

Easy Access to Grid Infrastructures

Dr. Harald Kornmayer (NEC Laboratories Europe)

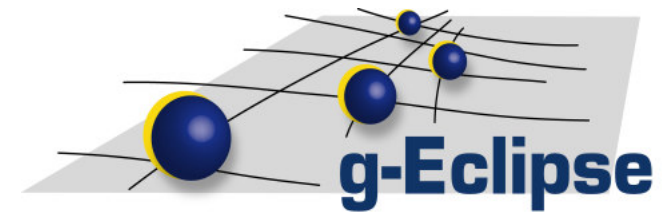
On behalf of the g-Eclipse consortium

WP11 Grid Workshop

Grenoble, France

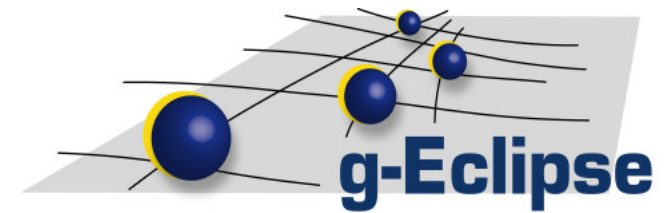
09th of December 2008

About me!



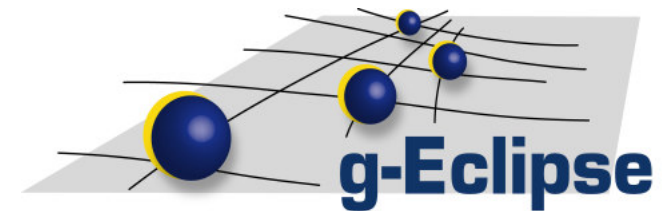
- Background in astro particle physics
- 2001: Build a Risk Management system (SOA with JAVA)
- Forschungszentrum Karlsruhe:
 - 2003: CrossGrid
 - 2004: EGEE
 - 2006: D-Grid
 - 2006: g-Eclipse
- Since 2007:
 - NEC Laboratories
 - IT Research Group

g-Eclipse – the idea

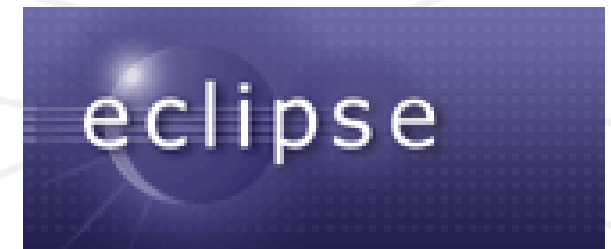


- Users want **easy access** to the system
- Users act within **different roles**
 - Grid applications users
 - Grid resource providers and operators
 - Grid application developers
- Users are middleware agnostic
 - Build a **middleware independent** framework
- Provide a general UI framework/eco system for the different Grid actors based on **a reliable platform**
 - (re-)use Eclipse and contribute!
 - gain OS independence (by using JAVA!)

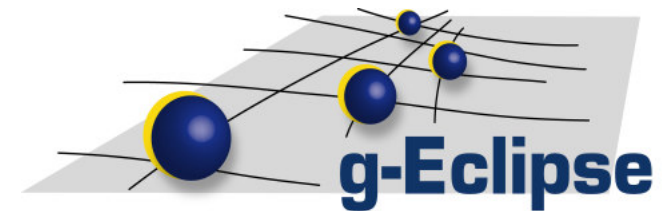
Eclipse



- Started in 2001
 - IBM donated their Java development framework as Open Source
 - Pure Java development
 - First industrial partners joined
- Eclipse Foundation started in 2004
 - Independent non-profit organization
 - Eclipse management organization
 - Councils (requirements, architecture and planning)
 - 10 projects (including > 50 subprojects)
 - With Eclipse 3.0 not only a Java IDE, but a general framework build for extension



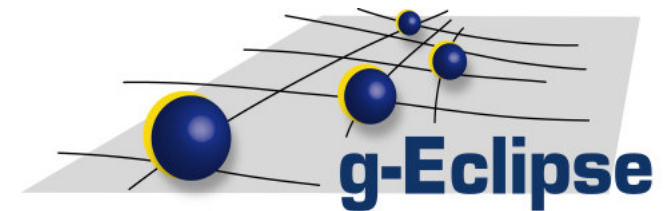
Can we learn from others?













- Are there other distributed systems supported by different middleware systems?
 - J2EE
 - (enterprise beans, JMS, web services, ...)
 - Based on specifications
 - implemented by different Vendors
 - Websphere, BEA, Oracle, JBoss, Geronimo, ...
 - Customers don't want to be vendor dependent
 - Use a generic middleware independent development tool
- Web Tools project (WTP) at Eclipse.org
- www.eclipse.org/webtools
 - Wizards, Editors, etc to simplify the development of Web applications
 - Independent on the underlying infrastructure

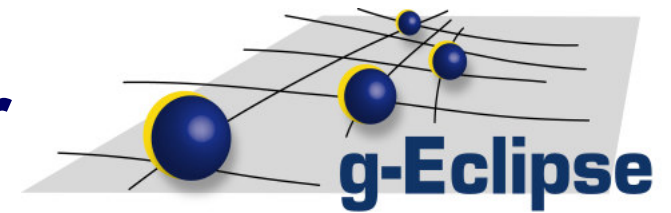


g-Eclipse – projects



- www.geclipse.eu
- Project funded by the European Commission (INFISO-32347)
- 8 partners
-    
-  
-    
- Until December 2008
- www.eclipse.org/geclipse
- Technology project at Eclipse Foundation
- Release 1.0.0 with stable API scheduled for next week (December 2008)
- Gathering community
 - i.e. in Grid communities
 - In Eclipse community

The reality for the Grid user

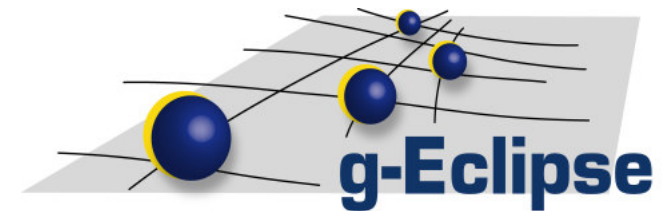


- Infrastructure for scientists were built in the past years
- Many application domains start using Grid infrastructures
- But...
 - Grid technology is complex
 - Different middleware systems are used
 - gLite, Globus, GRIA, UNICORE, ...
 - **Different programming paradigms**
 - Batch type systems vs. service oriented systems
 - Many programming languages



→ The threshold is too high for the “standard” user!

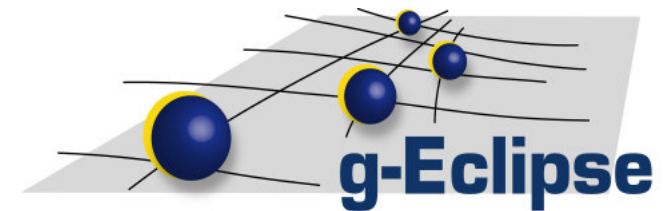
Grid application life cycle



- In most cases, e-Users (e-Scientists, e-Engineers, e-Stock Traders) have their application(s)
 - Legacy code written in different languages (FORTRAN, C, C++, ...)
- e-Users want to collaborate
 - A Virtual Organisation is build around a Virtual Computing Center on existing (and new) infrastructure
- e-Users create Grid projects
- e-Users want to interact with the Grid
 - without knowing all details!!
(development, deployment, testing, management, ...)
- → Tooling is necessary!!
 - Wizards, Editors, ...
 - Hide the complexity!!



