The NGS Grid Portal

David Meredith
NGS + Grid Technology Group, e-Science Centre,
Daresbury Laboratory, UK

d.j.meredith@dl.ac.uk

CCLRC

NGS Portal

The NGS portal can be used to access and interact with the HPC and Data resources available on the Grid via SSO (Certificates + myproxy):

- Browse for different applications available on a Grid this includes your own personal applications and pre-configured applications available on a particular Grid (e.g. the NGS is currently publishing applications within the NGS portal to be made easily available for its users).

- Submit/monitor compute jobs/applications.

- Access and move data around the Compute and Data Grid (Gridftp, srb).
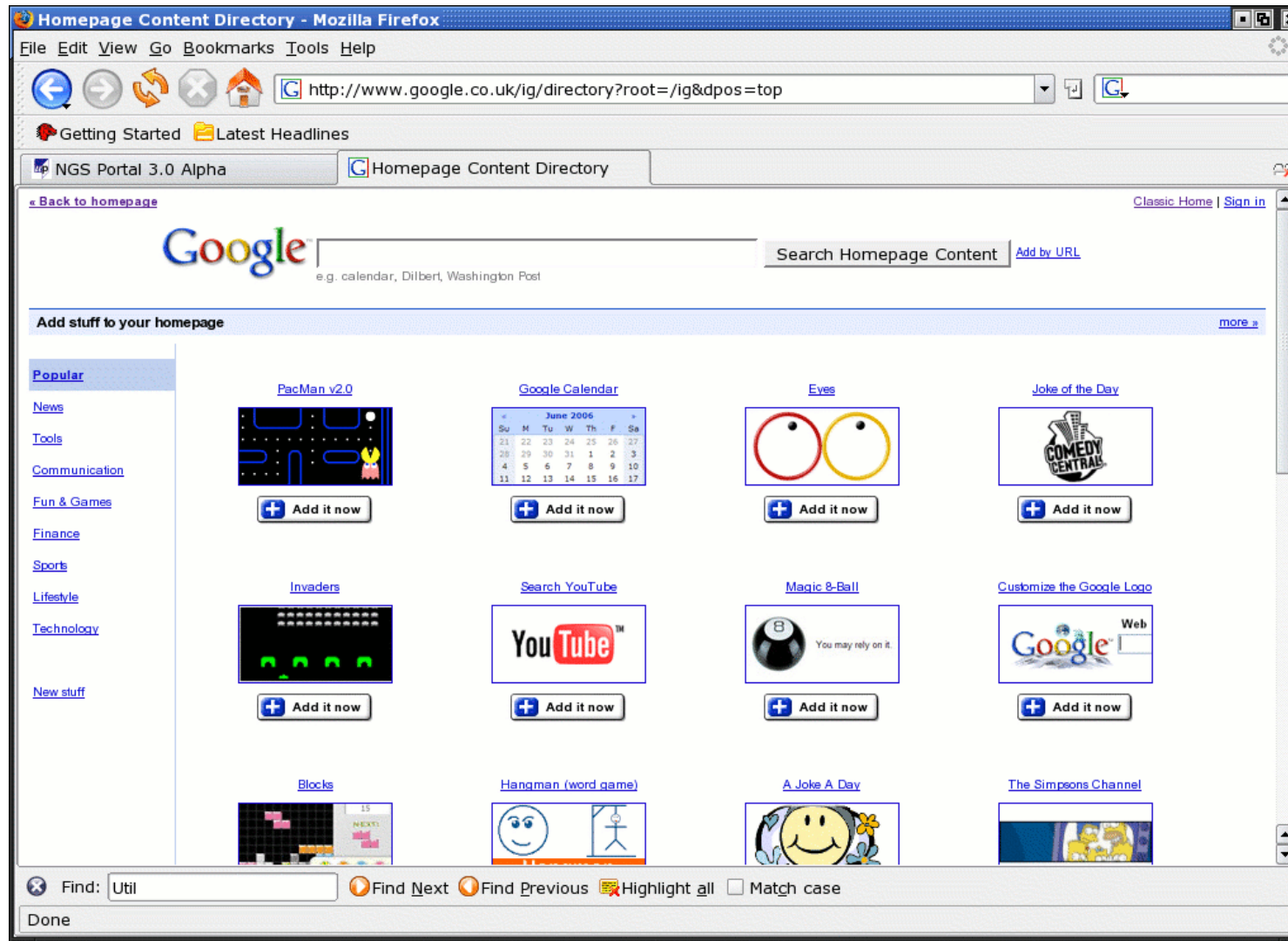
CCLRC

