Implementing Grid Standards

An inventory of standards and experiences from the development of the Advanced Resource Connector Grid middleware

The KnowARC Consortium

Prepared by the KnowARC project – co-funded by the European Commission
Sixth Framework Programme under contract 032691

Contents

1 Introduction

The KnowARC project, which began in 2006, aimed to take on the already successful ARC middleware, and further develop it as a simple but powerful Grid solution. A key part of this plan was to make it a strongly standards compliant middleware, both following existing standards and ensuring the project members contributed to the new standards emerging at the time.

The reason for this focus on standards is that the project believes that they are crucial to the growth and broader uptake of Grid technology. Standards allow interoperability, comparison of solutions and commercial competition. They help transition the Grid from a set of incompatible, one-off, black-box solutions to the beginnings of a new market and broad paradigm for computing.

One of the fundamental reasons for the continued funding of Grids and other emerging information technologies by the European Commission is their impact on the Knowledge Society and European Research Area – and ultimately their value to the European economy. Standards are an important factor in enabling this impact.

This is not to say that any standard is a good standard. One of the reasons KnowARC has concentrated so heavily on implementing as well as developing standards is the very low level of uptake many standards experience. For example, the Open Grid Forum – the major body attempting to define Grid standards, has only actually produced six formal standards recommendations, despite over 50 working groups. Formal standards are those that are actually implemented by two independent groups. The others remain unimplemented, at a preliminary stage or become branded obsolete due to lack of implementations in the real world, despite strong efforts from all the parties involved.

Furthermore, too often the standards-based multi-organisational work, such as interoperability, carried out between Grid efforts are on the level of 'Hello world' demos rather than actually of production quality. The same is seen even in the standards fully recommended by OGF. As stated later in this report, one of the six recommended standards, the Basic Execution Service (BED, GFD.108), is of interest and as far as possible implemented in ARC, but fundamentally not suitable or sufficient to describe production-mode implementation.

KnowARC is by no means the first to indentify this problem, but it is of particular concern to the projects strong commitment to a standards-based solution, which they feel is an absolute requirement for a Grid middleware solution with broad, long term potential. As a result they have tried to work within the framework of OGF to drive the organisation toward more realistic and useable standards. Two key examples here are the GLUE2 informational schema, which it is hoped will pass into a full recommendation in 2010, and other work by the OGF Production Grid Infrastructure working group (PGI-WG). This group states its objective is to "formulate a well-defined set of profiles, and additional specifications if needed, for job and data management that are aligned with a Grid security and information model that addresses the needs of production grid infrastructures."[1]

[1] See http://www.ogf.org/gf/group_info/view.php?group=pgi-wg

Achieving standards that are applicable to a production environment is crucial not only in allowing academic Grid projects to interact, but also in allowing their solutions to be taken up by the commercial sector. Standards would encourage and allow them to interface with to take up academic solutions, rather than make it as easy to implement their own solutions, ignoring the great deal of work done in the academic sector.

While these are long-term goals, this documents starts from a more low level position, addressing the existing standards, and how they could and were addressed in the ARC middleware. This document is based on Deliverable D3.3-1 KnowARC Standards Conformance Roadmap, released in July 2009.

2 Inventory of standard defining specifications

The inventory collects all the standard defining specifications which were surveyed and monitored over the three years of the KnowARC project in order to determine their relevance for ARC development. The specifications are grouped in three categories:

1. Implemented specifications
2. Specifications with potential future support
3. Specifications with no immediate relevance.

The first group contains those specifications which have been followed and implemented by some ARC components. The detailed standard-conformance status is provided for each such specification. The second group lists specifications of potential interest but not of priority at this stage. Finally, the last group contains documents of no current relevance.

2.1 Implemented specifications

This section contains those specifications with the highest relevance to ARC development. Achieving conformance with these standards, while being mindful of their scope and maturity, has been a priority for the project: The standardization efforts of the NorduGrid community have been targeted at these specifications.

2.1.1 WS-I Basic Profile 1.1

Organisation/Group	The Web Services-Interoperability Organization (WS-I)
Reference	http://www.ws-i.org/Profiles/BasicProfile-1.1.html
Status/Type	***F***inal material, version 1.1 was published in 2006-04-10

Short description:

The document specifies a profile for web service communication. It is based on other standards, such as SOAP and HTTP. Where the underlying standards are ambiguous or provide for multiple implementations, the specified profile tries to provide clarifications and implementation guidelines aimed at increased interoperability between various services.

The document focuses on giving the correct interpretation of SOAP 1.1, WSDL 1.1 and UDDI specifications, therefore the document refers to XML related specifications

(XML 1.0, Namespace in XML 1.0, XML Schema part 1 and 2). It also refers to the RFC2616 and RFC2965 because in the transport level this document prefers the HTTP.

The security part of the document adopts, but not mandates use of HTTPS (HTTP secured with either TLS 1.0 or SSL 3.0). So it refers to RFC2818 (HTTP over TLS), the SSL protocol version 3.0, RFC2246 (TLS 1.0) and RFC2459 (PKI), which are the basis of public key-based secure communications.

Relevance to ARC:

WS-I Basic Profile is the foundation for any WS-based software; therefore ARC development must comply with the specification as much as it is possible.

Current ARC conformance status:

The ARC SOAP implementation is based on libxml2 library. XML Level of compliance is as implemented in libxml2. Because most of SOAP messages are generated dynamically by the different service and client codes, it is difficult to define level of compliance. The only part of SOAP that is handled by HED is SOAP Fault. For that, both versions 1.1 and 1.2 are implemented. Regarding XML Schema for SOAP messages it is also the task of client and service code to keep messages compliant with the defined schemas. There is one important missing feature: the implementation of strict ordering in XML sequences. However, no significant problems are expected because of this shortcoming, because usually parsers do not care about ordering of elements.

Concerning the HTTP support - as defined in http://www.w3.org/Protocols/ - both versions 1.0 and 1.1 are supported. Not all features are implemented - only those identified as required for communication with widely deployed HTTP clients and services. HTTP implementation has support for chunked and non-chunked data transfers, ranges, and persistent connections.

As a result of an initial WS-I Basic Profile conformance evaluation it was found that the WS hosting environment framework of ARC (the HED component) satisfies most of the WS-I profile requirements. A non-exhausting study resulted that approximately 80 % of the requirements are fulfilled.

All the ad-hoc interoperability tests with other WS frameworks so far successfully passed.

Initial performance test that compares ARC and Axis2/C, a relatively new implementation of WS framework from Apache software foundation[2], has been carried out. The test shows the competitive performance capability of ARC and also demonstrates the interoperability between ARC and Axis2/C

Potential issues:

ARC does not make use of UDDI technology.