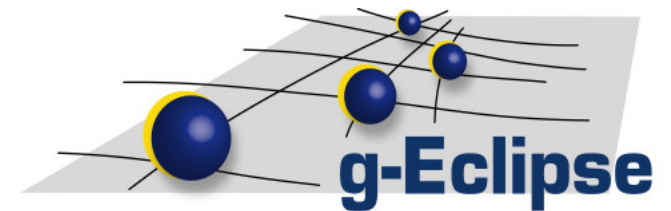
Grid Middleware



- Connects resources (computing, storage, (network), . . .) at different sites to one Grid infrastructure
- Provides services to access an infrastructure
- Common basic functionalities of middlewares:
 - Security layer for authentication and authorization
 - Transfer protocols for accessing and managing data
 - Brokers to distribute computing jobs on the infrastructure
 - Higher level services (SLAs, accounting, ...)
- Many different middlewares are available:
 - gLite, Globus Toolkit, UNICORE, GRIA, ...
- g-Eclipse supports

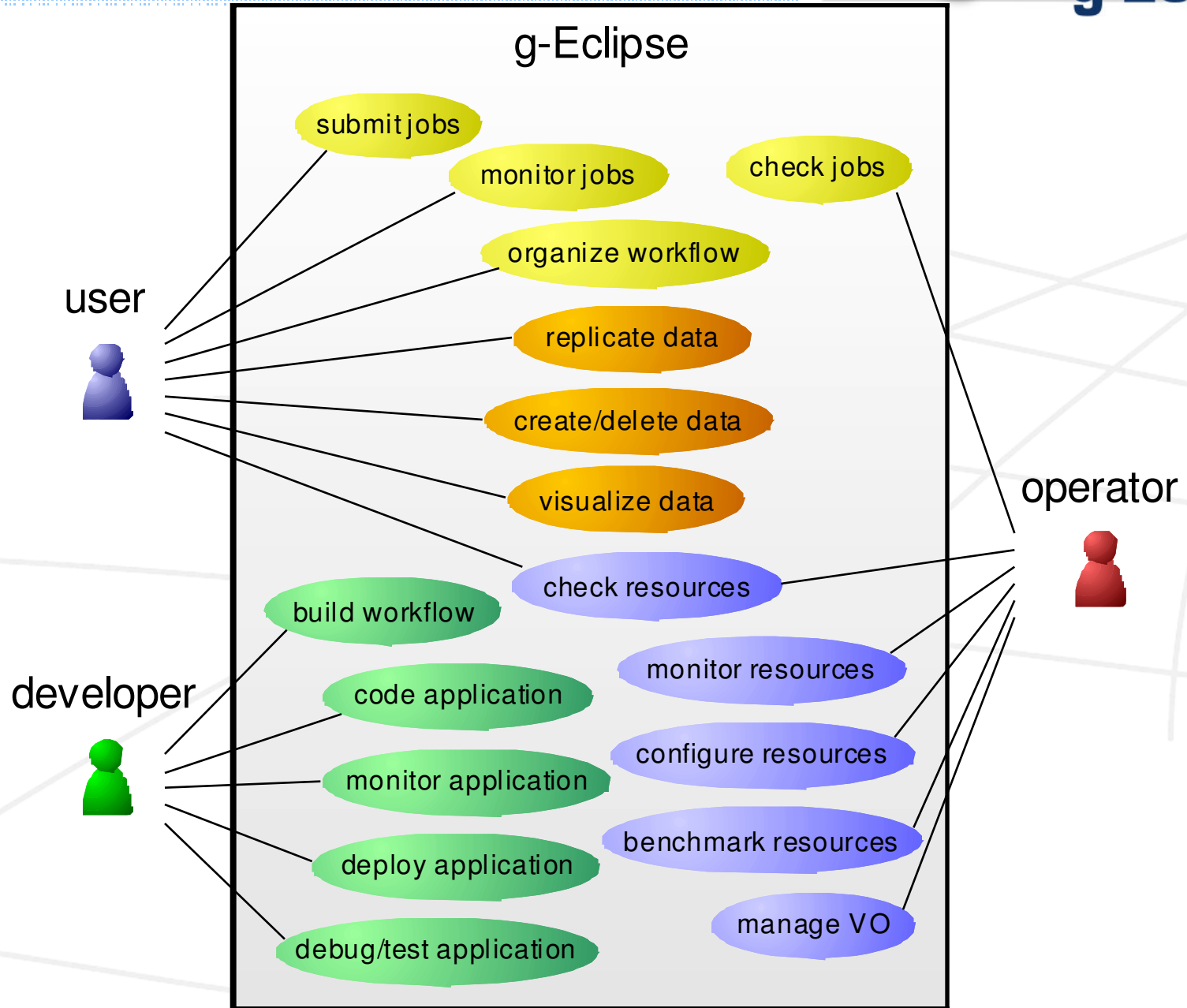
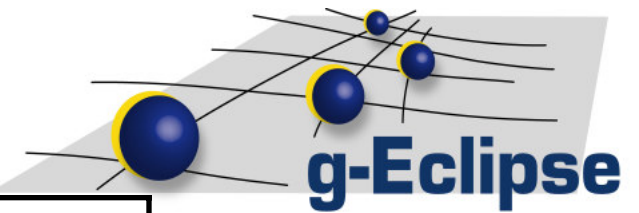


Demo 1

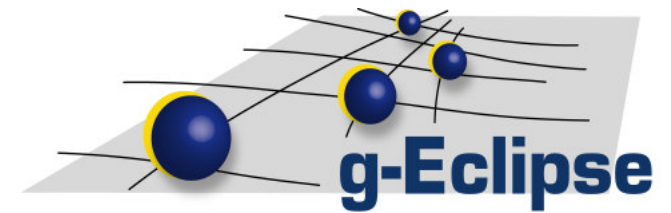


- Just use Amazon
- Create a Grid project
 - Including a VO
- Access to data
 - Edit a file remotely
- Rent a machine in a few seconds
 - Login there

Use cases

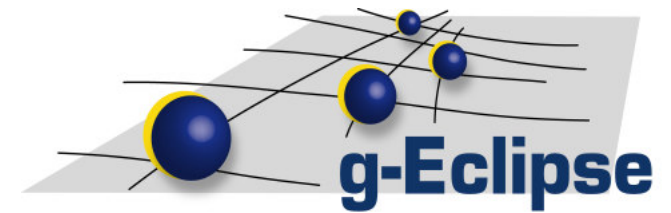


Roles and Contexts

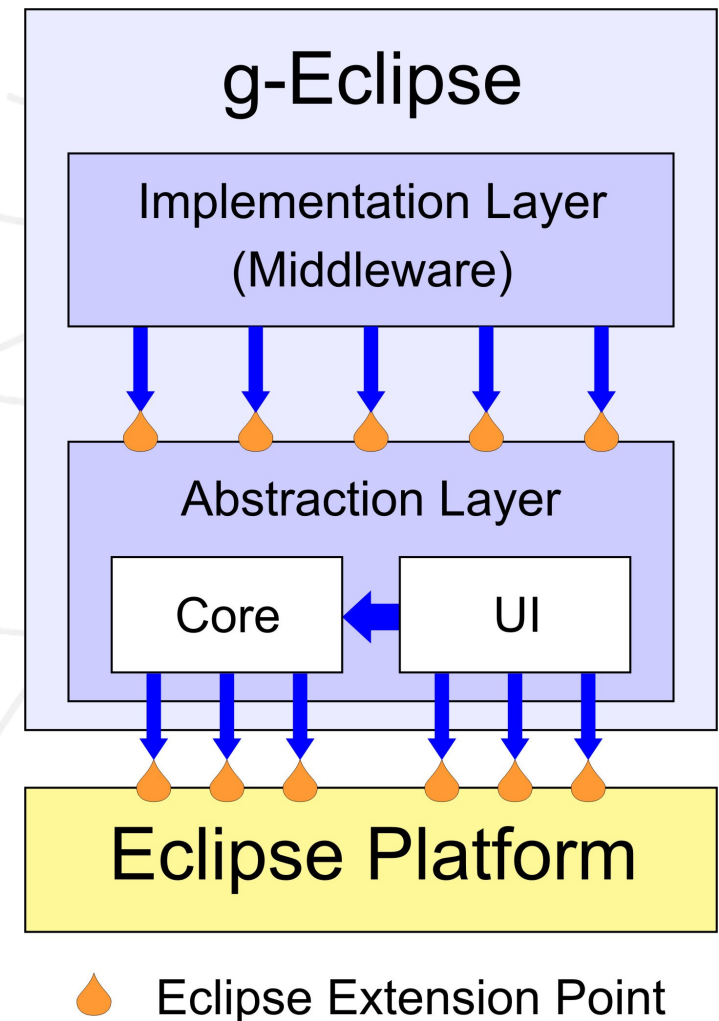


- Grid user plays different **roles**
 - Grid applications user
 - Grid resource provider and operator
 - Grid application developer
 -
- Grid user acts in different **contexts**
 - Virtual Organizations
 - Projects
 - ...
- g-Eclipse supports “Contextualization”
 - Depending on the user role/context a different set of tools is used by/presented to the user
- g-Eclipse supports “Customization”
 - Build the user-preferred workbench
 - Persistent over sessions

