Imperial College London

GridSAM - Job Submission and Monitoring Web Service

William Lee
London e-Science Centre
Department of Computing, Imperial College London

Agenda

- GridSAM Overview
- Job Submission Description Language
- Job Submission and Monitoring Port Types
- GridSAM Architecture
- GridSAM Implementation
- Planned Works
- Comments
- Summary

GridSAM Overview

Grid Job Submission and Monitoring Service

- What is GridSAM?
  - Part of the OMII managed programme
  - Client's Perspective
    - Job Submission and Monitoring Web Service
    - Standardised Job Description
    - Virtual file input and output sandboxes
    - Client-side submission clients
  - Developer's Perspective
    - Extensible JobManager API interfacing with existing Distributed Resource Managers (DRM)
    - Used as an embedded library
      - Job Submission Portal
      - Grid Applications
- What not?
  - A job scheduling system
  - Replacement of existing Grid resource management system.

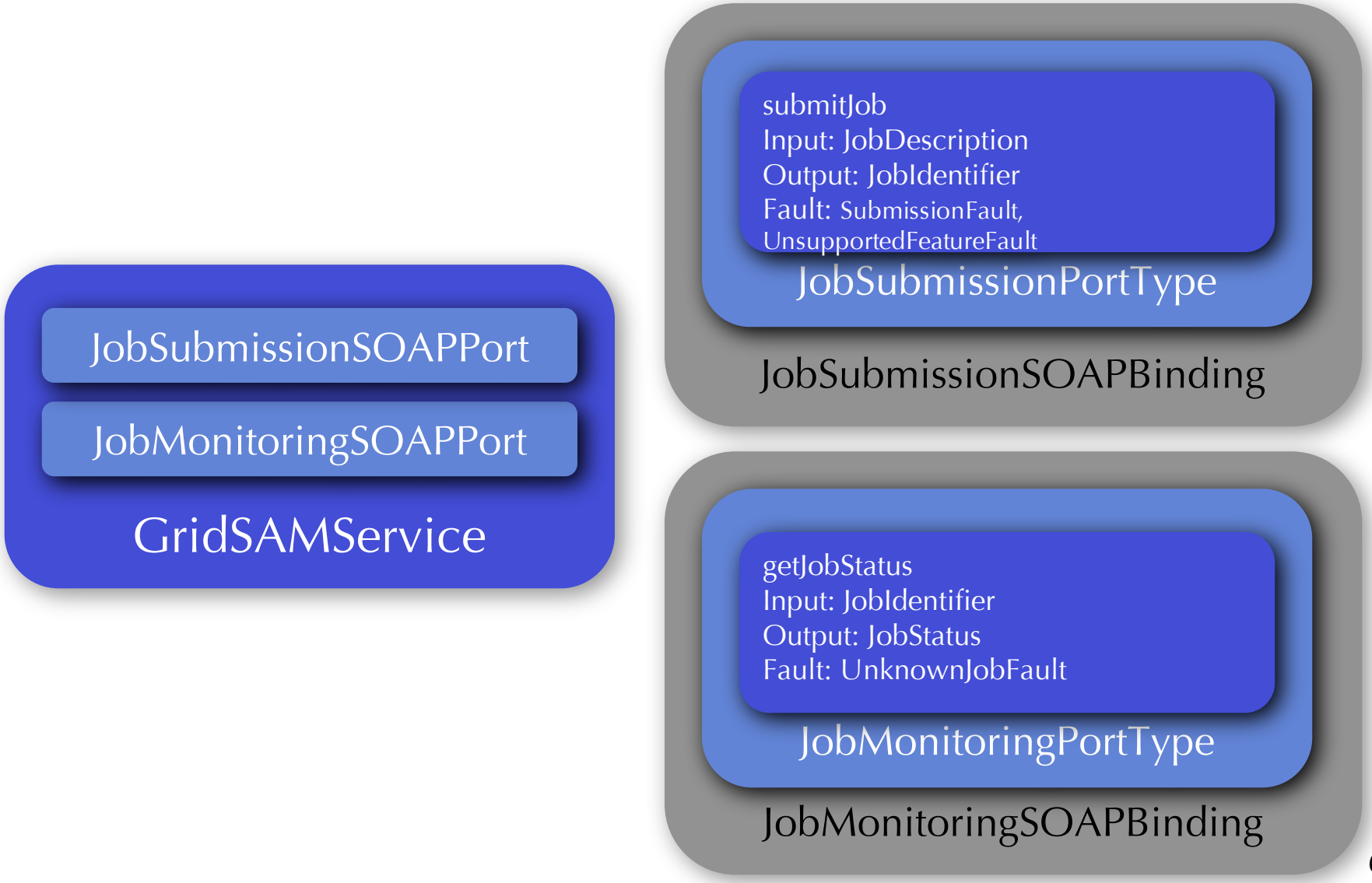
Job Submission Description Language

- Job Submission Description Language (JSDL)

