EU project: RI031844-OMII-Europe

Project no: **RI031844-OMII-Europe**

Project acronym: **OMII-Europe**

Project title: **Open Middleware Infrastructure Institute for Europe**

Instrument: **Integrated Infrastructure Initiative**

Thematic Priority: **Communication network development**

D:JRA1.10 The second yearly report of the job submission and job monitoring task

Due date of deliverable: Apr 2008
Actual submission date: Apr, 2008

Start date of project: **1 May 2006**

Duration: **2 years**

Organisation name of lead contractor for this deliverable: INFN

Revision [1.0]

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	**Public**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

For additional information see
http://omii-europe.com or contact **info@omii-europe.com**

Document Change History

Version	Date	Comment	Author/Partner
0.1	2008-04-02	Initial draft	Moreno Marzolla/INFN
0.2	2008-04-07	Second draft	Moreno Marzolla/INFN
0.3	2008-04-09	Third draft	Moreno Marzolla/INFN
0.4	2008-04-11	Fourth draft	Moreno Marzolla/INFN
0.5	2008-04-16	Fifth draft	Morris Riedel/FZJ Sven Van De Berghe/FLE
1.0	2008-04-24	Final version	Moreno Marzolla/INFN

Document Control Sheet

Document	**Title**: Second yearly report of the job submission and job monitoring task	
	ID: D:JRA1.10	
	Version: 1.0	**Status**: Final
	Available at:	
	Software Tool: OpenOffice.org, Microsoft Word	
	File(s): OMII-EU-DJRA1.10-v1.0.odt	
Authorship	**Written by:**	Moreno Marzolla/INFN (Editor) Morris Riedel/FZJ Sven Van De Berghe/FLE
	Contributors:	
	Reviewed by:	Fredrik Hedman/KTH David Scott/EPCC
	Approved by:	

Document Status Sheet

Version	**Date**	**Status**	**Comments**
0.1	2008-04-02	Draft	Initial version
0.2	2008-04-07	Draft	Second draft
0.3	2008-04-09	Draft	Third draft
0.4	2008-04-11	Draft	Fourth draft
0.5	2008-04-16	Draft	Fifth draft
1.0	2008-04-24	Final	Final version, niternal reviewers' comments incorporated

1 Executive Summary

This document describes the activity carried out in the Job Submission and Job Monitoring task during the second 12 months of the project; in particular, this report describes the work done from project month 12 (May 2007) to the end of project month 24 (April 2008). The aim of the Job Submission and Monitoring task is to analyze and implement two OGF specifications of standard interfaces for Job Submission and Monitoring services. Specifically, the OGSA—Basic Execution Service (BES) and Job Submission Description Language (JSDL) are standard interfaces for describing and managing computational jobs in Grid systems. The JRA1—Job Submission and Monitoring task within the OMII-Europe project has two main goals: the first one was the analysis of these specifications, identifying and proposing possible extensions which can be useful for OMII-Europe infrastructures. The second goal is the implementation of OGSA—BES and JSDL interfaces in the different Grid middleware platforms, namely gLite 3.1, UNICORE 6, and Globus GT4.

During the first year of the OMII-Europe project, we concentrated on the analysis and extension of the OGSA—BES and JSDL specifications. This work has already been documented in the milestone documents [MJRA1.7] [MJRA1.17] [MJRA1.8] and in the first year deliverable [DJRA1.7].

During the second year of the OMII-Europe project the JRA1/Job Submission and Monitoring task implemented the OGSA—BES and JSDL specifications into the CREAM Computing Element used in the gLite 3.1 middleware, and into the UNICORE6 system. Implementation of BES and JSDL into Globus GT4 was done by the JRA3/Infrastructure Integration task, as they had more expertise with Globus. We implemented the latest available versions of the abovementioned specifications, namely OGSA—BES version 1.0, and JSDL version 1.0. Both the BES and JSDL specifications are at their final revisions.

In the first year deliverable [DJRA1.7] we observed that the OGSA—BES specification (which was then available only as a working draft) seemed to be mature enough to be taken into consideration, as it provided a reasonable set of features. Given our experience with the implementation of the final BES/JSDL specifications, we can confirm our previous analysis. Nevertheless, we suffered from missing features or limitations which are currently being addressed in the feedback process of OGF. Another issue is the definition of a minimum security profile which a BES endpoint must support. In OMII-EU we adopted VOMS-SAML (developed by the JRA1/VOMS task) as a way to transfer users credentials to the BES endpoints. The second major limitation we observed is the lack within the JSDL specification of an appropriate data staging mechanism. Both these limitations will be discussed in more detail in the conclusions of this document.

The results of the JRA1/Job Submission and Monitoring task are the software components CREAM-BES and the OGSA-BES implementation for UNICORE 6. These software components (together with GLOBUS-BES extension of Globus GT4 developed by the JRA3/Infrastructure Integration task) have been put into the OMII-EU repository, so that users can access and install them. In addition, the OGSA-BES implementation of UNICORE 6 is available on UNICORE@SourceForge, which is the major UNICORE platform for release management.

Maintenance and development of CREAM-BES and UNICORE 6 will continue after the end of the OMII-EU project. UNICORE 6 is the new major release of the UNICORE system and includes by default the OGSA-BES interface. The OGSA-BES interface for CREAM, which has been developed by OMII-EU as an add-on to the standard gLite 3.1 CREAM service, will be merged into the main CREAM service, so that upcoming releases of CREAM will support OGSA-BES and JSDL by default. The OGSA-BES implementation in UNICORE 6 will be maintained by the UNICORE community, while the development of CREAM will continue as part of the EGEE-3 project in the gLite 3.1 middleware. At the time of writing, it is evaluated how the GLOBUS-BES interface implementation of JRA3 can be fed back to the Globus Community as well.

2 Table of Contents

3 Introduction

The JRA1/Job Submission and Monitoring task is part of the OMII-EU JRA1 Activity (Re-engineering of services). Its aim is to reengineer existing job submission components from three major Grid systems (gLite, UNICORE and Globus) to support emerging standard interfaces for job description and management.

Job submission services are among the most fundamental services provided by Grid systems. They allow users to submit computational jobs to the Grid, manage them and monitor their progress. Job management involves suspending, resuming, or removing a Grid job. Job monitoring involves checking the current status of the job. Moreover, job submission services also provide operations to handle the service itself, e.g. disabling further job submissions, check the service capabilities and so forth.

