

# RGLite

## A gLite plug-in for ROOT

Anar Manafov, GSI  
HEPCG Workshop, Apr 27 2006



[www.gsi.de](http://www.gsi.de)



[www.bmbf.de](http://www.bmbf.de)



[www.d-grid.de](http://www.d-grid.de)



[aliceinfo.cern.ch](http://aliceinfo.cern.ch)

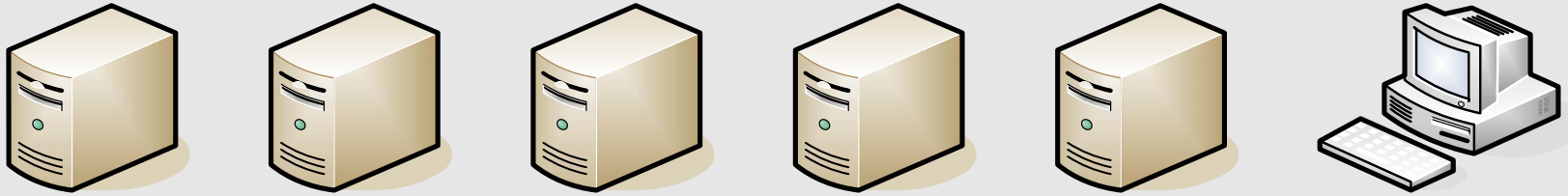


[root.cern.ch](http://root.cern.ch)



[public.eu-egee.org](http://public.eu-egee.org)

