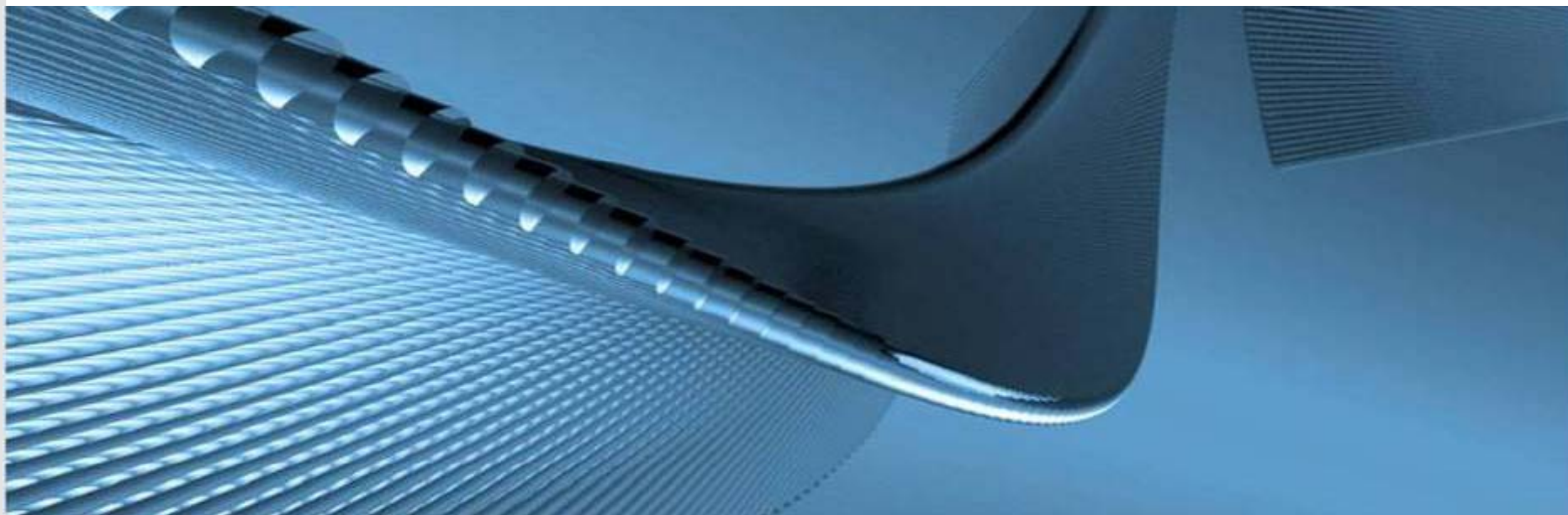


# Combining Cloud and Grid with a User Interface

Jie Tao  
Karlsruhe Institute of Technology  
[jie.tao@kit.edu](mailto:jie.tao@kit.edu)



# Outline

- Motivation
- The g-Eclipse Project
- Extending g-Eclipse for a Cloud Framework
- Initial Implementation: Accessing the Amazon Web Service
- Conclusion

# Motivation

- Various clouds co-exist
  - Amazon EC2, S3
  - Google App Engine
  - Microsoft Azure
  - Eucalyptus, Cumulus, Nimbus, OpenNybula
- Different user interfaces (GUI or command-line)
  - Working with several clouds: must know each interface
  - Interaction between clouds: not easy
- Moving grid to cloud
  - Running grid jobs on the cloud
  - Grid as a Service (RightScale)
- Goal: developing a generic, intuitive cloud user interface

# g-Eclipse: A General Framework for Accessing the Grid

- Many application domains start using Grid infrastructures
- But...
  - Grid technology is complex
    - Different systems are used
      - Middleware (Unicore, gLite, Globus, GRIA, ...)
      - Many separate tools (i.e for installation, monitoring, ...)
    - Different programming paradigms
      - Batch type systems vs. service oriented systems
      - Many programming languages
- e-Users want to interact with the grid infrastructure
  - Without knowing all details (development, deployment, testing, management, ...)
- → **Tooling is necessary!!**
  - **Wizards, Editors, ...**
  - **Hide the complexity!!**

# g-Eclipse: the Idea

- Provides a general UI framework/eco system
  - Designed for application users, resource providers, and application developers
  - Middleware independent core infrastructure + middleware extension
  - Allowing users to access the grid in a simple way
    - File transfer: drag&drop
    - Job submission: mouse click
- (Re-)use Eclipse and contribute
  - Eclipse is an eco system
    - Build for extension
    - More than a JAVA IDE
    - The biggest “coordinated” Open Source project
  - Gain OS independence (by using JAVA)