Further plans:

Perform interoperability tests with other WS frameworks, complete conformance

[2] http://ws.apache.org/axis2/c/

evaluation. Improve error handling.

2.1.2 XML Path Language (XPath) v1.0

Organisation/Group	W3C
Reference	http://www.w3.org/TR/xpath
Status/Type	Version 1.0 is W3C Recommendation published on 1999-11-16. This version is a stable, widely accepted specification. Version 2.0 has also achieved a W3C Recommendation status on 2007-01-23, nevertheless it is still less supported than v1.0.

Short description:

XPath is a language for addressing parts of an XML document. The language is based on a tree representation of the XML document, and provides the ability to navigate around the tree, selecting nodes by a variety of criteria. The primary purpose of XPath is to address parts of an XML document. In support of this, it also provides basic facilities for manipulating strings, numbers and booleans. XPath uses a compact, non-XML like syntax to facilitate use of XPath within URIs and XML attribute values. XPath operates on the abstract, logical structure of an XML document.

The most recent version of the language is 2.0, nevertheless version 1.0 is still the most widely-used one. XPath v2.0 is a superset of v1.0 with a backwards compatibility mode.

Relevance to ARC:

The status and characteristics of the ARC services are described by XML documents which are exposed through the Local Information Description Interface (LIDI) of the corresponding service. For the query of the information system, an XML-based query language was a natural selection. Furthermore, the ARC Information Index service (ISIS) also offers access to its database of registration records through XML queries. The WS-based ARC information system queries were decided to be XPath-based.

In particular, ARC use the limited set of WS-ResourceProperties as a primary interface to the local information system based on XML rendering of the GLUE v2 model. The QueryResourceProperties function of WS-ResurceProperties (as part of our LIDI implementation) supports multiple query languages. XPath is the most widely used and supported of them, e.g. libxml2 offers support for XPath but not for XQuery. XQuery is very complicated language. Therefore XPath was chosen over XQuery[3]. During the design phase it was concluded that XPath version 1.0 is sufficient, there is no additional benefit of using XPath Version 2.0 or XQuery.

The ISIS server supports XPath version 1.0 queries through its WS interface as well.

[3] XQuery language by W3C: http://www.w3.org/XML/Query/

Current ARC conformance status: The HED-based ARC server-side components use the XPath parser of the libxml2[4] C library which fully supports 1.0 version of XPath, the XPath functionality is available everywhere where *XMLNode* class is used. The InformationInterface class of infosys library makes use of XPath as the currently only supported query language thus XPath through the LIDI/WSRF interface was chosen as ARC's main information system query language. XPath support is also available in the client side through the *libarccommon* library which also comes with the *XMLNode* class. XPath is also the only supported way of querying the Information Indexing Service (ISIS) for information about registered resources through the ISIS WS-interface.
Potential issues: In connection with XPath one may expect potential performance bottlenecks that will require special attention.
Further plans: ARC complies with XPath v1.0 as much as libxml2 does; no further steps are needed apart from performance tests and potential problem fixing.

2.1.3 WS-Resource Framework v1.2 (WSRF)

Organisation/Group	*OASIS*/Web Services Resource Framework (WSRF) TC
Reference	http://www.oasis-open.org/committees/wsrf
Status/Type	Completed, WSRF version 1.2 became an OASIS standard in April 2006.
Short description: WS-Resource Framework (WSRF) is a family of OASIS published WS-specifications which aim at defining open framework for modelling and accessing stateful resources using Web Services. WSRF consists of the following specifications: • WS-Resource: includes the XML schema to describe resource properties as an XML document. • WS-ResourceProperties: defines an interface to get and put one or more resource properties document to and from web services. • WS-ResourceLifetime: contains additional interface to WS-ResourceProperties to manage lifetime of resource properties documents. • WS-ServiceGroup: provides a description of a general-purpose WS-Resource, which aggregates information about multiple WS-Resource or Web services.	

[4] Libxml2: http://xmlsoft.org/

- WS-BaseFaults: includes the recommended basic fault message elements definition, which includes some additional error metadata like timestamp, origin of fault, classification of fault etc.

The WSRF family together with the WS-Notification and WS-Addressing specifications are used as the main building blocks of the WS Distributed Management (WSDM) framework.

Relevance to ARC:

WSRF is broadly supported in the Grid community, several middlewares decided to use a subset of the technology to exchange resource/service information. That part of the WSRF is highly relevant for ARC due to interoperability with other middlewares such as Unicore. Implementing the basic property get/set semantics of WSRF resources is relatively simple.

WSRF is a highly relevant specification for ARC middleware, in particular for resource information and status publishing.

Current ARC conformance status:

The HED framework of ARC supports the query-like operations of the WS-ResourceProperties specification (comprising GetResourcePropertyDocument, GetResourceProperty, GetMultipleResourceProperties and QueryResourceProperties). These operations serve the basis of the Local Information Description Interface (LIDI) of every WS-based ARC service. Currently, the job execution service A-REX and the Information Indexing service ISIS come with a LIDI interface.

WS-BaseFaults are implemented in the ARC WS-RF library as responses to WS-RF requests.

The remaining part of the WSRF family, including the write-like operations of the *WS-ResourceProperties (PutResourcePropertyDocument, SetResourceProperties, InsertResourceProperties, UpdateResourceProperties, InsertResourceProperties, DeleteResourceProperties),* the *WS-ResourceLifetime* and the *WS-ServiceGroup* are not supported.

Potential issues:

The introduction of WS-RF had split the WS-* community due to its limited compatibility with the mainstream WS-I architecture and the overlap it introduced with existing WS-Man specifications. As of writing, WS-RF is still almost completely ignored outside of the grid community. In 2006, a convergence plan of WS-RF family and WS-Man was announced, which wanted to introduce the omission or rendering optional of the most controversial parts of the WS-RF specifications. Unfortunately, the convergence plan has not yet generated concrete specifications or implementations.

Further plans:

ARC will only support the most widely accepted minimal subset of the WS-RF family and provide those as part of the ARC service interfaces even if that would mean incomplete conformance with the specification. The currently supported subset is judged to be sufficient for ARC's purposes.

The future ARC development will also integrate the identified WSRF subset into

all major ARC services, though this is a low priority activity.

2.1.4 Web Services Addressing (WS-Addressing) v1.0

Organisation/Group	W3C, WS-Addressing WG
Reference	http://www.w3.org/TR/2006/REC-ws-addr-core-20060509 http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509 http://www.w3.org/TR/2007/REC-ws-addr-metadata-20070904
Status/Type	The WS-Addressing v1.0 group of specifications consist of three W3C Recommendation status documents: The "*Core*" and the "*SOAP binding*" were released together on 2006-05-09 while the "*Metadata*" on 2007-09-04.