# gLite R1.5

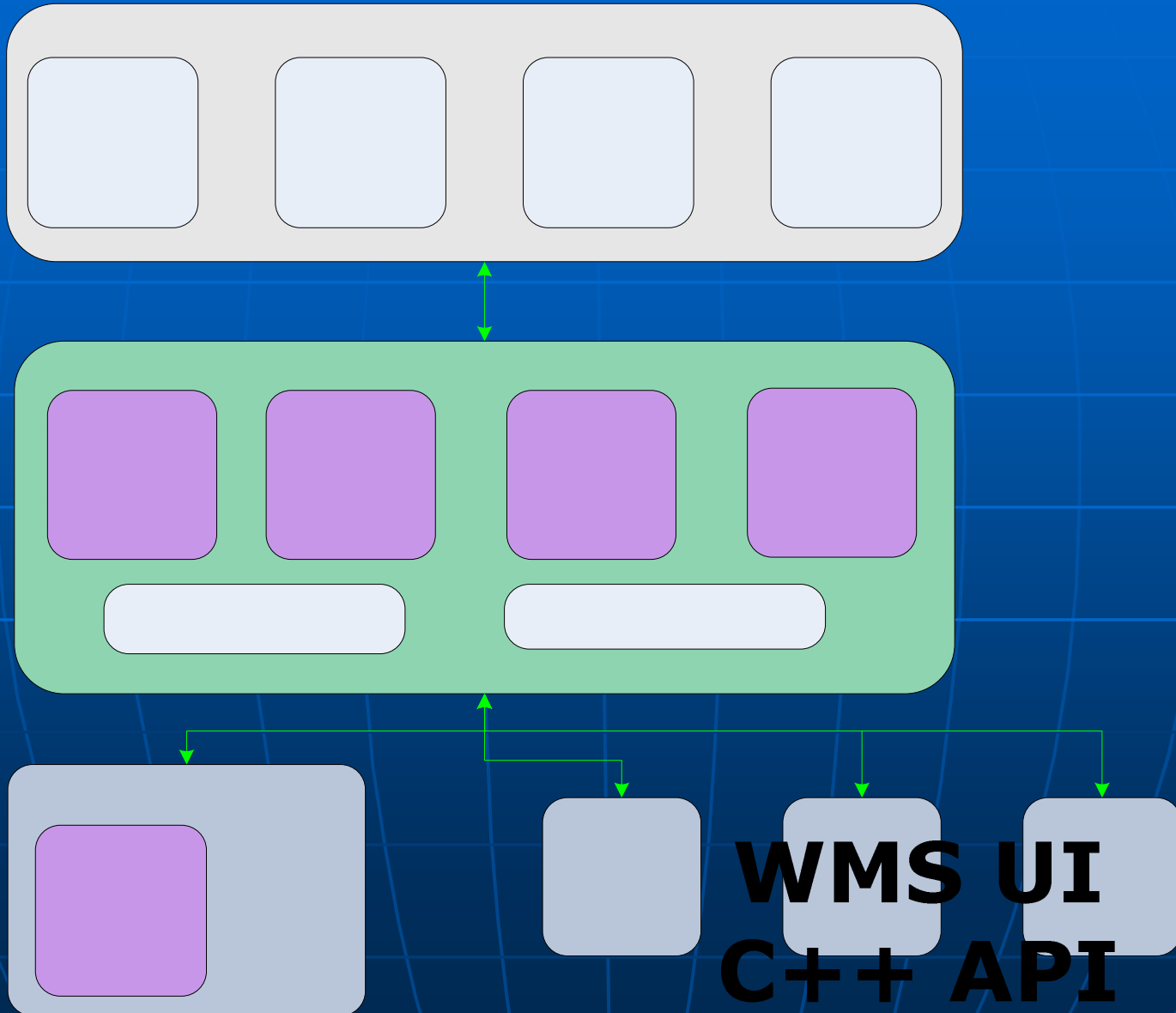


- development and testing environment;

## All gLite nodes installed as

- operational and providing essential resources needed for project development;
- the installation and configuration process turned out to be a difficult task
  - there is no support for this release and there are bugs in the MW,
  - lots of time has been spent on it;
- several tracked bugs were reported to "savannah" and fixed locally;
- the process of installation and configuration was documented in GSI's Wiki site (includes workarounds and tricks).

