Karlsruhe Institute of Technology

Combining Cloud and Grid with a User Interface

Jie Tao
Karlsruhe Institute of Technology
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

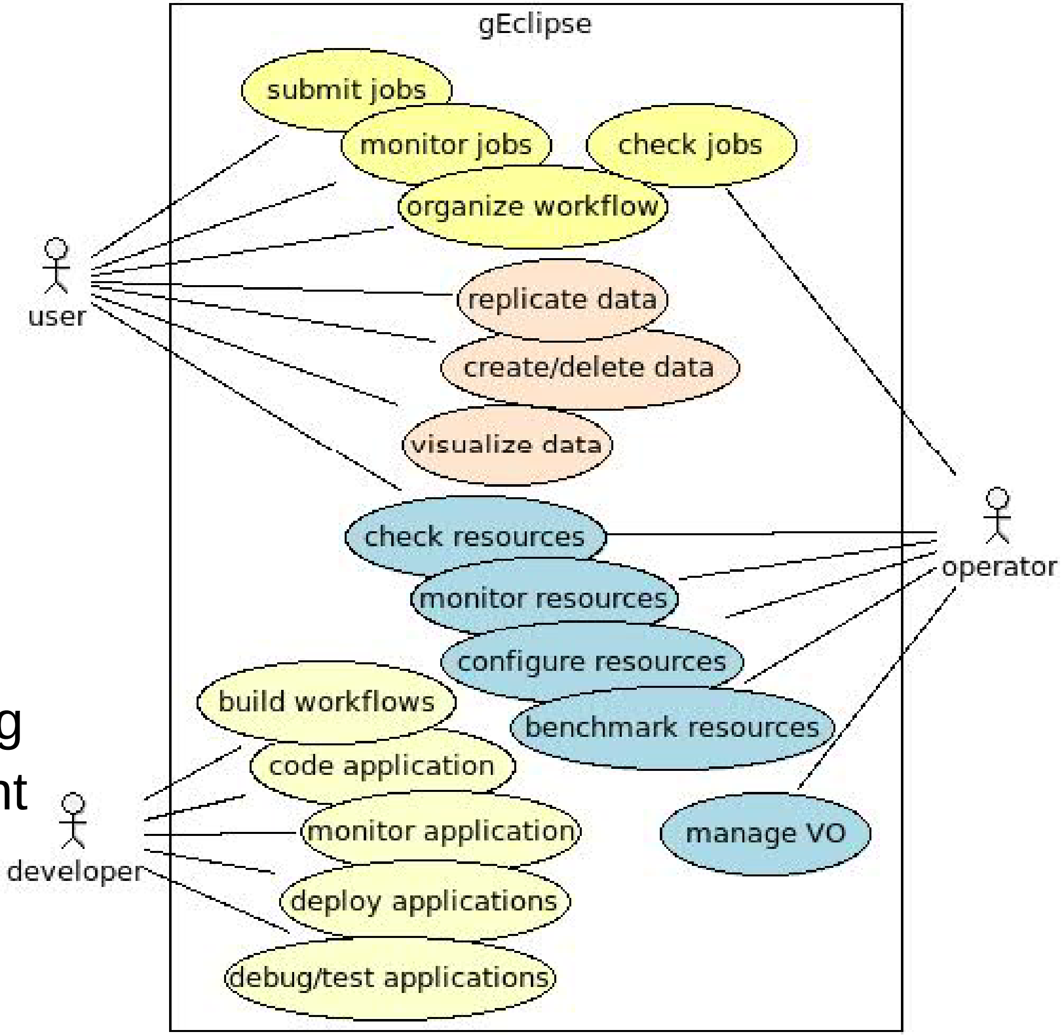
- Many application domains start using Grid infrastructures
- But…
  - □ Grid technology is complex
    - Different systems are used
      - □ Middleware (Unicore, gLite, Globus, GRIA, …)
      - □ Many separate tools (i.e for installation, monitoring, …)
  - □ Different programming paradigms
    - Batch 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

