

European Community Sixth Framework Programme

RESEARCH INFRASTRUCTURES
Integrated Infrastructure Initiative

DEISA DISTRIBUTED EUROPEAN INFRASTRUCTURE FOR SUPERCOMPUTING APPLICATIONS

CONTRACT NUMBER 508830

DEISA JRA7 2.1 Recommendations on workflow standards
Deliverable ID: DEISA-JRA7-2.1

REPORT

September / 2004

Project and Deliverable Information Sheet

DEISA Project	Project Ref. No: 508830	
	Project Title: Distributed European Infrastructure for Supercomputing Applications	
	Project Web Site: http://www.deisa.org	
	Deliverable ID: DEISA-JRA7-2.1	
	Deliverable Nature: Report	
	Deliverable Level: PU *	Contractual Date of Delivery: 31 / July / 2004
		Actual Date of Delivery: 9 / September / 2004
EC Project Officer: Mario Ramalho-Natario		

* - The dissemination level are indicated as follows: **PU** – Public, **PP** – Restricted to other 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).

Document Control Sheet

Document	Title: Recommendations on workflow standards	
	ID: DEISA-JRA7-2.1	
	Version: 1.0	Status: Public
	Available at:	
	Software Tool: Microsoft Word 2002	
	File(s): D-JRA7-2.1-Recommendations-workflow-standards.doc	
Authorship	Written by:	Roger Menday (RPM)
	Contributors:	Philipp Wieder (PW), Thomas Seed (TS), Terry Sloan (TMS)
	Reviewed by:	RZG
	Approved by:	DEC

Document Status Sheet

Version	Date	Status	Comments
0.1	27 / July / 2004	First draft	TS: Formatted document
0.2	6 / August / 2004	Draft	RPM: Revision of WS standards
0.3	10 / August / 2004	Draft	RPM: Grid section revision
0.4	12 / August / 2004	Draft	RPM: Misc additions, inc. PW contribution
0.5	16 / August / 2004	Draft	TS: Added WS glossary.
0.6	17 / August / 2004	Draft	TS: Added comments.
0.7	19 / August / 2004	Draft	TMS: Added comments
0.8	23 / August / 2004	Draft	RPM: Review of comments
0.9	25 / August / 2004	Draft	TS: Added PJG comments
0.10	25 / August / 2004	Final Draft	Submitted to RZG and DEC
1.0	9 / September / 2004	Public	TMS: Added RPM's updated section 6 following comments from DEC

Document Keywords and Abstract

Keywords:	DEISA, HPC, Grid, workflow, standards
Abstract:	<p>This document examines Grid standards in the areas of resource management and workflow which are relevant to JRA7 of the DEISA project. We assume a Web services framework for the work in JRA7 and thus we also cover Web services standards, including security.</p>

Copyright notices

© 2004 DEISA Consortium. All rights reserved. This document is a project document of the DEISA project. All contents are reserved by default and may not be disclosed to third parties without the written consent of the DEISA partners, except as mandated by the European Commission contract 508830 for reviewing and dissemination purposes. All trademarks and other rights on third party products mentioned in this document are acknowledged as own by the respective holders.

Table of Contents

Project and Deliverable Information Sheet.....	2
Document Control Sheet	2
Document Keywords and Abstract.....	3
Table of Contents	4
List of Figures.....	5
List of Tables.....	6
1. Introduction	7
1.1 Executive Summary	7
1.2 References and Applicable Documents.....	7
1.3 Document Amendment Procedure	8
1.4 List of Acronyms and Abbreviations	8
2. Background.....	11
3. A Glossary of Web Services Standards and related technologies	14
4. Web Services Standards.....	19
4.1 Web Services Security	19
4.2 Web Services Resource Framework	19
4.3 Web Services Workflow standards.....	20
4.4 OASIS Web Services Distributed Management TC	21
5. Grid Standards.....	22
5.1 Open Grid Services Architecture (OGSA)	22
5.2 Job Specification Description Language (JSDL).....	22
5.3 CIM based Grid Schema (CGS-WG).....	23
5.4 Grid Scheduling Architecture Research Group (GSA-RG)	24
5.5 Workflow Management Research Group (WFM-RG)	24
5.6 Grid Resource Allocation Agreement Protocol Working Group (GRAAP-WG)	24
5.7 OGSA Authorisation WG.....	26
5.8 Simple API for Grid Applications (SAGA-RG).....	26
6. Recommendation	27

List of Figures

Figure 1: JRA7 Software Architecture and HSM layer [1] (page 205).	11
Figure 2: User is confronted with heterogeneous sites.	12
Figure 3: User has a single interface to heterogeneous sites.	12

List of Tables

Table 1: Web Services standards and related technologies glossary 18

1. Introduction

This document, 'DEISA JRA7 2.1 Recommendations on workflow standards', is deliverable DEISA-JRA7-2.1 v1.0 from Task 2.1 'Assess and evaluate DRM/workflow standards' in Work Package 2 of the DEISA JRA7 activity [1].

1.1 *Executive Summary*

The objectives of DEISA JRA7 are to develop a means for coordinating and integrating OGSA (Open Grid Services Architecture)-based services for distributed resource management (DRM) in a heterogeneous environment, the Heterogeneous Service Management (HSM) layer, and to use this to integrate a variety of existing user-level tools to provide the necessary high-level services in:

- authentication, authorisation and accounting;
- job preparation, submission and monitoring;
- data movement for job input and output;
- other areas to be determined by DEISA user requirements.