# gLite API Wrapper library – UI provider



# gLite API Wrapper

## **gLite API Wrapper:**

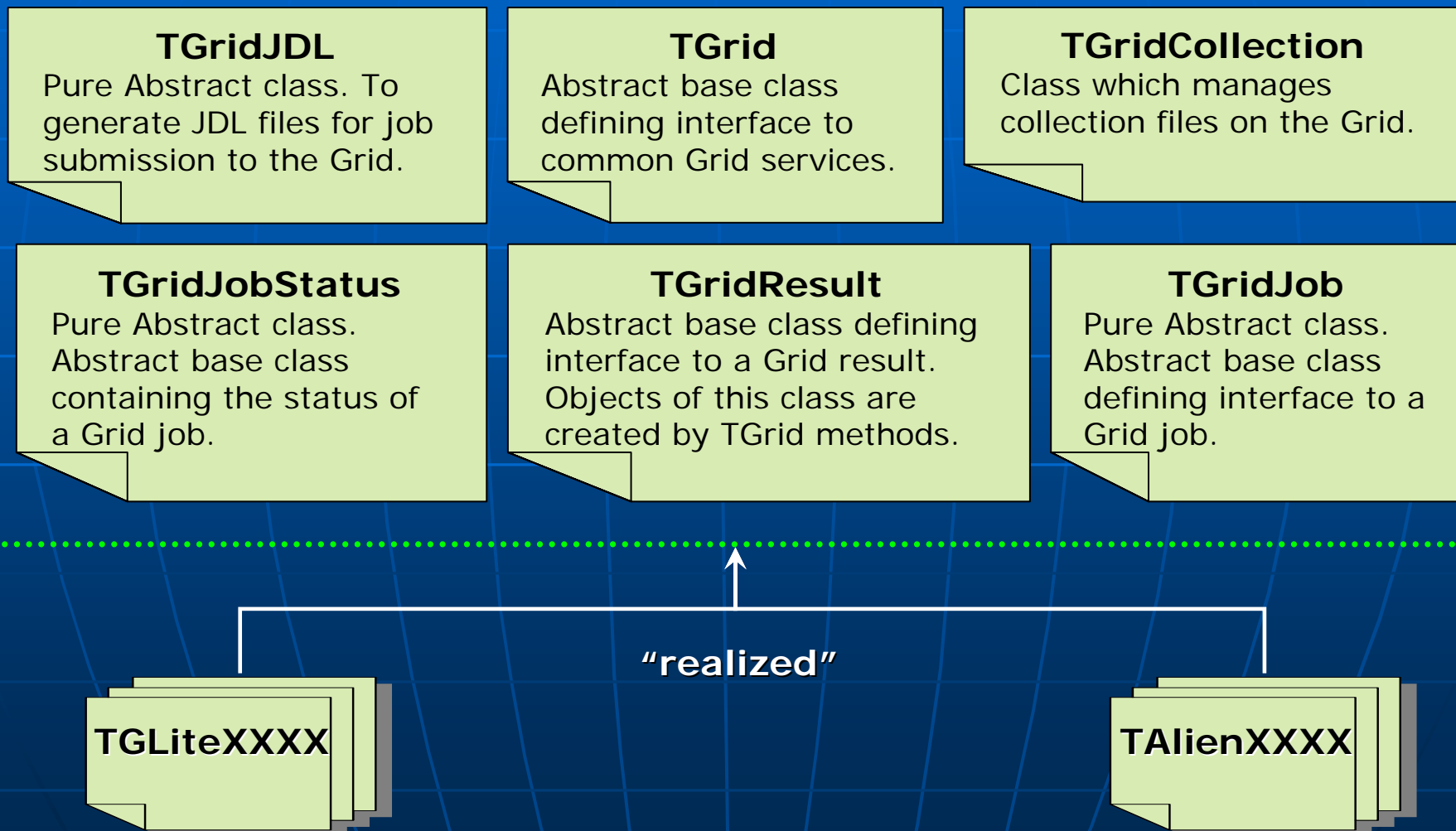
- encapsulates the gLite API,
- offers (implements) simple gLite UI functionality,
- implements an automation,
- implements modularity of the system:
  - simplifies the usage of gLite API,
  - simplifies the development and testing,
- by design it implements a number of managers:
  - main manager is a singleton class,
- supports the persistence and has internal log engine.

## **gLite API Wrapper development**

- C++ code.
- So far home-made tests are used. It will be switched to the Boost Test Library.
- The library building and installing is done using GNU's autotools.
- Version control system is GSI's central SVN.
- Code documentation is doxygen engine.
- Project documentation: GSI's Wiki site.
- Bug tracking will be done using Bugzilla.

# RGLite – ROOT plug-in

## Set of ROOT Interfaces to the Grid



# RGLite

- Development and tests are done with ROOT 5.10.
- Installation and registration of plug-in via a patch to ROOT src.
- The following interfaces partially implemented up to now:
  - TGrid -> TGLite,
  - TGridJob -> TGLiteJob,
  - TGridJobStatus -> TGLiteJobStatus.
- RGLite implements:
  - Job Submission (gLite WMSUI),
  - Job Status Retrieving (gLite LB),
  - Job Output Retrieving (gLite WMSUI).
- Code documentation is the ROOT style auto HTML gen.
- Project documentation: GSI's Wiki site.

