User and developer friendly access to Grids and Clouds

Ariel Garcia

P. Wolniewicz, H. Kornmayer, M. Stümpert

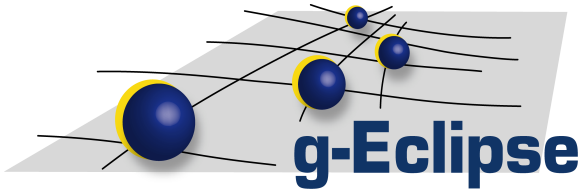
on behalf of the

g-Eclipse Consortium

The idea

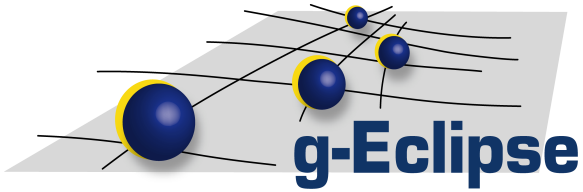
- Accessing a Grid is difficult
  - provide user-friendly UI for accessing Grids
- Many different middlewares are out there
  - provide extensible middleware-independent framework for accessing Grids
- Currently supported middlewares:
  - gLite - Batch oriented Grid for the scientific user
  - GRIA - Service-oriented infrastructure for industry and commerce
  - AWS elastic compute cloud EC2 and S3 storage – Cloud computing

Some facts



- Funded by European Commission, FP6
- Eclipse.org project (incubation)
- 20+ developers
- Monthly release cycle
- Currently at 1.0 release track
  - release candidate RC0 now
  - release 1.0 in December 2008