The Architecture

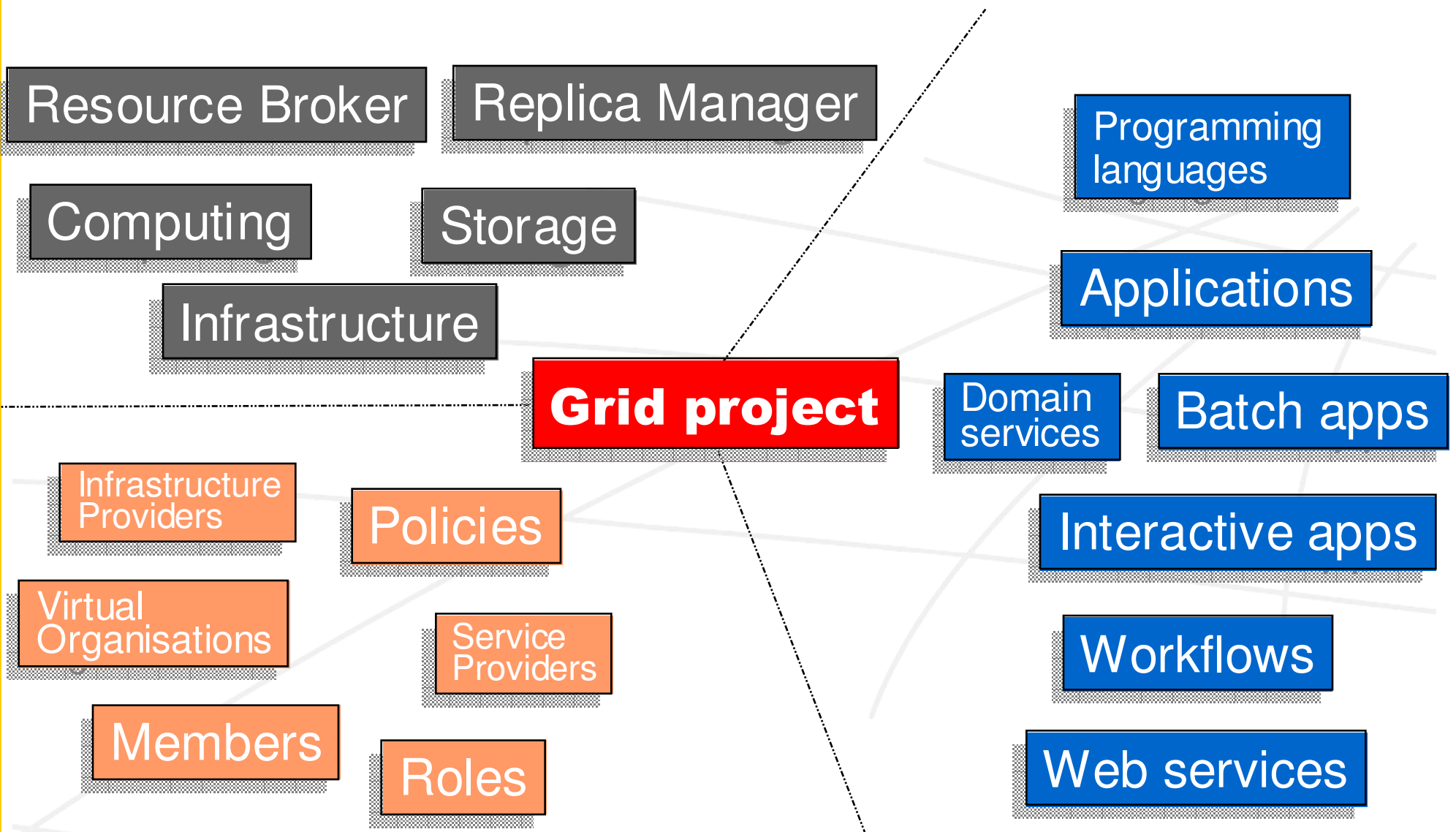
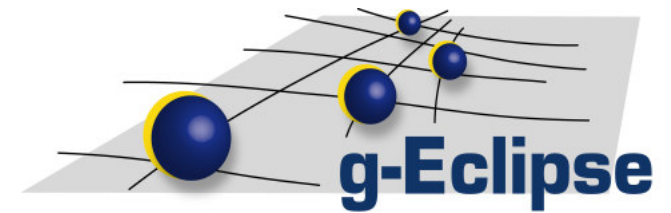


- Two layers:
 - Abstraction layer (Grid model):
 - Authentication/Authorization
 - Job management
 - Data management
 - Services
 - ...
 - Implementation layer:
 - Implements the model for specific middlewares/Grids
- UI is based on abstraction layer
 - UI looks the same for all middlewares

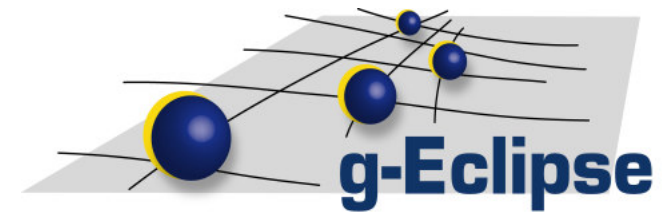


🔹 Eclipse Extension Point

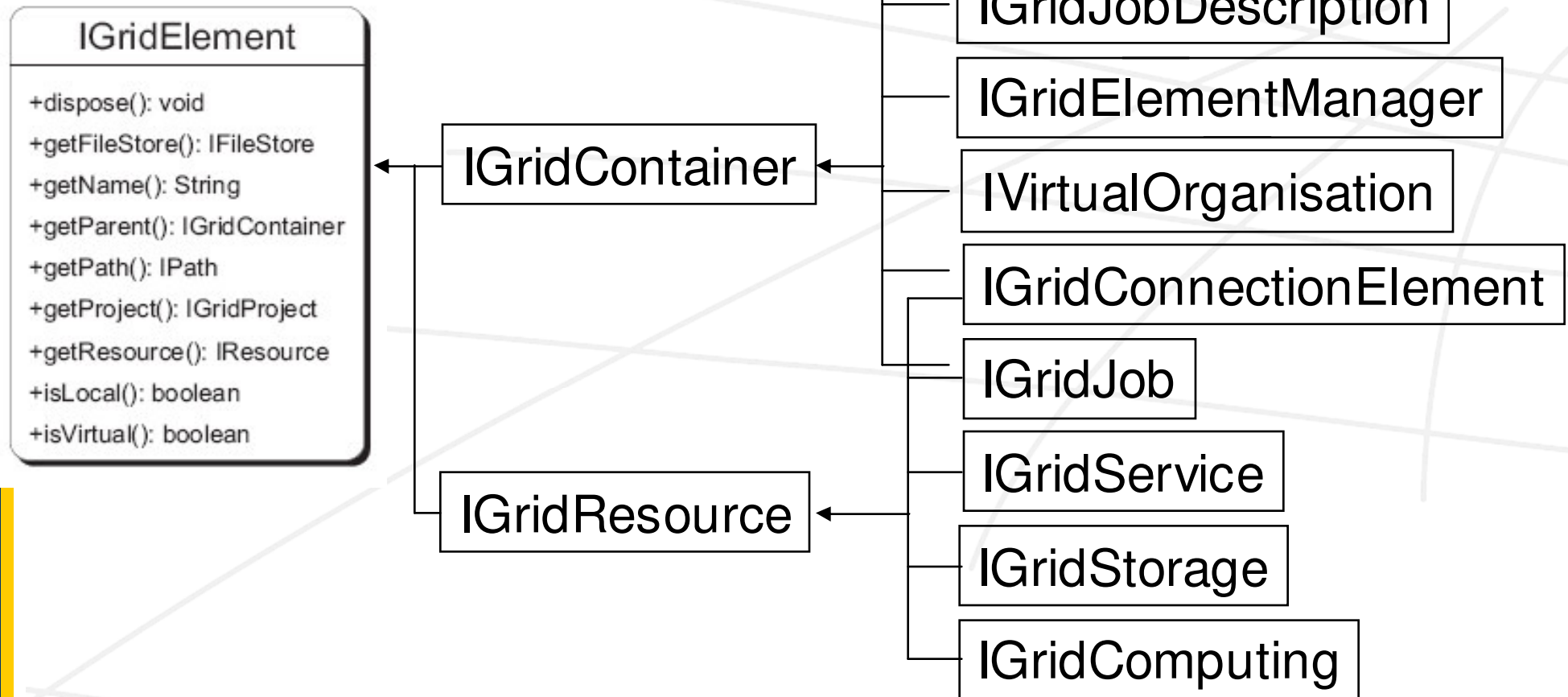
Grid project



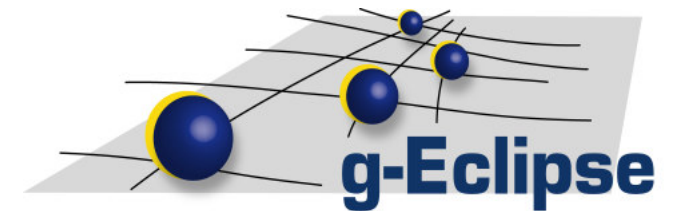
Grid Model



How to integrate the Grid into the Eclipse workbench?