This document is the deliverable from JRA7 Task 2.2 'Assess and evaluate DRM/workflow standards'. As such it identifies and assesses those Web and Grid standards that may be relevant to the JRA7 objectives.

Section 2 provides some background to JRA7 and the assessment of the standards covered in the document. Section 3 provides a glossary of the many Web services and related technologies mentioned in this document. Sections 4 and 5 assess the various Web and Grid standards of possible relevance to JRA7. Finally section 6 makes recommendations on using Web and Grid standards in JRA7.

1.2 *References and Applicable Documents*

- [1] DEISA Annex I – "Description of Work", November 5th 2003
- [2] DEISA JRA7 "Quality Plan", D-JRA7-1.1 v1.0
- [3] Web Services Security, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
- [4] Web Services Resource Framework, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf
- [5] Grid Laboratory Uniform Environment, <http://www.cnaf.infn.it/~sergio/datatag/glue/>
- [6] CIM based Grid Schema, <https://forge.gridforum.org/projects/cgs-wg/>
- [7] DMTF CIM, <http://www.dmtf.org/standards/cim/>

- [8] GGF Workflow Management Research Group, <http://www.isi.edu/~deelman/wfm-rg/>
- [9] GGF Grid Resource Allocation Agreement Protocol WG, <https://forge.gridforum.org/projects/graap-wg/>
- [10] WS-Agreement specification, <https://forge.gridforum.org/projects/graap-wg/document/WS-AgreementSpecificationDraft.doc/en/8>
- [11] WS-AgreementNegotiation specification, <https://forge.gridforum.org/projects/graap-wg/document/WS-AgreementNegotiationSpecificationDraft.doc/en/1>
- [12] SAGA Research Group, <https://forge.gridforum.org/projects/saga-rg/>
- [13] UniGrids Project, <http://www.unigirds.org/>
- [14] NaReGi project, <http://www.naregi.org/>
- [15] GGF Common Management Model Working Group, <http://forge.gridforum.org/projects/cmm-wg/>
- [16] Global Grid Forum, <http://www.ggf.org/>
- [17] JSDL working draft, <https://forge.gridforum.org/projects/jsdl-wg/document/draft-ggf-jsdl-spec/en/8>
- [18] GGF Job Submission Description Language Working Group, <https://forge.gridforum.org/projects/jsdl-wg/>
- [19] GGF Grid Scheduling Architecture Research Group, <https://forge.gridforum.org/projects/gsa-rg/>
- [20] GGF GSA-RG Use Cases, https://forge.gridforum.org/docman2/ViewCategory.php?group_id=133&category_id=714

1.3 Document Amendment Procedure

The document procedure is covered in the Quality Plan [2](Section 5.6).

1.4 List of Acronyms and Abbreviations

BP4WS	Business Process Execution Language for Web Services
BSS	Batch Sub System
CGS-WG	CIM based Grid Schema Working Group
CIM	Common Information Model

CMM-WG	Common Management Model Working Group
DEISA	Distributed European Infrastructure for Supercomputing Applications
DMTF	Distributed Management Task Force
DRM	Distributed Resource Manager (Multi-cluster/site BSS)
EC	European Community
FP6	Sixth Framework Programme
GGF	Global Grid Forum
GLUE	Grid Laboratory Uniform Environment
GSA-RG	Grid Scheduling Architecture Research Group
HPC	High Performance Computing
HSM	Heterogeneous Service Management
HTTP	HyperText Transport Protocol
HTTPS	HTTP over SSL
IETF	Internet Engineering Task Force
JRA	Joint Research Activity
JSIM	Job Submission Information Model
MOWS	Management of Web Services (WSDM specification)
MUWS	Management using Web Services (WSDM specification)
OASIS	Organization for the Advancement of Structured Information Standards
OGSA	Open Grid Services Architecture
OGSI	Open Grid Services Infrastructure
PKI	Public Key Infrastructure
QoS	Quality of Service
SOAP	XML-based protocol for exchanging information
SSP	Secure Socket Layer
TC	Technical Committee
TLS	Transport Level Security
UDDI	Universal Description, Discovery and Integration
UML	Unified Modelling Language
URI	Uniform Resource Indicator
W3C	World Wide Web Consortium
WSA	Web Services Architecture
WSBPPEL	Web Services Business Process Execution Language

WS-CAF	Web Services Composite Application
WS-CDL	Web Services Choreography Description Language
WS-CF	Web Services Coordination Framework
WSDL	Web Services Description Language
WSDM	Web Services Distributed Management
WSFL	Web Services Flow Language
WSI	Web Services Interoperability
WS-JDML	Web Services Job Description Markup Language
WSTXM	Web Services Transaction Management
XML	eXtensible Markup Language

2. Background

The aim of JRA7 is to define a Heterogeneous Service Management (HSM) layer and then integrate functionality from existing tools to provide a user-transparent view of the DEISA heterogeneous grid [1] (page 204). This HSM layer is shown in Figure 1 as the blue ellipses. The HSM can be regarded as a suite of suitable heterogeneous resource management interfaces for the DEISA Grid. These interfaces are based on existing and emerging standards from the Web services and Grid communities ([1], page 204). This layer will allow a single, common architecture for enabling such scheduling functions as advanced reservations, co-scheduling and agreement over qualities of service.