Portal and Portlets

- The NGS Grid portal extends a JSR-168 compliant portal container which hosts a selection of 'portlets.'

- Portlets are online-accessible applications that are hosted within a portal container.

- The list of portlets that are deployed to the portal make up the portal's overall functionality (users may be interested only in a selection of portlets)

- Main Benefit: Portlets facilitate the sharing and re-use of applications (168-compliant 3rd party portlets can be used within a portal as required).

- NGS and CCLRC e-Science Centre are developing a collection of portlets designed for the Computational and Data Grid.

- The NGS portal is a current implementation of these Grid portlets.
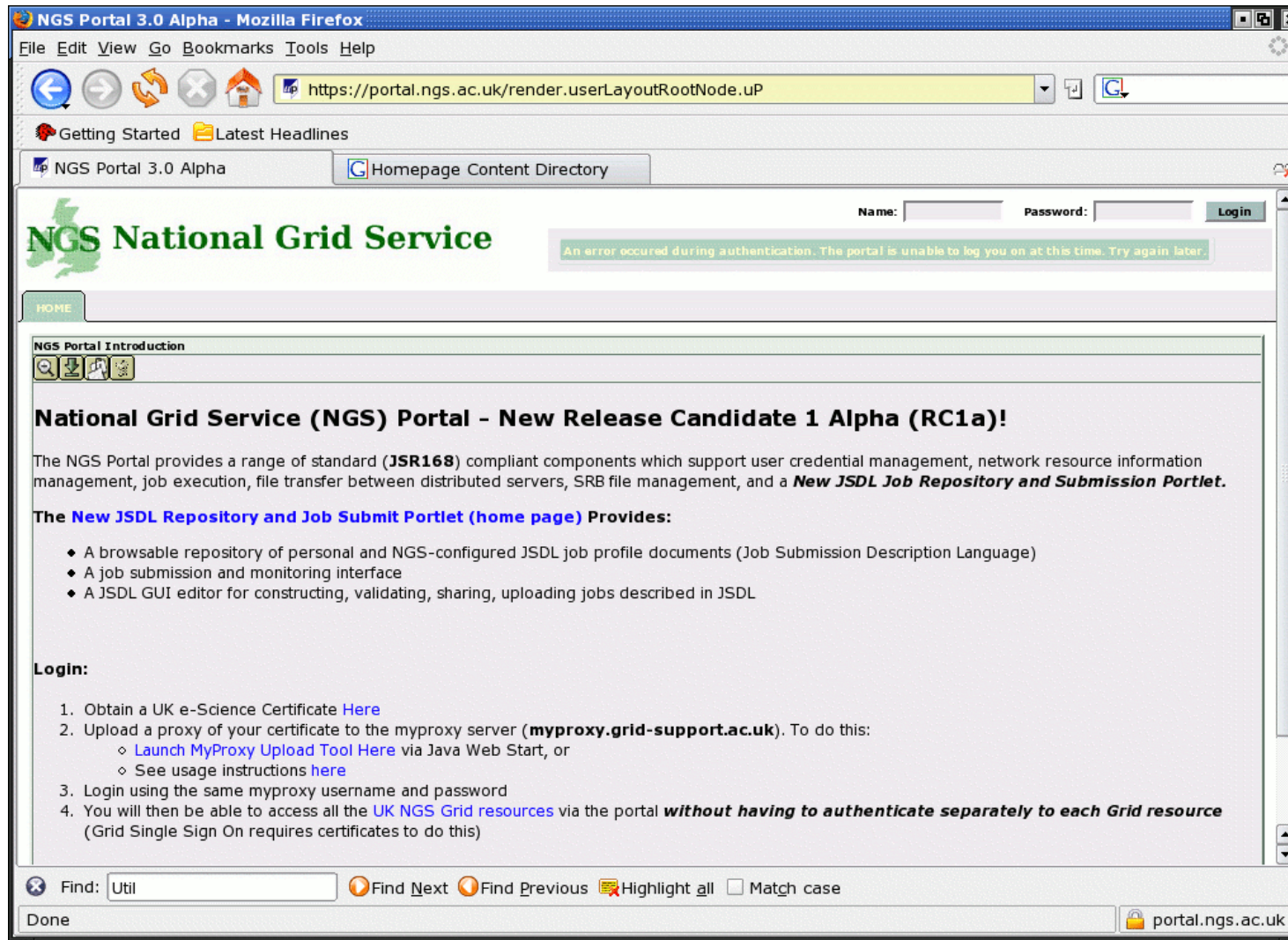
A Famous Portal + Portlets