# g-Eclipse: grid user roles & use cases

- 3 different roles

- User
- Operator
- Developer

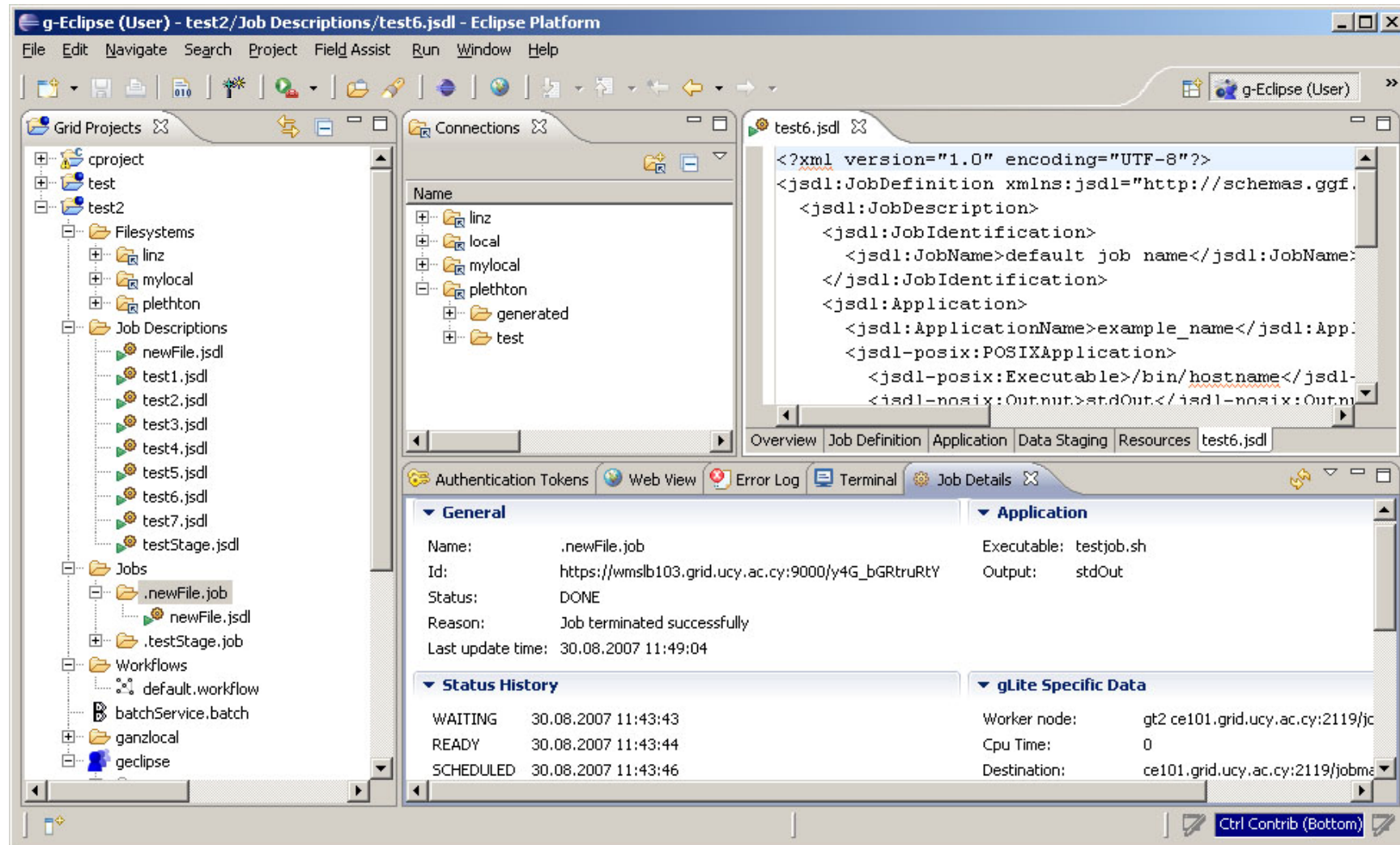
- In general...

- Job management
- Resource management
  - Files
  - Applications
  - Hardware
- Application deployment
- Infrastructure monitoring
- Application development
- Visualization tools





# g-Eclipse: screenshot



The screenshot displays the g-Eclipse IDE interface. The main window title is "g-Eclipse (User) - test2/Job Descriptions/test6.jsdl - Eclipse Platform". The interface includes a menu bar (File, Edit, Navigate, Search, Project, Field Assist, Run, Window, Help) and a toolbar. The left sidebar shows a project tree with folders like "Filesystems", "Jobs", and "Workflows", and files like "test1.jsdl" through "test7.jsdl". The "Connections" view shows a tree of file systems including "linz", "local", "mylocal", "plethton", and "test". The main editor displays the XML content of "test6.jsdl":