Short description:

The goal of WS-Addressing is to define transport-neutral mechanisms to address Web Services. It essentially consists of two parts: a structure for communicating a reference to a Web service endpoint, and a set of Message Addressing Properties that associate addressing information with a particular message. The WS-Addressing is specified in three documents: The "*Core"* specification of Endpoint References and Message Addressing Properties, the "*Binding*" of these properties to SOAP and the "*Metadata*" specification defining how the abstract properties in "*Core*" are described using WSDL, how to include WSDL metadata in End Point References, and how WS-Policy can be used to indicate the support of WS-Addressing by a Web service.

In the pure web service world the only way to address services is the URL. The WS-Addressing extends these capabilities to add a formal definition how the URL address of a service can be extended. The WS-Addressing introduces the End Point Reference (EPR) that describes the real address of a service and some properties. The properties relate to information about interaction between the EPRs and to metadata about the endpoints. The WSRF, WS-Notification and WS-Addressing specifications are used as the main building blocks of the WS Distributed Management (WSDM) framework.

Relevance to ARC:

KnowARC has chosen WS-Addressing, due to WSRF specifications, to represent service endpoints. Our goal is to add WS-Addressing support for all ARC services. In terms of implementation, of the three specifications, the *Metadata* is not related to ARC. This is because the dynamic SOAP parsing of the HED framework ARC does not generate service interfaces from WSDL.

Current ARC conformance status:

WS-Addressing Version 1.0 is fully supported by HED libraries. The HED framework of ARC comes with a WS-Addressing library which supports the "*http://www.w3.org/2005/08/addressing*" namespace. Every service that offers the LIDI interface (through the usage of the *InformationInterface* class implementing WS-RP) implicitly uses WS-Addressing as well.

WS-Addressing as a destination of SOAP message is also supported by SOAP MCC regardless of service. HED implements End Point Reference, WS-Addressing elements in SOAP header and WS-Addressing faults. That makes it a complete implementation.
Potential issues: None.
Further plans: Make sure service developers will use the available framework while creating new services. Implement WS-Addressing support for all the ARC services including the Chelonia Storage services.

2.1.5 OGSA Basic Execution Service (BES) v1.0 (GFD.108)

Organisation/Group	OGF, OGSA-BES WG
Reference	http://www.ogf.org/documents/GFD.108.pdf
Status/Type	Version 1.0 was released as an OGF proposed recommendation (P-REC) on 2007-08-07. The OGF group is working on Errata.
Short description: This document presents a specification for a service to which clients can send request to initiate, monitor and manage computational activities. The document defines three different service interfaces: BES-Management, BES-Factory and BES-Activity. BES-Management allows client to manage the BES container itself. BES-Factory is the main interface that allows creating and managing activities. These activities can be interpreted as a traditional grid job or a service deployment, depending on the backend of the BES-Factory. The BES-Activity interface duplicates most of the functions in the BES-Factory, however these functions work only on one activity. The status of the BES-Activity port type is rather unclear within BES, for example no normative schema is provided for this port type. BES also defines a basic state model and the way in which this model can be extended. BES requires that the state model extension should be backward compatible with the basic state model. BES also defines a simple but extensible information model. Basic attributes of BES-Factory describe the BES itself and the represented resources as well. BES uses JSDL to describe any kind of activity. The standard way to create an extension to BES is to make a BES-specific profile.	

Relevance to ARC:

The next generation job management service of ARC, named A-REX service offers its computational capability through a BES-like interface. The service specific part of the A-REX interface[5] is provided as an extended BES-Factory.

ARC client library should be able to communicate with BES-compliant resources.

Current ARC conformance status:

The A-REX job execution service implements an extended BES-compliant interface. The optional BES-Management and the undefined and abandoned BES-Activity port-types are not supported.

Below we present the details of the current BES support status of A-REX including the description of ARC specific extensions:

BES-Management operations are not supported. None of the three optional BES extensions (Idempotent Execution, Subscription to Notification Events, LifeTime Management) are implemented. The BES-Activity operations and the BES-Activity attributes (*Status, ActivityDocument, FactoryReference*) are not supported[6].

All the required BES-Factory attributes and some optional ones are supported: *IsAcceptingNewActivities, TotalNumberOfActivities, ActivityReference, TotalNumberOfContainedResources, NamingProfile, BESExtension, LocalResourceManagerType*, *OperatingSystem.*

The *"Basic state model"* of BES together with the *"State specialization mechanism"* is used and fully supported. ARC-specific sub-states (*Accepted, Preparing, Prepared Submiting, Executing, Executed, Finishing, Finished, Deleted, Killing*) are added as Activity sub-states. The current A-REX implementation does not use the *"Terminated"* BES state because it does not distinguish it from the "*Failed*" state.

A-REX supports all the BES-Factory operations (*CreateActivity, GetActivityStatuses, TerminateActivities, GetActivityDocuments, GetFactoryAttributesDocument*).

The required *Faults handling* is supported by A-REX.

From the non-normative WSRF rendering of BES A-REX only implements the WS-ResourceProperties.

A-REX introduced a new operation in addition to those provided by BES. It did that by defining its own port-type with the single operation *ChangeActivityStatus*. This operation provides a way to request simple transfers between states of jobs being processed.

There is also Delegation interface in A-REX introduced as an BES extension. It adds two operations (DelegateCredentialsInit and UpdateCredentials) and one element (DelegatedToken) which can be used in CreateActivity. The DelegatedToken

[5] Definition of the interfaces of the core ARC components, https://www.knowarc.eu/documents/Knowarc_D1.2-1_07.pdf

[6] Actually, BES-Activity operations and attributes are incompletely defined in the document. BES-Activity seems to be an orphan child of the BES interface.

element contains the public part of the delegated credentials. Since BES does not provide a standard mechanism to communicate the staging directory address (the so-called *sessiondirectory*) to the job submission client, ARC had to introduce its own convention-based solution: A-REX uses the return value of the *BES-Factory:CreateActivity* operation, the *ActivityIdentifier* to pass the directory address as a *JobSessionDir* element of the WSA-Endpoint to the submission client. On the client-side, the ARCLIB implements the corresponding BES client features needed to communicate with the A-REX service in the corresponding client component plugin. In addition, the Unicore client component plugin also implements BES client calls to interact with the BES interface of the Unicore services.
Potential issues: The BES specification, due to its deliberately minimalist scope, is not suitable for production environments: The profile does not cover the file staging scenarios; the ad-hoc information model of BES is not applicable for realistic systems. ARC had to introduce extensions to the basic state model, add a new operation and implement an ad-hoc staging solution. Furthermore, BES does not offer a solution for delegation and management of user credentials either (for this purpose ARC added a separate "delegation interface"[7]).
Further plans: Review the A-REX state model. Propose ad-hoc staging approach as a potential solution for the next BES version. Investigate the co-existence BES and the emerging GLUE v2 specification in order to overcome the limitation of the BES information model (Factory attributes).