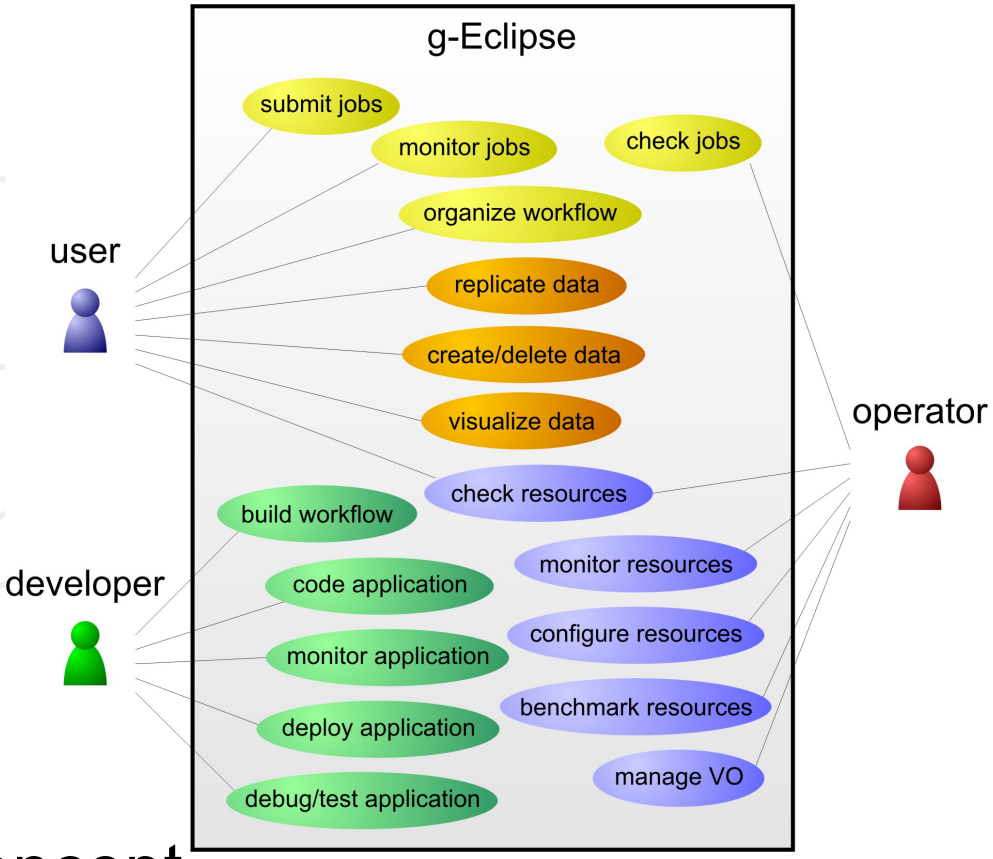
g-Eclipse domains



- User interface / Grid client
  - graphical user interface for accessing Grid infrastructures
- Framework / API
  - collection of pure Java classes for developing client- and server-side applications for the Grid

The g-Eclipse client

- Support for
  - Grid User
    - job management
    - data management
  - Grid Operator
    - site administration
    - user administration
  - Grid Developer
    - compile/debug apps.
    - deploy apps.
- Eclipse's perspective concept



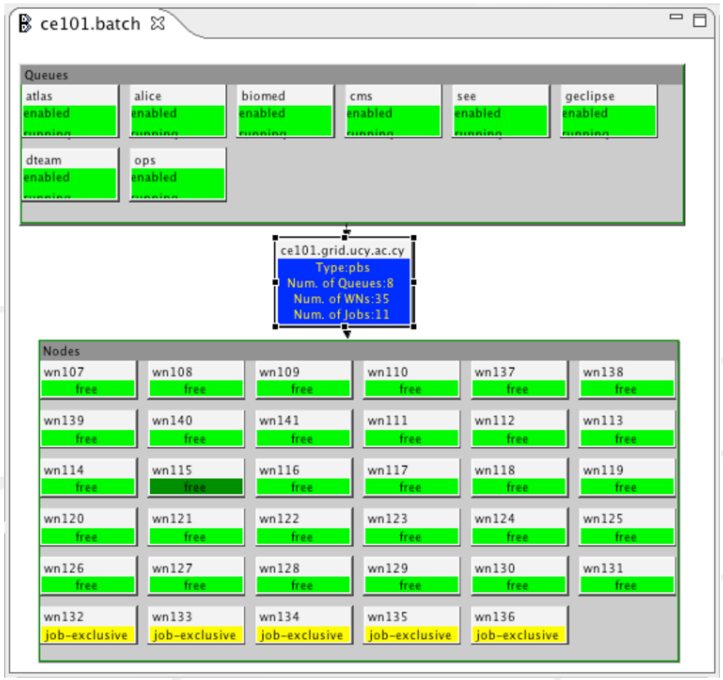
The user perspective

- Data management
  - Files/folders create/save/copy/move/delete
    - GridFTP
    - SRM (WS standard)
    - LFC, own native Java implementation
    - GRIA data stagers
    - AWS S3
  - 3rd party transfers
  - Transfer manager
    - can restart unfinished transfers

The user perspective

- Job management
  - Job description creation and editing
    - JSDL standard compliant editor, full support
    - JDL supported (gLite)
  - Job submission, status monitoring
    - WMS/Cream (gLite)
    - JobService (GRIA)
  - Parametric jobs support
- Workflows
  - Dedicated workflow editor
  - Submission and status, just like ordinary job!
- Data visualisation
  - Using VTK (and SRS3D)