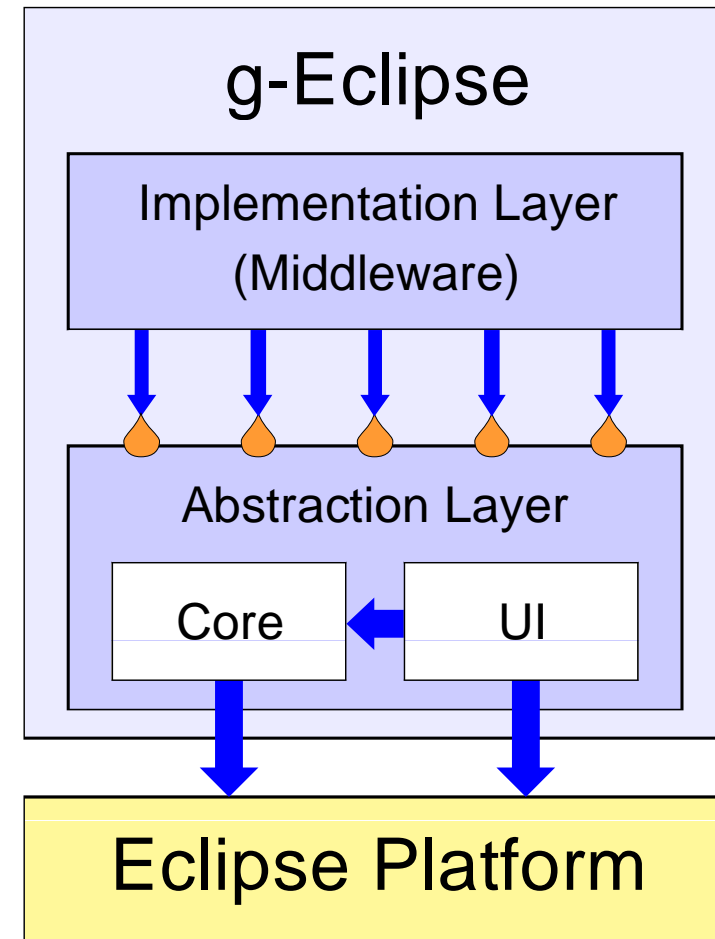
```
<?xml version="1.0" encoding="UTF-8"?>
<jsd1:JobDefinition xmlns:jsdl="http://schemas.ggf.
<jsd1:JobDescription>
  <jsd1:JobIdentification>
    <jsd1:JobName>default job name</jsdl:JobName>
  </jsdl:JobIdentification>
  <jsd1:Application>
    <jsd1:ApplicationName>example_name</jsdl:App
    <jsd1-posix:POSIXApplication>
      <jsd1-posix:Executable>/bin/hostname</jsdl-
      <jsdl-posix:Output>stdout</jsdl-posix:Outm
```

Below the editor, the "Job Details" panel is visible, showing information for a job named ".newFile.job".

General		Application	
Name:	.newFile.job	Executable:	testjob.sh
Id:	https://wmslb103.grid.ucy.ac.cy:9000/y4G_bGRtruRtY	Output:	stdout
Status:	DONE		
Reason:	Job terminated successfully		
Last update time:	30.08.2007 11:49:04		
Status History		gLite Specific Data	
WAITING	30.08.2007 11:43:43	Worker node:	gt2 ce101.grid.ucy.ac.cy:2119/jc
READY	30.08.2007 11:43:44	Cpu Time:	0
SCHEDULED	30.08.2007 11:43:46	Destination:	ce101.grid.ucy.ac.cy:2119/jobme

# g-Eclipse: architecture overview

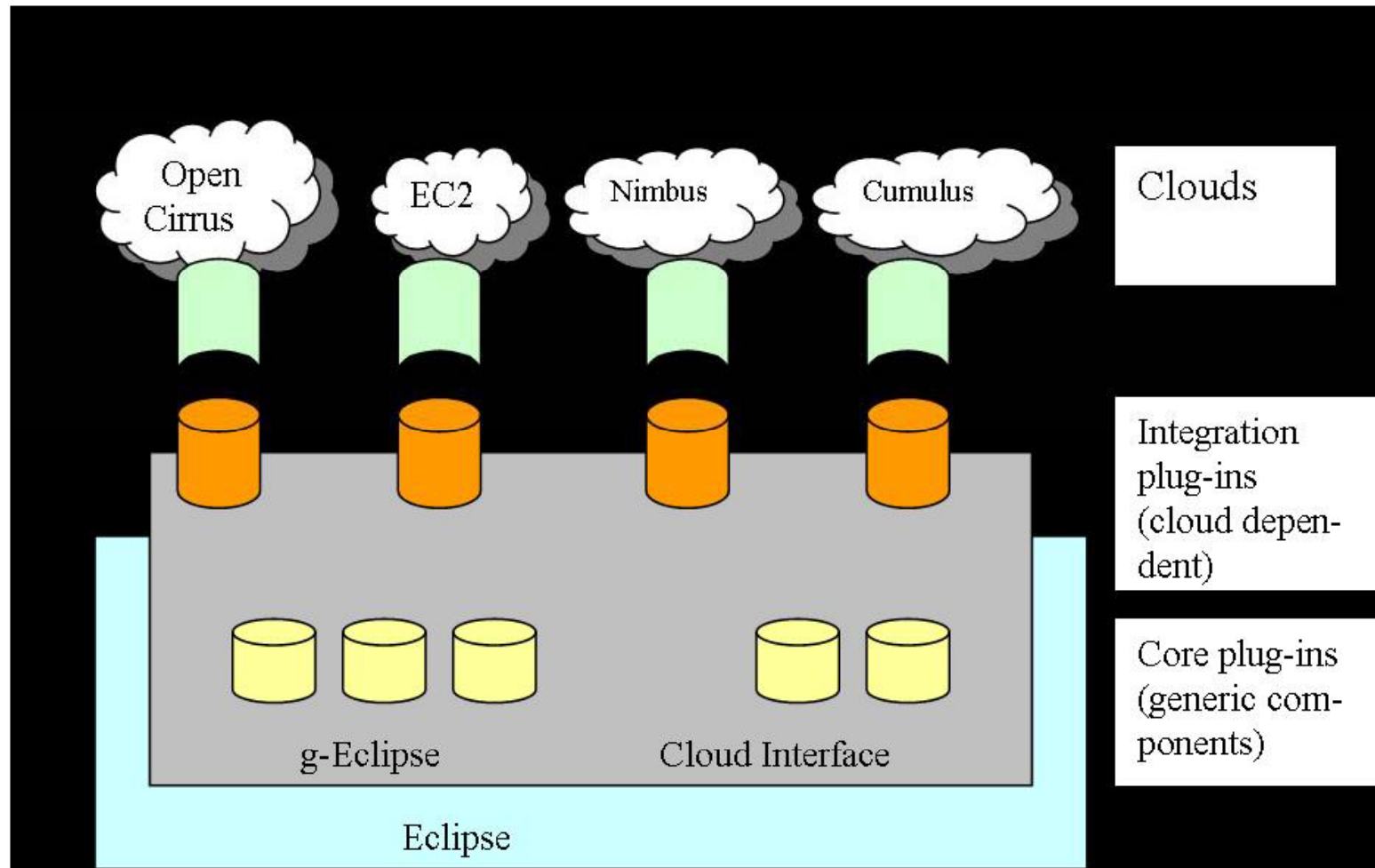
- Abstraction Layer
  - Core functionalities, e.g.
    - Authentication/Authorization
    - VO management
    - Data management
    - Job submission
  - Common user interface, e.g.
    - Views
    - Wizards
    - Dialogs
    - Preference pages
- Implementation Layer
  - Extended core functionalities
  - Middleware specific functionalities