2.1.6 Job Submission Description Language (JSDL) Specification v1.0 (GFD.136)

Organisation/Group	OGF, JSDL WG
Reference	http://www.ggf.org/documents/GFD.136.pdf (obsoletes GFD.56) http://www.ggf.org/documents/GFD.140.pdf http://www.ggf.org/documents/GFD.56.pdf (obsolete)
Status/Type	The Version 1.0 was published in November 2005 as an OGF Recommendation document. An Errata to the version 1.0 was released 2008-07-28 which obsoletes the GFD.56. JSDL became a full recommendation with the publication of the GFD.140 experience document on 2008-09-14.
Short description:	

[7] Definition of the interfaces of the core ARC components, https://www.knowarc.eu/documents/Knowarc_D1.2-1_07.pdf

Document provides an XML schema with quite a wide set of elements, to describe computational jobs, including pre- and post-staging of data files. Schema lacks some functionality that was intentionally excluded in order to limit its scope. Despite that, many basic use cases can be covered. Schema also allows for unlimited extensions.

The Errata version has minor editorial changes to fix typographical errors and other inaccuracies; clarifications on the definitions of various elements, in particular the meaning of several of the Resources elements.

Relevance to ARC:

JSDL is the chosen native job description language of the A-REX execution management service. JSDL is meant to be used for communication among ARC components dealing with job instances (e.g. job submission client and A-REX). Agreed extensions are necessary. However we don't expect end-users specifying their job descriptions using JSDL directly.

Current ARC conformance status:

JSDL version 1.0 is fully supported by both the Grid Manager and A-REX. Furthermore, the ARCLIB client side library also supports JSDL. Full support in our case means that all the required JSDL elements are parsed and interpreted and most of the optional elements are also supported. Non-interpreted elements currently are simply ignored without any special treatment (no exception is thrown)

There are several extensions have been being created over core JSDL either within or outside OGF. ARC currently supports the JSDL HPC Profile Application Extension (Section 2.1.8) and has created its own NorduGrid JSDL extension[8]. Support for other draft JSDL extensions (e.g. Section 2.2.5) is being considered. In case of overlapping or conflicting profiles and extensions the ARC server side components apply an ordered processing. Currently this is implemented in case of the HPC Application Extension and POSIX extension of core JSDL.

ARC client side components are capable handling and submitting jobs defined in JSDL to execution services. Furthermore, the underlying ARCLIB offers some pre-processing capability and transformation among different job description languages, for example Globus RSL, gLite JDL and JSDL.

The currently unsupported, optional elements of JSDL version 1.0 (supported neither on the client nor the server side) are: *Description, JobAnnotation*[9] subelements of *JobIdentification. ApplicationName, ApplicationVersion, Description* subelements from *Application* element. The *POSIXApplication* element's non-supported subelements are: *WorkingDirectory, FileSizeLimit, CoreDumpLimit, DataSegmentLimit, LockedMemoryLimit, OpenDescriptorsLimit, PipeSizeLimit, StackSizeLimit, UserName, GroupName*. The non-supported subelements of the *Resource* are: *FileSystem, IndividualCPUSpeed, TotalVirtualMemory, TotalDiskSpace, TotalResourceCount*. The non-supported subelement of the *DataStaging* element is *CreationFlag*.

For the sake of completeness we also copy here the non-supported elements from

[8] Chapter 4 and Appendix B in NORDUGRID-MANUAL-4, http://www.nordugrid.org/documents/xrsl.pdf

[9] *JobAnnotation* is deprecated and may be removed in future versions of JSDL

HPCProfile Application Extension OGF specification (Section 2.1.8): *WorkingDirectory*, *UserName.* To overcome the limited scope of JSDL version 1.0 NorduGrid found necessary the introduction of the following extensions: *jsdl-arc:CredentialServer, jsdl-arc:RemoteLogging, jsdl-arc:Reruns, jsdl-arc:Notify, jsdl-arc:ProcessingStartTime, jsdl-arc:AccessControl, jsdl-arc:LocalLogging* within the *JobDescription* element. The *jsdl-arc:GridTimeLimit, jsdl-arc:RunTimeEnvironment, jsdl-arc:Middleware, jsdl-arc:CandidateTarget* with *jsdl-arc:HostName* and *jsdl-arc:QueueName, jsdl-arc:SessionFileTimePart* of the *Resources* element. And the *DataStaging* element was extended by the *jsdl-arc:IsExecutable* subelement. Two of the extended attributes (*jsdl-arc:ProcessingStartTime and jsdl-arc:GridTimeLimit)* are not yet interpreted by ARC.
Potential issues: Due to its deliberately limited scope there are numerous extensions emerged. "Raw JSDL" alone is hardly used in production deployments. The rather ambiguous semantics and the limited scope of JSDL elements necessitate the creation of profiles and extensions. The large number of profiles and extensions make it difficult to create a clean job description. The smooth co-existence of the profiles and extensions has not yet been investigated. Neither was JSDL designed for or suitable for exposure directly to end-users. JSDL is not synchronized with the GLUE v2 specification. Requesting resources described by the GLUE v2 information model via a JSDL document is not straightforward.
Further plans: ARC has no plan to exclusively support the above listed currently non-interpreted JSDL version 1.0 elements, because many of them are simply not relevant. The project will implement the two non-supported extensions *(jsdl-arc:GridTimeLimit, jsdl-arc:ProcessingStartTime)*. It will also provide a high-level user-side JSDL editor which will hide the complexity of the underlying JSDL format. Finally, ARC plans to support exception handling for non-supported JSDL elements, schema violation, as part of A-REX functionality possible using strict XML validation. Design a general solution to deal with non-supported optional JSDL elements on the server side where the user may specify the tolerance level of JSDL processing.

2.1.7 HPC Basic Profile (HPC-BP) v1.0 (GFD.114)

Organisation/Group	OGF, OGSA-HPCP WG
Reference	http://www.ogf.org/documents/GFD.114.pdf http://www.ggf.org/documents/GFD.124.pdf
Status/Type	Version 1.0 was released as an OGF proposed

	recommendation (P-REC) on 2007-08-28. An experience document (GFD.124) was published on 2008-02-21.