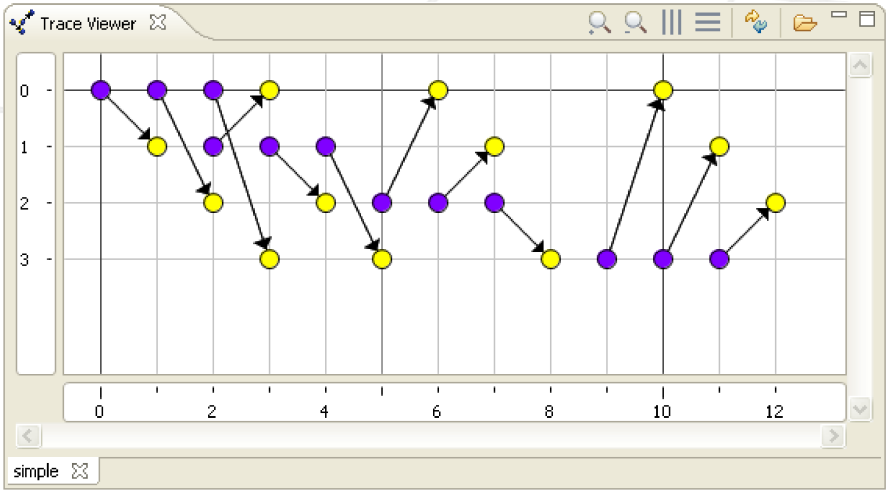
The operator perspective



- Site administration
  - Queue management
    - PBS/Torque
  - Job management
  - Infrastructure monitoring
  - Infrastructure testing
  - Infrastructure benchmarking
  - Service level agreement editor (in preparation)
- User administration
  - VO management (in preparation)

The developer perspective

- Application development
  - Remote compiling
  - Remote debugging
    - normal Eclipse debugging perspective!
  - Analyzing MPI applications
    - traceviewer



- Application deployment

How does it look like?

