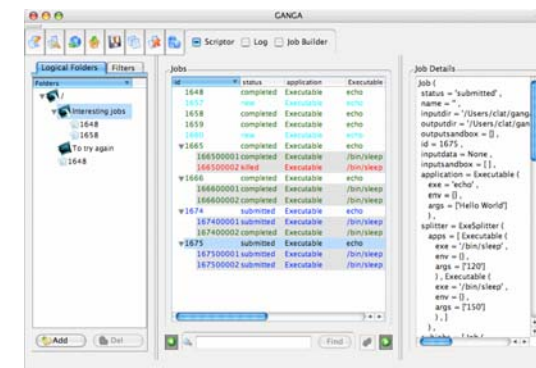
In [5]: status = j.submit()
Ganga.GPDev.Lib.Job : INFO submitting job 1

In [6]: Ganga.Lib.LCG : INFO Job 1 Ready at host.matrix.sara.nl:2119/jobmanager-gba-atlas - Fri Sep 9 12:20:13 2005
Ganga.Lib.LCG : INFO Job 1 Submitted at host.matrix.sara.nl:2119/jobmanager-gba-atlas - Fri Sep 9 12:20:19 2005
Ganga.Lib.LCG : INFO Job 1 Running at host.matrix.sara.nl:2119/jobmanager-gba-atlas - Fri Sep 9 12:22:08 2005
Ganga.Lib.LCG : INFO Job 1 Done (Success) at host.matrix.sara.nl:2119/jobmanager-gba-atlas - Fri Sep 9 12:24:28 2005

In [6]: print jobs
Statistics: 1 jobs
-----
ID      status      name
-----
# 1     completed   HelloFromBackend
         host.matrix.sara.nl:2119/jobmanager-gba-atlas

In [7]: print file(j.outputdir + "stdout").read()
Hello from host.matrix.sara.nl - Fri Sep 9 13:20:27 CEST 2005

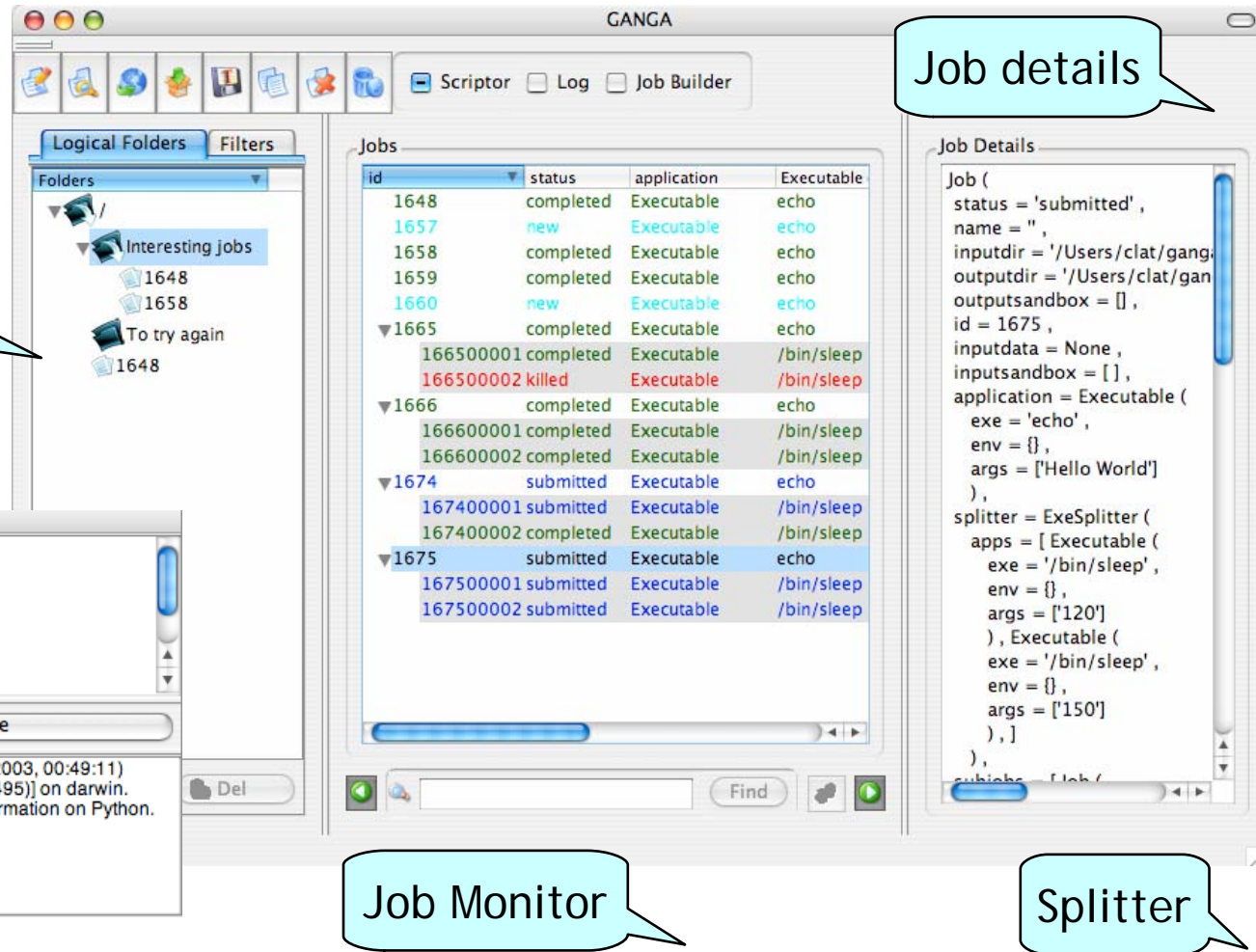
In [8]:
    