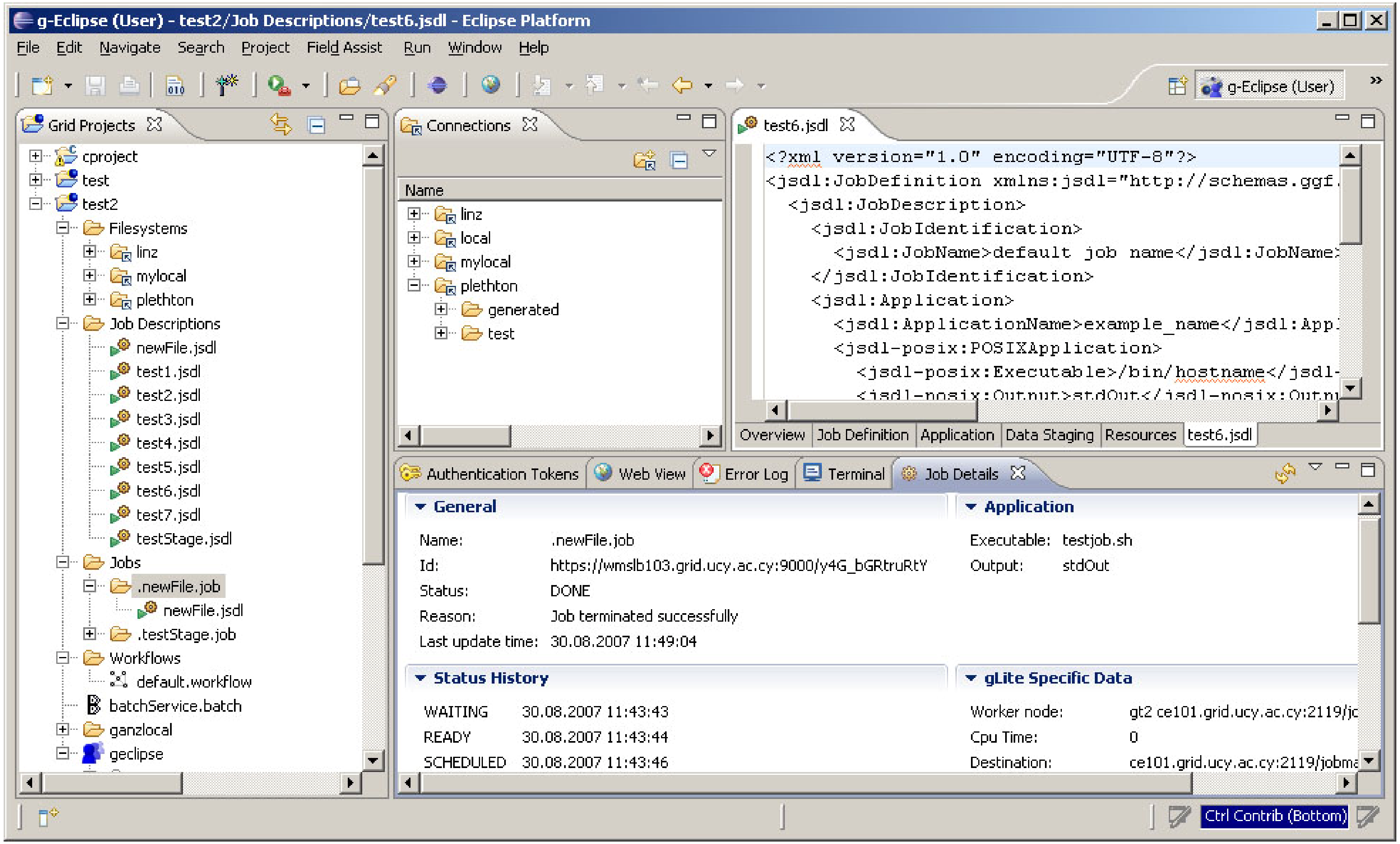
- Provides a general UI framework/eco system
  - Designed for application users, resource providers, and application developers
  - Middleware independent core infrastructure + middleware extension
  - Allowing users to access the grid in a simple way
    - File transfer: drag&drop
    - Job submission: 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



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

General

Name: .newFile.job
Id: https://wmslb103.grid.ucy.ac.cy:9000/y4G_bGRtruRtY
Status: DONE
Reason: Job terminated successfully
Last update time: 30.08.2007 11:49:04

Application

Executable: testjob.sh
Output: stdOut

Status History

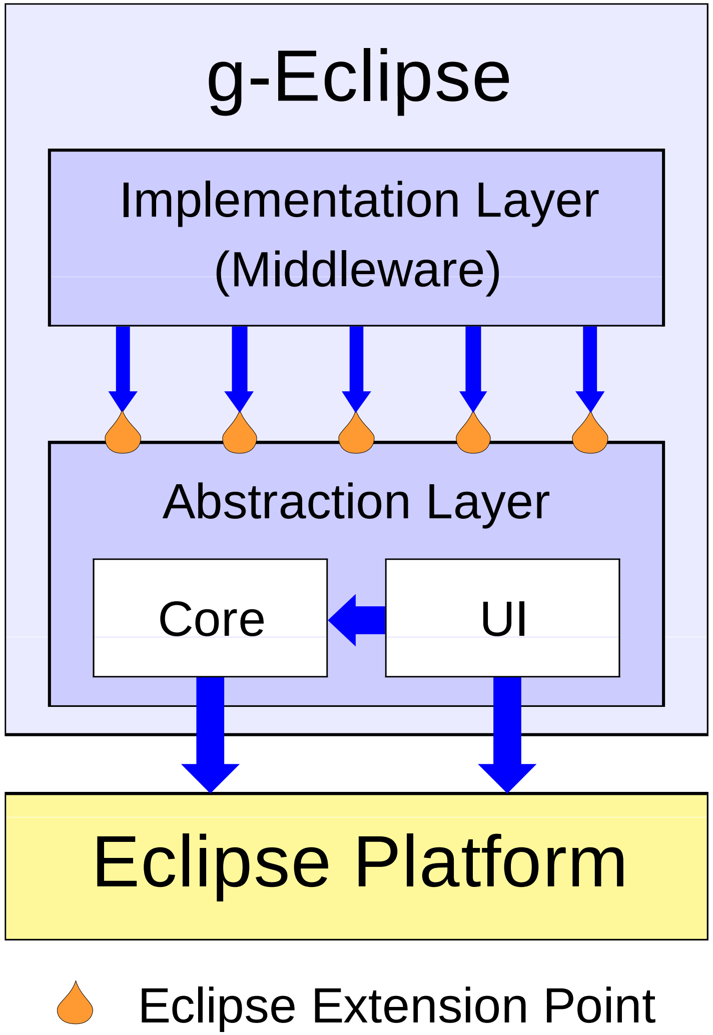
```
WAITING     30.08.2007 11:43:43
READY       30.08.2007 11:43:44
SCHEDULED   30.08.2007 11:43:46
```