Mounted File Systems

Virtual Organisation

Computing Elements

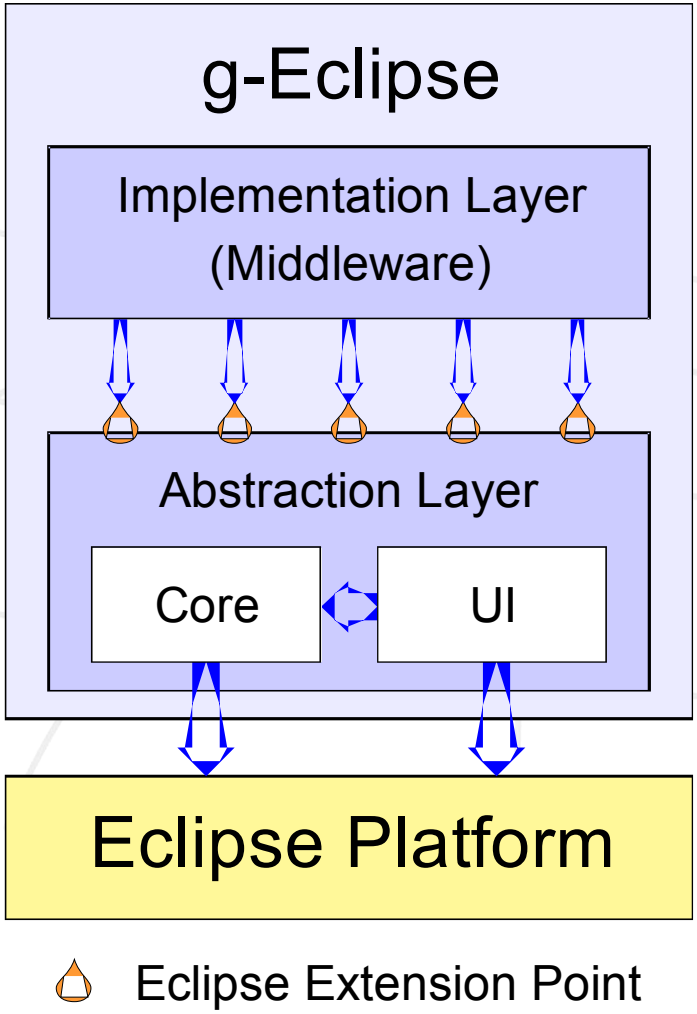
Services

Storage Elements

- MyGridProject
  - Connections
    - Linz
    - Poznan
  - Job Descriptions
    - MyJSDL.jsdl
  - Jobs
    - .MyJSDL.job
      - MyJSDL.jsdl
  - Workflows
    - MyWorkflow.workflow
  - geclipse
    - Computing
      - CE @ a01-004-128.gridka.de:2119/jobmanager-pbspro-geclipse
      - CE @ ce-1-fzk.gridka.de:2119/jobmanager-pbspro-geclipse
      - CE @ ce-2-fzk.gridka.de:2119/jobmanager-pbspro-geclipse
      - CE @ ce-fzk.gridka.de:2119/jobmanager-pbspro-geclipse
      - CE @ ce.egee.man.poznan.pl:2119/jobmanager-lcgpbs-geclipse
      - CE @ ce101.grid.ucy.ac.cy:2119/jobmanager-lcgpbs-geclipse
      - CE @ egee-ce1.gup.uni-linz.ac.at:2119/jobmanager-pbs-geclipse
    - Services
      - Info Services
        - BDII @ ldap://bdii101.grid.ucy.ac.cy:2170
      - Job Status Services
      - Job Submission Services
        - WMS @ https://wmslb101.grid.ucy.ac.cy:7443/glite_wms_wmproxy_server
      - Other Services
    - Storage
      - SE @ egee-ce1.gup.uni-linz.ac.at
      - SE @ gridka-dCache.fzk.de
      - SE @ plethon.grid.ucy.ac.cy
      - SE @ se1.egee.man.poznan.pl