 Eclipse Extension Point



# Extending g-Eclipse for Clouds

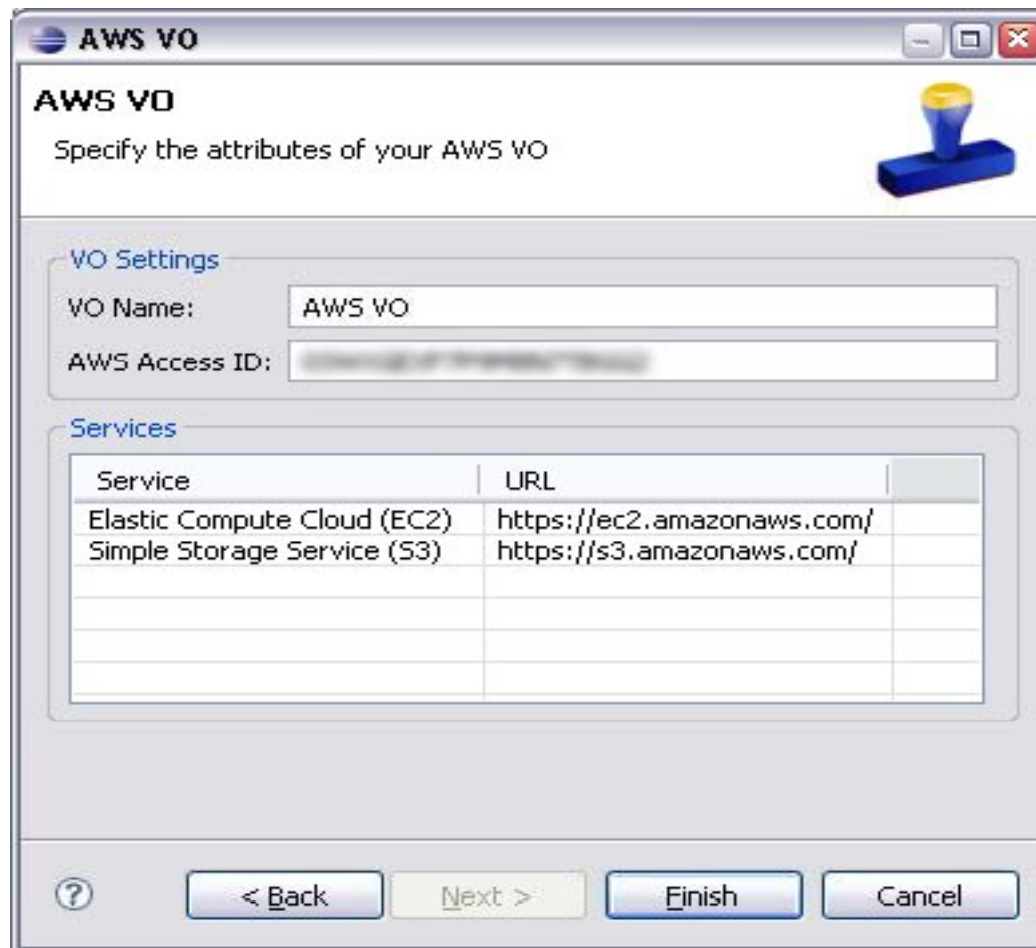


# Extending g-Eclipse for Clouds (cont.)

- An additional folder: service description
  - g-Eclipse core
    - Common interfaces for cloud service
    - Extension of VO management
  - g-Eclipse UI:
    - Multi-layer editor for service specification, supporting
      - HaaS, SaaS, HPC as a Service, Storage as a Service, Grid as a Service
    - Actions: service request, service execution, service termination
    - Views: available services, attributes, ...
- Templates for application development (Map/Reduce, web service...)
- Middleware extension
  - Specific implementation for individual clouds
    - Wizards and views for authentication token, service deployment/execution/termination, ....