Different Grid middleware platforms have different interfaces for the job submission service, which makes interoperability extremely difficult. This is clearly undesirable for at least two reasons:

- from the software engineering point of view, different, non-interoperable job submission components make software reuse very difficult, if not impossible. So, different Grid middleware developers often rewrite code which basically does the same thing over and over again;

- from the resource usage point of view, non-interoperable job submission components make resource sharing among different middlewares (and thus different natures of computational resources, i.e. massively parallel vs. embarrassingly parallel resources) extremely difficult. It is often not possible for one Grid to use available resources on a different Grid just because the job submission components cannot directly interoperate.

Traditionally, interoperability issues have been solved using ad hoc, point-to-point solutions (mostly involving *adapter* components), which lead to the well-known short-term interoperations in the past. Thus, these ad hoc solutions do not guarantee interoperability, but only *interoperation.* In our vision, sustainable interoperability has a much broader scope and can only be achieved using open standard interfaces.

In order to address these issues the Open Grid Forum (OGF) started the development of two specifications: the Job Submission Description Language (JSDL) [JSDL] and OGSA—Basic Execution Service (BES) [BES] specification. JSDL is an XML-based notation for describing computational jobs, while OGSA—BES is a Web Services-based standard interface for a Job Submission and Monitoring service.

During the first year of the OMII-EU project, the JRA1/Job Submission and Monitoring task analysed the OGSA—BES and JSDL specifications, in order to identify any missing functionality; at that time JSDL was available as version 1.0 final specification, while OGSA—BES was still a working draft specification. Both JSDL and OGSA—BES were considered powerful enough to be implemented in Grid middlewares [MJRA1.7][MJRA1.17].

During the second year of the project, the OGSA—BES specification reached a stable state as final specification (version 1.0), thus becoming suitable candidates for actual implementation. This implementation of BES and JSDL into the OMII-EU middlewares was the goal of the JRA1/Job Submission and Monitoring task for the second year of the project.

This task has strong relations with the JRA3—Task 2 Infrastructure Integration and JRA3—Task 1 Security tasks. The implementation of the OGSA—BES interface in gLite, UNICORE and Globus is an important step to achieve interoperability between these different systems. OGSA—BES-enabled job submission components can be used as standard components which can be easily integrated (plugged-in) into existing infrastructures. This in turn allows for the usage of this components in scientific use cases, also known as interoperability scenarios in JRA3 - Task2 Infrastructure Integration.

The JRA3—Task 1 Security gave guidelines for authentication and authorization mechanisms which will enable secure job submissions to OGSA—BES services. In fact, as already observed in [MJRA1.17], the OGSA—BES specification does not mandate any security mechanism. It is then essential to couple OGSA—BES with suitable security mechanisms in order to propagate (and enforce) authentication and authorization information along with job submission and management requests.

In OMII-EU, we decided to use the newly developed SAML-based Virtual Organization Membership Service (VOMS) (provided by the JRA1/VOM task) as security set-up to be used with OGSA-BES-enabled job submission components. Thus, both UNICORE 6 and the CREAM-BES services have been made capable of interacting with the VOMS-SAML service to request and validate user credentials.

Another important interaction within OMII-EU is between the JRA1/Job Submission task and the JRA2/Identification of new services activity. Already in the beginning of the project, JRA3 – Task2 in collaboration with JRA2 identified missing component capabilities (integration phase 1 in JRA3-Task2). Hence, it turned out that the resource description model provided by the OGSA—BES specification was inadequate to fully describe actual computational resources and job requirements. One of the “missing pieces” identified by the JRA2 activity during the first year of the project was precisely a standard mechanism for resource representation, to be used also (but not only) in BES—compliant job submission components. The GLUE2.0 schema developed within JRA2 is being implemented in the UNICORE 6 and CREAM-BES systems as an extension for resource modelling within BES services. This is being undertaken in collaboration with JRA3 – Task 2 that requires this feature for one scientific scenario in the context of the EU-IndiaGrid interoperability scenario.

Finally, during the second year of OMII-EU we also had strong interactions with the JRA1/Portal Interface task, as the OMII-EU portal application was extended to be able to submit jobs to OMII-EU reengineered, OGSA-BES-enabled services.

4 Summary of first year

During the first year of the OMII-Europe project, the JRA1—Job Submission and monitoring task analysed and started implementations of the JSDL and OGSA—BES specifications. JSDL is a standard notation for describing computational jobs; OGSA—BES is a specification for an interface to job submission and management services. OGSA—BES-compliant services accept JSDL documents and instantiate new jobs according to the received request. These services also provide operations for job management such as cancelling, suspending, resuming and status querying of jobs.

The JSDL and OGSA—BES specifications are an extremely important step towards interoperability between Grid systems. Together with the outcomes of the JRA3 activity, the OMII-Europe project could provide the building blocks to create a “federated system” of Grids sharing resources to end-users that require resources in multiple Grids. We have numerous scenarios where jobs originating on a Grid system can be transferred (provided that the users which originated them are authorized to do so) to another Grid in the federation seamlessly, as if they were executed and managed on the originating system. Initial scenarios like that are tested within the multi-platform Grid infrastructure of JRA3.

During the first year of the project we assessed the feasibility of implementing OGSA—BES and JSDL in order to achieve interoperability. The result of this assessment was positive: OGSA—BES and JSDL were complete enough to be implemented in the gLite, UNICORE and Globus middlewares as alpha prototypes. Furthermore, the OGSA—BES implementation provided by the GridSAM job execution service was integrated into UNICORE 6 as a virtual site [MJRA1.8]. An important missing piece (security) was investigated within the JRA3—Security activity of the OMII-Europe project. It turned out that VOMS is one important cornerstone of the mentioned security profile and thus initial development in JRA3—Task 2 focus on an interoperability of the OGSA—BES interfaces with VOMS.

5 Review recommendations for second year

During the first review, some recommendations were made to the JRA1 activity. In this section we describe the recommendations and how they were addressed during the second year of the project.

R: "Better align the decisions about the selection of versions of external middleware across the project activities. Avoid the use of different versions of a middleware whenever possible."

As far as the JRA1/Job Submission task is concerned, the reference middleware versions have been defined as gLite 3.1 and UNICORE 6. Both are the most recent versions of the gLite and UNICORE middlewares, respectively, and thus are the most appropriate platforms where additional features can be implemented. This guarantees that the implementation effort from OMII-EU will not get lost, as gLite 3.1 and UNICORE 6 will be subject to further developments by the respective communities even after the end of the OMII-EU project.

R: "Consider the implementation of a project wide inventory that defines the current version to be used by all OMII-Europe developments."

The project-wide versions of the middlewares have been accepted in accordance with the centralised OMII-Europe middleware version inventory.