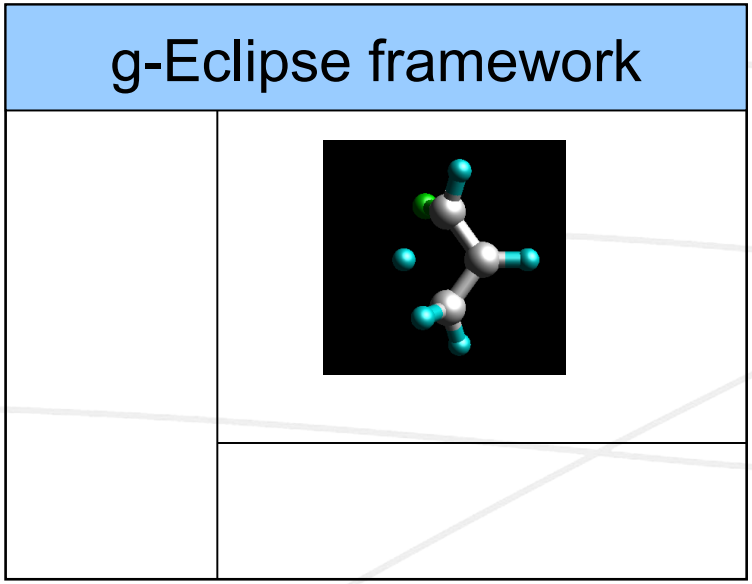
Developing on g-Eclipse

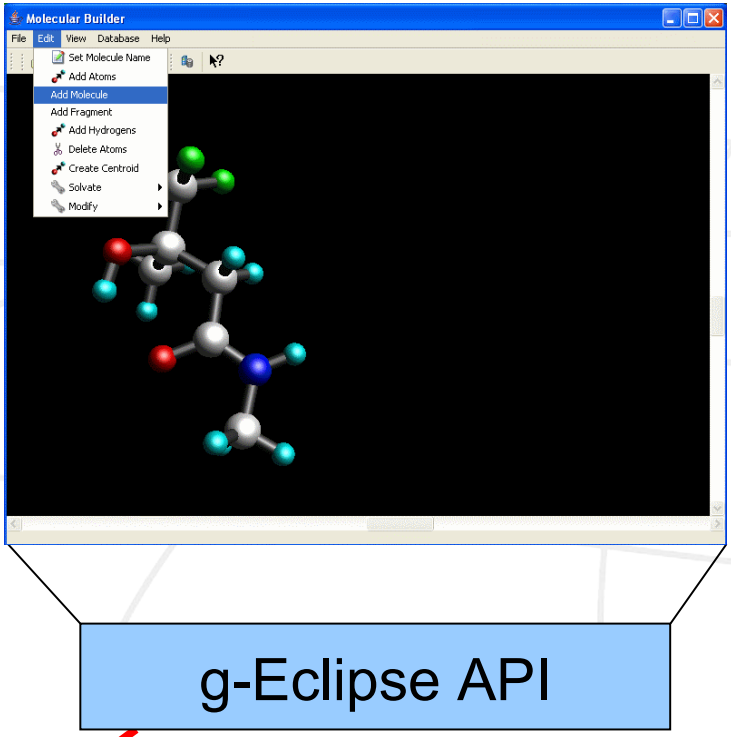
- Abstraction layer
  - core functionality, e.g.
    - authentication/authorization
    - VO management
    - data management
    - job submission
  - common user interface, e.g.
    - views
    - wizards
    - dialogs
    - preference pages
- Implementation layer
  - implementing core functionality
  - middleware specific functionality