# Initial Implementation: Accessing the Amazon Web Service (I)

- AWS VO implementation



The screenshot shows a window titled "AWS VO" with a subtitle "Specify the attributes of your AWS VO". The window is divided into two main sections: "VO Settings" and "Services".

**VO Settings**

VO Name:

AWS Access ID:

**Services**

Service	URL
Elastic Compute Cloud (EC2)	https://ec2.amazonaws.com/
Simple Storage Service (S3)	https://s3.amazonaws.com/

At the bottom of the window, there are four buttons: a help button (question mark), a "< Back" button, a "Next >" button, and a "Cancel" button.

# Initial Implementation: Accessing the Amazon Web Service (II)

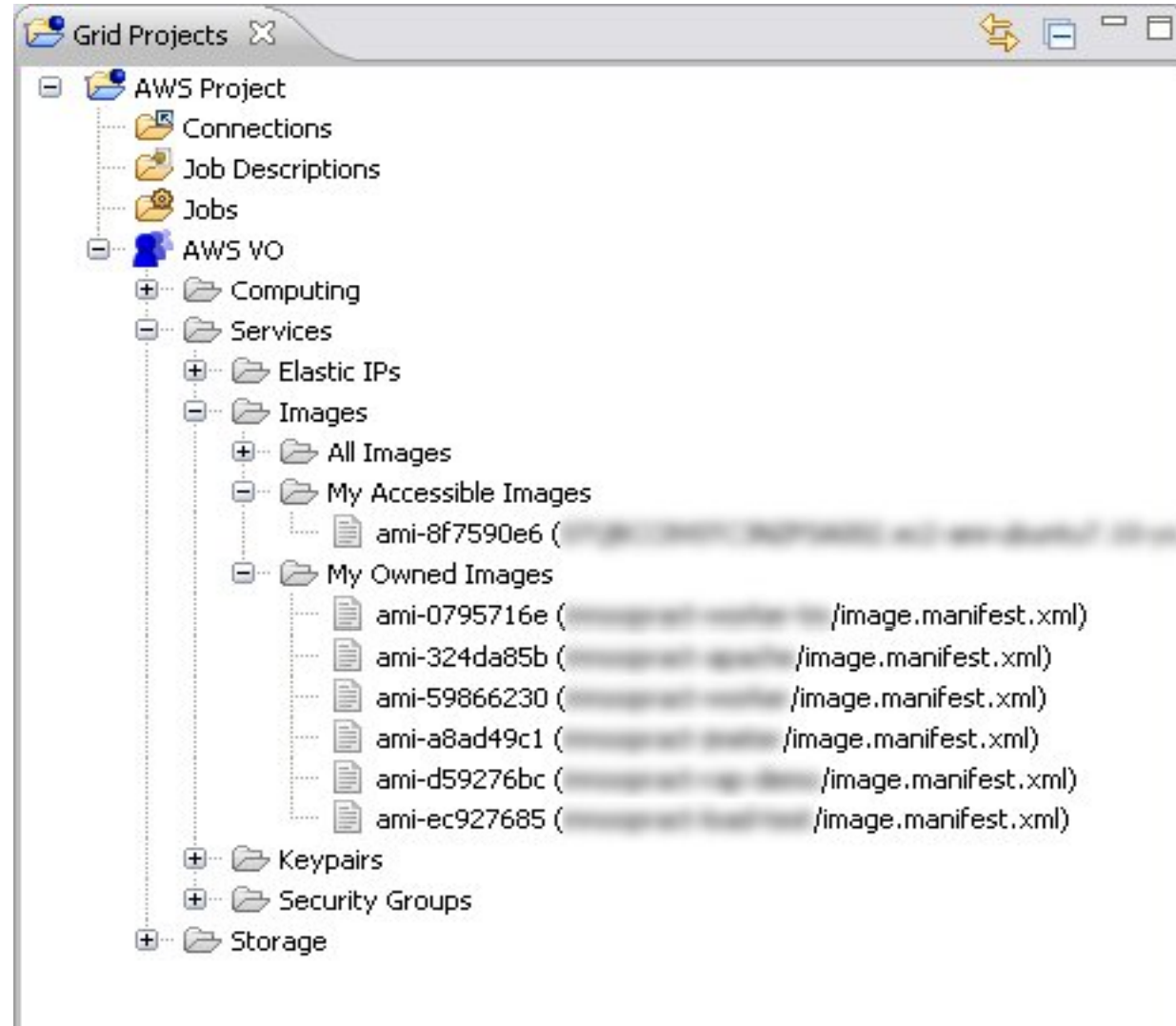
- Authentication token for AWS credential



The screenshot shows a Windows-style dialog box titled "Authentication Token". The main heading is "Create new authentication token" with a sub-instruction "Provide authentication credentials" and a padlock icon. Below this is a section titled "Amazon Web Service Credentials" containing two input fields: "Access ID:" and "Secret ID:". At the bottom, there are four buttons: a help button (question mark), a "< Back" button, a "Next >" button, and an "Finish" button. A "Cancel" button is also present at the bottom right.