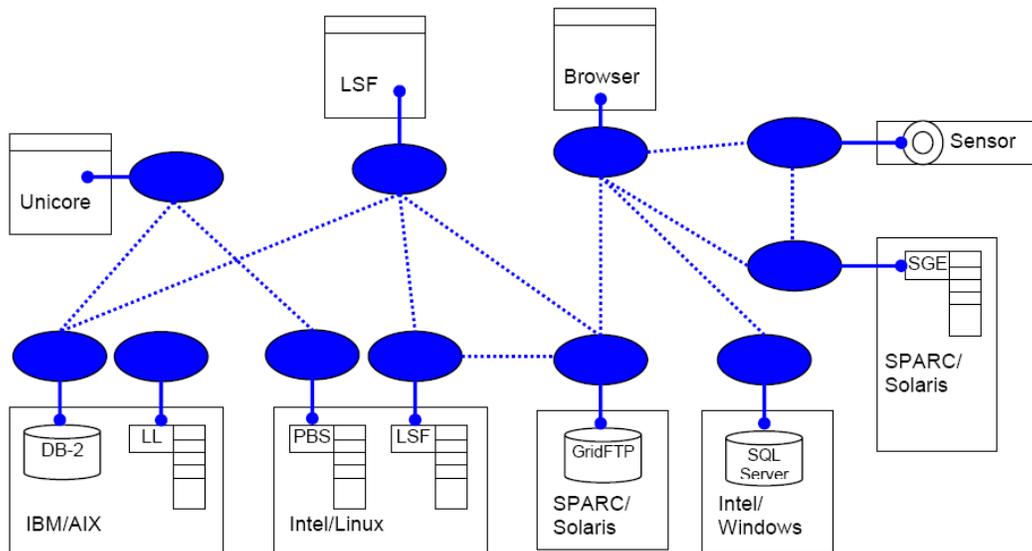


Figure 1: JRA7 Software Architecture and HSM layer [1] (page 205).

Currently heterogeneous sites present themselves to the user as a multitude of different operating systems and resource management tools as illustrated in Figure 2.

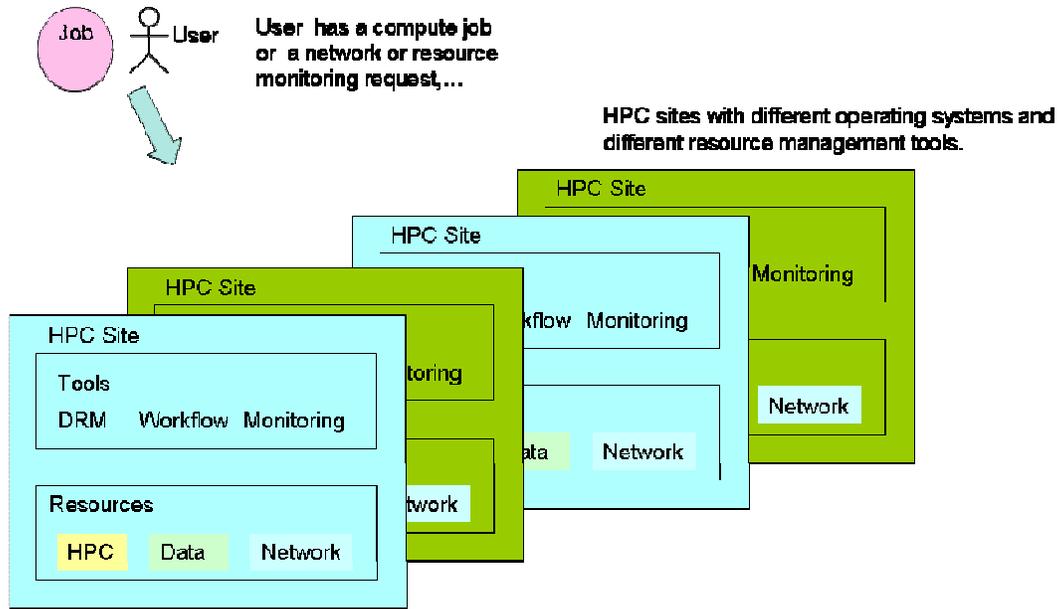


Figure 2: User is confronted with heterogeneous sites.

With the HSM layer in place the user can have a consistent uniform interface to the DEISA heterogeneous sites as shown in Figure 3. This uniform layer will be standards based.