```



GUI based on dockable windows

Logical Folders

Scriptor



The screenshot shows the Ganga GUI interface with several dockable windows:

- Logical Folders:** A tree view showing folders like 'Interesting jobs' and 'To try again' with sub-items like '1648'.
- Scriptor:** A window for editing and running scripts. It shows a Python script:


```
import time
for x in range(10):
    j = Job()
    j.submit()
    time.sleep(2)
```

 Below the script is an 'Execute' button and a terminal window showing the shell prompt 'The PyCute shell running Python 2.3 (#1, Sep 13 2003, 00:49:11) [GCC 3.3 20030304 (Apple Computer, Inc. build 1495)] on darwin. Type "copyright", "credits" or "license" for more information on Python. >>>'
- Job Monitor:** A table displaying job status and details:

id	status	application	Executable
1648	completed	Executable	echo
1657	new	Executable	echo
1658	completed	Executable	echo
1659	completed	Executable	echo
1660	new	Executable	echo
1665	completed	Executable	echo
166500001	completed	Executable	/bin/sleep
166500002	killed	Executable	/bin/sleep
1666	completed	Executable	echo
166600001	completed	Executable	/bin/sleep
166600002	completed	Executable	/bin/sleep
1674	submitted	Executable	echo
167400001	submitted	Executable	/bin/sleep
167400002	completed	Executable	/bin/sleep
1675	submitted	Executable	echo
167500001	submitted	Executable	/bin/sleep
167500002	submitted	Executable	/bin/sleep
- Job details:** A window showing the configuration for a specific job:


```
Job (
  status = 'submitted',
  name = "",
  inputdir = '/Users/clat/ganga',
  outputdir = '/Users/clat/ganga',
  outputsandbox = [],
  id = 1675,
  inputdata = None,
  inputsandbox = [],
  application = Executable (
    exe = 'echo',
    env = {},
    args = ['Hello World']
  ),
  splitter = ExeSplitter (
    apps = [ Executable (
      exe = '/bin/sleep',
      env = {},
      args = ['120']
    ), Executable (
      exe = '/bin/sleep',
      env = {},
      args = ['150']
    ), ]
  ),
  subjobs = Job (
```

Job details

Job Monitor

Splitter