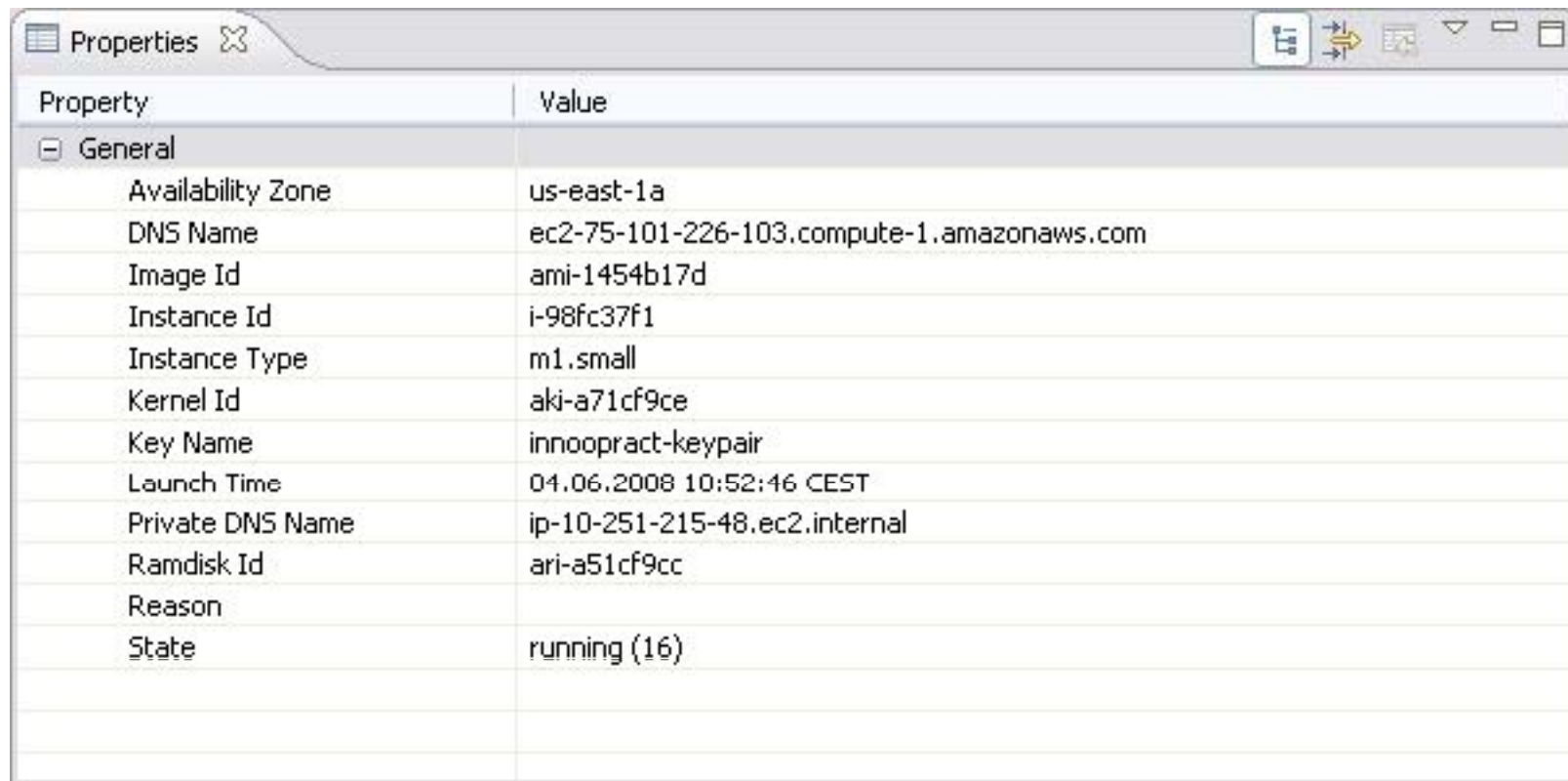
# Initial Implementation: Accessing the Amazon Web Service (III)