gLite Specific Data

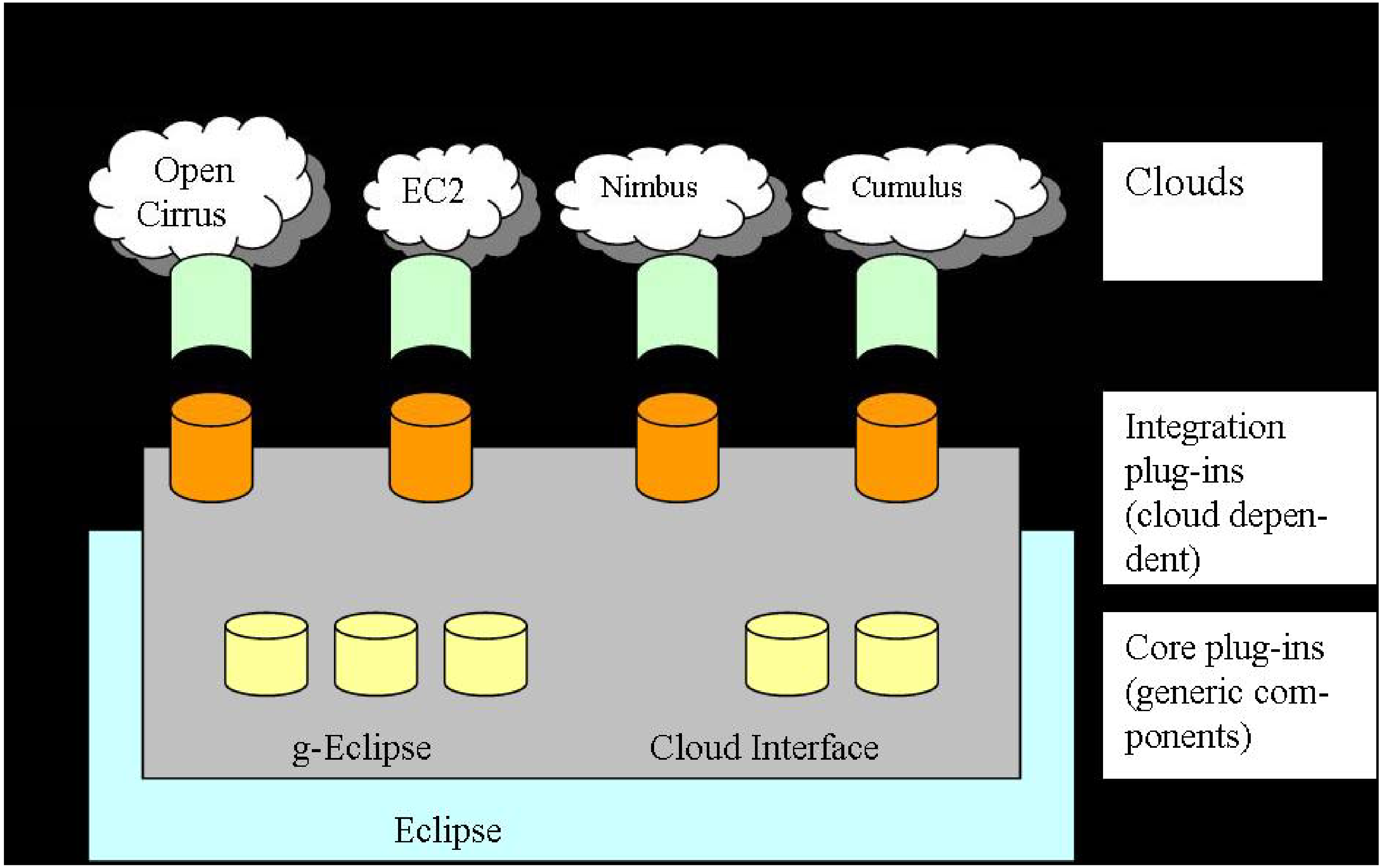
Worker node: gt2 ce101.grid.ucy.ac.cy:2119/jc
Cpu Time: 0
Destination: ce101.grid.ucy.ac.cy:2119/jobma