R: "Refer clearly in documentation to specific versions not only to the basic name (e.g., UNICORE 6 instead of UNICORE)."

This has been done after the decision to adopt gLite 3.1 and UNICORE 6 as reference platform for implementing the OGSA-BES/JSDL specifications. All the documentation produced in the second year of the OMII-EU project clearly states the adopted versions of the gLite and UNICORE middlewares.

R: "In future versions of the yearly management report of the re-engineering activity, describe the issues faced and how they were addressed with particular emphasis on cross-activity issues."

The whole project adopted a new template for the monthly reports, with specific sections for describing issues and how they were addressed.

6 Objectives for second year

In the second year of the project, the goal was to provide full support for OGSA—BES and JSDL, with additional support for security (authentication and authorization), by using the outcomes of the JRA3 activity. OGSA—BES and JSDL compliant job submission and management services developed within JRA1/Job Submission will be used by the JRA3—Task 2 task for the important interoperability tests and in so-called interoperability scenarios that demonstrate how the components work in real scientific use cases (with EGEE and DEISA). Implementation of OGSA—BES and JSDL has been done for the gLite 3.1 CREAM Computing Element and in the UNICORE 6 middleware. Implementation in Globus has been done by the OMII-EU JRA3/Infrastructure Integration task, as they had both the required expertise and also could devote some funded effort on the implementation.

We identified some major users for the reengineered components: firstly, other OMII-EU activities and tasks depend on the outcome of the JRA1/Job Submission task. Most notably, the JRA3 activity with a wide variety of use cases always requires the OGSA-BES implementation as one fundamental component. Thus, as already stated, job submission components are important elements for the JRA3/Infrastructure Integration task; moreover, job submission components will be assessed both by the Quality Assurance activity and by the JRA4/Benchmarking activity.

The JRA1/Portal Interface developed specific modules for OMII-EU portal applications based on GridSphere to support submission to OGSA-BES endpoints. Also, the JRA1/Component Exchange task used OGSA-BES-enabled components in the CROWN Grid Scheduler. Moreover, as stated in the official Description of Work, re-engineered components will be returned to the originating middlewares so that the final users of these components will be the middleware developers and the middleware final users. In fact, the European A-WARE project [AWARE], for instance, has already used the OGSA-BES implementation in UNICORE 6 for their bus-oriented submission system. The adoption by Grid middleware providers will have the positive effects of guaranteeing the sustainability of the JSDL/BES implementation effort past the end of the OMII-EU project, and that other projects such A-Ware can use the outcome of it.

7 Approach

The first year of the OMII-EU project was essentially used in the analysis of the BES and JSDL specifications, contributing to the standardization process and planning their deployment into the UNICORE 6 and gLite 3.1 middlewares. The actual implementation of these specifications took place during the second year.

In more detail, the goals for the second year have been the following:

1. add support for the OGSA—BES version 1.0 and JSDL version 1.0 specifications into UNICORE 6 and into the CREAM job management service from gLite 3.1;

2. implement a common authentication mechanism on top of the OGSA-BES interface, in accordance with the outcome of the JRA3/Security task; This is extremely important to allow for real usage within infrastructures like EGEE and DEISA in the near future;

3. provide technical support to the internal users of the BES services (the JRA1/Portal, JRA1/Component Exchange, JRA3/Infrastructure Integration, and JRA4/Benchmarking tasks).

Given the limited manpower available, the likelihood of changes in both the specifications or the requirements for the reengineered components, and the need to provide alpha or beta releases as soon as possible for dissemination or testing purposes, we decided not to rely on traditional rigid software development processes (such as the *waterfall* software development model).

Instead, we informally adopted an *evolutionary programming* approach which consists of fast incremental development of new features of the system, such that these new features can be quickly tested and deployed (see Illustration 1). Similarly, for the documentation we relied on the task wiki page [JRA1WIKI] to keep all the relevant information up-to-date.

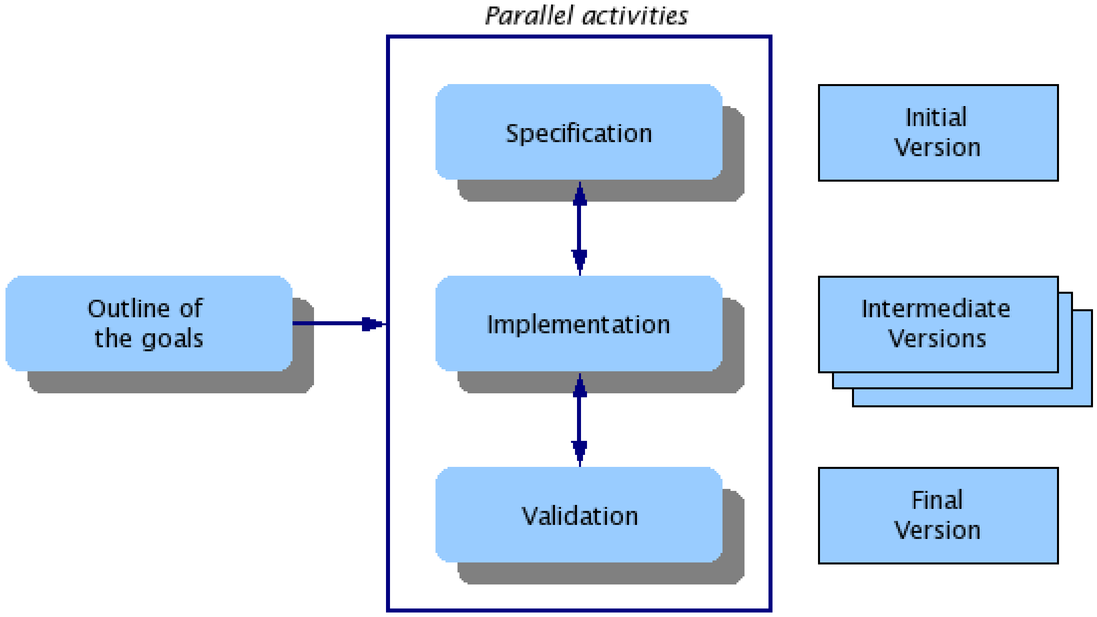

Illustration 1: JRA1/Job Submission internal process view

Illustration 1 depicts the internal process view of the JRA1/Job Submission and Monitoring task. The general goals of this task were stated in the OMII-EU Description of Work; in particular, according to the Milestones [MJRA1.9] and [MJRA1.10] we were required to implement the JSDL and OGSA-BES specifications at project months 18 and 20 respectively.