Structuring the Grid



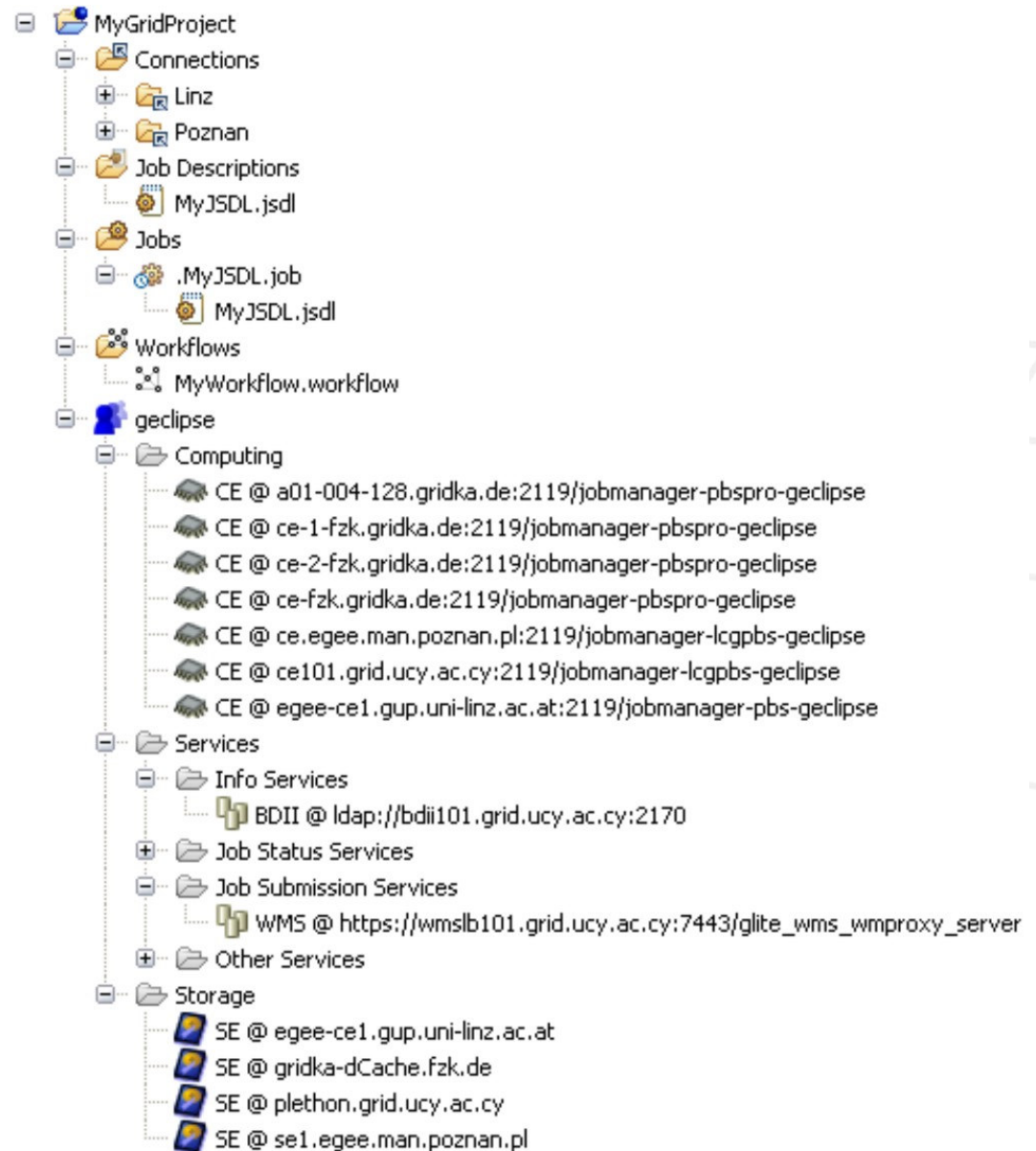
Mounted File Systems →

Virtual Organisation →

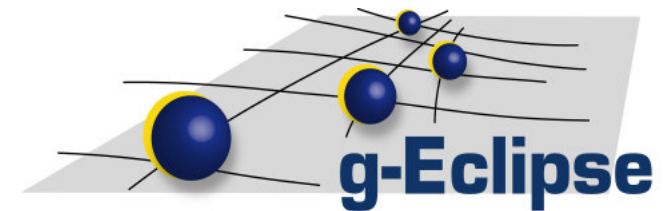
Computing Elements →

Services →

Storage Elements →



User perspective



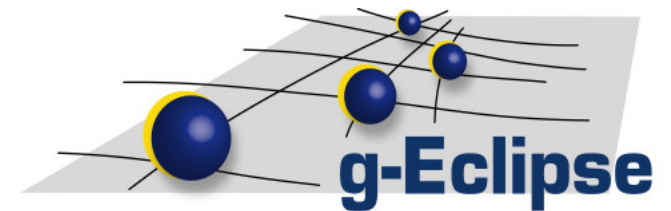
The screenshot shows the g-Eclipse IDE interface with several views and editors. The main workspace is divided into several panes:

- Grid Projects:** A tree view showing the project structure, including "MyFirstgEclipseProject" and "geclipse".
- Job Status View:** A view showing the status of jobs, with a list of jobs like ".osgitest.job" and ".osgitest[1].job".
- JSDL Editor:** An editor for editing JSDL documents, showing a list of elements to be manipulated.
- Resources Page:** A page for managing resources, showing a list of elements to be manipulated.
- Glue Info view:** A view showing information about computing elements, including a list of elements like "a01-004-128.gridka.de:2119/jobmanager-p".
- GGUS Web view:** A web browser view showing the GGUS website, including a navigation menu and a logo.
- Auth Token View:** A view showing authentication tokens, including a list of tokens like "Globus Proxy#01".

Callouts point to various components:

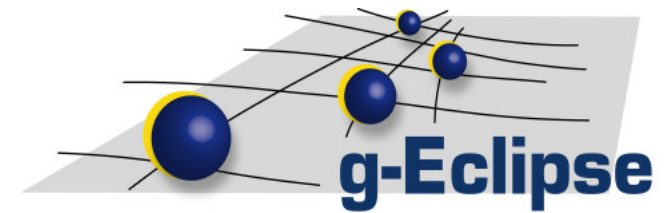
- Grid project view
- Data Connection
- Job Descriptions
- Jobs
- VO resources
- VO computing resources
- VO services
- VO storage resources
- Glue Info view
- GGUS Web view
- Auth Token View

Demo 2



- Just use EGEE
- Create a Grid project
 - Including a VO
- Access to data
 - Edit a file remotely
- Rent a machine in a few seconds
 - Login there

Grid Resource Provider



- How can a site A support a new VO with computing resources?