Select Portlets of Interest (customisation)

NGS portal

After Login – Grid portlets

JSDL Repository / Job Submit Portlet

1. A browsable JSDL database of personal and shared JSDL job profile documents (Job Submission Description Language, i.e. 'job recipes/templates').

   - JSDL can be browsed for, selected and loaded in order to run applications on the Grid (loaded either 'out-of-the-box' or, more usually loaded and modified/tweaked as required).

   - JSDL can be searched for by category of interest in the portal (e.g bioinformatics, chemistry, tutorials/examples).

   - JSDL documents can be pre-configured and published by the portlet administrator(s) to be made available to all other users.

2. A JSDL GUI editor for constructing, validating, sharing, uploading jobs described in JSDL.

3. A Grid job submission and monitoring application (currently, only Globus but more Grid middleware providers are being added, e.g. GridSam/WSRF).

CCLRC

JSDL – Job Submission Description Language

```
<jsdl:Application>
    <jsdl:ApplicationName>gnuplot</jsdl:ApplicationName>
    <jsdl-posix:POSIXApplication>
        <jsdl-posix:Executable>
            /usr/local/bin/gnuplot
        </jsdl-posix:Executable>
        <jsdl-posix:Argument>control.txt</jsdl-posix:Argument>
<jsdl-posix:Argument>DavesControlFile.txt</jsdl-posix:Argument>
        <jsdl-posix:Input>input.dat</jsdl-posix:Input>
        <jsdl-posix:Output>output1.png</jsdl-posix:Output>
    </jsdl-posix:POSIXApplication>
    </jsdl:Application>
    <jsdl:Resources>
        <jsdl:IndividualPhysicalMemory>
<jsdl:LowerBoundedRange>2097152.0</jsdl:LowerBoundedRange>
        </jsdl:IndividualPhysicalMemory>
        <jsdl:TotalCPUCount>
            <jsdl:Exact>1.0</jsdl:Exact>
        </jsdl:TotalCPUCount>
    </jsdl:Resources>
     ....
```

1. XML Schema language for describing compute jobs in a platform independent language (XML).

2. Is agnostic of middleware - no dependencies on Globus, WSRF, gLite (portal that is generic and not tied to any particular set of Grid technologies).