Starting from these goals, the tightly coupled activities of working out the system specification, implementing the specifications and validating the implementation has been carried out. This concurrent activity resulted in the incremental release of software components, which were made available through the OMII-EU software repository (http://repository.omii-europe.org/). In turn, JRA3 – Task 2 tracked the outcomes of this activity to establish a multi-platform infrastructure for interoperability tests and use cases in conjunction with other components of OMII-Europe.

During the implementation phase we encountered some issues, some of which were middleware-specific, and others were related to the tool we used. On the CREAM-BES side, one of the main difficulties was the lack of a *delegation* feature in the BES specification. The gLite middleware is heavily based on the concept of delegation: the user can temporarily (and securely) transfer his/her credentials to a specific service, which can subsequently issue requests on behalf of the user. This is extremely useful in the Computing Element, such as CREAM: the user delegates credentials to the CE, which can subsequently stage data in/our on behalf of the user. In this way it is possible for the CE to access files residing on a separate Storage Element using secure authentication.
The OGSA-BES specification is—by design—completely agnostic with respect to security considerations; for this reason, and for compatibility with other middlewares which do not make use of proxy credential delegation, it was necessary to modify CREAM-BES so that the underlying CREAM engine worked without a delegated proxy. Since the current version of CREAM-BES only

supports plain FTP to stage data, it turned out that a valid delegated proxy is not strictly necessary in the main CREAM engine. So, we were able to let CREAM-BES create a “fake” (but structurally correct) delegated proxy just to make the underlying CREAM backend work as expected.

Since UNICORE 6 integrates SAML-based delegation in such a way that it is decoupled from the OGSA-BES implementation full credential delegation is possible. However, UNICORE on the other hand encountered problems in the state model mapping between the UNICORE 6 execution backend and the specified OGSA-BES state model. This, however, was solved by using both state models optionally for the service.

We also encountered problems related to interoperability issues among different Web Service tools. CREAM (and CREAM-BES) has been developed using the Apache Axis2 toolkit [AXIS2], while UNICORE 6 is based on XFire [XFIRE]. It turned out that both tools in some rare circumstances generate wrong code used to serialize/deserialize the SOAP messages. Other times there just was an XML namespace mismatch which prevented an otherwise valid request to be interpreted from the other side. All these problems were addressed thanks to the cooperation of all the developers in this activity.

We also encountered significant difficulties while dealing with SAML assertions in the OGSA-BES services. In particular, there have been issues when trying to validate the signature of SAML assertions. The problem has eventually been addressed and solved thank to the cooperation of the JRA1/Job Submission task with JRA1/VOM, JRA1/Portal Interface, JRA1/Accounting and JRA3 activities.

8 Community involvement

The OGSA-BES implementations in the various middlewares are quite central to the whole project, adopters in the overall Grid community as well as the overall standardization process in OGF. The outcomes of this activity have been used in interoperability demonstrations of the High Performance Computing – Basic Profile (HPC-BP) at Supercomputing 2006 and 2007.

Also, the implementations occupy a central position in the use cases of JRA3, namely the WISDOM interoperability scenario, the EU-IndiaGrid interoperability scenarios, and in the recently started EUFORIA interoperability scenario. Installations of these components are now available at numerous sites within these scenarios. All of them use the fact that the OGSA-BES components in gLite and UNICORE 6 can bridge EGEE and DEISA. This means that the components (along with a suitable security set-up such as the common security profile of JRA3) allow for a seamless usage of massively parallel resources (i.e. DEISA) and compute farm resources (i.e. EGEE).

More recently, the European A-WARE project has started to use the component for their bus-oriented workflows that submit computational jobs to gLite and UNICORE 6.

In addition, partners in the German D-Grid (after an OMII-Europe training event) are evaluating the OGSA-BES components for their special use cases within the German National Grid. This is especially interesting since this Grid infrastructure is deploying UNICORE, gLite and Globus in parallel. Hence, they require interoperability for these systems and thus OMII-Europe components are well suited to this infrastructure.

9 Progress

The main goal the JRA1/Job Submission and Monitoring task during the second year is to support the OGSA—BES and JSDL specifications into the OMII-EU middlewares. The results of this activity are the re-engineered job submission components, which have to be made available through the OMII-EU software repository.

These goals of the JRA1/Job Submission and Monitoring task correspond to the following project milestones which were due at M18 and M20 respectively. The outcome of the milestones has been the availability of the components (in alpha and beta stage) in the repository.

Milestone	Due on	Delivered on
MJRA1.9: *"Implementation of JSDL into OMII-Europe middleware together with the identified extensions"*	M18	M18
MJRA1.10: *"OMII-Europe supports BES plus the required extensions"*	M20	M20

An overview of the results of the milestones is provided in the following sections.

9.1 MJRA1.9: Implementation of JSDL into OMII-Europe middleware

The Job Submission Description Language (JSDL) [JSDL] is an XML-based notation for describing the requirements of computational jobs for submission to Grid environments. The JSDL notation is defined by means of a normative XML Schema that facilitates the expression of those requirements as a set of XML elements. The JSDL specification is motivated by the need to achieve interoperability between different Grid job management systems; in fact, it is not uncommon that the same user community uses different Grid systems at the same time, each with its own notation for describing jobs. In this scenario, a common, standardized notation such as JSDL is clearly desirable.

Illustration 2 shows the typical *interoperability* scenario for job submission using the BES/JSDL specifications. Note that different BES/JSDL compliant clients are able to submit jobs to any service exposing a BES interface [BES]; this is different from the typical *interoperation* approach between different Grid systems, which on the other hand would be based on ad hoc, point-to-point adapters.

Note that, in general, the same execution service may expose a ”legacy” interface and a BES-compliant one. This is useful both for backward compatibility purposes, and because the BES/JSDL specifications allow for very basic functionalities only; for example, more complex job types (e.g., parametric jobs, or job collections with dependencies) are outside the scope of the current JSDL specification, while they are supported by many execution services through their legacy interfaces.