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



Extending g-Eclipse for Clouds

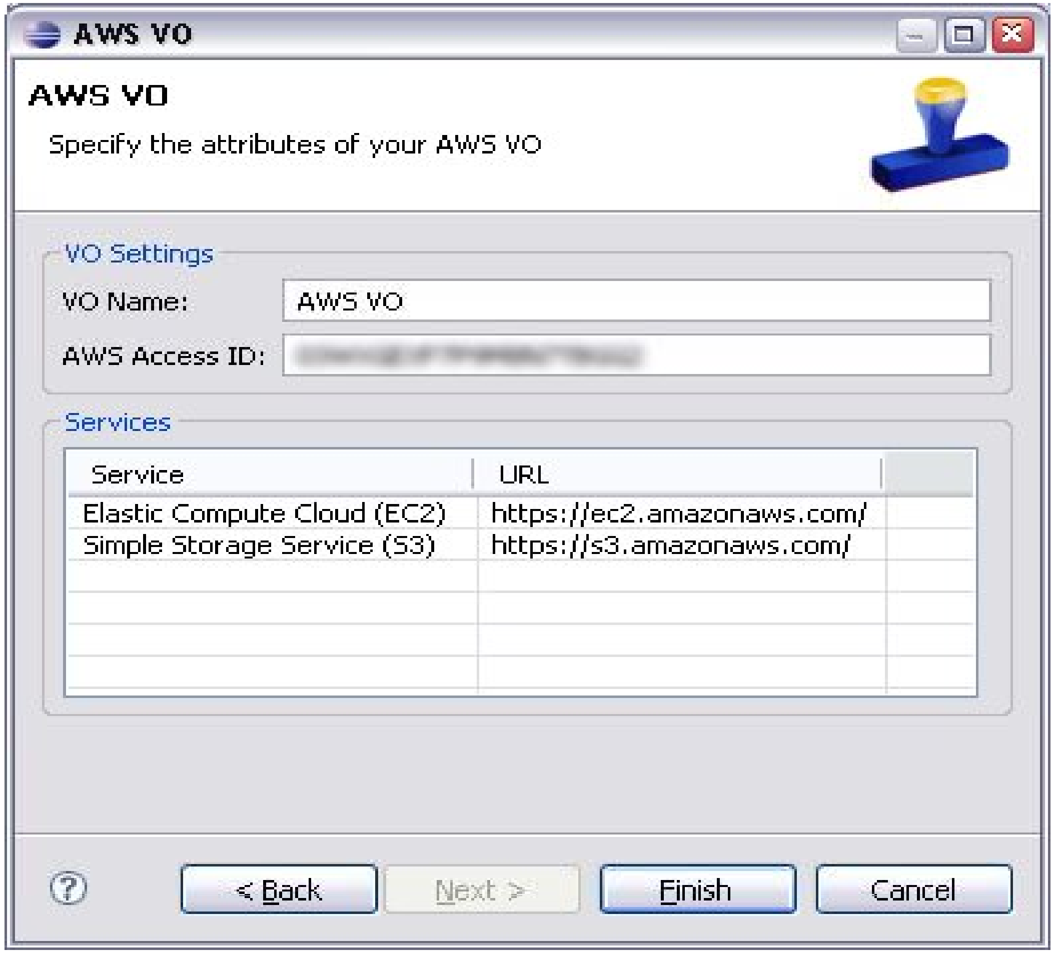

Open Cirrus
EC2
Nimbus
Cumulus
Clouds
Integration plug-ins (cloud dependent)
Core plug-ins (generic components)
g-Eclipse
Cloud Interface
Eclipse

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



AWS VO

Specify the attributes of your AWS VO

VO Settings

VO Name: AWS VO

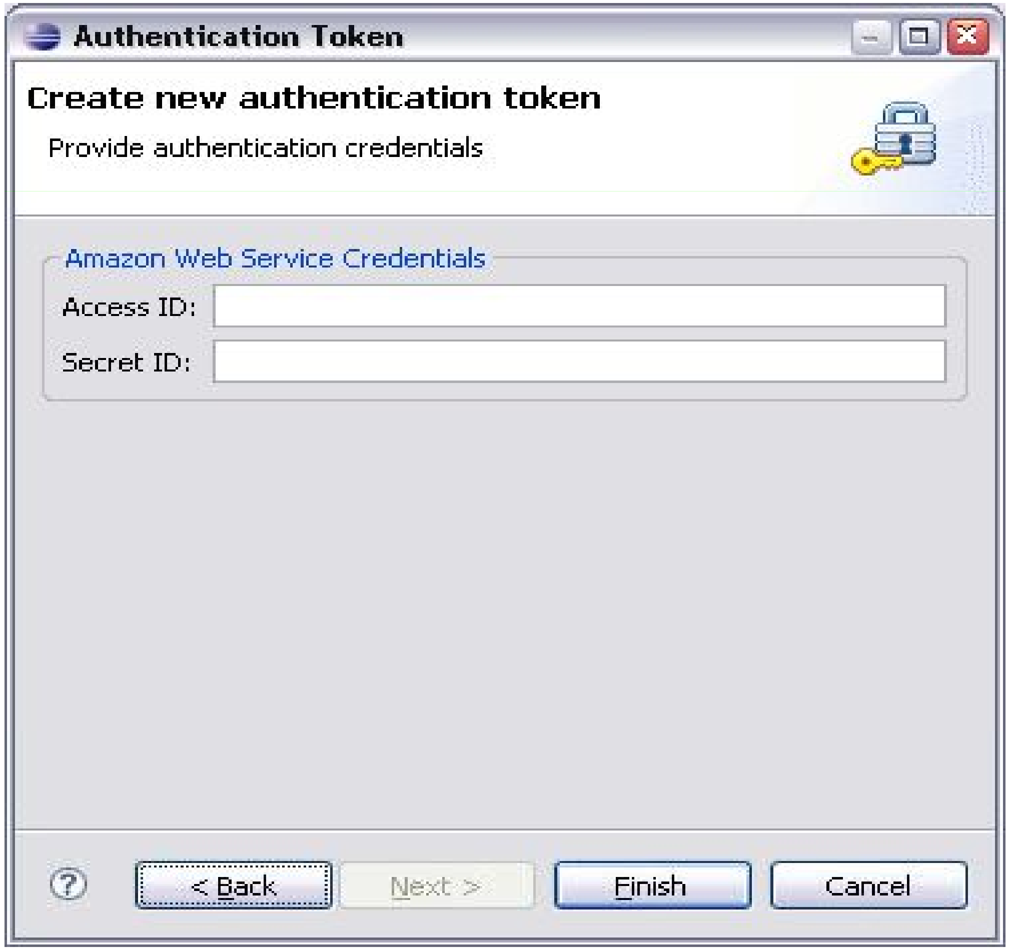
AWS Access ID: [redacted]