Set up a queuing system:

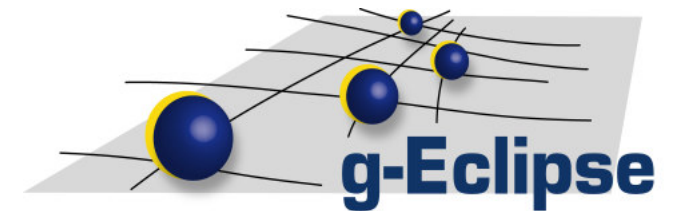
- Old: Know all the details of the queuing system

```
[root@ce201 root]# qmgr -c 'create queue test2 queue_type=executi
on' ; qmgr -c 'set queue test2 resources_max.walltime=48:00:00' ;
qmgr -c 'set queue test2 resources_max.cput=72:00:00' ; qmgr -c
'set queue test2 acl_group_enable=true' ; qmgr -c 'set queue test
2 acl_groups= +sec'
[root@ce201 root]#
```

- New: Configure the batch system on site with g-Eclipse

- Set up a VO specific queue
- Drain queues
- Manage Cluster nodes

Operator perspective



g-Eclipse (Operator) - huschel/batchService.batch - Eclipse Platform

File Edit Navigate Search Project Run Window Help

100%

Grid Projects

- huschel
 - Connections
 - Job Descriptions
 - Jobs
 - batchService.batch
 - geclipse
 - Computing
 - Services
 - Storage

Glue Information Vi

- Computing Elements
- Sites
- Storage Elements

batchService.batch

The diagram shows a central blue box representing the job configuration for "egee-ce1.gup.uni-linz.ac.at". It has the following properties: Type:pbs, Num. of Queues:5, Num. of WNs:4, and Num. of Jobs:5. Five arrows point from this central box to five worker nodes on the right, each in a yellow box: egee-wn1 (free), egee-wn2 (free), egee-wn3 (free), egee-wn4 (job-exclusive), and a fifth node (free). On the left, five green boxes represent the state of different sites: biomed (enabled, running), dteam (enabled, running), geclipse (enabled, running), ops (enabled, running), and voce (enabled, running). Dashed arrows point from each of these site boxes to the central job configuration box.

Terminal

```
SSH: hakor@cvs.fzk.de
hakor@savannah:~$ _
on Mar 26 21:49:04 UTC 2007 1686

The programs included with the Debian
system are free software;
the exact distribution terms for each
re described in the
individual files in /usr/share/doc/*/*c

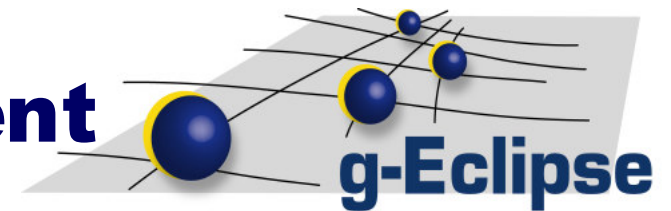
Debian GNU/Linux comes with ABSOLUTELY
NTY, to the extent
permitted by applicable law.
Last login: Thu Jan 24 11:41:54 2008 f
tz.it.necclab.eu
hakor@savannah:~$
```

Properties

Property	Value
Kernel:	2.6.9-67.EL.cernsmp
Name:	egee-wn2.gup.uni-linz.ac.at
Num. of CPU:	2
RAM:	514432kb
Running jobs	0
State:	free

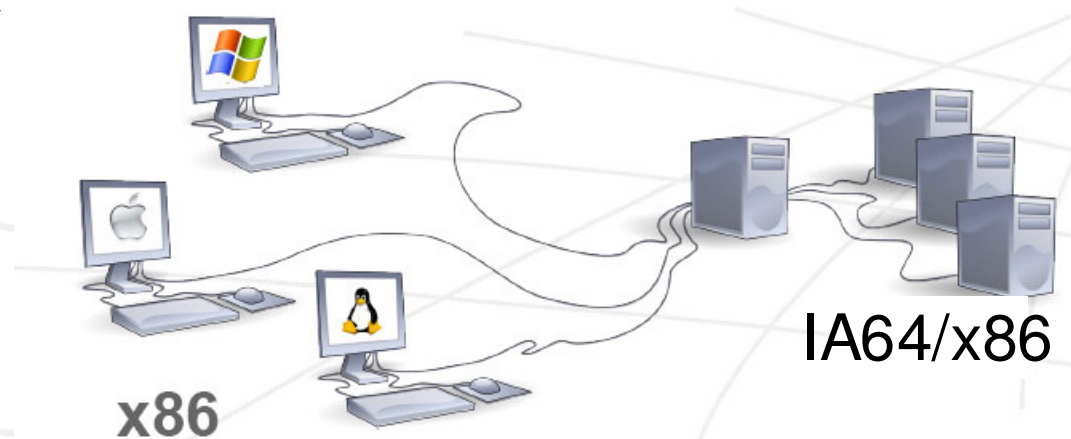
start | eclipse | g-Eclipse (Operator) - ... | g-Eclipse - Access the... | 6:28 PM

Grid Application development



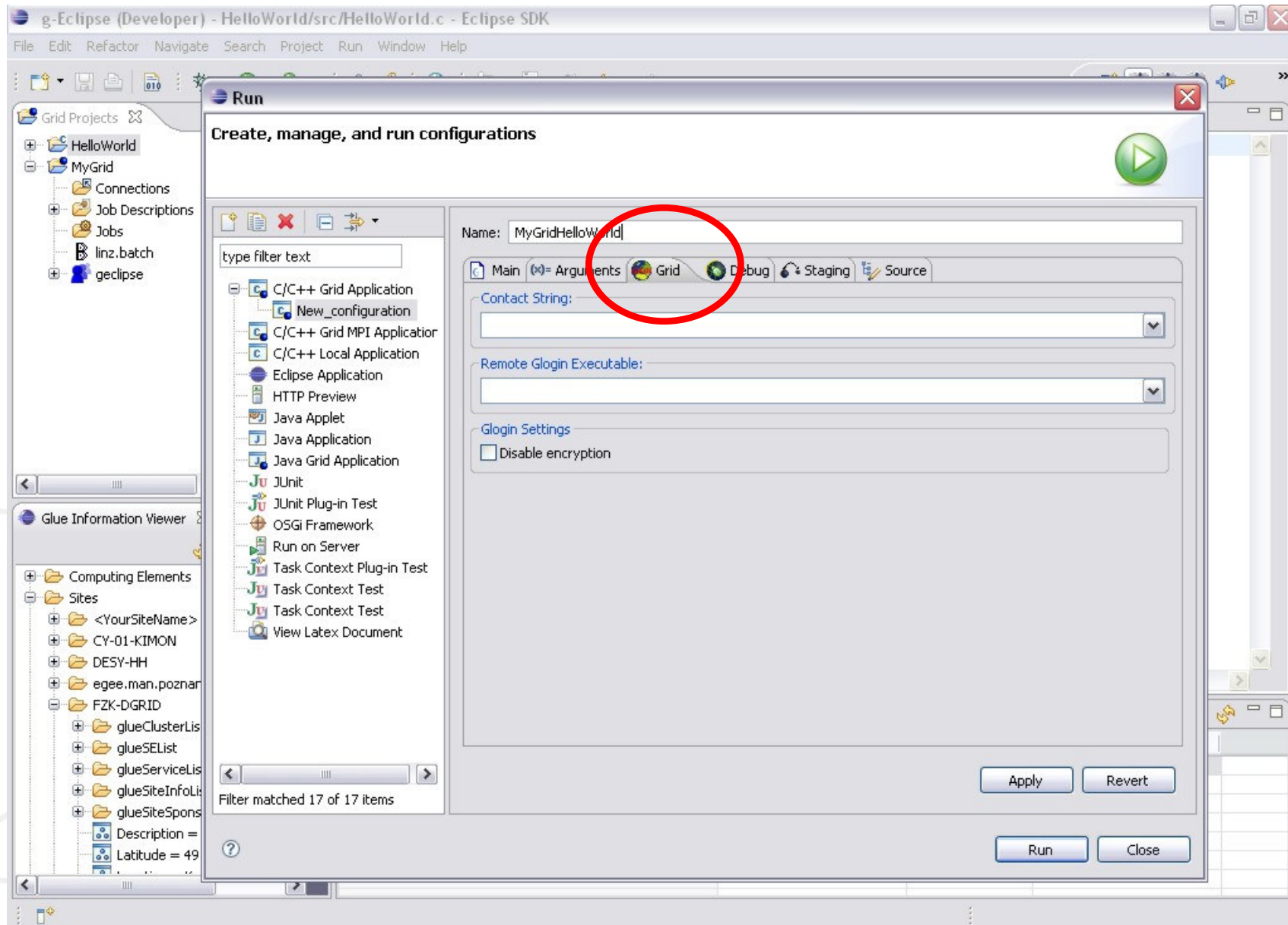
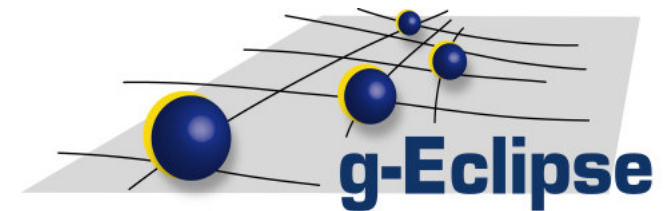
How to “gridify” a Legacy application?