- Project view



# Initial Implementation: Accessing the Amazon Web Service (IV)

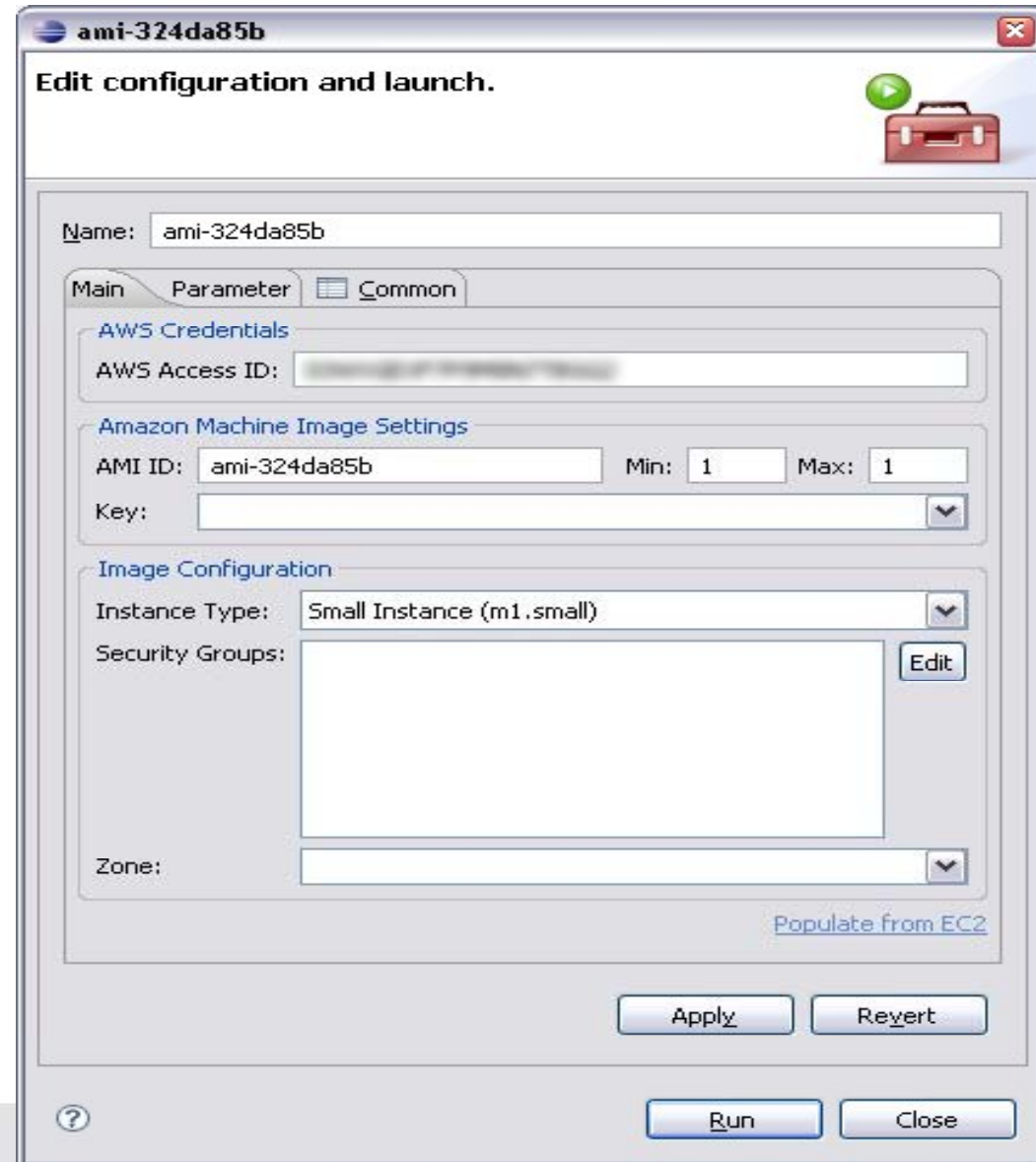
- AMI properties



Property	Value
General	
Availability Zone	us-east-1a
DNS Name	ec2-75-101-226-103.compute-1.amazonaws.com
Image Id	ami-1454b17d
Instance Id	i-98fc37f1
Instance Type	m1.small
Kernel Id	aki-a71cf9ce
Key Name	innoopract-keypair
Launch Time	04.06.2008 10:52:46 CEST
Private DNS Name	ip-10-251-215-48.ec2.internal
Ramdisk Id	ari-a51cf9cc
Reason	
State	running (16)