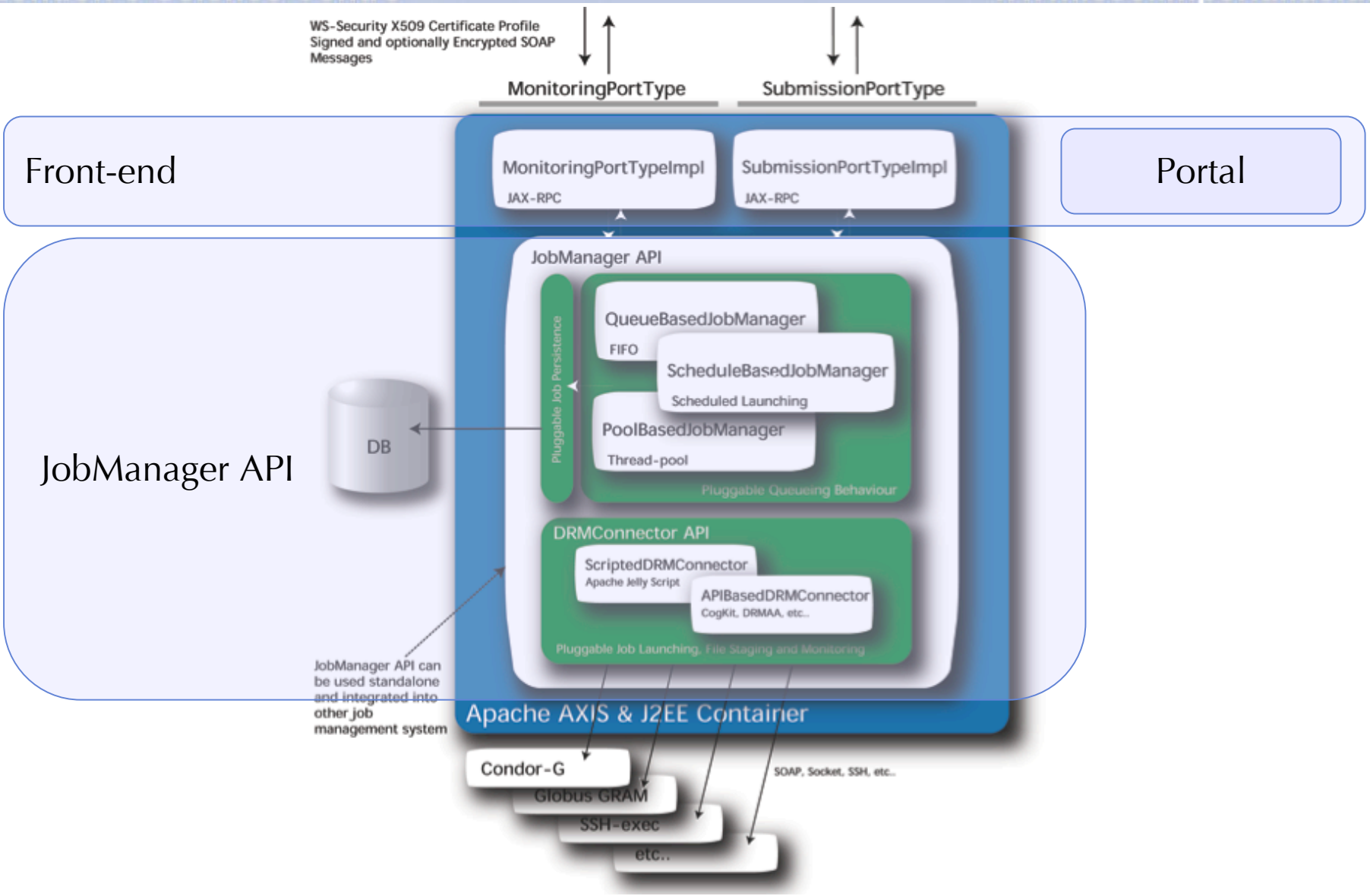
Job Submission Description Language


```
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:job xmlns:jsdl="http://www.gridforum.org/JSDL"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.gridforum.org/JSDL jsdl.xsd">
    <jsdl:JobIdentification>
        <jsdl:JobName>This is my job - a short description
        </jsdl:JobName>
    </jsdl:JobIdentification>
    <jsdl:Application>
        <!-- type should be xsd:token not xsd:string -->
        <jsdl:ExecutableName type="jsdl:onPath">mycommand</jsdl:ExecutableName>
        <jsdl:Argument>-somearg</jsdl:Argument>
        <jsdl:Argument>inputfile.txt</jsdl:Argument>
        <jsdl:StdOut>output.txt</jsdl:StdOut>
        <jsdl:StdErr>error.txt</jsdl:StdErr>
    </jsdl:Application>
    <jsdl:DataAttributes>
        <jsdl:File>
            <jsdl:FileName>inputfile.txt</jsdl:FileName>
            <jsdl:Source>http://mysite/some/public/file.txt</jsdl:Source>
        </jsdl:File>
        <jsdl:File>
            <jsdl:FileName>output.txt</jsdl:FileName>
            <jsdl:Target>gridftp://mygftpserver/incoming/output.txt</jsdl:Target>
        </jsdl:File>
        <jsdl:File>
            <jsdl:FileName>error.txt</jsdl:FileName>
            <jsdl:Target>ftp://myftpserver/incoming/output.txt</jsdl:Target>
        </jsdl:File>
    </jsdl:DataAttributes>
</jsdl:job>
```