Short description:

The HPC Basic Profile is a document that is used to describe how a particular set of specifications are composed in order to solve a basic use case of High Performance Computing (HPC) systems. The Profile consists of references to existing specifications, along with any clarifications of the contents of those specifications, restrictions on the use of those specifications, and references to any normative extensions to those specifications. In particular, the Profile covers the co-existence of JSDL v1.0 (Section 2.1.6), the JSDL HPC Profile Application Extension v1.0 (Section 2.1.8) and BES (Section 2.1.5).

Relevance to ARC:

HPC Basic profile is a highly relevant OGF specification since it represents the first coherent attempt to describe the full chain of a computational job execution starting from the job description till the actual execution on a high performance computing resource. The covered use-case of the HPC basic profile is a starting point for every real-life computational application.

The HPC-BP specification, as a first complete job management profile, has gained a proof-of-concept status within the grid community.

Current ARC conformance status:

The current ARC implementation can claim conformance with *"http://ogf.org/profiles/hpc-basic/1.0/username-token"* because the "TLS/SSL with Username-Password Client Authentication" requirement of HPC Basic Profile is met.

ARC server- and client-side components also conform to *"http://ogf.org/profiles/hpc-basic/1.0/x.509-certificate-token"*as the X.509 support implemented within the ARC security framework meets all the requirements of the profile.

The profile's requirements related to the BES specification (Section 2.1.5) are fully met. The functionality provided by the optional *BasicFilter extension* introduced in the profile will be offered through the WSRF interface in ARC, therefore it is not relevant.

ARC currently does not provide server-side support for the following required JSDL version 1.0 elements: *OperatingSystem, CPUArchitecture, CandidateHosts, ExclusiveExecution.* However the listed elements are already parsed on the client side.

ARC partially understands the JSDL HPC Profile Application Extension Version 1.0 required by the HPC-BP

Potential issues:

Even though HPC-BP is considered and marketed as the most successful OGF specification it is not applicable for production grid environments: the profile does not cover the file staging scenarios; the ad-hoc information model of the underlying BES is not applicable for realistic systems. HPC-BP is mostly appropriate for batch-system level job management or rather restricted "hello grid" like interoperability demonstrations.

Further plans:

Eventually, ARC will comply fully with all the JSDL requirements and add support for the not yet processed elements, though this is a low priority.

2.1.8 JSDL HPC Profile Application Extension, v1.0 (GFD.111)

Organisation/Group	OGF, JSDL WG
Reference	http://www.ogf.org/documents/GFD.111.pdf
Status/Type	Version 1.0 was released as an OGF proposed recommendation (P-REC) on 2007-08-28 together with the HPC-BP v1.0.

Short description:

This document specifies the semantics and structure of the HPC Profile Application. The HPC Profile Application is an extension to JSDL v1.0 that is used to describe an executable running as an operating system process.

The extension is actually a further simplification of the POSIX extension of JSDL (part of the JSDL specification) by eliminating UNIX-specific parts. The purpose of the simplification is to define an extension to JSDL 1.0 for describing a simple HPC application that is made up of an executable file running within an operating system process. It shares much in common with the JSDL POSIXApplication, but removes some of the features that present barriers to interoperability.

The document includes the normative XML Schema for the HPC Profile Application, along with examples of documents based on this schema.

Relevance to ARC:

A simple JSDL structure that defines simple HPC Applications in an operating system-independent way is relevant for ARC, and job execution services should properly interpret the JSDL structure.

Current ARC conformance status:

The A-REX computing service can partially understand job requests formulated using this extension. The following elements are supported from the profile: *Executable, Argument, Input, Output, Error, Environment.*

The profile's requirement related to the *WorkingDirectory* element so that *"This element MUST specify an absolute pathname and the consuming system SHOULD raise a fault if the working directory specified does not exist",* makes the element useless in ARC environment which by design wants to hide the local, non-grid related system configuration from the Grid layer. Therefore our implementation always raises a fault when this attribute is set and clients are recommended not to use this attribute.

The profile's requirement related to the *UserName* element so that "*this element defines the user name to be used when executing the application*" implies that it is not implementable within the ARC framework since grid entities are not allowed to request local user IDs. Therefore, the consumer of the JSDL always raises a fault

when this element is specified in the job request ad users are advised not to use the element.
Potential issues: A rather simplistic JSDL profile, most of the real applications can't be described by such a JSDL element. The *HPCProfileApplication* and the *POSIXApplication* are exclusive profiles. If either of them defined then the other should not be processed. Within the ARC implementation *HPCProfileApplication* has precedence over the *POSIXApplication.*
Further plans: Though only partial conformance is provided, no further work on this profile is feasible or necessary.

2.1.9 GLUE Specification v2.0 (GFD.147)

Organisation/Group	OGF, GLUE WG
Reference	http://www.ggf.org/documents/GFD.147.pdf (conceptual model) https://forge.gridforum.org/sf/docman/do/downloadDocument/projects.glue-wg/docman.root.public_comment/doc15219 (not in sync with the GFD.147)
Status/Type	Version 2.0 was released as an OGF proposed recommendation document on 2009-03-03. The data model specific renderings (XML, SQL, LDAP) have not yet been updated and represent the status as of the public comment version of the Glue 2 specification.
Short description: The GLUE v2 information model presents a conceptual information model for Grid entities described in natural language enriched with a graphical representation using UML Class Diagrams. As a conceptual model, this is meant to be implementation-independent. Mapping to concrete data models such as XML Schema, LDAP, relational and RDF are provided as an Appendix. The GLUE v2 information model is based on the experience of several modelling approaches being used in current production Grid infrastructures (e.g., GLUE Schema 1.x, NorduGrid schema, Naregi model). GLUE v2 defines the so called "main entities" and their specializations for computing and storage capabilities.	
Relevance to ARC: An informational model is a cornerstone of any grid implementation. The GLUE v2 modelling and its rendering to concrete data models are highly relevant for ARC development. ARC services should comply with the GLUE v2 specifications when it comes to service & resource descriptions, resource discovery and even monitoring. Naturally, the next generation ARCLIB should also support GLUE v2.	
Current ARC conformance status:	