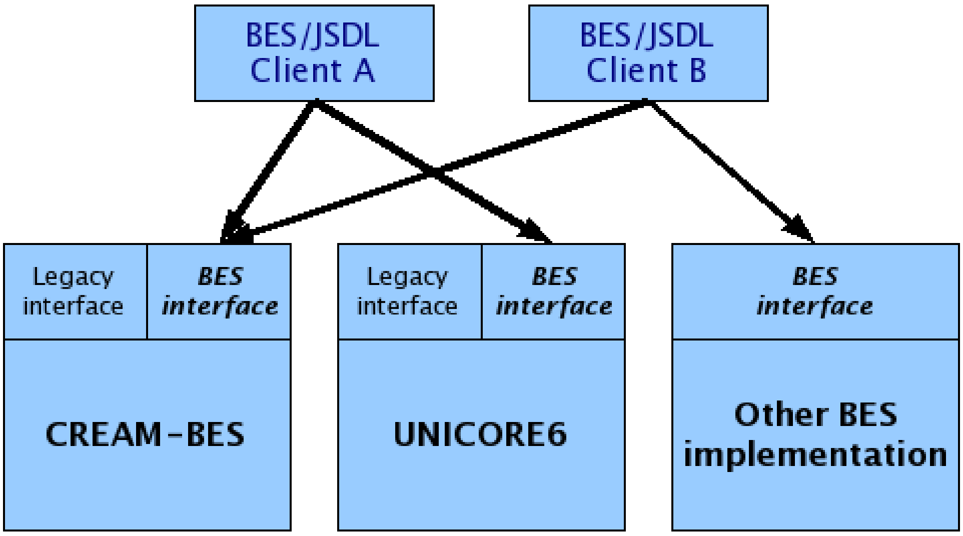

Illustration 2: Interoperability using BES/JSDL

CREAM-BES is a BES-enabled version of the CREAM Computing Element [CREAM]. CREAM is being developed by the EGEE/EGEE2 project for inclusion in the gLite 3.1 middleware distribution. CREAM is a Java-based computing execution service, which exposes a Web Service-based interface.

The “legacy” CREAM interface is not BES, and its standard job description notation is based on Condor classads rather than JSDL. The reason is that the development of CREAM started before the standardization work which led to BES and JSDL started. Moreover, the requirements for inclusion of

CREAM in other gLite 3.1 middleware included support for the classad-based Job Description Language (JDL), for compatibility with the rest of the gLite 3.1 infrastructure.

One of the goals of the OMII-Europe project was to reengineer some of the existing job execution services to include support for the BES and JSDL specifications. As far as CREAM is concerned, we decided to add BES support on the top of the core of the existing CREAM software, as to keep the code for the legacy service effectively separated with the BES/JSDL code (see Illustration 3).

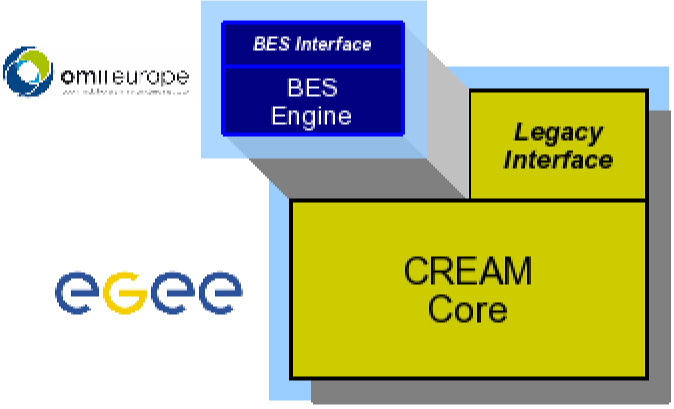

Illustration 3: CREAM-BES structure

Illustration 3 depicts the structure of the CREAM-BES server. As can be seen, CREAM-BES is made of two distinct components:

1. the legacy CREAM server (in yellow); this is exactly the same software used in gLite 3.1, and needs to be installed according to the standard CREAM installation procedure (see the CREAM-BES installation procedure described at http://grid.pd.infn.it/omii/cream-bes). The legacy CREAM is being developed by the EGEE-2 project for the gLite 3.1 middleware.
2. the BES interface with the corresponding implementation for CREAM. Implementation of the BES interface for CREAM (in blue in the picture) is being developed by the OMII-Europe project. The implementation of the BES interface uses the CREAM core; note that the BES interface coexists alongside with the legacy CREAM one (so both interfaces can be used at the same time).

There are some advantages in keeping the core CREAM physically separated from the BES interface and implementation. Specifically:

1. The BES/JSDL specifications evolved at a different rate than the development of the legacy CREAM service which is being included in gLite 3.1. It was thus necessary to decouple the development of the BES/JSDL interface from that of the CREAM core.
2. The CREAM-BES/JSDL interface does not require any modification on the CREAM core, so that the new interface does not introduce additional bugs, and does not interfere with the testing of CREAM as done by the EGEE-2 certification team.
3. Development of the BES/JSDL interface can make use of the latest CREAM version available. We use the ETICS [ETICS] system for building CREAM and CREAM-BES. The

> CREAM-BES ETICS component includes the latest version of the legacy CREAM as an external component. This means that every build of CREAM-BES uses the latest version of the CREAM core, so that CREAM-BES automatically gets all the bug fixes and improvements from the current CREAM release.

CREAM-BES currently implements JSDL version 1.0; furthermore, it implements the HPC-P (HPC Profile Application Extension) [HPCP].

As far as UNICORE is concerned, a study of the the OGSA-BES interface reveals that it is similar to the Target System Service (TSS) and Job Management Service (JMS) of the rather proprietary UNICORE Atomic Services (UAS) of UNICORE 6.

While the UAS does have WS-RF compliant message exchanges and the submit operation takes a JSDL document as parameter, the syntax of the UAS operations is proprietary. The OGSA-BES interfaces offer UNICORE 6 a standardized syntax for Web service operations dealing with job control and management which can be implemented on top of the same execution backend XNJS as the UAS.

The Web service layer is well isolated from the lower-level execution engine. All OGSA-BES invocations are transmitted to the server-side within the SOAP body, except the pieces of information within the SOAP header that are used to address specific job WS-Resource instances and also the security tokens (i.e. SAML assertions, see JRA1 VOM Activity). As shown in Illustration 4, the integrated OGSA-BES UNICORE 6 interface has three services: Activity, Factory and Management services.

The BES Factory Service is responsible for the creation and control of a set of activities (described using JSDL). In the UNICORE 6 implementation, this Web service interface uses the newly developed and enhanced (X)NJS as an execution backend that provides significant performance and scalability improvements over UNICORE 5. The CreateActivity(JSDL) operation of the BES-Factory Service leads to the creation of a job resource within the NJS that represents the computational job described by the JSDL.

At the Web service level, this resource can be controlled and monitored by using the BES Activity Service operations, e.g. by using the Terminate() or GetStatus() operations. In addition, the OGSA-BES implementation for UNICORE includes a BES Management Service. This service improves the functionality of UNICORE in terms of remote administration. An administration client, for instance, can be used to access this service and control whether new jobs can be submitted to UNICORE or not. To support this, the BES operations StopAcceptingNewActivities() and StartAcceptingNewActivities() are supported by UNICORE 6.