Applications with g-Eclipse

- Application inside g-Eclipse
- Application on top of gEclipse





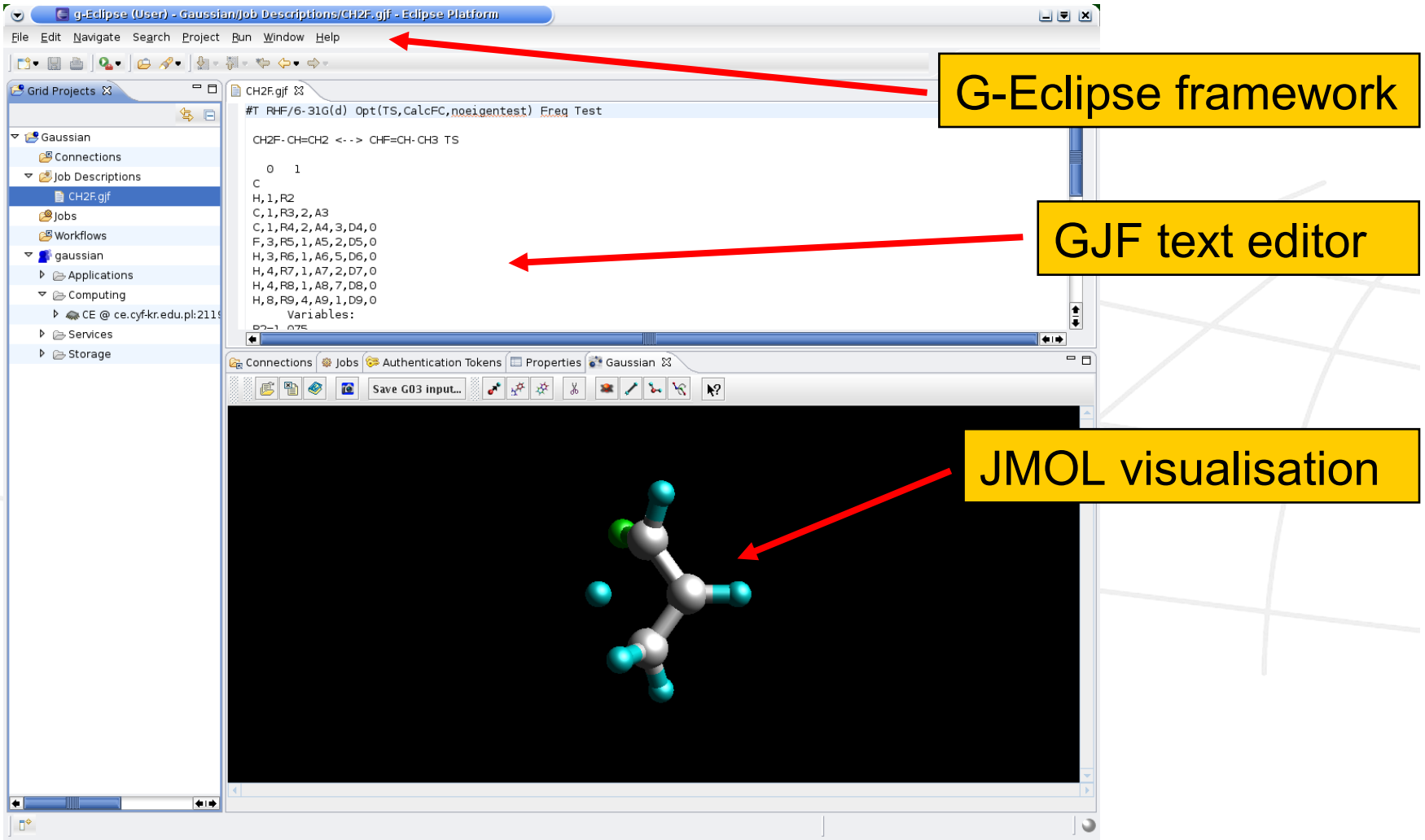
g-Eclipse API

GRID

Application inside g-Eclipse

- Application is plugged into g-Eclipse framework
- Benefit from the user-friendly graphical interface for accessing Grid infrastructures
- Only few optional elements should be added:
  - editor for input files/parameters
  - submission support
  - data visualisation