GridSAM Service



GridSAM Architecture



Standards

- WS-I Basic and Security Profiles
  - XML, XSD
  - WSDL (document/literal)
  - SOAP
  - WS-Security
- WS-Notification*
- JSDL*

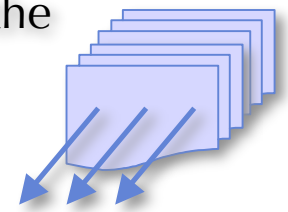
Implementation
Front-ends

- J2EE
  - Implemented using the J2EE JAX-RPC Web Service API.
    - Tested on the Sun Application Server v.8
  - Exploits clustering and Java Message Queue
  - Cross-deployable on other J2EE compliant servers
- Apache Axis on Tomcat Server
  - Porting to Apache Axis Web Service integrating with the OMII base distribution
- Core functionalities reside in JobManager API, front-ends are pure wrappers providing front-line security to the underlying system.

Implementation
JobManager API

- JobManager implementation - pluggable queuing behaviour
  - CalendarBasedJobManager
    - Passes job to DRMConnector according to a timetable

  - ThreadPoolBasedJobManager
    - Pool of threads taking jobs off the queue passing them to the DRMConnector



  - JMSJobManager
    - Java Message Queue based job queuing. Allows DRMConnectors to be clustered using J2EE features

Implementation
DRMConnector API

- Interacting with existing DRM or providing new launching mechanism.
  - ScriptBasedDRMConnector
    - An helper implementation third-party developers can extend.
    - Converts declarative JSDL into an executable set of actions in the form of an *Apache Jelly* script.
      - Staging in files
      - Generating lower-level script
      - Invoking external commands
      - Staging out files
    - In-line XML parsing of JSDL using XPath or Xquery
    - Exploits the wealth of "Jelly Tags" (plugins) available from the open-source community.
    - Aids diagnostic

JSDL to Apache Jelly
Description to Instruction


```
<j:jelly xmlns:j="…" xmlns:f="…" xmlns:h="…" xmlns:jsdl="…">
        <!-- stage-in files -->
        <f:ftp from="ftp://myserver/myfile.dat" to="${job.home}/input1.dat"/>
        <h:http method="get" url"http://myserver/myfile2.dat" to="${job.home}/input2.dat"/>
        <!-- use XPath to extract information from JSDL document -->
        <x:set select="$jsdl/jsdl:job/jsdl:Application/jsdl:ExecutableName" var="executable"/>
        …
        <c:condorsubmit var="clusterid">
                Executable=${executable}
                Arguments=${args}
                        …
                Queue
        </c:condorsubmit >
        …
        <!-- Upon job termination, stage-out files -->
        <c:condorstatus cluster="clusterid" status="done" >
                <g:gftp from="${job.home}/output.dat" to="gsiftp://myserver/myfile.dat"/>
        </c:condorstatus>
</j:jelly>
```