1. Develop them on your local computer as a separate JDT/CDT project
2. After a code change, compile them locally **and on a remote Grid resource**
3. (if needed debug them locally or on a remote Grid site)
4. Deploy the application

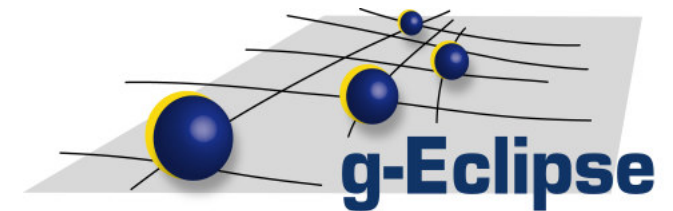


Usage of gLogin introduce some firewall issues!!!

Developer perspective



Developer perspective II



The screenshot displays the Eclipse IDE in the Developer perspective, configured for debugging. The interface is divided into several panes:

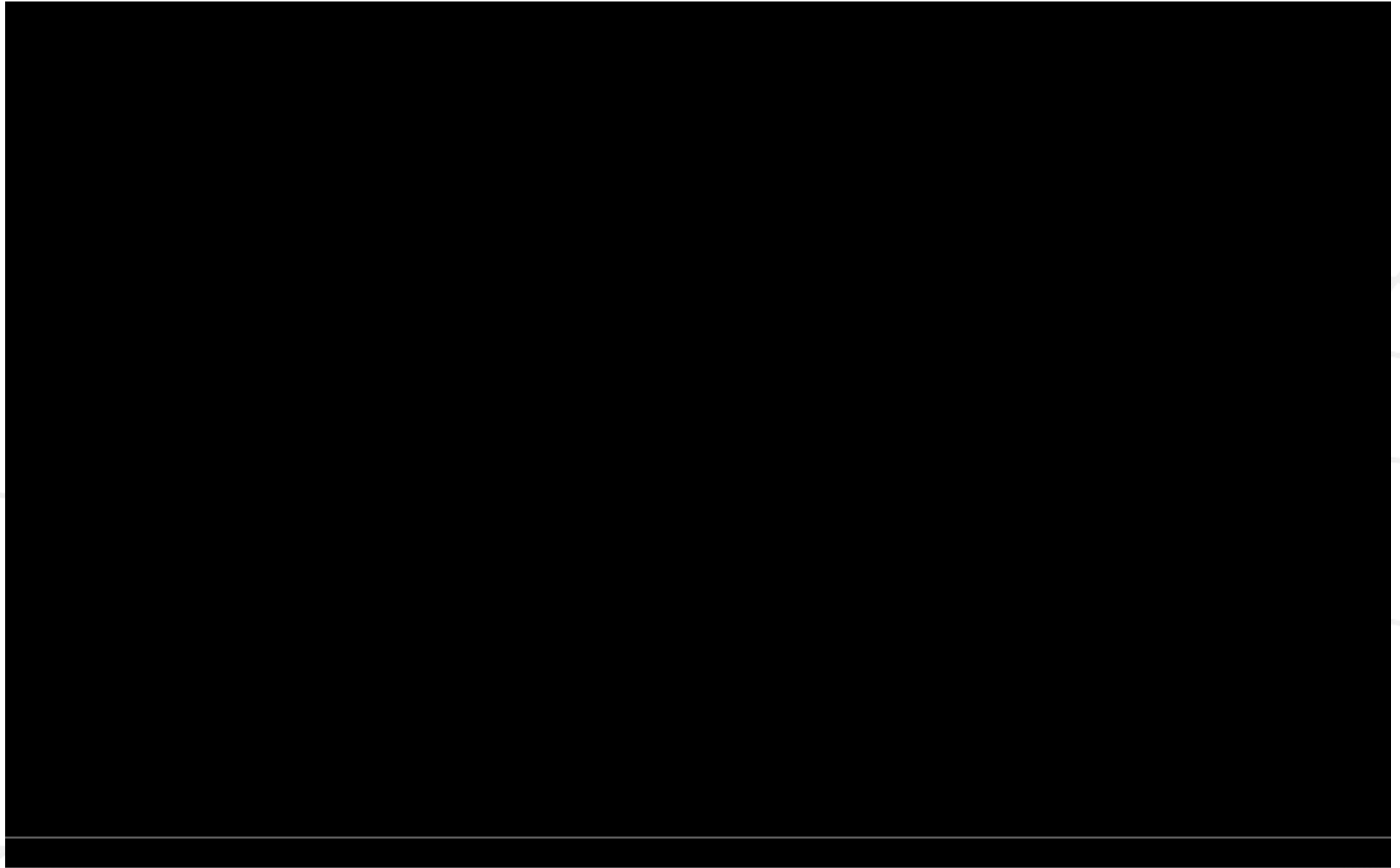
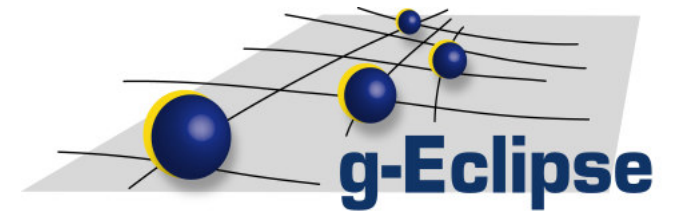
- Debug Console:** Located at the top left, it shows the current debugging session. A red circle highlights the toolbar icons for stepping through code (Step Over, Step Into, Step Return, Step Out) and the Run button.
- Variables View:** Located at the top right, it displays the current state of variables. The variable `i` is highlighted in yellow, with a value of `1`.
- Code Editor:** The central pane shows the source code of `simple2.c`. The current line of execution is `printf("%i\n", i);`, which is highlighted in blue. The code is as follows:

```
int main(int argc, char **argv)
{
    int i;

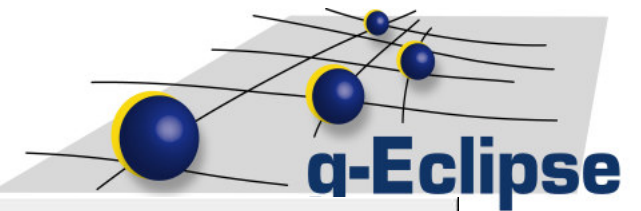
    for(i=0; i<10; i++){
        printf("%i\n", i);
    }

    return 0;
}
```
- Outline View:** Located at the bottom right, it shows the project structure, including `stdio.h` and `main`.
- Console:** Located at the bottom, it displays the output of the program, showing the path and time of execution: `simple2 [C/C++ Grid Application] /home/koe/runtime-New_configuration/simple2/Debug/simple2 (1/4/07 2:55 PM)`.

Visualisation



Visualisation



File Edit Navigate Search Project Run SLA Menu Diagram Services Samples Window Help

g-Eclipse (...)

Grid Projects

- myProject
 - Connections
 - Job Descriptions
 - Jobs
 - Visualisation
 - FromTemplate.vtk.pipeline
 - exProtein.pharma
 - headsqExample.vtk.pipeline
 - gedipse

exProtein.pharma

Connections Authentication Tokens Jobs Visualisation

Chain(alpha-thrombin (small subunit))
 H0B3A5H(alpha-thrombin (large subunit))
 Chain(hirugen)

Selection
 Alz600
 C20

Coloring: Disulfide
 Disulfide
 no annotation

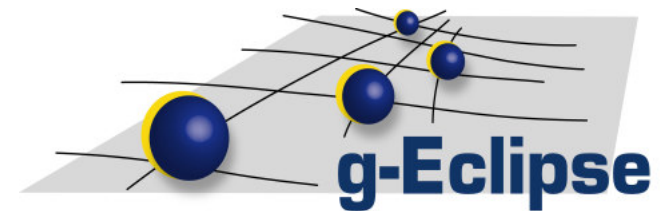
Property	Value
Disulfide	
conflict_1	
conflict_2	
variant_1	
variant_2	
act_site	
disulfid_1	
disulfid_2	
disulfid_3	
peptide	
propsep	
signal	
chain_1	
chain_2	
carbohydr	
site	
domain	
region	
mod_res	
uniprot P00734 P0073...	
1AFE-L	
Disulfide	
conflict_1	
conflict_2	
variant_1	
variant_2	
act_site	
disulfid_1	
disulfid_2	
disulfid_3	
peptide	
propsep	
signal	
chain_1	
chain_2	
carbohydr	
site	
domain	
region	
mod_res	
uniprot P00734 P0073...	
1AFE-H	