3. GGF / OGF Standard.

4. JSDL documents can be validated against the JSDL and JSDL POSIX XSD Schema to ensure its correctness

CCLRC

JSDL – Job Submission Description Language XSD Schema

```xml
<!--==============================================-->
<xsd:complexType name="Environment_Type">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="name" type="xsd:NCName" use="required"/>
            <xsd:attribute name="filesystemName" type="xsd:NCName" use="optional"/>
            <xsd:anyAttribute namespace="##other" processContents="lax"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
<!--==========================================-->
<xsd:complexType name="Argument_Type">
    <xsd:simpleContent>
        <xsd:extension base="xsd:normalizedString">
            <xsd:attribute name="filesystemName" type="xsd:NCName" use="optional"/>
            <xsd:anyAttribute namespace="##other" processContents="lax"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```
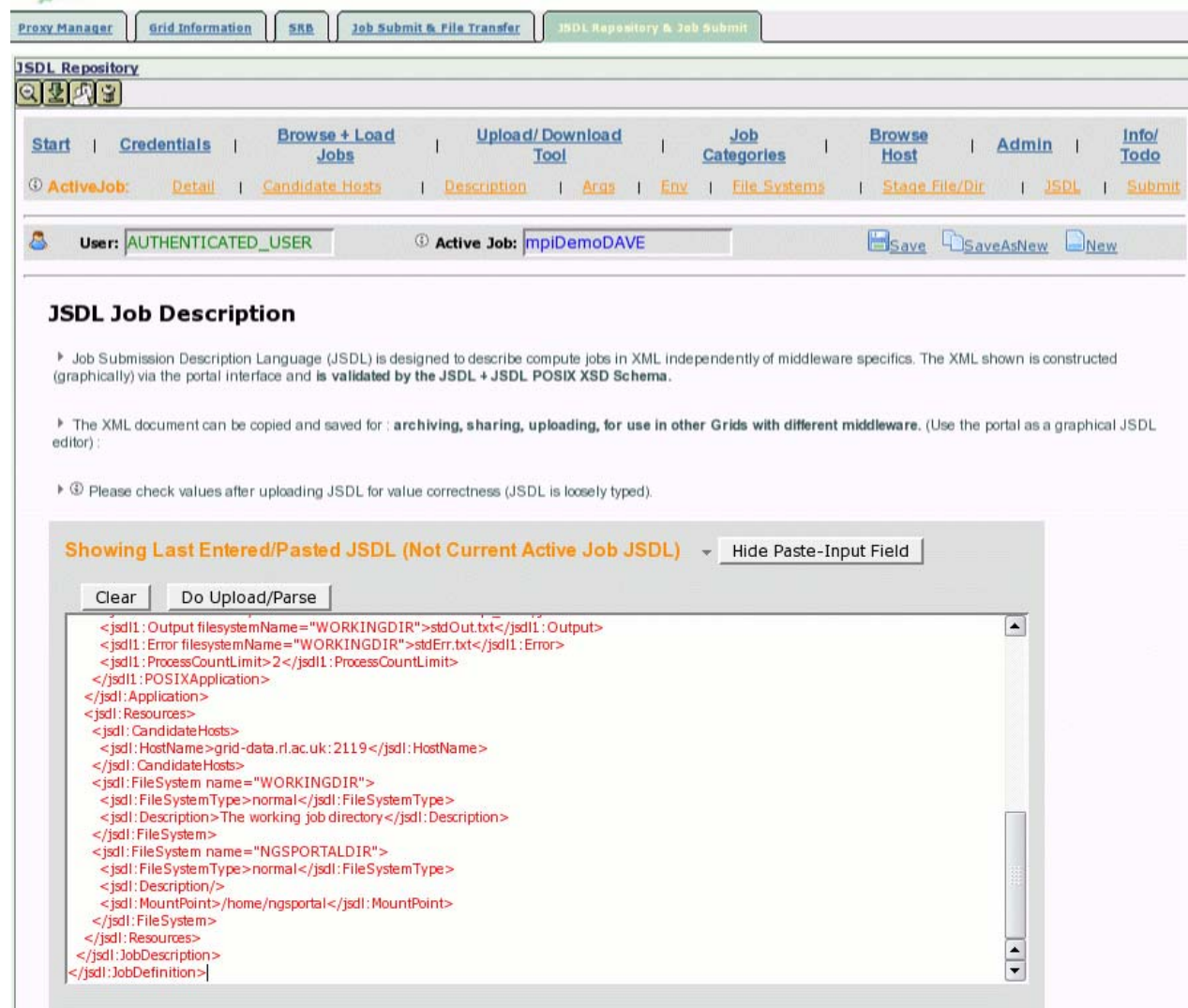
JSDL Repository / Database

Select category of interest – e.g. 'RAL Bioinformatics.'

Browsing the JSDL database for public and personal job profiles.

List jobs, read job descriptions and load a job to initialise the 'Active Job.'

CCLRC

Active Job's JSDL

The Active Job JSDL is automatically created, updated and validated by the portal by changing parameters in the portal GUI.

The portlet acts as a JSDL GUI editor

CCLRC

Upload / Share JSDL

JSDL documents can be uploaded to the portal in order to initialise the Active Job.

Validation errors and messages are displayed in the interface when uploading JSDL.

The portlet allows the sharing of job profiles and JSDL between users and user communities.

CCLRC

Active Job Detail

Input fields are filled out for pre-configured applications.

Changes to the parameters in the GUI will update the generated JSDL automatically.

Input fields are taken from the JSDL and JSDL-POSIX extension schemas.

CCLRC

Environment Variables / Arguments

Environment Variables

Setup environment variables required to run the job e.g. **"NGSMODULES" (how to)**

Edit	⊕ Name	⊕ Value	✕
▸ Edit	TMP	/tmp	☐
▸ Edit	NGSMODULES	envVarValue1	☐
			Close Detail

Add Environment Variable:

ⓘ Name:

ⓘ Value:

[Add]   [Add NGSMODULES]

Add required env vars, e.g. 'NGSMODULES' – used to configure application environment

Command Line Arguments

▸ Enter or paste arguments below
▸ Use Space and/or Line separated values
▸ Use double quotes if space is required in a single argument, e.g. "a and b"