```

Ganga
In [1]: status = os.system( "/bin/echo `hostname -f` - `date`" )
pcff.hep.phy.cam.ac.uk - Sat Feb 11 09:17:39 GMT 2006

In [2]: j1 = Job( application = Executable(), backend = LCG(), name = "Hello" )

In [3]: j1.application.exe = "/bin/echo"

In [4]: j1.application.args = [ "Hello from `hostname -f` - `date`" ]

In [5]: status = j1.submit()
Ganga: INFO      submitting job 1

In [6]: Ganga: INFO      Job 1 Waiting at None - Sat Feb 11 09:18:14 2006
Ganga: INFO      Job 1 Ready at lcgce01.nic.ualberta.ca - Sat Feb 11 09:18:30 2006
Ganga: INFO      Job 1 Scheduled at lcgce01.nic.ualberta.ca - Sat Feb 11 09:19:01 2006
Ganga: INFO      Job 1 Running at lcgce01.nic.ualberta.ca - Sat Feb 11 09:22:26 2006
Ganga: INFO      Job 1 Done (Success) at lcgce01.nic.ualberta.ca - Sat Feb 11 09:29:49 2006

In [6]: !cat $j1.outputdir/stdout
Hello from thuner069.ualberta - Sat Feb 11 02:20:05 MST 2006