CYVAGKPGDFGYCDLNYCEEAVEEETGDGLDESDRAIEGRTATSEYQTFNPRTFGSGEADGLRPLFEKKSLEDK
 TFGSGEADGLRPLFEKKSLEDK

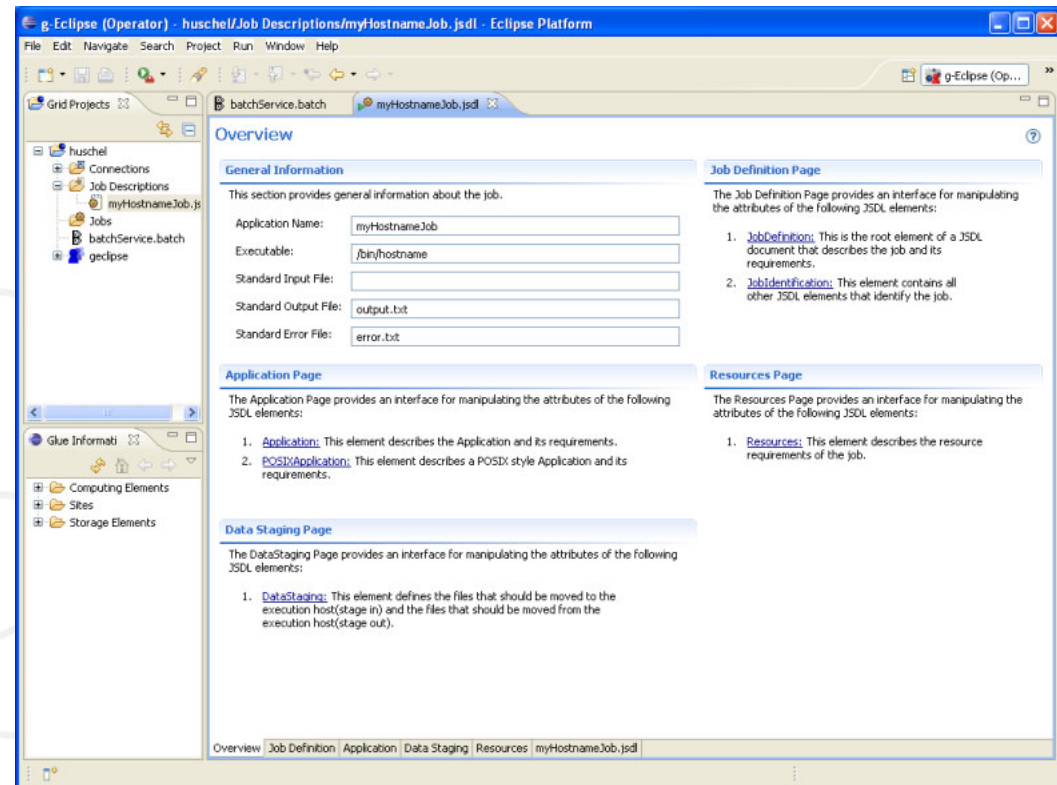
conflict_2 CONFLICT

SRS 3D Viewer

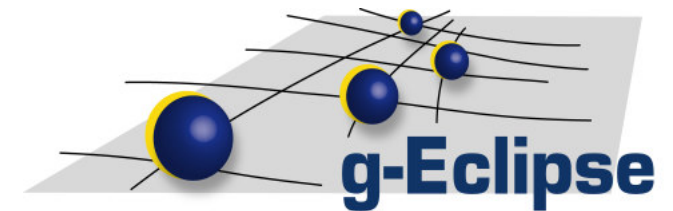
Standards



- JSDL editor
 - Multitpage editor following the OGF JSDL standard
 - Submission to different middleware possible
 - gLite:
XSLT transformation
- GLUE schema browser
 - Browse through your resources
- Eclipse is based on OSGi
 - Enables dynamic code deployment

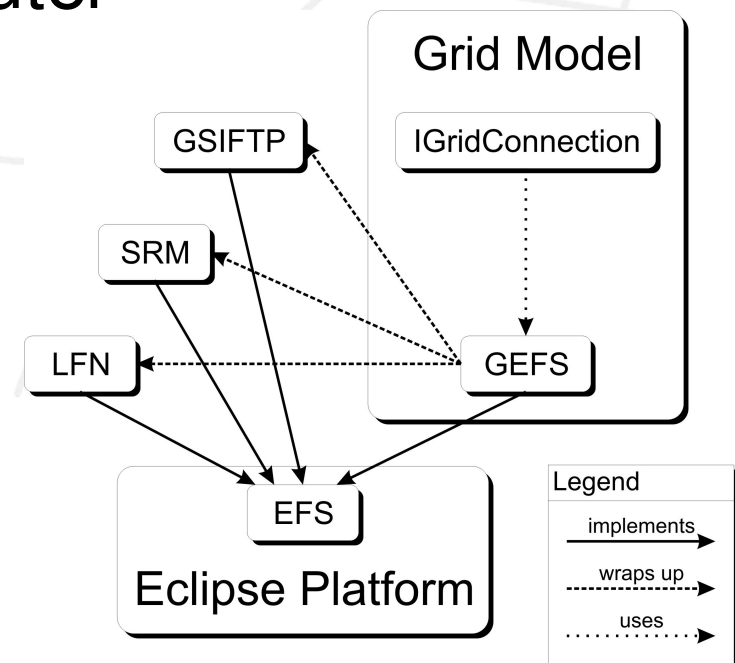


Interoperability

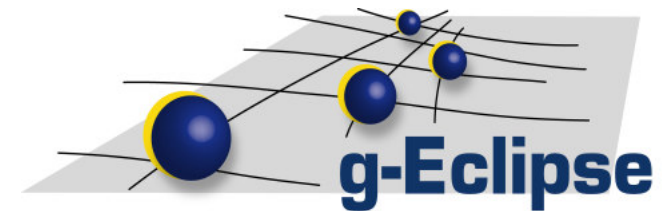


- g-Eclipse workspace can contain project with different Grid flavors
- g-Eclipse Authentication framework manage the “single sign on” on request
- g-Eclipse enables transfer from/to different Grid flavors and from/to local computer

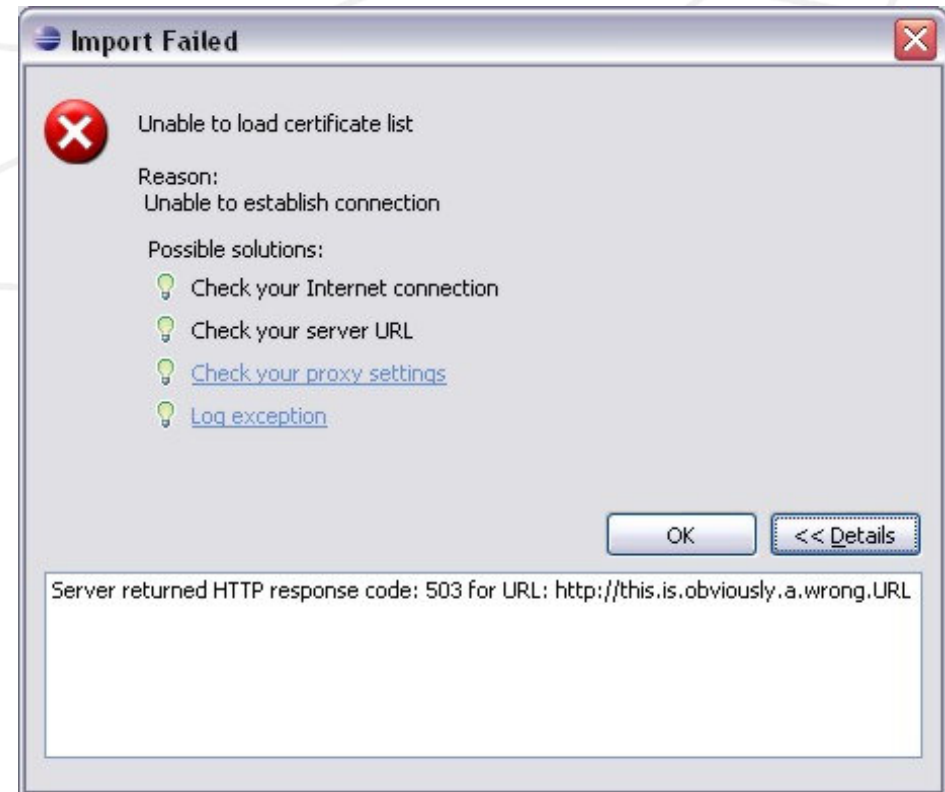
- By using the EFS (Eclipse Files System) implementation