Services

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

Initial Implementation: Accessing the Amazon Web Service (II)

- Authentication token for AWS credential



Authentication Token

Create new authentication token

Provide authentication credentials

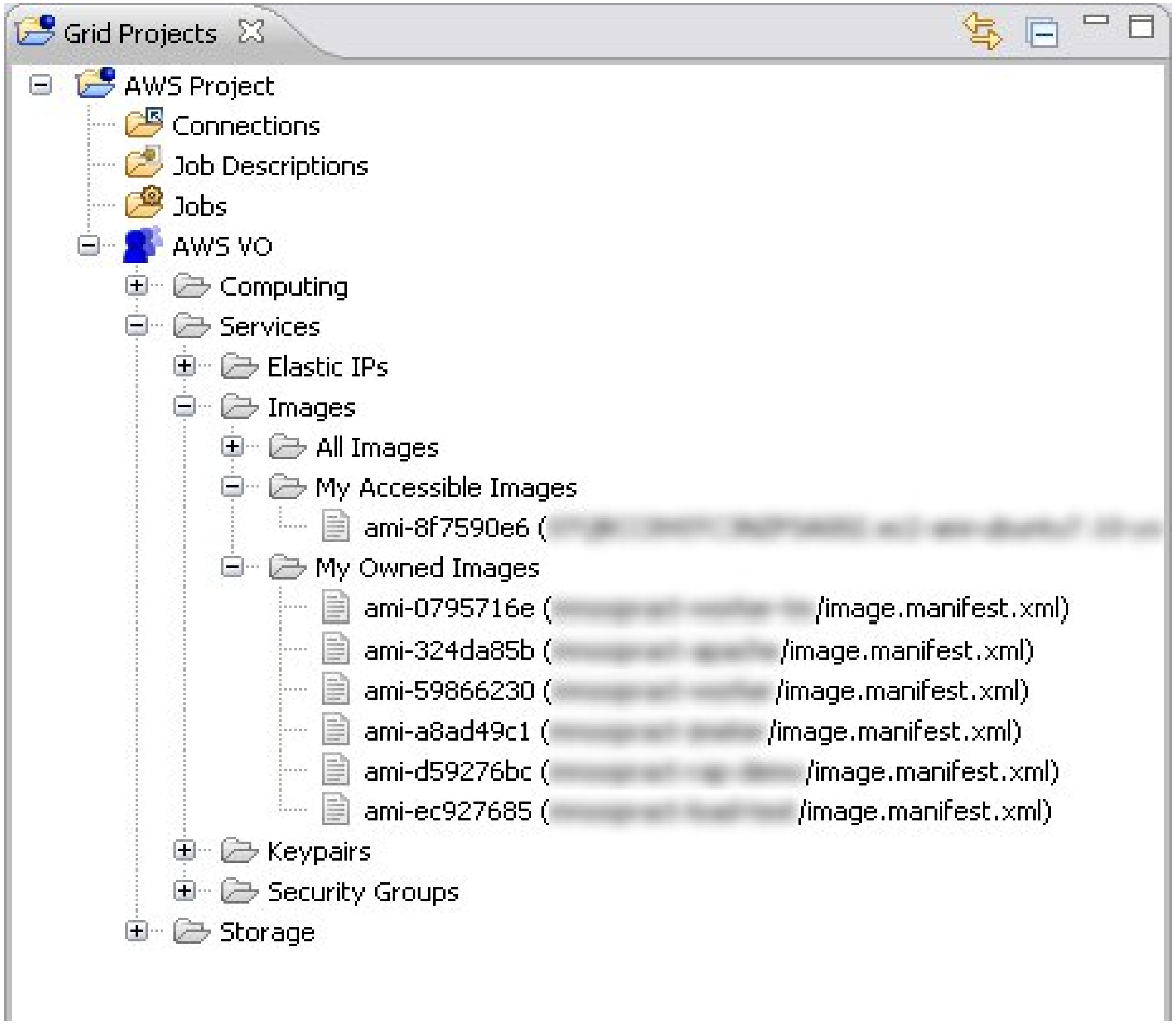
Amazon Web Service Credentials

Access ID:

Secret ID:

Initial Implementation: Accessing the Amazon Web Service (III)

- Project view

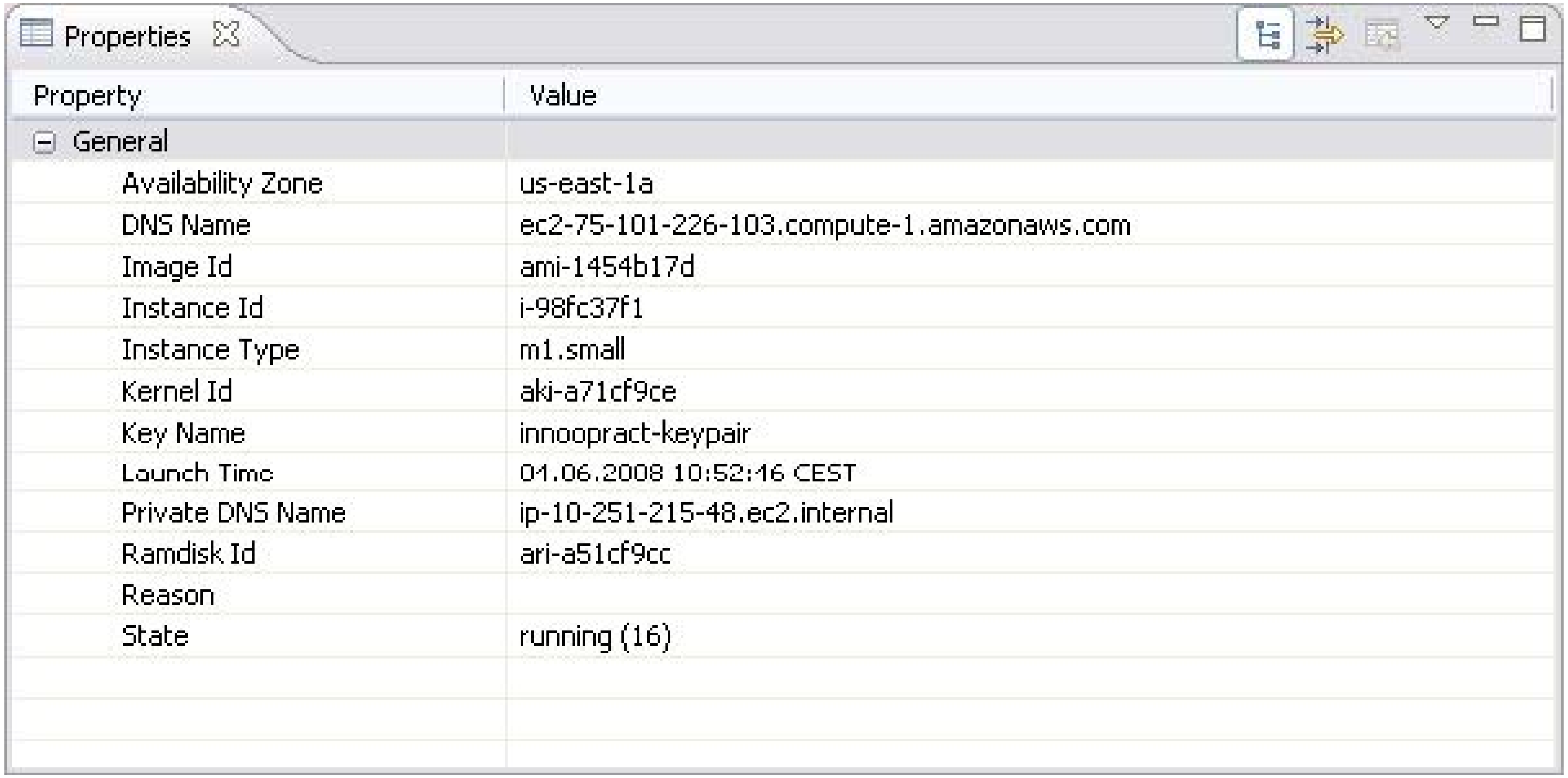


Grid Projects

- AWS Project
  - Connections
  - Job Descriptions
  - Jobs
  - AWS VO
    - Computing
    - Services
      - Elastic IPs
      - Images
        - All Images
        - My Accessible Images
          - ami-8f7590e6 ([redacted]
        - My Owned Images
          - ami-0795716e ([redacted]/image.manifest.xml)
          - ami-324da85b ([redacted]/image.manifest.xml)
          - ami-59866230 ([redacted]/image.manifest.xml)
          - ami-a8ad49c1 ([redacted]/image.manifest.xml)
          - ami-d59276bc ([redacted]/image.manifest.xml)
          - ami-ec927685 ([redacted]/image.manifest.xml)
      - Keypairs
      - Security Groups
    - Storage