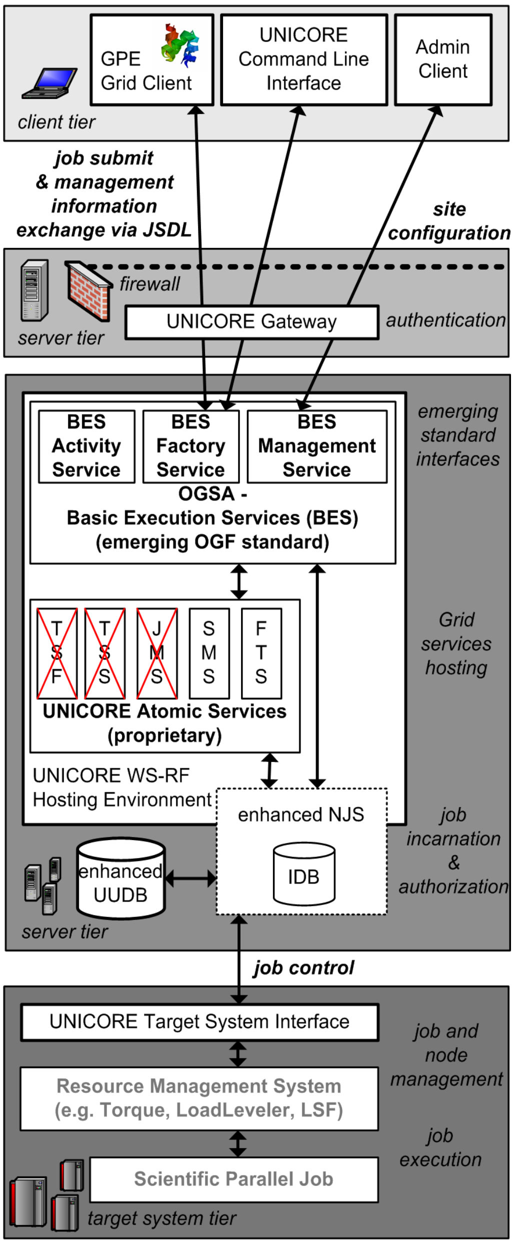
GPE Grid Client
UNICORE Command Line Interface
Admin Client
client tier
job submit & management information exchange via JSDL
site configuration
firewall
UNICORE Gateway
authentication
server tier
BES Activity Service
BES Factory Service
BES Management Service
OGSA - Basic Execution Services (BES) (emerging OGF standard)
emerging standard interfaces
TSF
TSS
JMS
SMS
FTS
UNICORE Atomic Services (proprietary)
Grid services hosting
UNICORE WS-RF Hosting Environment
enhanced NJS
IDB
enhanced UUDB
job incarnation & authorization
server tier
job control
UNICORE Target System Interface
Resource Management System (e.g. Torque, LoadLeveler, LSF)
Scientific Parallel Job
job and node management
job execution
target system tier

Illustration 4: UNICORE 6 with a deployed OGSA–BES services. In OMII–Europe the proprietary UNICORE Atomic Services are partly replaced by OGSA–BES services.

UNICORE 6 is also compliant with the HPC-Basic Profile. This profile specifies the usage of the OGSA-BES interface in conjunction with certain extensions to JSDL. UNICORE 6 implements the HPC Application Extension Specification [HPCP] as an extension to JSDL that is used to describe an executable running as an operating system process. Basically it shares much in common with the JSDL POSIXApplication, but removes some of the features that present barriers to interoperability by using the XML element jsdl-hpcpa:Executable and other similar elements.

The Resources JSDL element describes the resource requirements of the job. CREAM-BES is able to check whether the requirements can be satisfied; to do so, CREAM-BES checks the capabilities of the attached batch queues. Such capabilities can be queried using the gLite infoprovider scripts, or by querying the OpenPegasus-based infoprovider. Unfortunately, the infoprovider scripts do not report all the available capabilities of each batch queue; thus, there are some requirements which CREAM-BES is unable to satisfy, because it cannot verify the relevant parameter. For example, the IndividualNetworkBandwidth JSDL element is used to specify the bandwidth requirement of each individual resource in bits per second. However, none of the currently available information providers report such information, thus CREAM-BES is unable to enforce that requirement.

The BES CreateActivity operation is used to submit a new job to the BES service. If the requirements for the job cannot be satisfied, the BES service raises an UnsupportedFeatureFault fault; the fault message contains the resource requirements which could not be satisfied. In [MJRA1.9] we described the support status for each of the requirement elements of the JSDL specification. While both CREAM-BES and UNICORE 6 support a subset of these requirements, implementation of all of them cannot be done at the moment without a complete rewrite of the underlying information system of CREAM and UNICORE. In fact, the unsupported requirement elements are those whose value cannot be probed by the specific middleware platform, and thus the job submission service is unable to guarantee that those elements are honoured if specified inside a valid JSDL document.

During the implementation of the OGSA-BES interface (which includes JSDL) it became clear that BES covers less job execution functionality than either CREAM or UNICORE. In most job management use cases, data needs to be staged-in before execution and after the job execution the results need to be staged-out to a dedicated (permanent) location. Therefore, file transfers and storage management have a close relationship to every job management system on computational Grids or e-Science infrastructures.

The OGSA-BES specification has left this concern to implementers. But both file transfer and storage management are needed if complete JSDL descriptions are to be supported. Therefore, the OGSA-BES implementation within UNICORE 6 also provides a file transfer service (FTS). The FTS provides Streamable ByteIO (SByteIO) and Random ByteIO (RByteIO) data access mechanisms to support job submissions that rely on staged data. These specifications are standardized by the OGF and thus lay the foundation for data staging capabilities in interoperability scenarios and cross-Grid use cases. In CREAM-BES, a simple file staging mechanism has been implemented based on plain FTP; Other, more secure and scalable alternatives will be implemented in the future.

To sum up, the core JSDL elements matched quite well with the existing functionalities provided by the native job description notations used by UNICORE and CREAM; thus, for simple jobs, JSDL provides most of the features which were available in the legacy notations. However, there are also features provided by JSDL which cannot be easily implemented without rewriting significant amounts of existing code. In particular, JSDL allows users to specify several attributes for the execution environment on which the job will be executed (e.g., file system layout, local user ID, network bandwidth requirements etc.) which cannot be directly controlled by UNICORE or CREAM execution services. Such features are thus currently not supported in UNICORE 6 and CREAM-BES.