In [7]: 

```

Time on local machine (Cambridge)

Create job

Specify executable

Specify arguments

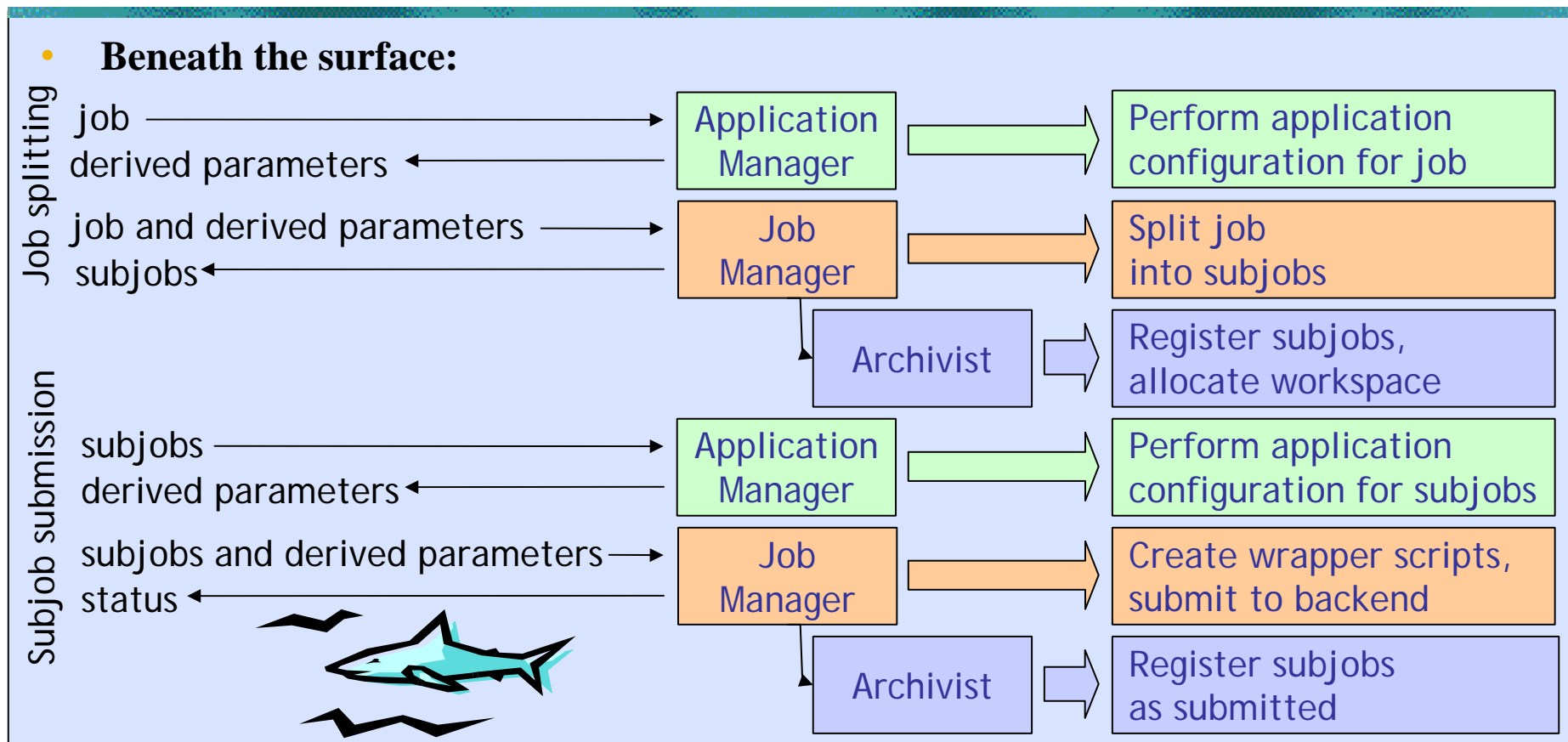
Submit job to LCG

Status on LCG monitored

Python allows combining of shell and Python syntax

Time on remote machine (Alberta)

- **User command: `job.submit()`**
- **Outcome: job submitted, split into subjobs, command status returned**



- **User command: `job.remove()`**
- **Outcome: job/subjobs killed if submitted/running, records deleted from repository, workspace deallocated, command status returned**

- **Beneath the surface:**

job
status

Job
Manager

Query status of job/subjob,
kill if submitted/running

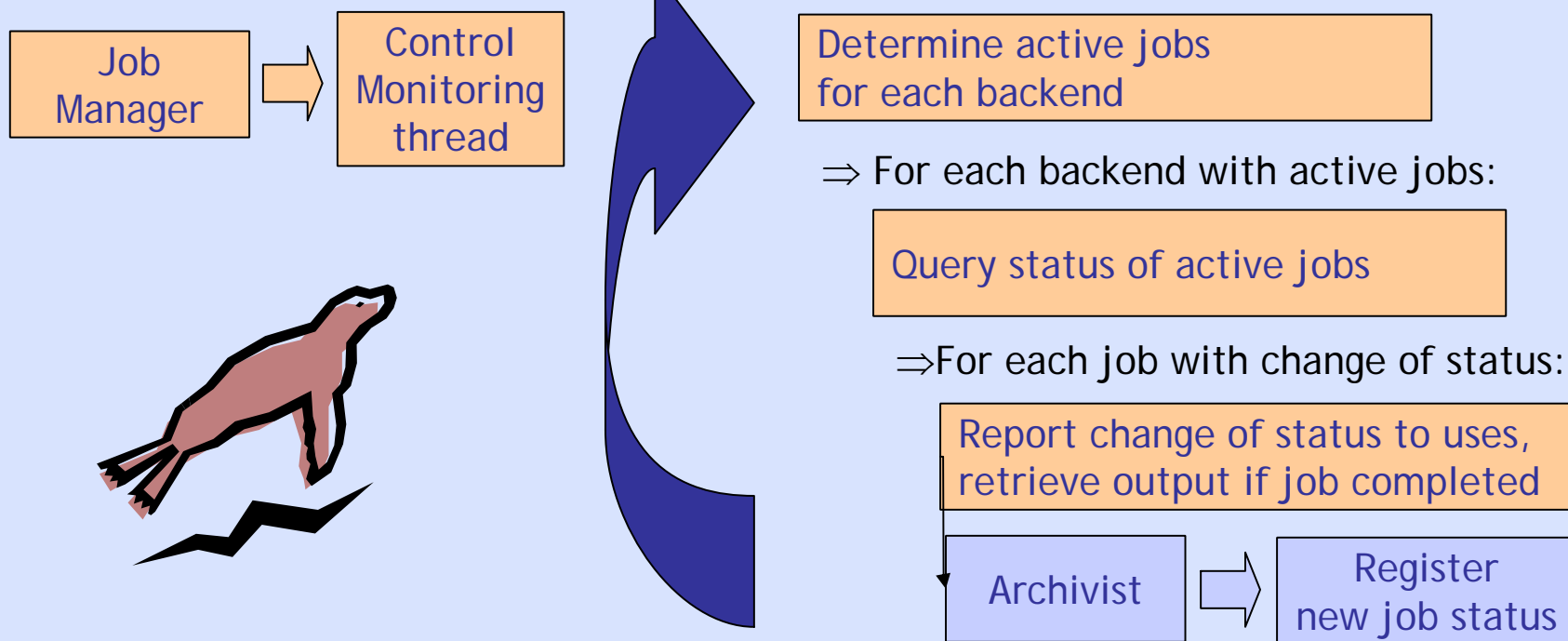
Archivist

Delete job/subjobs from repository,
deallocate workspace



- User action: none
- Outcome: changes in job status reported/updated periodically, output retrieved automatically when job completes

• **Beneath the surface:**

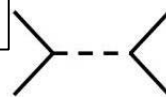


New session started

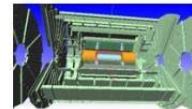
```
In [1]: templates
Out[1]: Statistics: 7 templates
```

#	id	status	name	application	backend
#	1	template	Hello	Executable	LCG
#	2	template	EvGen	Athena	LCG
#	3	template	Simul	Athena	LCG
#	4	template	Digi	Athena	LCG
#	5	template	Reco	Athena	LCG
#	6	template	ESDtoAOD	Athena	LCG
#	7	template	SusyPlot	Athena	LCG

List templates



Event generation



Detector simulation



Digitisation



Reconstruction



Data filtering



Physics analysis