# Initial Implementation: Accessing the Amazon Web Service (V)

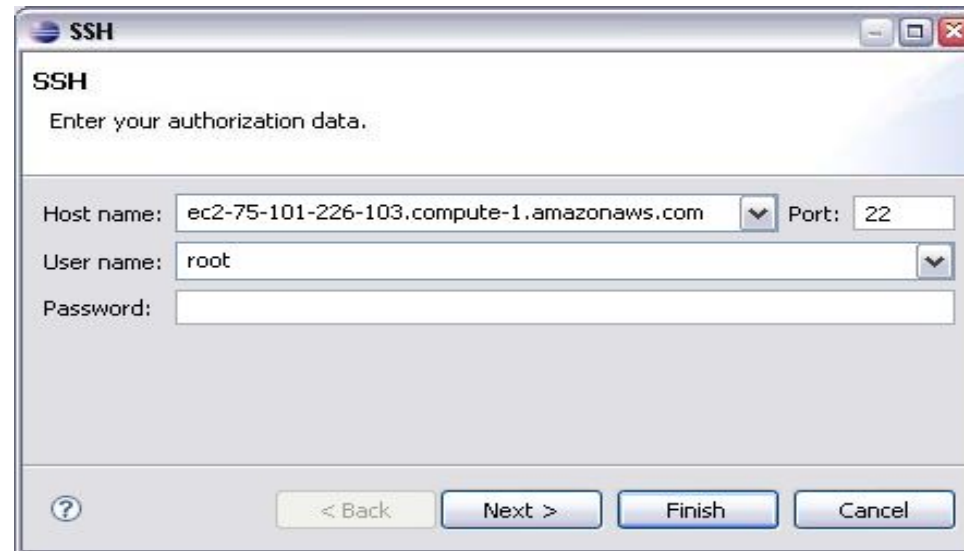
- Service execution: launching an AMI





# Initial Implementation: Accessing the Amazon Web Service (IV)

- Remote login



SSH

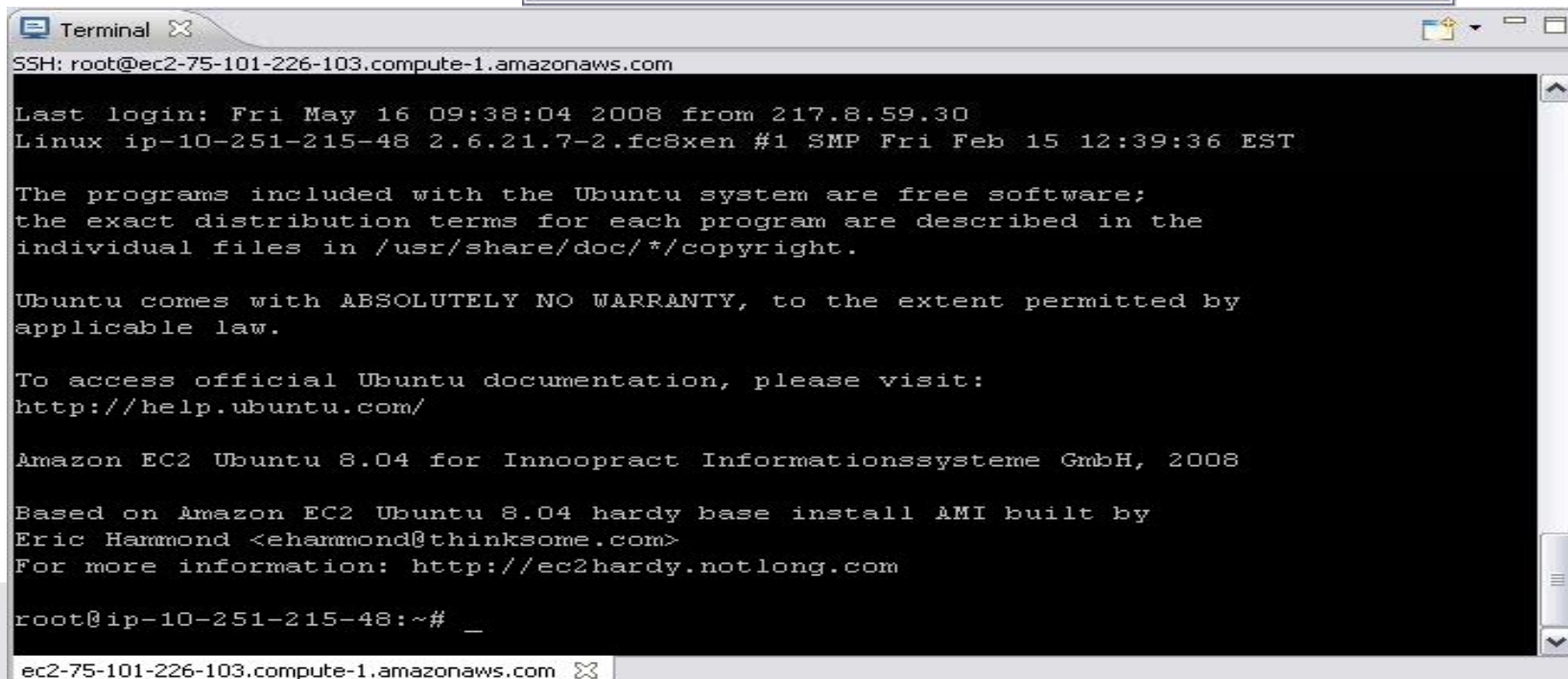
Enter your authorization data.

Host name:  Port:

User name:

Password:

< Back Next > Finish Cancel



```
SSH: root@ec2-75-101-226-103.compute-1.amazonaws.com
Last login: Fri May 16 09:38:04 2008 from 217.8.59.30
Linux ip-10-251-215-48 2.6.21.7-2.fc8xen #1 SMP Fri Feb 15 12:39:36 EST

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/

Amazon EC2 Ubuntu 8.04 for Innoopract Informationssysteme GmbH, 2008

Based on Amazon EC2 Ubuntu 8.04 hardy base install AMI built by
Eric Hammond <ehammond@thinksome.com>
For more information: http://ec2hardy.notlong.com

root@ip-10-251-215-48:~# _
```

# Conclusion

- g-Eclipse is a generic framework for simplifying the access to the grids
- Extending g-Eclipse for accessing the clouds
  - An easy way to interact with computing clouds
  - A bridge across clouds and grids
- Core extensions and specific implementation for AWS and Eucalyptus
- Further work
  - Full functionality of the cloud user interface
  - Connecting more clouds