On the other hand, we observed that many existing execution services (including UNICORE and CREAM) provide additional functionalities which are not covered by the JSDL specification; the most notable one is the ability to describe and execute job collections with dependencies described as DAGs (Directed Acyclic Graphs). Since it is quite natural to leverage the power of the Grid to execute such "complex" kind of jobs, we are currently investigating solutions to overcome these limitations of JSDL. Most likely, these limitations will be addressed by developing appropriate JSDL extensions (using the standard JSDL extension mechanisms).

9.2 MJRA1.10: OMII-Europe supports BES

The OGSA Basic Execution Service specification defines a Web Service interface for submission and management of computational jobs (called *activities*) on execution services. The BES specification defines two Web Service port-types as follows (the third Activity portType has no operations and is thus available for proprietary extensions):

- the **BES Management** port-type is used to control the BES service itself; it contains two operations for starting and stopping accepting new submissions. This port-type is normally used by the BES service administrators.
- the **BES Factory** port-type is used by ordinary clients to request creation of new activities, manage and monitors the activities, and ask for the capabilities of the BES service.

The following table contains the list of all defined BES operations, divided according to the port-types they belong to:

BES-Management Port-type	
`StopAcceptingNewActivities`	Request that the BES stop accepting new activities
`StartAcceptingNewActivities`	Request that the BES start accepting new activities
BES-Factory Port-type	
`CreateActivity`	Request the creation of a new activity
`GetActivityStatuses`	Request the status of a set of activities
`TerminateActivities`	Request that a set of activities be terminated
`GetActivityDocuments`	Request the JSDL documents for a set of activities
`GetFactoryAttributesDocument`	Request XML document containing BES properties

BES uses JSDL as the notation used to describe activities. The JSDL specification [JSDL] can be used to describe computational jobs and their requirements. The CreateActivity operation takes a JSDL document as a parameter, and instantiates a new activity within the BES service.

As stated in the previous section, in CREAM-BES we added a separate, BES-compliant interface to the legacy gLite 3.1 CREAM service. The interface has access to all the functionalities provided by the legacy CREAM core, so that all the features of CREAM apply to CREAM-BES as well. The BES interface represents a subset of the legacy CREAM one, so implementation of BES was not particularly difficult. The more challenging part was the implementation of the VOMS-SAML based security layer in CREAM-BES.

As already shown in Illustration 4, the integrated OGSA-BES interface of UNICORE 6 consists of three standardized services. The BES-Factory Service is responsible for the creation and control of activities described with JSDL. The invocation of its CreateActivity() operation leads to the creation of a job resource within the enhanced NJS that represents a computational job described by the JSDL. The underlying enhanced NJS accepts JSDL with extensions required by the HPC-BP and thus UNICORE is HPC-BP compliant. This JSDL is converted to Resource Management System (RMS) (e.g. Torque, LoadLeveler) specific commands on the target system.

On the Web services level, the job resources can be controlled and monitored by using the BES-Activity Service, for instance by using the Terminate() or GetStatus() operations. The BES-Management Service can be used to control whether new jobs can be submitted or not. Finally, UNICORE 6 supports the WS-RF rendering of the OGSA-BES specification, which includes that OGSA-BES activity properties of the internal WS-RF resource model are provided and supported by UNICORE 6. Hence, beside the mandatory operations defined in OGSABES, UNICORE 6 also provides operations according to the WS-Resource properties. This allows service properties to be requested according to the WS-RF specification for jobs submitted through the OGSA-BES interface.

Both UNICORE 6 and CREAM-BES support user authentication using X509 proxy certificates with VOMS-SAML headers in the SOAP request, while UNICORE 6 can also use full X509 certificates in conjunction with VOMS-SAML headers in the SOAP messages. As stated above, VOMS extensions are fundamental because access control policies and user mapping is performed also at the VO level. The BES interfaces for UNICORE 6 and CREAM-BES can be accessed using TLS (HTTPS) encrypted connections.

The BES services authenticate the user according to the DN contained in the certificate used during the TLS handshake. Once the user is authenticated, the CREAM-BES and UNICORE 6 services need to associate VO information to that user in order to authorize the use of resources (e.g., batch queues contained in the BES service). VO information is extracted from the SAML assertions which must be contained into the SOAP header used to invoke BES operations. More specifically, in order to access the the BES services, the user needs to perform the following steps:

1. The client requests a SAML assertion from the VOMS-SAML service. The user needs to access VOMS-SAML with appropriate credentials (X509 certificate) in order to get the SAML assertions.
2. The VOMS-SAML service identifies the client from the supplied credentials. It then supplies a SAML assertion (signed with the server certificate in order to guarantee its authenticity).
3. The client inserts the SAML assertions into the SOAP header which is then used to access the BES interface. CREAM-BES and UNICORE 6 extract the SAML assertions (which, among other things, contains the Virtual Organization information of users they belongs to) and decides whether the user is authorized to perform the requested operation.

The situation is summarized in Illustration 5, assuming in this case that the client application is the GridSphere portal developed by the JRA1/Portal Interface task.

The authorization within UNICORE 6 uses Extensible Access Control Markup Language (XACML)-based policies in conjunction with SAML assertions, while CREAM-BES relies on the gJAF framework for SAML-based authorization decisions.

10 Lessons learned

The following lessons have been learned during the second year of the JRA1/Job submission and Monitoring task:

1. JSDL lacks a suitable mechanism for securely staging data. As a temporary workaround, many current BES implementations rely on plain FTP to stage data in and out from the execution node. This is clearly not appropriate (nor secure) for production use.

2. BES currently lacks a minimum agreed security profile. That is security considerations are outside the BES specification, so that one can claim BES compliance without actually being interoperable with a different producer's BES client, because they use different security settings. Whilst the HPC-BP defines initial security, it was identified by the JRA3 activity that this security set-up is not realistic for infrastructure integration purposes. Therefore, a common security profile providing more advanced security set-ups has been defined.

3. Secure data staging probably requires a suitable mechanism for the client to delegate its credentials to the BES service, so that the BES service can execute operations (e.g., access remote files) on behalf of the client. This is heavily used in e.g. gLite (the legacy CREAM exposes a separate delegation port-type which must be used by clients to delegate their credentials), but this is not defined by the BES specification. UNICORE 6, on the other hand, uses SAML delegation and thus third-party delegation tokens are also included in the SOAP header.

4. Web Service technologies, while increasingly common, are becoming quite complex to implement properly. Tooling support for interface generation from WSDL or XML Schemas have various kind of bugs or incomplete support for the underlying specification, which sometimes result in hard-to-debug issues. Full interoperability between services based around different tools (e.g., Axis2 vs. XFire) is not always guarantee without proper workarounds.