# RGLite

```
lxial24.gsi.de
[anar@grid8 anar]$ root -b
*****
*                                     *
*      W E L C O M E  to  R O O T      *
*                                     *
*   Version   5.10/00      1 March 2006 *
*                                     *
* You are welcome to visit our Web site *
*      http://root.cern.ch            *
*                                     *
*****

Compiled on 23 April 2006 for linux with thread support.

CINT/ROOT C/C++ Interpreter version 5.16.8, February 9, 2006
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
TGrid::Connect("glite");
Info in <TGLite::TGLite>: gLite API Wrapper engine has been successfully initialized.
TGridJob *job = ggrid->Submit("JDLs/simple.jdl");
JDL: JDLs/simple.jdl
Info in <TGLite::Submit>: Job successfully submitted
Info in <TGLite::Submit>: NativeJobID https://grid12.gsi.de:9000/u1SdtUjX3kv9EQR1DAJRHg; ROOT JobID 1;
TGridJobStatus *status = job->GetJobStatus();
status->GetStatus()
Info in <TGLiteJobStatus::GetStatus>: Native JobID = https://grid12.gsi.de:9000/u1SdtUjX3kv9EQR1DAJRHg
Info in <TGLiteJobStatus::GetStatus>: Job status is [3]; gLite status code is "Ready"; gLite status string is "matching resources found"
Info in <TGLiteJobStatus::GetStatus>: Job status is kWAITING
(const enum TGridJobStatus::EGridJobStatus)1
status->GetStatus()
Info in <TGLiteJobStatus::GetStatus>: Native JobID = https://grid12.gsi.de:9000/u1SdtUjX3kv9EQR1DAJRHg
Info in <TGLiteJobStatus::GetStatus>: Job status is [6]; gLite status code is "Done"; gLite status string is "execution finished, output is available"
Info in <TGLiteJobStatus::GetStatus>: Job status is kDONE
(const enum TGridJobStatus::EGridJobStatus)5
TGLiteJob* job_glite(job);
job_glite->GetJobOutput("/tmp");
root [7]
```

# RGLite

lxial24.gsi.de | lxial24.gsi.de (1) | lxial24.gsi.de (2)

```
[anar@grid8 anar]$ root -b
```

```
*****  
*                               *  
*   W E L C O M E  to  R O O T   *  
*                               *  
*   Version   5.10/00       1 March 2006   *  
*                               *  
*   You are welcome to visit our Web site *  
*       http://root.cern.ch             *  
*                               *  
*****
```

Compiled on 23 April 2006 for linux with thread support.

CINT/ROOT C/C++ Interpreter version 5.16.8, February 9, 2006

Type ? for help. Commands must be C++ statements.

Enclose multiple statements between { }.

```
TGrid::Connect("glite");
```

```
Info in <TGLite::TGLite>: gLite API Wrapper engine has been successfully initialized.
```

```
TGridJob *job = gGrid->Submit("JDLs/proofd.jdl");
```

```
JDL: JDLs/proofd.jdl
```

```
Info in <TGLite::Submit>: Job successfully submitted
```

```
Info in <TGLite::Submit>: NativeJobID https://grid12.gsi.de:9000/KHC0g77C3hyg0Xt1UZyh0A: ROOT JobID 1:
```

```
TGridJobStatus *status = job->GetJobStatus();
```

```
status->GetStatus()
```

```
Info in <TGLiteJobStatus::GetStatus>: Native JobID = https://grid12.gsi.de:9000/KHC0g77C3hyg0Xt1UZyh0A
```

```
Info in <TGLiteJobStatus::GetStatus>: Job status is [3]; gLite status code is "Ready"; gLite status string is "matching resources found"
```

```
Info in <TGLiteJobStatus::GetStatus>: Job status is kWAITING
```

```
(const enum TGridJobStatus::EGridJobStatus)1
```

```
status->GetStatus()
```

```
Info in <TGLiteJobStatus::GetStatus>: Native JobID = https://grid12.gsi.de:9000/KHC0g77C3hyg0Xt1UZyh0A
```

```
Info in <TGLiteJobStatus::GetStatus>: Job status is [6]; gLite status code is "Done"; gLite status string is "execution finished, output is available"
```

```
Info in <TGLiteJobStatus::GetStatus>: Job status is kDONE
```

```
(const enum TGridJobStatus::EGridJobStatus)5
```

```
TGLiteJob* job_glite(job);
```

```
job_glite->GetJobOutput("/home/anar/");
```

## TODO (till Aug 2006)

- To complete job submission using WMPProxy API.
- To implement basic I/O operations using gLite I/O API.
- To investigate an interface for interactive job submission:
  - find out a way to communicate with interactive job.
- To analyse current status and to define development plan for the next year.