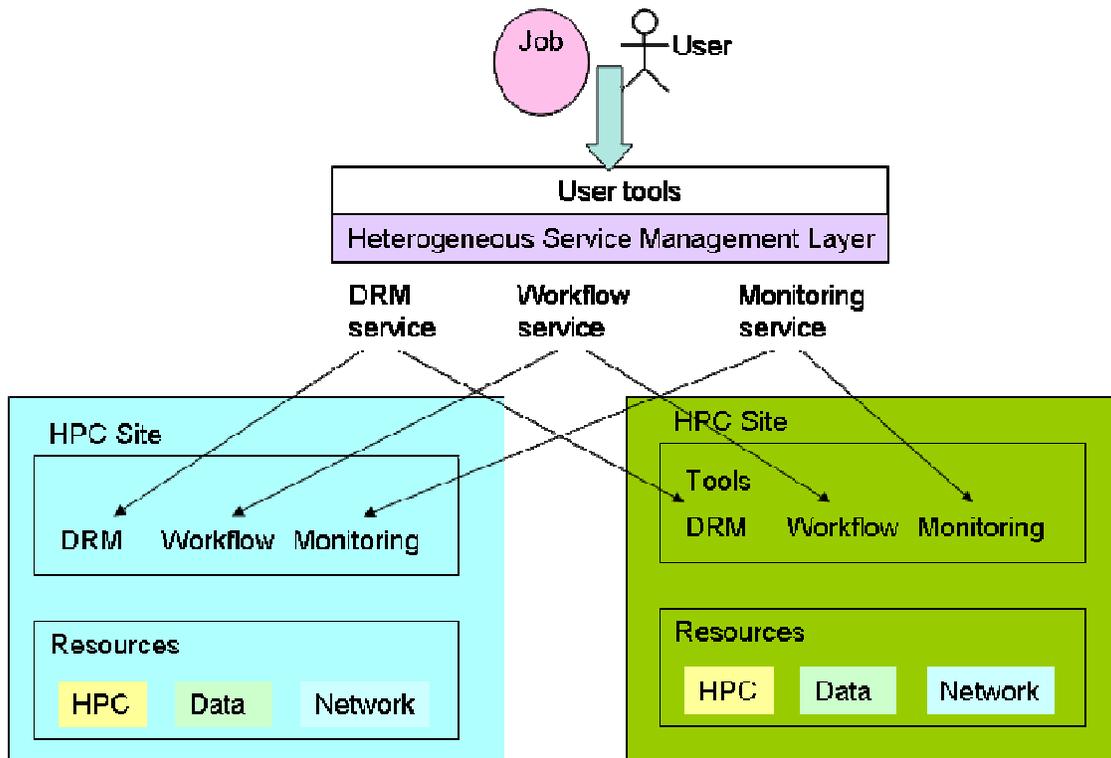


Figure 3: User has a single interface to heterogeneous sites.

The general strategy for JRA7 is for the HSM layer to be based on Web Services standards. Grid services are layered on top of this core Web services infrastructure, and build upon standards from the Grid and Web services communities. Standards developed in the Grid community should therefore have an XML rendering (which most of the active GGF working groups already have), in order to be considered for use by JRA7.

We can look to the working groups of the Global Grid Forum as being most relevant for resource management standards. For security services, we can perhaps leverage much of the work in the area of Web Services Security. We currently observe some convergence between the Web service and Grid communities, and so, regardless of origin, we should evaluate all standards necessary to get the job done.

One clear observation we can make at this point is that no specification exists as an island. In the Web services realm, the specifications are designed to be *composable*. This means that in the case of SOAP as a messaging format, the application messages in the SOAP body are augmented by security, assurances (transacted, reliable), etc, content contained in the header of the message.

Any service that we wish to develop will need to be built on a supporting infrastructure. Essential parts of this infrastructure are the security mechanism and the information model. Since JRA7 proposes to use Web services, equally important is how security and information are realized in a Web services context. Taking the example of job submission, a design of such a service is likely to have to utilise standards, not just job submission languages, but additionally resource information models, security, policy declarations, etc.

We acknowledge that there are many standards, and from many different sources. We attempt to refine the list of potential standards by being mindful of what is possible within the scope of JRA7.

3. A Glossary of Web Services Standards and related technologies

Table 1, below, is a brief overview of some Web Services standards and related technologies of interest to JRA7. For each item where applicable the table also refers to the item's relevant standards body.

Short Name	Standards Body	Description
HTTP/HTTPS	IETF	HTTP is a communication protocol for exchanging information. HTTPS is a secure version based on TLS. HTTP(S) is often the default transport associated with Web services.
SOAP	W3C	(Originally an acronym for Simple Object Access Protocol but now just SOAP). XML-based information that can be used for exchanging structured and typed information between peers in a decentralized, distributed environment.
TLS	IETF	Transport Layer Security: provides communication privacy over the Internet, preventing eavesdropping, tampering or message forgery.
UDDI	OASIS	UDDI is a mechanism for registering and discovery of Web services.
Web service		A software application that is identified by a URI, with interfaces and bindings that are capable of being described, and discovered, as XML artefacts, and supports direct interaction using XML based messages over Internet based protocols.
WSA	W3C	Web Services Architecture identifies the functional components and their relationships to provide an interoperable architecture.
WS-Addressing	W3C Submission	WS Addressing is a transport neutral mechanism to address Web services (Similar to WS-MessageDelivery). Recommends the use of WS-Security to provide end-to-end secure communication.
WS-Agreement	GGF Draft	Web Services Agreement specification defines a language and protocol for advertising service provider capabilities and creating and monitoring service level agreements.

WS-AgreementNegotiation	GGF Draft	Web Services Agreement Negotiation specifies how a consumer and a service provider negotiate to create a service level agreement, WS-Agreement, for a consumer request.
WS-AtomicTransaction	Proprietary	Web Services Atomic Transaction provides the definition of the atomic transaction coordination type that is to be used with the WS-Coordination specification.
WS-Authorization	OASIS	Web Services Authorization describes how to specify access policies for a Web service. Part of the original WS Security roadmap submitted to the OASIS WS Security TC.
WS-BaseFaults	OASIS	Web Services Base Faults specifies the base fault to be returned when a fault must be raised in a message exchange.
WS-BaseNotification	OASIS	Web Services Base Notification specifies the core interfaces and message exchanges a notification producer and consumer must conform to. This is part of the WS-Notification family.
WSBPEL / BPEL4WS	OASIS	Web Services Business Process Execution Language provides a language for the formal specification of business processes and business interaction protocols, enabling Web services to support Business transactions.
WS-BrokeredNotification	OASIS	Web Services Brokered Notification specifies the interfaces and message exchanges an intermediary notification producer, that is one that is able to publish messages on behalf of others, and consumers must conform to. This is part of the WS-Notification family.
WS-BusinessActivity	Proprietary	Web Services Business Activity provides the definition of the business activity coordination type that is to be used with the WS-Coordination specification.
WS-CAF	OASIS	Web Services Composite Application Framework defines a generic and open framework for applications that contain multiple services used in combination (composite applications).
WS-CDL	W3C	Web Services Choreography Description Language is for composing interoperable peer-to-peer collaborations between any type of Web service participant where ordered message exchanges result in accomplishing a common business goal.