Initial Implementation: Accessing the Amazon Web Service (IV)

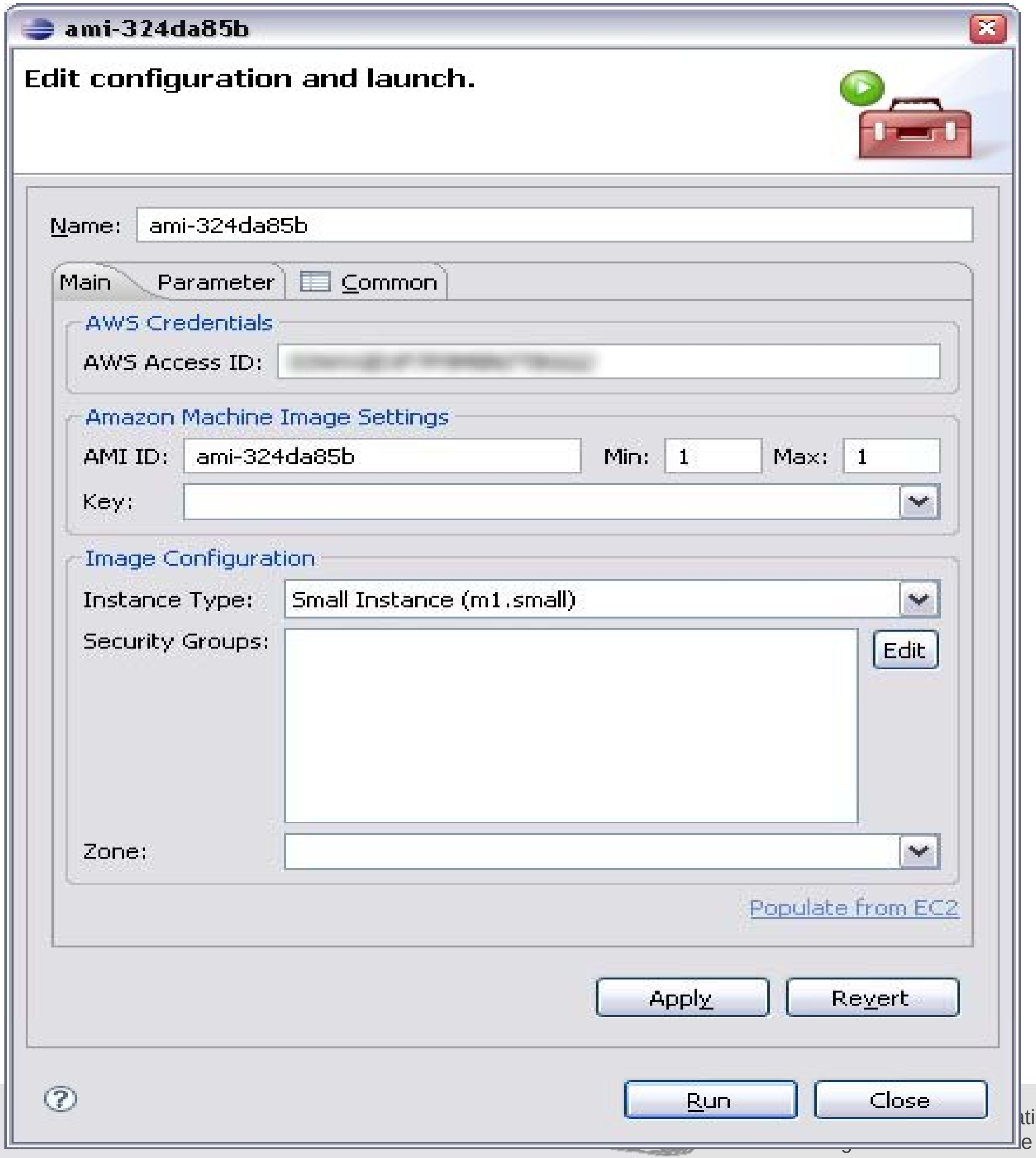
- AMI properties



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	01.06.2008 10:52:16 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



ami-324da85b

Edit configuration and launch.