The A-REX job execution service publishes resource and job information according to the XML-rendering of the GLUE version 2 information model through its WSRF-based LIDI interface. The GLUE version 2 information is generated by the so-called information provider scripts. The current support status is initial and rather partial; most of the GLUE2 elements are only partially supported or not supported at all. The current list of non-(fully) supported GLUE2 elements is given below: *MaxMultiSlotWallTime, MaxTotalJobs, MaxRunningJobs, MaxWaitingJobs, MaxPreLRMSWaitingJobs, MaxSlotsPerJob, MaxDiskSpace, MaxStageInStreams, MaxStageOutStreams, SchedulingPolicy, MaxMainMemory, GuaranteedMain-Memory, MaxVirtualMemory, GuaranteedVirtualMemory, EstimatedAverageWaitingTime, EstimatedWorstWaitingTime, Preemption, DefaultStorageService, ReservationPolicy, Tag attributes of the ComputingShare* entity; *Reservation, WorkingAreaMultiSlotTotal, WorkingAreaMultiSlotFree, WorkingAreaMultiSlotLifeTime, TotalPhysicalCPUs, TotalLogicalCPUs, WorkingAreaTotal, CacheTotal, TmpDir, ScratchDir, ApplicationDir* attributes of the *ComputingManager* entity; *Name, Repository, State, RemovalDate, License, Description, BestBenchmark, ParallelSupport, MaxSlots, MaxJobs, MaxUserSeats, FreeSlots, FreeJobs, FreeUserSeats* attributes of the *ApplicationEnvironment* entity; *ComputingManagerSubmissionTime, StartTime, ComputingManagerEndTime, UserDomain attributes of the ComputingActivity element; HealthState, HealthStateInfo, StartTime, Downtime** attributes of the *ComputingEndpoint* entity. Furthermore, support for the following entities are completely missing: *ExecutionEnvironment, ApplicationHandle, Benchmark, UserDomain, AccesssPolicy, MappingPolicy, Location, Contact, ToStorageService and all StorageElement related entities.* Besides the information providers, another ARC component, the ARCLIB also uses GLUE v2: The internal classes representing resources and jobs in the ARC client library are also based on the GLUE v2 information model, but their structure is flattened with respect to the full specification.
Potential issues: It is not yet fully understood how GLUE v2 will work together with BES, JSDL, WS-RF.
Further plans: Provide a complete implementation of the GLUE v2 information model as part of the XML-based ARC service and activities (grid jobs) advertisements.

2.1.10 Usage Record – Format recommendation (GFD.98)

Organisation/Group	OGF, UR WG
Reference	http://www.ogf.org/documents/GFD.98.pdf
Status/Type	The OGF proposed recommendation (P-REC) was released on 2007-02-22.
Short description:	

The document defines a common format for exchanging basic accounting and usage data over grid instantiations. The format aims to encompass both job-level accounting and aggregate accounting. This record format is intended to facilitate the sharing of usage information among grid sites, particularly in the area of job accounting. The document describes the requirements in natural language form for a Usage Record standard. The usage record is then represented in an XML format.

The document does not address how these records should be used, nor does it attempt to dictate the format in which the accounting records are stored at a local site; instead, it defines a common exchange format. Furthermore, nothing is said regarding the communications mechanisms employed to exchange the records, i.e. transport layer, framing, authentication, integrity, etc

Relevance to ARC:

A Usage Record (UR) is meant to hold user identity and resource consumption data for a single job. JURA, the usage logging & accounting component of ARC generates records for each job, complying with the OGF UR format recommendation. The generated records are transported to the SGAS logging server (LUTS). ARC introduced UR extensions for additional data that proved to be essential for accounting scenarios.

Current ARC conformance status:

The 2007-02-22 release of GFD.98 is supported by JURA. All required elements and most optional ones are filled. All storage volume data has accuracy of 1 kB, and all time data has accuracy of 1 s.

Among the "Basic" properties of UR, *ProcessId* and *Charge* is missing. Among the "Differentiated" properties, the following are missing: *Network*, *Disk*, *Swap*, *Processors*, *TimeDuration*, *TimeInstant* and *ServiceLevel*. Some other resource usage data may be missing as well, if not supported by the specific batch system.

If supported by the batch system, *CpuDuration* can have two values: one with attribute *usageType="system"* and another one with *usageType="user"*. *Memory* can have three values: one with *metric="average"*, *type="virtual"*, another one with *metric="max"*, *type="physical"* and a third one with *metric="average"*, *type="physical"*.

Upon storing URs, LUTS puts an extra time stamp on each record in the attribute *creationTime* not defined by the recommendation.

An extension was added to the UR, holding data for requested runtime environments: an arbitrary number of *RuntimeEnvironments* elements in a custom ARC namespace, *"http://www.nordugrid.org/ws/schemas/ur-arc"*, contain a string value: the dash-separated "NAME-VERSION" pair.

Potential issues:

This document itself declares that "it does not address aggregation, summary records, "grid job" records, consolidated records, or anything other than an atomic resource consumption instantiation." Version 1.0 is kept too batch-system-specific: despite our comments essential grid-like properties are not included in the record.

Further plans:

Rely on the specification as much as possible when implementing the accounting

service of the next generation ARC. Follow the development of the version 2 of the UR and try to incorporate our extensions.

The ARC community will also track the development of the next version of the UR and try to incorporate our extensions and extend the user identity data with additional necessary information, e.g. VOMS info. We will also check if there are any conflicts with consumers of stored URs (clients of LUTS such as the current initial version of LUTS Federation or the conversion tools for EGEE accounting system) due to reliance on the non-standard records generated by legacy components.

2.1.11 GridFTP Protocol Description v2.0 (GFD.47)

Organisation/Group	OGF, GridFTP-WG
Reference	http://www.ggf.org/documents/GFD.47.pdf
Status/Type	Published in May 2005; OGF Recommendation document. No implementation in GT4 found, but libraries are now present.

Short description:

This document defines new features introduced in GridFTP v2 as compared to GridFTP v1 – checksum calculation and enhanced data transfer mode. It is important to note that limitation on direction of data transfer in extended mode seems to be removed.

Relevance to ARC:

The new version is relevant to the ARC GridFTP server and to the clients, which should make use of the new features – at least all the services that act on behalf of a user. As long as GridFTP is the de facto standard for data transfers in many Grid setups[10], it is important to keep ARC GridFTP server up to date with most important features of it.

Current ARC conformance status:

The recent version of the Globus Toolkit comes with libraries that support Version 2 of the protocol. ARC data components are compliant with the GridFTP v2 specification via the Globus Toolkit libraries.

To satisfy the transfer neutrality requirement of WS-based ARC components, support for the GridFTP transfer protocol is being implemented in a modular manner in both server and client components.

Potential issues:

The protocol implementations suffer from hard-to-meet firewall requirements, though version 2 is somewhat more firewall-friendly. There is a formal issue as well, as the proposal never reached the official recommendation status due to the insufficient number of implementations and absence of the corresponding experience documents. The GridFTP2 specification is thus degraded by OGF to the “historical” status.