```
/usr/local/applications/bioinformatics/EBI/data/rel_tpa_inv_01_r89.dat seqr.out -osformat fasta
```

[Update]   [Clear]

Configured Arguments
/usr/local/applications/bioinformatics/EBI/data/rel_tpa_inv_01_r89.dat
seqr.out
-osformat
fasta

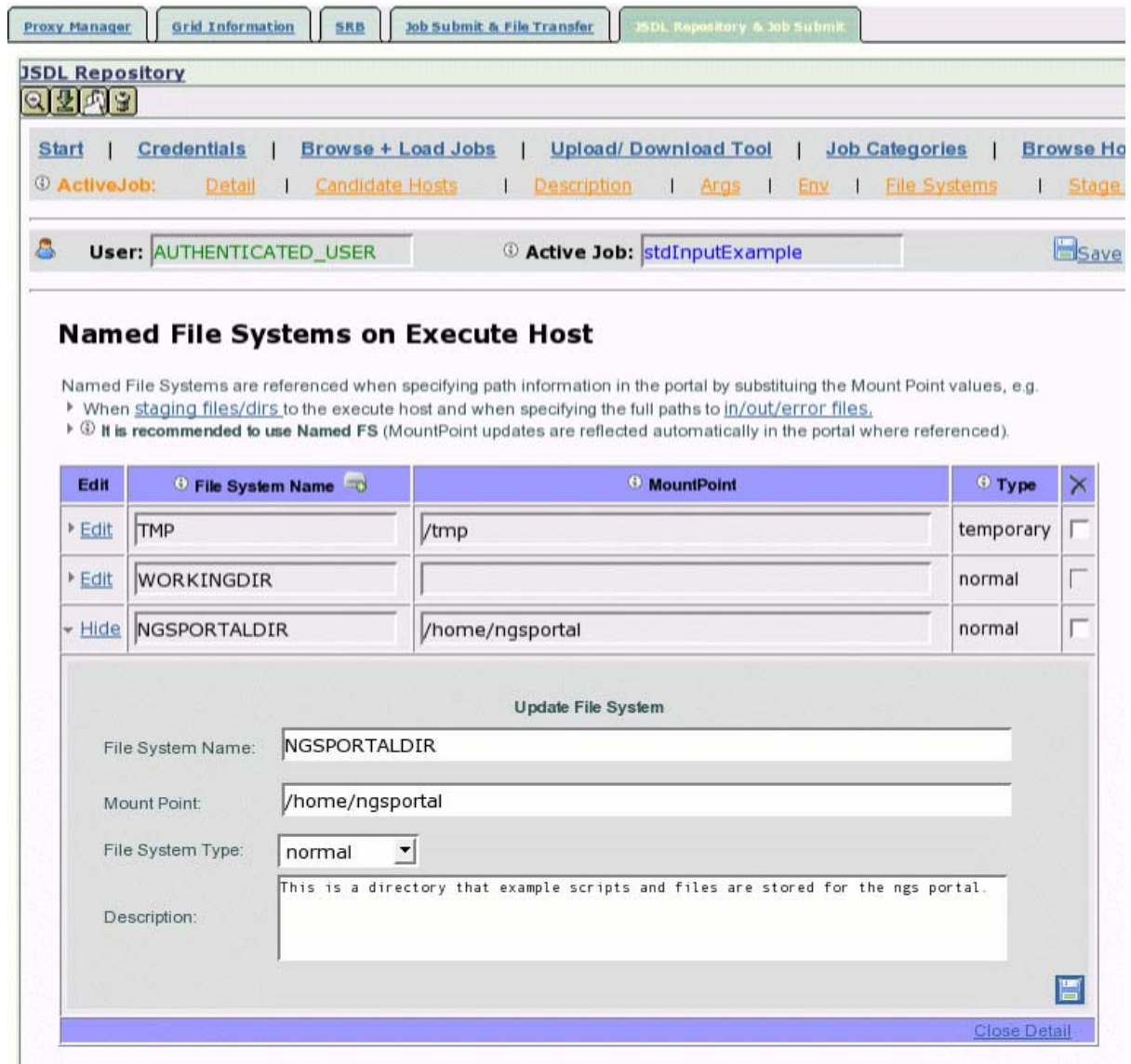Paste and parse command line arguments (space and/or line separated values)

CCLRC

Named File Systems

Named file systems used to declare mount points that are required on the consuming system.

File system names are referenced throughout the portlet (and JSDL doc) for substituting mount points where required.

Changes to a FS mount point will be updated automatically throughout the portal/JSDL.

Used when specifying path info e.g., locations to files/dirs, stage data locations etc.

CCLRC

Stage Data

Compile a list of required data (i.e. data that should be copied to the consuming system from remote locations across the Grid prior to job execution).

Data is staged relative to named file systems.