WS-CF	OASIS	Web Services Coordination Framework specifies a coordination framework whereby distributed parties can coordinate their actions.
WS-Context	OASIS	Web Services Context is a context framework for managing the propagation of context within a distributed messaging exchange.
WS-Coordination	Proprietary	Web Services Coordination describes an extensible framework for providing protocols that coordinate the actions of distributed applications.
WSDL	W3C	Web Services Description Language is an XML based mechanism for describing the types, messages, operation and transport binding that a Web service supports.
WSDM	OASIS	Web Services Distributed Management is defined as a set of capabilities for discovering the existence, availability, health, performance, and usage, as well as the control and configuration of a Web service, and includes using Web services to manage distributed resources.
WS-Federation	OASIS	Web Services Federation describes how to federate trust. Part of the original WS Security roadmap submitted to the OASIS WS Security TC.
WSFL	Proprietary	Web Services Flow Language: superseded by WSBPEL.
WSI-Profile	WSI	Web Services Interoperability Basic Profile is a set of non-proprietary Web services specifications and how they should be used to promote interoperability. These are: SOAP, HTTP and HTTPS, XML, WSDL and UDDI
WS-JDML	No	This London e-Science Centre project is a Web services based job submission tool.
WS-Manageability		Superseded by WSDM.
WS-MessageDelivery	W3C Discussion	Web Services Message Delivery is a transport neutral mechanism for sending SOAP messages to a Web service (Similar to WS-Addressing).
WS-Notification	OASIS	Web Services Notification is a family of specifications that define a topic based publish and subscribe mechanism for Web services.
WS-Policy	OASIS	Web Services Policy Framework is a general purpose and extensible means to describe and communicate the policies of a Web service. Part of the original WS Security roadmap submitted to the OASIS WS Security TC.

WS-Privacy	OASIS	Web Services Privacy uses WS Trust, WS Policy and WS Security. Part of the original WS Security roadmap submitted to the OASIS WS Security TC.
WS-Reliability	OASIS	Web Services Reliability provides reliable messaging using SOAP: guaranteed delivery, no duplicates and guaranteed message ordering.
WS-RenewableReferences	OASIS	Web Services Renewable Reference decorates WS-Addressing to add policy information to enable retrieval of a new reference to a WS-Resource.
WS-Resource Framework / WSRF	OASIS	Web Services Resource Framework is a suite of specifications that expresses the relationship between stateful resources and Web services.
WS-ResourceLifetime	OASIS	Web Services Resource Lifetime specifies mechanisms for managing a WS-Resource's lifetime
WS-ResourceProperties	OASIS	Web Services Resource Properties provides the definition of a WS-Resource and the interface to its properties.
WS-SecureConversation	OASIS	Web Services Secure Conversation specifies how a Web service and a requestor can communicate securely. Part of the original WS Security roadmap submitted to the OASIS WS Security TC.
WS-Security	OASIS	Web Services Security aims to specify how to communicate privately with a Web service and also how to exchange security information. Version 1.0 includes SOAP Message Security, Username Token Profile and X.509 Token Profile.
WS-ServiceGroup	OASIS	Web Services Service Group specifies an interface to a heterogeneous collection of Web services.
WS-Topics	OASIS	Web Services Topics defines a mechanism to organize and categorize items of interest for subscription. This is part of the WS-Notification family.
WS-Transaction	Proprietary	Superseded by WS-AtomicTransaction and WS-BusinessActivity.
WS-Trust	OASIS	Web Services Trust describes how to establish both direct and brokered trust relationships. Part of the original WS Security roadmap submitted to the OASIS WS Security TC.
WS-TXM	OASIS	Web Services Transaction Management is a suite of transaction models built upon WS-CF.

X.509 PKI	IETF	Public Key Infrastructure, a means to use Public Keys for privacy.
XLANG	Proprietary	Superseded by WSBPEL. Web Services for Business Process Design.
XML	W3C	Extensible Markup Language is a means for interoperable data exchange.
XML Encryption	W3C	Encrypting/decrypting digital content (including XML documents and portions thereof)
XML Signature	W3C	XML compliant syntax used for representing the signature of Web resources and portions of protocol messages (anything referencable by a URI) and procedures for computing and verifying such signatures.

Table 1: Web Services standards and related technologies glossary.

4. Web Services Standards

This section briefly describes and assesses various Web Services standards and specifications of possible relevance to the JRA7 objectives.

4.1 Web Services Security

The basic premise of Web Services Security is that no new security technology has been created to solve the challenges. Well-established security techniques have been given an XML rendering ready for Web service usage.

Security for Web services can be achieved at two levels - either at the transport level or at an XML level. Transport layer security uses existing Internet protocols to secure the traffic between the Web service and the client application.

For a number of important reasons Web service security is often preferred over transport-level security. Whilst it is possible to offer a WSDL binding to HTTPS for example, the 'recommended' route is to use WS-Security. WS-Security forms the foundation for the security of Web services. [WS-Security](#) is a *protocol neutral* mechanism for securing SOAP messages. It builds upon [XML-Signature](#) and [XML-Encryption](#), and also specifies how security tokens can be associated with messages. WS-Security also provides end-to-end message security as all the security information is contained in the message itself. Such a message can now pass securely through multiple intermediaries on its way to the eventual endpoint.