5. Having examined the usage scenarios of JRA3, we see a clear advantage in using OGSA-BES for the infrastructures EGEE and DEISA in the near future. This common interface between UNICORE and gLite (and Globus) enables use cases that were previously unrealizable. This ill lead, one hopes, to new breakthroughs in e-Science and thus can be seen as major success in this activity by enabling scientists with seamless access to the infrastructures.

11 Conclusion

The goal of the JRA1/Job Submission and Monitoring task was to re-engineer existing job submission and management components according to the emerging standards. During the second year of the OMII-EU project, we implemented the OGSA Basic Execution Service (BES) version 1.0 and Job Submission Description Language (JSDL) version 1.0 into two existing job management systems for two different Grid platforms, namely UNICORE 6 and the CREAM-BES Computing Element from gLite 3.1.

UNICORE 6 represents the most recent version of the UNICORE system, which means that UNICORE now provides BES and JSDL support out of the box for support for both is part of the core package within UNICORE 6. This package can be obtained from the OMII-Europe repository and UNICORE@SourceForge. Early versions have already been evaluated at some DEISA sites. Also some sites within D-Grid as well as other European projects have already started to work with UNICORE 6 in general and the OGSA-BES implementation in particular.

CREAM-BES represents an extension of the CREAM Computing Element from gLite 3.1; CREAM-BES uses the legacy CREAM core, and adds an additional BES interface to the legacy one (which is still accessible). Future plans include the adoption of updated versions of the BES/JSDL specifications, in order to keep the software aligned with the most recent interface specification available, and the adoption of appropriate standard extensions whenever they are considered useful for the middlewares users. CREAM-BES will also be merged into the main CREAM code branch. This means that, similarly to what has been done with UNICORE 6, future versions of CREAM will support BES/JSDL out of the box, without the need to install additional components. This will ensure that the code developed by OMII-EU will be used and maintained by the CREAM developers in the gLite 3.1 middleware.

Finally, the standardization of the OGSA-BES interface and its integration into the major Grid middleware systems gLite and UNICORE are a major success for Grids, especially within Europe. That means that the fundamental job submission technologies in conjunction with an advanced security set-up (from JRA3) can be used in real production use cases on EGEE and DEISA in the near future.

In addition, the standardization allows end-users to use these two rather different infrastructures seamlessly. The use cases of JRA3 – Task 2 already have shown that the interfaces can be used in real scientific scenarios (e.g. WISDOM, EUFORIA, EU-IndiaGrid).

12 References

[M:JRA1.7] *Definition of the required extensions needed to JSDL to satisfy OMII-Europe requirements*, OMII-EU Milestone Document M:JRA1.7, available on the project site: http://www.omii-europe.org/

[M:JRA1.17] *BES will be evaluated with respect to its adoption in the middleware of the OMII-Europe partners*, OMII-EU Milestone Document M:JRA1.17, available on the project site: http://www.omii-europe.org/

[M:JRA1.8] *GridSAM integration into UNICORE: architecture and implementation available*, OMII-EU Milestone Document M:JRA1.8, available on the project site: http://www.omii-europe.org/

[D:JRA1.9] *The first yearly report on the Job Submission and Job Monitoring task*, OMII-EU Deliverable D:JRA1.9, available on the project site: http://www.omii-europe.org/

[MJRA1.9] *Implementation of JSDL into OMII-Europe middleware together with the identified extensions*, OMII-EU Milestone MJRA1.9

[MJRA1.10] *OMII-Europe supports BES plus the required extensions*, OMII-EU Milestone MJRA1.10

[SHAH07] Z. Ali Shah, P. Andreetto, S. van de Berghe, A. Ferraro, F. Hedman, V. Li, M. Marzolla, Shabaz Memon, Shiraz Memon, M. Riedel, D. Snelling, K. Stamou, A. Streit, B. Twedell, V. Venturi, *Open Standards-based Interoperability of Job Submission and Management Interfaces across the Grid Middleware Platforms gLite and UNICORE*, Proceedings of International Interoperability and Interoperation Workshop (IGIIW) 2007 at 3rd IEEE International Conference on e-Science and Grid Computing, Bangalore, India, December, 2007, IEEE Computer Society, ISBN 0-7695-3064-8, pp. 592—599

[MEMON07] M.S. Memon, A.S. Memon, M. Riedel, B. Schuller, D. Mallmann, B. Tweddell, A. Streit, S. van den Berghe, D. Snelling, V. Li, M. Marzolla, P. Andreetto, *Enhanced Resource Management Capabilities using Standardized Job Management and Data Access Interfaces within UNICORE Grids*, Proceedings of 3rd Workshop on Scheduling and Resource Management for Parallel and Distributed Systems SRMPDS 2007, ICPADS'07 - The 13th International Conference on Parallel and Distributed Systems Hsinchu, Taiwan, December, 2007, Volume 2, 5—7 dec 2007, pp. 1—6, ISBN 978-1-4244-1890-9

[RIEDEL07] M. Riedel, B. Schuller, D. Mallmann, R. Menday, A. Streit, B. Tweddell, M.S. Memon, A.S. Memon, B. Demuth, Th. Lippert, D. Snelling, S. van den Berghe, V. Li, M. Drescher, A. Geiger, G. Ohme, K. Benedyczak, P. Bala, R. Ratering, A. Lukichev *Web Services Interfaces and Open Standards Integration into the European UNICORE 6 Grid Middleware* Proceedings of 2007 Middleware for Web Services (MWS 2007) Workshop at 11th International IEEE EDOC Conference “The Enterprise Computing Conference”, 2007, Annapolis, Maryland, USA, to appear

[AXIS2] Apache Axis2 project, http://ws.apache.org/axis2/

[XFIRE] Codehaus XFire, http://xfire.codehaus.org/

[JRA1WIKI] OMII-EU JRA1/Job Submission and Monitoring task wiki page: http://grid.pd.infn.it/omii/

[OPENSAML] The OpenSAML Toolkit, https://spaces.internet2.edu/display/OpenSAML

[ETICS] ETICS home page, http://etics.web.cern.ch/etics/

[JSDL] A. Savva (editor), Job Submission Description Language (JSDL) Specification, Version 1.0, GGF November 2005, available at http://www.gridforum.org/documents/GFD.56.pdf

[BES] I. Foster et al., OGSA Basic Execution Service, Version 1.0, August 2007. available at http://www.ogf.org/documents/GFD108.pdf

[HPCP] M. Humphrey et al., JSDP HPC Profile Application Extension, Version 1.0, october 2006

[AWARE] An easy Way to Access grid Resources Project, http://www.a-ware-project.eu/