Implementation

ScriptBasedDRMConnector Implementation

- DRMConnector implementation - DRM-specific plugins
  - ForkDRMConnector
    - Spawns job locally
    - Provides limited requirement specification
  - CondorDRMConnector
    - Launches job through Condor
    - Exploits requirement matching capability in Condor
    - Submits jobs to Globus resources through Condor-G
  - Others
    - SGE
    - PBS
    - etc..

Job Persistence

- Job Persistence Layer
  - JobQueue
    - InMemoryJobQueue
      - For testing or embedded use
    - JDBCJobQueue
      - Job information and state changes are persisted to RDBMS using Java JDBC API
    - JMSJobQueue
      - Use Java Messaging Queue to persist durable job information.

File Transfers

- Currently supported file transfer protocols
  - Input: http(s)://, ftp://, file://
  - Output: ftp://, file://
- Planned support
  - Gridftp: gridftp://
- Others
  - Secure FTP: sftp://
  - WebDAV: webdav://
  - etc..

Security

- WS-Security - Message Level Security
  - User signs SubmitJobRequest message.
  - Server checks message integrity and authorise access using the J2EE role-based access control or the OMII authorisation mechanism.
- HTTPS - Transport Level Security
  - Encrypted data stream
  - X509-certificate mutual authentication on the portal front-end

Planned Works

- Jobs are submitted under a mapped native user account
- Investigate secure delegation in order to perform secured file staging on behalf of users
- Job state notification
- Continuously track development of JSDL
- Full use of Condor-G functionality through CondorDRMConnector
- Provides experimental supports for other DRM systems

Comments

<Application/>

- Executable/@type
  - Is JSDL going to define some common types?
- Executable/@name
  - Is it a path or symbolic name?
    - If it is a path, how to interpret path separator. If relative? Is it relative to the working directory?
- Argument
  - Must spell out 'meaning' of ordering in the spec, not in the schema.

Comments
<DataAttributes/>

- File/Source || File/Target
  - Is an URI expressive enough to describe the location of the source/target?
  - What about security? Who to stage the jobs?
  - Multiple File/Source is a choice, multiple File/Target is replication.

Comments

<Resource/>

- Semantics of the Resource Attributes
  - A description for the submission system to decide to honour. Exact interpretation or failure handling is out-of-scope of JSDL.
    - Hard requirement?
      - How about a mustUnderstand attribute across all resource attributes?
    - Just a hint to the scheduler.

Comments
Extensions

- Do we need substitution?
  - Useful in constructing input/output file names in relation to the submitted process
  - Can be done externally by a pre-processor. Generates multiple JSDL documents.
- Third-party extensions
  - Attribute extension: namespace qualified attribute
  - Element extension: Could use URI instead of string to identify the extension

Description vs Instruction

Conclusion

- GridSAM
  - is a Job Submission and Monitoring Web Service
  - is a Job Management API embeddable in other applications
  - uses a standardised Job Description
  - provides a standard-based approach to interface with existing DRM systems
  - supports a variety of file transfer protocols for staging files in-and-out of jobs.

Demonstration

- Demonstration
  - Web-based job submission and monitoring
  - Command-line Web Service submission and monitoring

For more information

- London e-Science Centre
  - http://www.lesc.imperial.ac.uk
- Open Middleware Infrastructure Institute
  - http://www.omii.ac.uk

Acknowledgements

- Director: Professor John Darlington
- Research Staff:
  - Nathalie Furmento, Stephen McGough
  - William Lee, Jeremy Cohen
  - Marko Krznaric, Murtaza Gulamali
  - Asif Saleem, Laurie Young, Jeffrey Hau
  - David McBride, Keith Sephton
- Others:
  - Steven Newhouse, Yong Xie, Gary Kong
  - James Stanton, Anthony Mayer
- Contact:
  - http://www.lesc.ic.ac.uk/iceni
  - e-mail: lesc@ic.ac.uk