For WS-Security, the W3C standards, XML-Certificate and XML-Encryption are the foundation technologies. XML-Certificate is used for representing digital signatures in an XML form. XML-Encryption ensures message confidentiality, by encrypting data, and also stores the result as XML.

A number of other specifications exist in the area of Web service security. For example, WS-Policy, WS-Trust, SAML, XACML, WS-SecureConversation, WS-Federation, etc. How these specifications may be useful to the OGSA is addressed by a number of groups at the GGF, for example see section 5.7. Many of these specifications address issues such as cross-domain trust issues, which, to some extent, in the 'special' environment of DEISA may not be such an issue. Thus, it is quite likely that the core WS-Security is the only standard necessary for JRA7.

4.2 Web Services Resource Framework

The Web Services Resource Framework (WSRF) [4] differs from its predecessor OGSII (Open Grid Services Infrastructure), as it *firmly* places the specification in the Web service domain. OGSII, on the other hand, had a number of quirks and oddities that placed it somewhat on the edge of mainstream Web service standards. As such, WSRF is the classic example of a specification with origins in Grid computing making the jump to 'mainstream' Web services.

However, many of the requirements for WSRF came from the Grid community. Indeed, given the brief for JRA7, the functionality offered by WSRF is very likely to be of interest to JRA7.

WSRF introduces the idea of a number of WS-Resources that can be present behind the façade of a ‘normal’ Web service interface. WSRF discusses the ‘implied resource pattern’ that provides a mechanism for referencing a particular WS-Resource when interacting with a service. An example of such a WS-Resource might be a running job and the implied resource pattern allows us to have a ‘conversation’ with the WS-Resource, where previous actions on the WS-Resource are ‘remembered’ as part of the state of the WS-Resource.

WS-ResourceProperties allows a particular WS-Resource to expose additional XML content regarding the state of the service. A client can read, query and update this content using the ResourceProperties porttype. One clear application of this is that the resource information (defined by the information model) can be exposed using this mechanism. The WS-ResourceProperties specification defines what it means to be a WS-Resource.

WS-ResourceLifetime addresses the following aspects of a WS-Resource’s lifecycle:

- Creation – no explicit message exchange is defined, however a factory pattern is mentioned whereby any Web service operation can create a new WS-Resource and make the endpoint reference available.
- Identity – this is the stateful resource identifier that is part of the endpoint reference. There is no specification for what this identity should be; it is design and implementation dependant.
- Destruction – this covers both immediate and scheduled destruction of the WS-Resource

WS-RenewableReference specifies a mechanism that allows an endpoint reference that has expired to be renewed. This involves decorating a WS-Addressing endpoint with a WS-Policy assertion that allows the endpoint to be renewed.

WS-ServiceGroups specification defines how collections of references to heterogeneous Web services can be represented and managed. The specification also covers expression of membership rules, constraints, and classifications. The ServiceGroup interfaces can be composed with new interfaces to provide more specialized interaction with the group, for example provide a method `find()` that is invoked on each member.

WS-BaseFaults defines the base fault type that is returned by any of the operations defined in the WS-Resource suite of specifications when a fault occurs. This gives consistency to the fault reporting in WS-Resource.

4.3 Web Services Workflow standards

In recent times, the BPEL4WS (Business Process Execution Language for Web Services) specification has emerged as the leading candidate for expressing a workflow of Web services interactions. This specification consolidates previous efforts from

Microsoft (XLANG) and IBM (Web Services Flow Language). There are a number of open source and commercial implementations of BPEL4WS.

(Also, see Section 5.5 on Grid Workflow standards)

4.4 OASIS Web Services Distributed Management TC

The GGF group CMM-WG [15] was initially chartered to define how manageable resources and resource managers are represented within a grid. Since then this work merged with an OASIS Technical Committee for standardisation as Web Services Distributed Management (WSDM). This committee is looking at Management of Web Services (MOWS) and Management Using Web Services (MUWS). The MUWS spec is based on WSRF - to provide access to native manageability. It also aims to provide functionality such as metrics, resource states and inter-resource relationships. The 1.0 specification is expected in October 2004. The CMM-WG is now working on an analysis of management in OGSA with the intention of creating a management framework for OGSA.

The work of the WSDM TC will produce two specifications: one focused on MUWS and the other on MOWS. The MUWS specification provides a framework for using Web services to manage general IT resources but does not specify a management information model. MOWS, on the other hand, builds on MUWS to provide an information model for Web services, defining the management capabilities and attributes of Web service entities.

5. Grid Standards

The Global Grid Forum (GGF) [16] is the current authority on Grid standards. The following sections cover various groups operating as part of GGF.

5.1 *Open Grid Services Architecture (OGSA)*

The Open Grid Services Architecture (OGSA) working group of the GGF is perhaps our first port of call for examining the activities of the GGF. This group is concerned with capturing the high level requirements to produce an OGSA architecture roadmap document that defines, scopes, and outlines requirements for key services in a Grid. OGSA makes no normative statements.

The OGSA process and results will clearly be of interest to JRA7. It is also true that some of the analysis conducted within JRA7 would potentially be interesting use-case material for the OGSA group.

Under the ‘umbrella’ of the Open Grid Services Architecture, there are many other groups active at the GGF focusing on more specific areas. These are described in the following sections.