The source URI can be either specified manually or, more normally, browsed for in the 'Browse Host' page.

Browse Hosts

Browse remote Grid hosts for stage data.

Select files and directories that should be copied to the consuming system via GsiFtp (more protocols to be supported inc srb, ftp, webdav).

# Candidate Hosts

Candidate hosts are consuming systems that can be nominated to run the Active Job.

The candidate host list can be compiled from a personal host list and from a default host list (available to all users).

GridFtp Upload / Download / File Transfer Tool

Transfer files to/from your desktop and a remote Grid ftp server via 'drag-n-drop' – akin to Windows explorer for the Grid
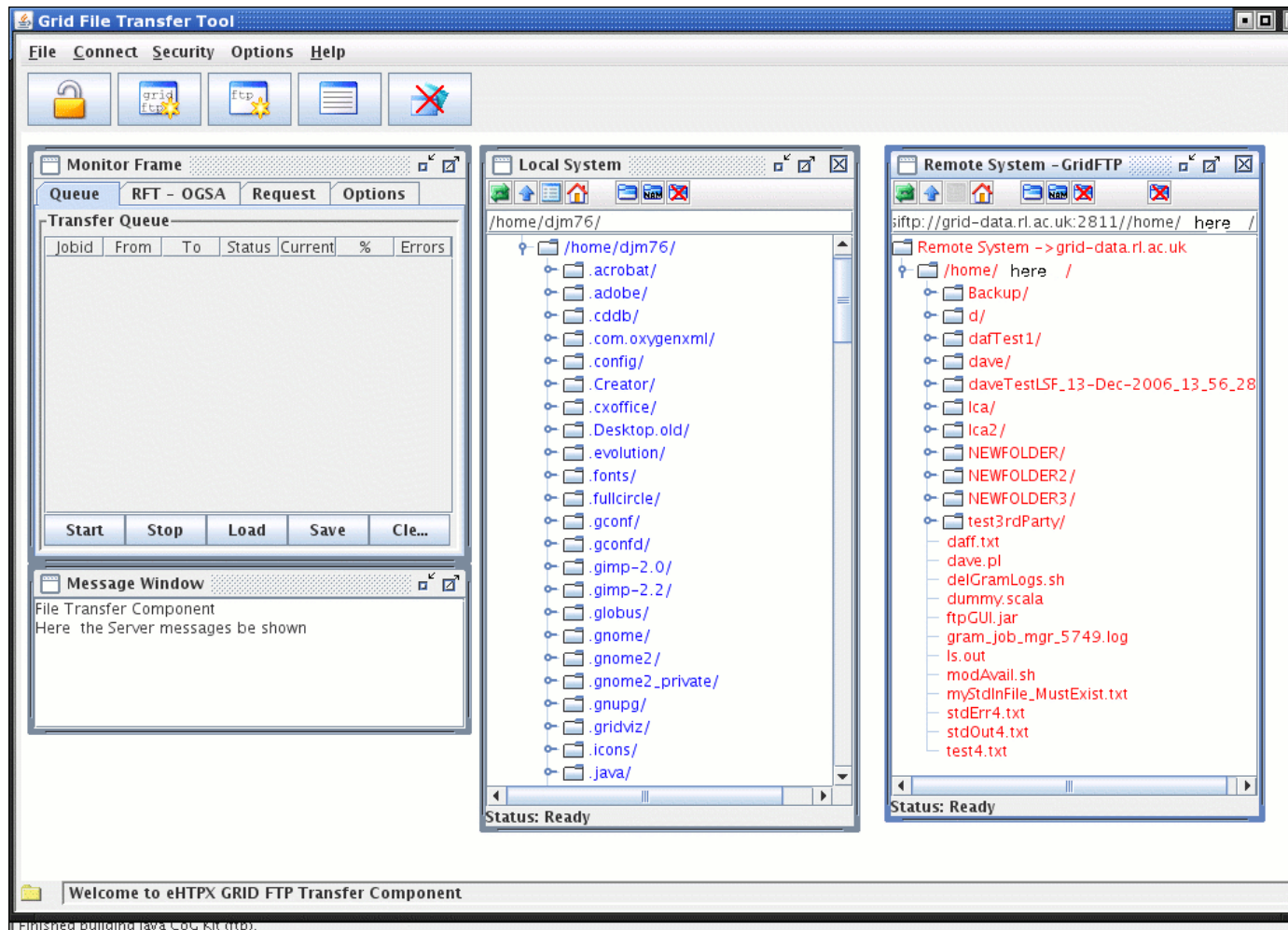
TODO / Future

1. Extend the portlet to support the registration of new interfaces, i.e. register application specific interfaces designed for specific applications. (Note, new interfaces will build JSDL in same way – will use underlying portlet functionality in same way).

2. Extend the interface for:-

   - a) Parameter sweep style jobs (parameter replacement in jsdl).

   - b) Data visualisation via the portal interface.

3. Portal is designed to be generic and not tied to any particular set of Grid technologies. Extend portal to support more Grid + Web protocols (e.g. SRB, WebDav, GridSam, WSRF).

4. Release the portlets for use in other projects / Grids.

   Final note - A portal is only as good as the underlying deployed infrastructure…. portal development often involves debugging the underlying consuming systems and middleware

Software Stack

JSF (Java Server Faces) interface + MVC control layer (Http session and request scope data)

Spring v2.0 managed business objects (singleton + prototype injected object graphs, declarative transaction demarcation, data source management).

C3p0 db connection pooling

ORM (object relational mapping) - JPA (Java persistence API) + Hibernate 3.2 for domain model.

Java CogKit for Globus API

Apache XMLBeans for JSDL xml-object data binding framework

CCLRC