Application inside g-Eclipse

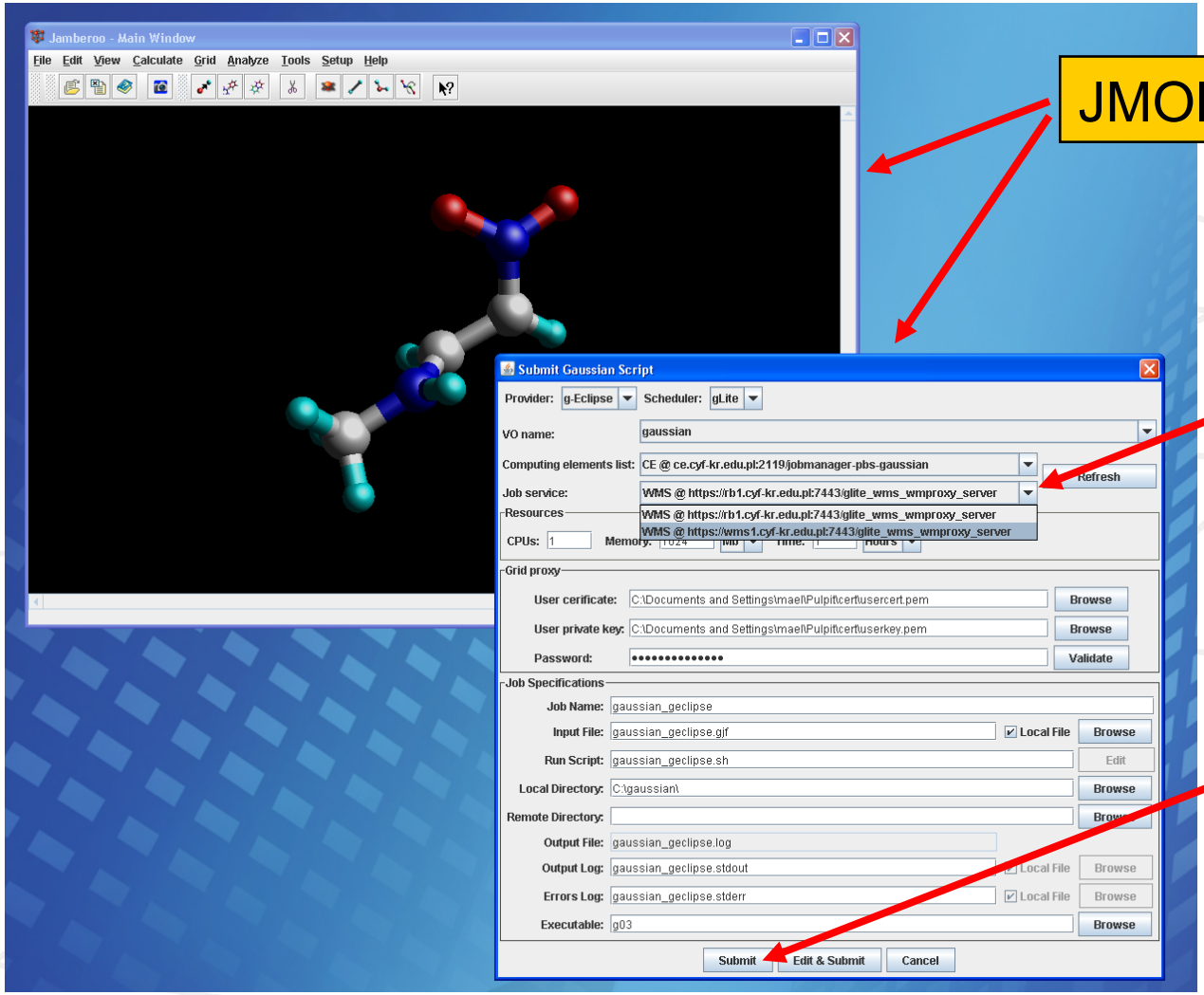


- Grid job description is created from GJF file on the fly

Application on top of g-Eclipse

- Enhance existing applications with Grid support
- Application has its out GUI and calls g-Eclipse API for accessing Grid resources.
- Application is started as Eclipse Rich Client Application
- Can access other bundles provided by g-Eclipse

Application on top of g-Eclipse



JMOL application

Data provided by G-Eclipse libraries

Action will be delegated to g-Eclipse libraries

Authn. and authz.

- AAI support
  - Globus proxies (X509)
  - **VOMS proxies**
  - GRIA keystores
  - AWS tokens
  - trusted certificates management

```
VomsProxyDescription desc = new VomsProxyDescription();
desc.setVo( vo );
desc.setCertFile( "/home/user/.globus/usercert.pem" );
desc.setKeyFile( "/home/user/.globus/userkey.pem" );
desc.setLifetime( 86400 );

AuthenticationTokenManager manager =
    AuthenticationTokenManager.getManager();
IAuthenticationToken token = manager.createToken( desc );
token.validate();
token.setActive( true );
```

VOs and jobs

- VO support

```
VomsVoCreator creator = new VomsVoCreator();
creator.setVoName( "geclipse" );
creator.setVoHost( "dgrid-voms.fzk.de" );
creator.setVoPort( 15009 );
creator.setVoHostDN(
    "/O=GermanGrid/OU=FZK/CN=host/dgrid-voms.fzk.de" );
creator.setVoInfoService(
    URI.create( "ldap://iwrbdii.fzk.de:2170" ) );

IVoManager manager = GridModel.getVoManager();
IVirtualOrganization vo
    = ( IVirtualOrganization ) manager.create( creator );
```

- Job management

```
WMSClient wmsClient = WMSClient.getClient( wmsClientUri );
JobIdStructType jobId = wmsClient.submitJob( my_jsdl, null );
LBClient lbClient = LBClient.getLBClient( lbClientUri );
JobStatus status = lbClient.getJobStatus( jobId.getId() );
```

Collaboration

- We welcome reuse and new developments
  - use of g-Eclipse as a library
    - for RCP applications
    - for server-side services which need Grid access
    - ...
  - new middleware implementations
  - new components
- EU project ends in 2008
  - ongoing proposals
  - BUT keeps going as Eclipse project
    - ongoing support
  - gathering community, users & developers

The end

Thank you for listening!

Questions?