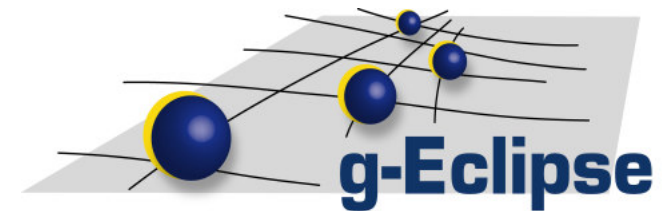
Manage Complexity



- By providing solution to common problems on Grid infrastructures
 - g-Eclipse provides an extended problem reporting mechanism based on the Eclipse core exception
 - Problems have associated solutions
 - Solutions may be
 - passive: just a descriptive text
 - active: provide an action that helps the user to solve the problem, e.g. open an associated preference page



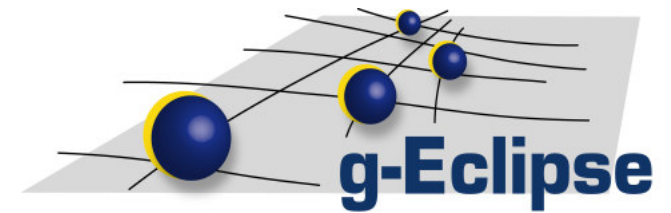
Current project status



- g-Eclipse is an official Eclipse Technology Project
- First release (0.5) available since September 2007
- Currently working in 1.0 milestone phase (RC3 available)
- Final 1.0 release planned for December 2008

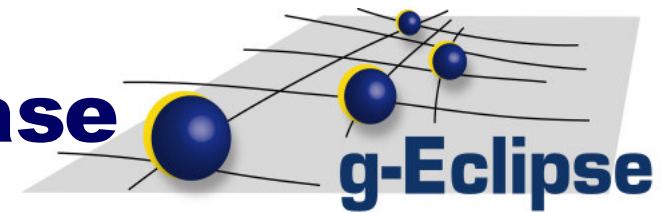
- Architecture and Grid model stable since mid of 2007
- First middleware implementation (gLite) stable since end of 2007
- Second middleware implementation (GRIA) started in the beginning of 2008
- Third middleware implementation (AWS) started in Spring 2008

User communities



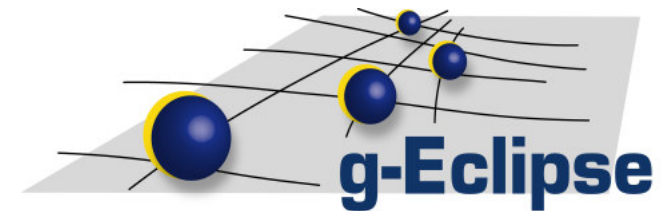
- Interest from many other project
 - Contact with China for GT4 implementation
- g-Eclipse will be used for university lecture in Romania
- Collaboration with other EC projects
 - i.e. SIMDAT, EGEE, DORII, IS-ENES, ...
- Collaboration with other Eclipse projects
 - Parallel Tool Platform (support for MPI on HPC resources)
 - SOA Tool Platform
 - Swordfish Runtime

Short summary for 1.0 release



- Finish second middleware implementation
 - Prove of middleware independent conception
- Implement some first industry relevant applications on top of g-Eclipse
 - BAE application for fluid dynamics
 - Pharmaceutical application by NEC
- Provide an interface for Amazon's EC2 and S3
 - Be not only middleware-independent but also Grid-independent
 - Introduce the world of cloud computing in Eclipse

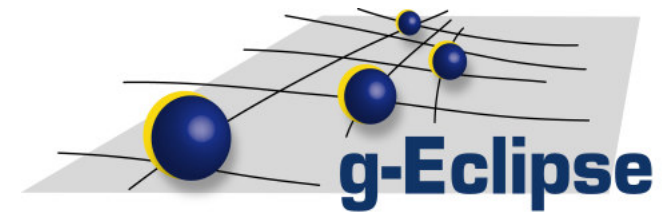
Contribute



- Use our tool and send us feedback!
 - We do it the Eclipse way!
 - Webpage www.eclipse.org/geclipse or www.geclipse.eu
 - Newsgroup
 - <http://dev.eclipse.org/newslists/news.eclipse.technology.g-eclipse/>
 - Developer mailing list
 - <https://dev.eclipse.org/mailman/listinfo/geclipse-dev>
 - Bugzilla
 - <https://bugs.eclipse.org/bugs>

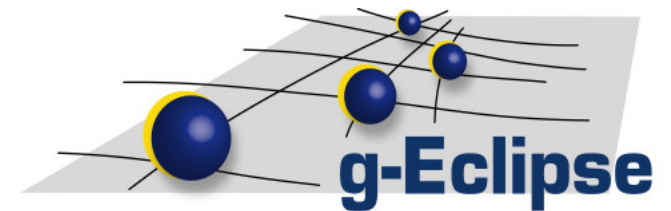
- Bring your application!!!!
 - Contact {at} geclipse.eu

Outlook – g-Eclipse



- g-Eclipse has the potential to become as fundamental on the Grid scientist's desktop as the web browser was for the internet
- By making no difference between academic Grids and commercial Clouds, g-Eclipse is the tools for **Groud** computing
(Grid + Cloud = Groud)
- With the help of the 1919 Eclipse the theory of relativity was verified, g-Eclipse will help to prove ...???

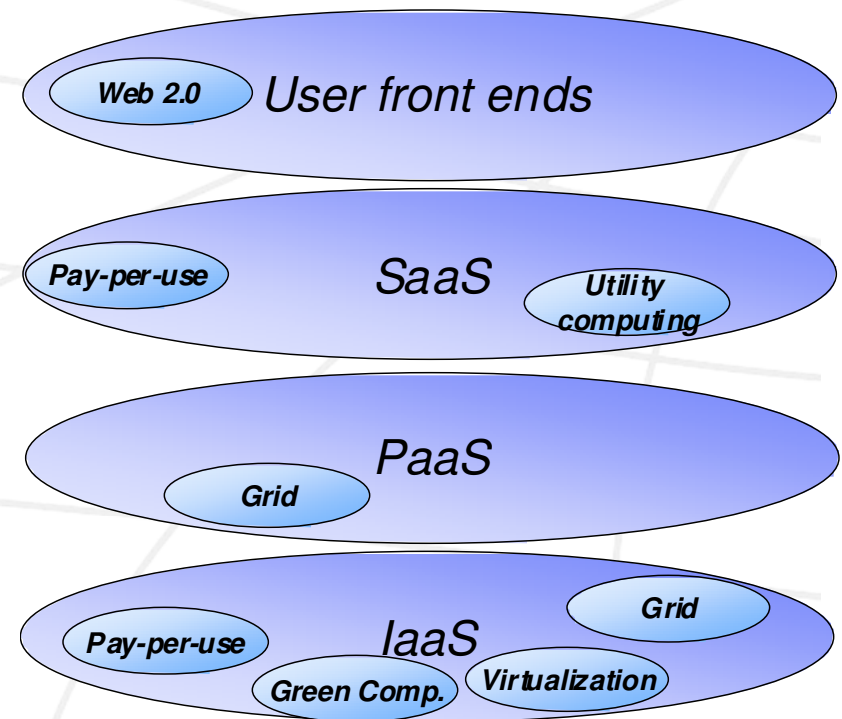
Outlook – Grid in general



- How will Grids look in the 64-Core-CPU time?
 - Service instead of Batch!
 - 10000 foot view

- The challenge will be the management on different levels in the XaaS world (XaaS = Everything as a Service)

- Cloud/Utility will be based on Virtualization with a lot of Grid behind the scene.



→ The Heaven starts beyond the Cloud

Empowered by Innovation

NEC