[10] For example, GridFTPv2 support is essential for implementing distributed storage facility in the framework of Nordic DataGrid Facility and is important for interoperability with EGEE project.

Further plans: The new features of the GridFTP protocol version 2 need to be tested in both ARC clients and services in order to see whether the Globus libraries are properly used. Interoperability with non-ARC clients and services implementing the version 2 protocol must be confirmed.

2.1.12 The Storage Resource Manager Interface (SRM) Specification v2.2 (GFD.129)

Organisation/Group	OGF, Grid Storage Management WG (GSM-WG)
Reference	http://www.ggf.org/documents/GFD.129.pdf
Status/Type	Version 2.2 was released as an OGF proposed recommendation (P-REC) on 2008-04-15. Multiple implementations available (BeStMan, Castor-SRM, dCache-SRM, DPM, StoRM) and there are also two interoperability test clients (http://datagrid.lbl.gov/).
Short description: This specification defines interfaces for managing hierarchical data storage system (mostly made for tape storage with disk cache). It defines methods for managing storage space of various kinds (volatile, durable, permanent) and data stored. Specification includes operations like space allocation, file creation, copy, staging, etc. Service implementing this specification does not provide data transfer interface. Other service/protocol is needed for that.	
Relevance to ARC: Storage management capability is an important feature of ARC. In conjunction with relatively wide acceptance of SRM it is important to provide implementation of this interface on both client and server side.	
Current ARC conformance status: The libarcdata2-based ARC client tools support both v1.1 and v2.2 data transfer and directory functionality of SRM. The following operations are supported: *srmMkdir, srmRmdir, srmLs, srmMv, srmPrepareToGet, srmBringOnline, srmPrepareToPut, srmReleaseFiles, srmPutDone, srmAbortRequest, srmAbortFiles, srmPing.* Space management and permission functions are not supported, nor will they be supported. The Chelonia ARC storage system currently offers no SRM support.	
Potential issues: During implementation of SRM v2.2 one must take into account that although the specification document provides descriptions of all operations their mutual relationship is not clearly described. Also, some statements may be unclear for those outside of the SRM community. Hence, it is extremely important not to rely solely on the document but to also consult SRM experts and use existing testing tools extensively.	

Further plans: Continue to monitor libarcdata2 compatibility with current SRM 2.2 service implementations, and adapt to any changes. Implement server side SRM v2.2 interoperability layer for Chelonia ARC Storage system. Ensure interoperability with non-ARC SRM clients and services.

2.1.13 Internet X.509 Public Key Infrastructure (PKI), Proxy Certificate Profile (RFC3820)

Organisation/Group	IETF, Network Working Group
Reference	http://rfc.net/rfc3820.html
Status/Type	RFC, Standards Track.
Short description: Defines the standard (proposed) way to produce Proxy Certificates (PC). It also defines the way to put delegation restriction policies into a PC. But it does not offer any language for policies.	
Relevance to ARC: ARC must support such Proxy Certificates. Delegation restriction is especially important for providing fine-grained delegation. Handling Proxy Certificates is key area of the ARC security framework.	
Current ARC conformance status: Two aspects are involved about Proxy certificate: On one hand, the generation of RFC compliant proxy certificate. The implementation in the client utility (called *arcproxy*) for generating proxy certificate can be used to generate the RFC3820 compliant proxy, as well as generating the Globus-legacy proxy. Also this client utility can be used to contact a VOMS server to generate a VOMS proxy, and to contact MyProxy server to delegate a credential to MyProxy server and acquire a delegated proxy certificate from MyProxy server. Moreover, fine-grained delegation policy can be embedded into proxy extension for further restrictions (going behind the usual restriction on time) in order to provide fine-grained delegation. The second aspect is the consumption and usage of an RFC compliant proxy certificate. In this respect one of the key components in ARC – the TLSMCC can consume an RFC proxy when doing TLS/SSL communication, but not the Globus legacy proxy. The support of RFC3820 in TLSMCC is based on OpenSSL implementation. Unfortunately only OpenSSL version > 0.9.7f supports RFC3820, so for older OpenSSL versions, the consumption of RFC3820 is not supported. The consumption of RFC compliant proxy certificate is also supported by some services such as job execution service (A-REX) and Chelonia storage services, where the proxy is sometimes used for secure communication through GSIMCC (that is, the secure communication implemented via GSS-API by the Globus toolkit) rather than TLSMCC (TLS based communication). The policy embedded in proxy extension	

(using an ARC proprietary format) is enforced by services to achieve fine-grained restriction. The Globus-legacy proxy is only consumed by the GSIMCC (which is in charge of GSI based communication) for interaction with those services that requires GSI based communication.
Potential issues: The ARC component dealing with consumption of proxy is based on OpenSSL[11] which does not support RFC3820 under version 0.9.7f. Therefore for those OpenSSL versions, RFC3820 proxy certificate can not be correctly consumed.
Further plans: As ARC conforms to the specification, no further work is needed.

2.1.14 WS-I Basic Security Profile (BSP) v1.1

Organisation/Group	The Web Services-Interoperability Organization, Basic Security Profile Working Group
Reference	http://www.ws-i.org/Profiles/BasicSecurityProfile-1.1.html http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html
Status/Type	Version 1.1 is under group approval since 2007-0-20, while version 1.0 was released as "final material" on 2007-03-30. There are no major or incompatible changes introduced in Version 1.0.
Short description: The profile defines measures to be taken for providing integrity and confidentiality of SOAP-based communication. It mostly focuses on message-level security, which allows routing of SOAP messages without decrypting them. Basic recommendations for transport-level security are also included. But there are no requirements. This profile defines an interoperability specification that addresses transport level security, SOAP message level integrity and some other related security considerations for WS-I Basic Profile 1.0 and 1.1, Simple SOAP binding profile 1.0, and Attachments Profile 1.0. This profile covers two scenarios: security inside SOAP message, by embedding WS-Security inside SOAP header; security at transport layer, by using HTTP over TLS (RFC 2818), TLS 1.0 (RFC 2246) or SSL 3.0.	
Relevance to ARC: This profile is relevant for the ARC core component development; the security features provided by the core should be as transparent to applications as needed. This profile is relevant in every ARC component where transport level or message level security is needed. Transport-level security is considered as our first priority during the development	

[11] OpenSSL: The open source toolkit for SSL/TLS, www.OpenSSL.org

of the next generation ARC. It provides adequate solution but lacks some flexibility – messages can't be routed without being decrypted, it is impossible to secure different parts of message by different credentials, etc. Despite that transport level security is enough in many use cases, it is almost only option in transferring large quantities of (non-SOAP) data and much easier to implement. Message level security currently is mostly useful for interoperability with other projects and will be incorporated during more distant stages of development.
Current ARC conformance status: ***:*** In terms of transport level security, ARC WS components satisfy the "*Transport Layer Mechanism*" of the profile via the TLSMCC component. In terms of message level security, ARC WS components satisfy the "Username Token", "X509 Certificate Token" and "SAML Token" profiles.
Potential issues: None.
Further plans: Some of the message level security profiles could not be completely implemented because the WS-Security specification referred by WS-I Basic Security Profile does not define clearly some aspects, e.g. such as the schema of SAML2.0 tokens. Therefore interoperability tests against other WS implementation are necessary.