Name: ami-324da85b

AWS Credentials

AWS Access ID: [redacted]

Amazon Machine Image Settings

AMI ID: ami-324da85b Min: 1 Max: 1

Key:

Image Configuration

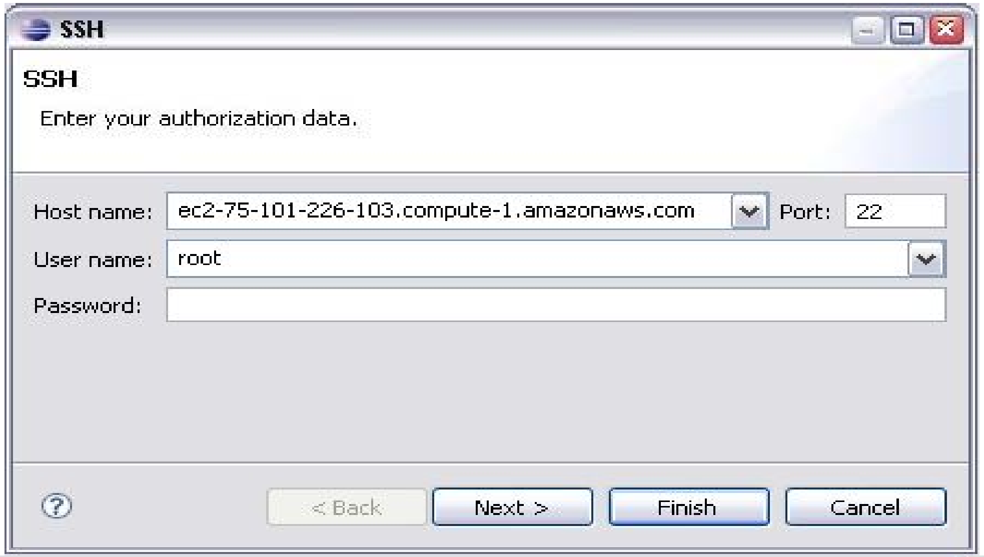
Instance Type: Small Instance (m1.small)

Security Groups:

Zone:

Initial Implementation: Accessing the Amazon Web Service (IV)

- Remote login



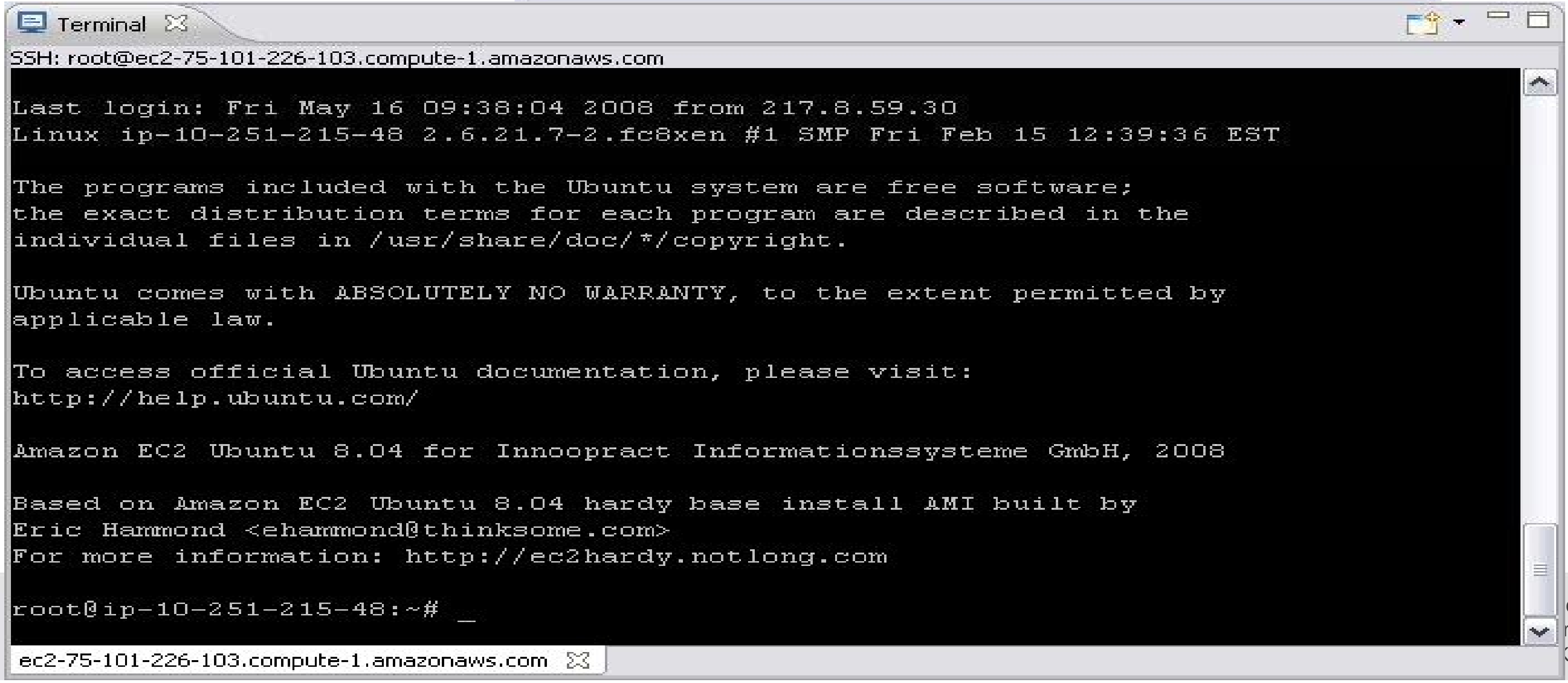
SSH

Enter your authorization data.

Host name: ec2-75-101-226-103.compute-1.amazonaws.com Port: 22

User name: root

Password:



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 Innopract 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