5.2 *Job Specification Description Language (JSDL)*

To motivate the specification of a Job Submission Description Language (JSDL) in the context of the Global Grid Forum, one may cite the abstract of the respective GGF working draft (currently draft-ggf-jsdl-spec-0.4.4 [17]):

“This specification document details the semantics and structure of the Job Submission Description Language (JSDL), which is used to describe the requirements of computational jobs for submission to resources, particularly in Grid environments, though not restricted to the latter.”

Apart from the specification itself the JSDL Working Group (JSDL-WG) [18] provides a normative XML schema for the instantiation and validation of jobs in a Grid environment and job examples covering some specific use cases.

By reaching its main objective, the specification of a JSDL accepted as a “standard” within the Grid community, the JSDL-WG aims to overcome the shortcomings of the current status of job submission in heterogeneous Grids, mainly the lack of interoperability: A broad variety of proprietary job (submission) descriptions exist and prevent jobs from being submitted across heterogeneous Grids. Once system developers and vendors agree on using JSDL it is envisaged that a variety of different job managers like schedulers, brokers or job execution systems share JSDL as a common job description format.

A JSDL document is a job template, therefore specific job instance information is not contained in it, that has to be maintained by the respective job manager. Concerning the content of a JSDL job description (the JSDL attributes) please refer to the previously mentioned document [17]. One further aspect should be mentioned here: JSDL does not

contain workflow information, a job instance which conforms to the JSDL specification can be seen as an atomic entity within a workflow (like a single (X)RSL request in Globus or a task in UNICORE). The only attribute which can be seen as contradictory to the atomic character of a JSDL job is the `File` attribute which allows files to be staged-in or staged-out, a task which may be seen as atomic itself.

Concerning the potential impact of JSDL it has to be stated that a number of groups at GGF make use of the JSDL Working Group output, as well as projects like the Japanese NaReGI [14] project and the EC FP6 project UniGrids [13], which develop job managers based on JSDL. JSDL is therefore clearly relevant to the objectives of JRA7.

5.3 CIM based Grid Schema (CGS-WG)

The DMTF CIM [7] is one of the approaches in the industry for enabling the management of real world managed objects that applies the basic structuring and conceptualization techniques of the object-oriented paradigm. CIM is capable of defining manageable objects that occur in actual computer and network environments. CIM consists of:

- a specification that describes a modeling language and syntax for defining manageable objects,
- a management schema for those objects,
- a protocol that encapsulates CIM syntax,
- schema to provide access to those manageable objects,
- a compliance document for interoperability between vendor implementations.

CIM has cross-platform and cross-vendor support. There are mappings for the CIM definitions to XML and LDAP, both core technologies in the frame of Grid computing. There are trends in the OGSA initiative to use CIM.

Within the context of JRA7, some careful consideration is necessary. UNICORE has been proposed as the standard high-level tool for accessing the resources. UNICORE defines its own information model. A JRA7 service in the HSM must either use the normal UNICORE information model, map another model on to the UNICORE resource model, or influence the future development of UNICORE to support a new model.

The Job Submission Information Model (JSIM) schema is based on the “job” schema in DMTF’s Common Information Model (CIM), version 2.8. It includes a UML diagram of the classes associated with job submission, the managed object format (MOF) for those classes, and an XML representation of the UML.

JSIM describes the managed objects and their relationships for managing the execution and monitoring of batch jobs in a Grid environment. The CIM 2.8 schema for jobs and processing is the foundation for the development of this model. The adoption of such a model would mean that a CIM-aware client could inspect the status of queues and jobs on a grid resource, regardless of the brand of scheduler being utilized. The usefulness of such an approach to JRA7 is clear. Note, in this regard UNICORE refines its own job submission information model, and so the previous comments regarding its use apply here too.

5.4 Grid Scheduling Architecture Research Group (GSA-RG)

The Grid Scheduling Architecture Research Group at GGF (GSA-RG) [19] states in its charter that it has the objective to define a Scheduling Architecture for Grids which enables the cooperation of the diverse entities involved in a Grid scheduling process, as there are users, scheduling and resource management services at different levels, brokers, negotiation services, etc. Key aspects of the research of this group are:

- Interaction between (classical) resource management and data management
- Co-allocation
- Agreement negotiation and advance reservation
- Evaluation and integration of work carried out by other GGF groups related to a Grid scheduling architecture

Currently the group is processing the uses cases [20] provided by the community and deriving requirements for a scheduling process and related components.

As a research group GSA has long-term objectives; a specification of a scheduling architecture may take a while. One crucial issue concerning the success of an architecture proposed by this group will be cooperation with the OGSA-WG. The OGSA-WG aims to define the overall Grid architecture, something which includes scheduling and resource management aspects as well. Therefore if both groups collaborate then the GSA-RG will provide scheduling domain-specific input to OGSA.

5.5 Workflow Management Research Group (WFM-RG)

The Workflow Management Research Group of the GGF [8] is good collection of resources relating to grid workflow and the associated standards. It would be fair to say that there are a number of options in this area. The Group's aim is to produce a taxonomy of workflow issues and systems. Currently the group is gathering details on workflow usage.

There is one line of argument which suggests that standardisation of Grid workflow is less important than that of regular job submission, for example. After all, one can clearly see the advantage of coordinating services into a workflow assuming each aggregated service implements a standard interface. On the other hand aggregation of (standards compliant) workflow services to form a 'super' workflow would appear to be a less essential activity.

It is clear that this activity is in the early stages and as such cannot be adopted in any initial developments in JRA7.