```
In [2]: j4 = Job( templates[ 7 ] )
```

Create job from template

```
In [3]: j4.application
Out[3]: Athena (
```

List application properties

```
atlas_area = '' ,
packages = [ 'PhysicsAnalysis/SUSYPhys/SUSYPhysAlgs/SUSYPhysAlgs-00-04-02/' , 'PhysicsAnalysis/SUSYPhys/SUSYPhysUser/SUSYPhysUser-00-04-01/' , 'PhysicsAnalysis/SUSYPhys/SUSYPhysUtils/SUSYPhysUtils-00-04-01/' ] ,
pool_catalog = '' ,
atlas_release = '10.0.2' ,
option_files = [ '/afs/cern.ch/user/h/harrison/public/gangainputs/SusyPlot_joboptions.py' ] ,
options = [] ,
user_area = '/afs/cern.ch/user/h/harrison/cmtuser' ,
evt_max = -1 ,
outputdata = None
)
```

User packages

Application directives (job-options file)

```

Ganga
In [4]: status = j4.submit()
Ganga: INFO      submitting job 4

In [5]: Ganga: INFO      Job 4 Waiting at None - Sat Feb 11 11:57:59 2006
Ganga: INFO      Job 4 Ready at grid10.lal.in2p3.fr - Sat Feb 11 11:58:14 2006
Ganga: INFO      Job 4 Scheduled at grid10.lal.in2p3.fr - Sat Feb 11 11:58:30 2006
Ganga: INFO      Job 4 Running at grid10.lal.in2p3.fr - Sat Feb 11 11:59:49 2006
Ganga: INFO      Job 4 Done (Success) at grid10.lal.in2p3.fr - Sat Feb 11 12:08:17 2006