5.6 Grid Resource Allocation Agreement Protocol Working Group (GRAAP-WG)

The Web Services Agreement Specification (WS-Agreement) and the Web Services Agreement Negotiation Specification (WS-AgreementNegotiation) are working drafts of the Grid Resource Allocation Agreement Protocol Working Group (GRAAP-WG) [9]. These specs have been evolved out of the so-called OGSi-Agreement and they are GGF Recommendation candidates.

“The goals of WS-Agreement are to standardize the terminology, concepts, overall agreement structure with types of agreement terms, agreement template with creation constraints and a set of port types and operations for creation, termination and monitoring of agreements, including WSDL needed to express the message exchanges and resources needed to express the state.”

This statement, taken from the WS-Agreement specification [10], describes the main goal of this effort. The essence of this approach is the provision of support for the agreement management in Grid environments where the state of resources is potentially dynamic and resource users and providers have to agree on the resource usage terms on a per case basis. “Agreement” can be taken literally in this case, the parties involved agree on state-dependent guarantees concerning the service provided and possible quality attributes (the QoS).

A simple request/response agreement may be sufficient for some of the use cases that occur in a Grid environment: a user may, for example, just want to reserve an instrument like a visualization cave for a day without specifying any other requirement. This may lead to the agreement provider looking up the schedule, reserving the cave and informing the user about the day. And voilà, the agreement has been established. In this case the WS-Agreement specification is sufficient. But imagine a scenario where the user wants to reserve the cave exactly after his data has been processed on a system that is also part of the workflow and he does not want to pay much and work at nights. This may involve a multi-step negotiation phase, something where the WS-AgreementNegotiation specification comes into play. Its goals

“are to standardize the terminology, concepts and protocols for negotiation and renegotiation of agreements, including WSDL needed to express the message exchanges.” [11]

The two specifications together introduce a layered concept of agreement provider, manager and negotiator. This allows the decoupling of the actual service provider and the agreement manager on the one hand and the negotiator on the other, a concept that makes it easy to switch between different negotiation strategies. Both specifications define the respective interfaces as Web Service Description Language (WSDL) documents.

Currently the WS-Agreement document is in a more mature state than the WS-AgreementNegotiation document. It is planned that WS-Agreement enters GGF's public comment period (60 days for Recommendations) before the end of 2004. Looking at the OGSA draft specification one can see the importance of WS-Agreement within this architecture since many of the OGSA services make use of WS-Agreement. In addition, that potential implementations of the specification are realized or discussed in a number of diverse projects.

As discussed in the original JRA7 proposal [1], this area is of clear importance to the JRA7 objectives.

5.7 OGSA Authorisation WG

This group is looking at Web services security specifications such as SAML, XAMCL and WS Security (all OASIS standards) and how these can be best used within OGSA.

The objective of the OGSA Authorization WG is to define the specifications needed to allow for basic interoperability and plugability of authorization components in the OGSA framework. There are a number of authorization systems emerging in the Grid today (Akenti, PERMIS, CAS, VOMS, Cardea, etc.), these specifications will allow these solutions to be interchangeably used with middleware that requires authorization functionality.

This group will leverage authorization work that is ongoing in the Web services world (e.g. SAML, XACML, the WS Security suite) and define a specification for how these should be used for Grid services.

5.8 Simple API for Grid Applications (SAGA-RG)

The SAGA-RG is concerned with defining a simple API which offers a layer of abstraction on top of the Grid middleware, and which can be used from a programming environment.

“These developers, however, have their own scientific agendas to pursue and often cannot spare the time or resources to fully investigate the vast wealth of Grid technologies and APIs which currently exist. They would rather be presented with a simple API close to the programming paradigms and interfaces they are used to for the fairly simple common operations they need to perform, e.g. remote job submission, file transfer, etc. A Fortran application programmer wants to see a call very much like: `call fileCopy (source, destination)`” [12]

Such an API would significantly lower the entry barrier for those wishing to take advantage of new Grid capabilities, and offer standards based tooling for integrating these functionalities into their own environment.

6. Recommendation

For job submission, JSDL is quickly shaping-up to be the standard language for job submission. However, software support for JSDL is currently lacking. For example, the LoadLeveler resource manager may support JSDL as a job description at some point, but currently it is missing such an interface.

WS-Agreement is a comprehensive specification allowing for the negotiation of an agreement regarding usage of a resource. Currently it is in draft status, however, WS-Agreement plays a important role in the work of the OGSA-WG and its progress should be closely monitored during the timeframe of JRA7.

Job submission and its subsequent monitoring can be cleanly modelled as a long running interaction with a WS-Resource. Therefore, firstly such a web service should leverage the various WSRF specifications. The WSRF looks promising as a re-working of the Open Grid Services Infrastructure, and during the early stages of JRA7 we should see tooling supporting WSRF emerge.

WS-Security (and other) standards can be positioned to secure such an interaction and indeed this looks like a strong candidate. Also, a strategy for the staging in and out of files to the web service must be considered too.

It is extremely difficult at this stage to make recommendations regarding which standards are ready for use. The reasons for this are twofold. Firstly, we are yet to decide the services comprising the HSM. Secondly, although we have witnessed good progress in the standards process, in many cases we await such standards to be finalised and supported by existing software.

Mindful of the comments above, in the area of resource management and workflow we make a number of strong suggestions. We suggest a web services based infrastructure. Using the composable nature of the WS specification we recommend the additional capability offered by WSRF and security using the mechanisms of WS_Security. For the description of jobs, we propose that JSDL be adopted.

Finally, the key to the success of JRA7 is to continuously monitor the progress of the standardisation activities, and to carefully weigh the risks before adopting.