In [5]: !ls $j4.outputdir
gangademo_susyplot_hist.root  stderr  stdout

In [6]: jobs
Out[6]: Statistics: 4 jobs

-----
# id   status   name   application  backend  backend.status  backend.actualLCE
# 1   completed  Hello  Executable   LCG      Done (Success)   lcgce01.nic.ualberta.ca
# 2   completed  Bu2D0K  DaVinci     LCG      Done (Success)   t2-ce-02.inl.infn.it
# 3   completed  Bu2D0K  DaVinci     Dirac     outputready     LCG.CERN.ch
# 4   completed  SusyPlot  Athena     LCG      Done (Success)   grid10.lal.in2p3.fr

In [7]: 

```

Submit job to LCG

Status in LCG monitored

Output retrieved automatically from LCG

List jobs

Jobs from previous session(s) remembered

- **Ganga tutorials organised for both ATLAS and LHCb**
- **Ganga demo at EGEE 06,**
- **Ganga at EGEE-EELA (Rio 2004)**
- **Ganga Demo at the EGEE User Forum**



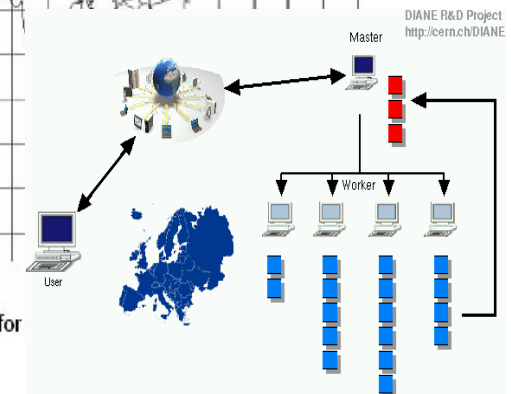
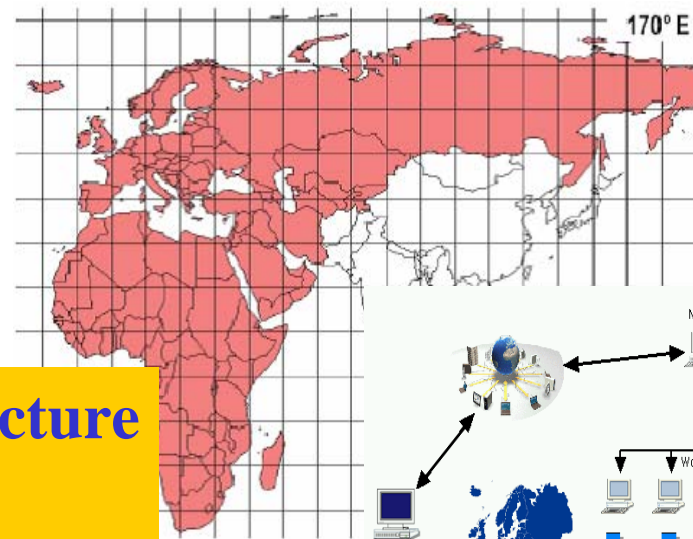
CERN, September 2005



Cambridge, January 2006

- **Ganga tried out by more than 100 people**
- **Feedback positive**
 - “Very handy way to organise job submission” (ATLAS user)
 - “Clever and nicely designed” (LHCb user)
- **Small but growing group of people regularly using Ganga (also from a laptop)**
 - LHCb analyses of up to 10^6 events run successfully on Grid using Ganga
 - ...

- **International Telecommunication Union**
 - ITU/BR: Radio-communication Sector
 - “management of the radio-frequency spectrum and satellite orbits for fixed, mobile, broadcasting and other communication services
- **RRC-06 (15 May–16 June 2006)**
 - 120 countries will negotiate the new frequency plan
 - a part of a new international agreement
 - introduction of digital broadcasting



**Use of EGEE technology and infrastructure
Presented at the EGEE User Forum
GANGA is an important component**

Geant4 Physics Validation Use of the GRID Resource

Patricia Mendez Lorenzo
CERN (IT-GD) / CNAF

Geant4 Bio-Medical Development
Geant4 Physics Validation

INFN Genova, 13-20 July 2005

www.eu-egee.org



INFSO-RI-508833



Enabling Grids for E-science

Geant4 in LCG

- **Electromagnetic and Hadronic physics** are fundamental features to be properly simulated in Geant4, however they are **extremely CPU demanding**

- Number of **events** and **energy** depending:
 - 1 event of 1GeV ~ 0.03 sec (2.4GHz machine)
 - 1 event of 300 Gev ~ 9-10 sec

GRID

- **Goal during the Software Validation:**

Comparison some shower observables between the two different Geant4 versions and check statistical significant changes

- **Applications in LCG:**

First application last December 2004

Second application end of June 2005

GRID

- A total amount of about 3 years of CPU time (1GHz machine)
- Very small output for the whole production: 15-20 GB

INFSO-RI-508833

Genova 13^a-20^a July

Patricia Mendez Lorenzo

- **Geant4 is a toolkit for the simulation of the passage of particles (radiation) through matter.**
 - Its areas of application include high energy, nuclear and accelerator physics, as well as studies in medical and space science.
- **Use of EGEE technology and infrastructure to run large validation of the G4 software package**
 - GANGA is an important tool to run these productions and also to ensure a lightweight bookkeeping

Large Scale Virtual Screening of Drug Design on the Grid

Fighting against Avian Flu

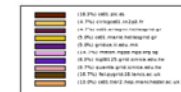
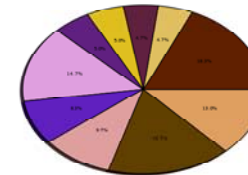
Hung-Chun Lee and Yin-Ta Wu



Some statistics of DIANE jobs

- Submitted Grid jobs: 300
- *Healthy* jobs: 261 (87%)
- Total number of dockings: 40210
- Total CPU time: 55684848 sec (1.76 year)
- Job duration: 249746 sec (2.9 days)

Grid Job Distribution



Academia Sinica Grid Computing

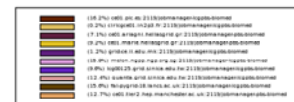
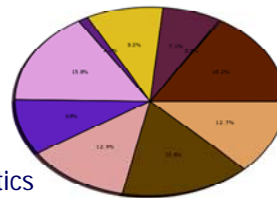
Logging and bookkeeping ... Thanks GANGA

```
Statistics: 325 jobs slice("DIANE_6")
-----
# id      status      name      subjbs      application      backend      backend.actualCE
# 1610    running     DIANE_6   Executable  LCG            melon.ngpp.ngp.org.sg:2119/jobmanager-lcgpbs-
# 1611    running     DIANE_6   Executable  LCG            node001.grid.auth.gr:2119/jobmanager-lcgpbs-b
# 1612    running     DIANE_6   Executable  LCG            polgrid1.in2p3.fr:2119/jobmanager-lcgpbs-biom
# 1613    failed      DIANE_6   Executable  LCG            polgrid1.in2p3.fr:2119/jobmanager-lcgpbs-sdj
# 1614    submitted   DIANE_6   Executable  LCG            ce01.ariagni.hellasgrid.gr:2119/jobmanager-pb
# 1615    running     DIANE_6   Executable  LCG            ce01.pic.es:2119/jobmanager-lcgpbs-biomed
# 1616    running     DIANE_6   Executable  LCG            ce01.tier2.hep.manchester.ac.uk:2119/jobmanag
# 1617    running     DIANE_6   Executable  LCG            clrlegce03.in2p3.fr:2119/jobmanager-lcgpbs-bi
```

plain table of GANGA job logging info.

- Helpful for tracing the execution progress and Grid job errors
- Fairly easy to visualize the job statistics

Contribution by Computing Element



Use of EGEE technology and infrastructure to run large validation of the G4 software package

GANGA is an important tool also to ensure a lightweight bookkeeping