2.1.15 Security Assertion Markup Language (SAML) v2.0

Organisation/Group	OASIS Security Services Technical Committee
Reference	http://www.oasis-open.org/committees/security/
Status/Type	Version 2.0 was approved in March 2005. An errata was released on 2007-08-14.

Short description:

Security Assertion Markup Language (SAML) is an XML standard for exchanging authentication and authorization data between security domains, that is, between an identity provider and a service provider. SAML is an XML-based framework for communicating user authentication, entitlement, and attribute information. As its name suggests, SAML allows business entities to make assertions regarding the identity, attributes, and entitlements of a subject. In practical terms, SAML consists of a set of specifications and XML schemas, which together define how to construct, exchange, consume, interpret, and extend security assertions for a variety of purposes.

SAML assumes the principal (often a user) has enrolled with at least one identity provider. This identity provider is expected to provide local authentication services to the principal. However, SAML does not specify the implementation of these local services; indeed, SAML does not care how local authentication services are implemented (although individual service providers most certainly will).

SAML standard specification consists of "assertions and protocols", "bindings", "profiles", "metadata", "authentication context", etc.

Relevance to ARC:

This specification is relevant in ARC components in terms of authentication, attribute exchanging, and authorization.

In detail, for authentication, in the SOAP message layer, the service could require a client to provide a SAML Token (defined according to the WS-Security SAML Token profile) that is embedded inside SOAP message header in order to authenticate client, and the client could also require the service to provide a SAML Token. Moreover, common identity federation solutions, such as Shibboleth 2.0, that are based on SAML (in particular, SAML2.0 Web Single sign on profile) can facilitate the users' authentication process by using their existing community credentials (such as username/password) rather than being bothered to apply an X.509 certificate.

For authorization, because of the requirement on attribute-based authorization support, the following role deployment scenarios are envisioned: client, policy enforcement point (PEP, inside service), attribute authority (AA), policy decision point (PDP). A client will contact the AA to query the attributes of group(s) which the client belongs to, and get back an attribute assertion. This process will be implemented using the "Assertion Query/Request Profile" (paragraph 6 of the SAML 2.0 profile specification) and "Assertion Query/Request Protocol" (paragraph 3.3 of SAML 2.0 core specification). Then the client will access the service along with its credential and attribute assertion. Once service gets access request, PEP will then query the PDP to get the authorization decision for this request. When the PDP makes authorization decision, it could try to query other policy repository to get authorization policy. Both the above two query processes can be supported by using "SAML profile of XACML 2.0".

SAML is widely accepted; the Liberty Alliance, the Internet2 Shibboleth project, and the OASIS Web Services Security (WS-Security) committee have all adopted SAML as a technological underpinning for various purposes. SAML support is relevant for interoperability purposes.

Current ARC conformance status:

The ARC security framework implements the "User agent" and "Service Provider" elements of the "SAML2 Web Browser SSO profile" that can interoperate with Shibboleth's "Identity Provider" implementations. In more detail, in terms of "authentication context", ARC implements the "password-protected transport" and "previous session" schemes. In terms of "profiles", ARC implements the SAML2 Web Browser SSO profile and for "assertion and protocol", all of the "assertion and protocol" elements inside the "SAML2 Web Browser SSO profile" are supported. Finally, in terms of "metadata", all of the metadata related to the "SAML2 Web Browser SSO profile" is supported.

The WS-Security related SAML token profile is also implemented for message level authentication.

Moreover, the "*Attribute Query Profile*" is supported for querying the attributes from attribute authority (AA), such as VOMS SAML service, and ARC's own test implementation of an attribute authority service.

Potential issues:

SAML is a large and complex specification that can satisfy different use cases.

Further plans:
Support is planned for the policy querying between PDP and policy repository, and authorization decision querying between PDP and PEP, for which "*Authorization Decision Query Profile*" will be supported. The authorization decision query will be based on SOAP binding and a SAML authorization decision authority.

2.1.16 MyProxy Protocol (GFD.54)

Organisation/Group	OGF
Reference	http://www.ggf.org/documents/GFD.54.pdf
Status/Type	Published (November 2005); OGF Experimental document with wide deployment.
Short description:	
MyProxy is a remote storage for storing user's credentials. This document describes the solution used to store, retrieve (delegate) and query user's credentials. The protocol uses proxy delegation to transfer credentials without transferring private keys. There is also a very popular implementation of the credential repository available (called MyProxy as well).	
Relevance to ARC:	
Credential repositories can be relevant services on an ARC-enabled Grid. Services acting on behalf of users should be able to obtain credentials from these repositories. Currently the MyProxy implementation offers such a service and has a wide user base. Common use cases include automatic management of long-term jobs submitted with short-term credentials. In no way should ARC be limited to MyProxy. There should be an infrastructure for plugging in support for various credential storage systems, including non-GSI ones.	
Current ARC conformance status:	
The client side functionality of MyProxy protocol has been implemented in the "*arcproxy*" utility. The MyProxy protocol is directly used instead of using the MyProxy client API. GSIMCC is configured on the client side in order to be compatible to the GSI based communication.	
Potential issues:	
None.	
Further plans:	
Although the client side functionality of MyProxy protocol has now been integrated into the client utility, it is necessary that this functionality be implemented as an application interface (i.e. as part of the ARCLIB) so that it can be directly called by other client utilities. MyProxy will be used by the A-REX job execution service to renew client's credentials. This feature will be implemented through the MyProxy API. Also creating a general framework to handle credential repositories and offering a MyProxy plug-in for this framework, is generally useful. There are no plans to	

implement server-side MyProxy.

2.1.17 GSS-API Extensions (GFD.24)

Organisation/Group	OGF / Security Area
Reference	http://www.ggf.org/documents/GFD.24.pdf
Status/Type	Revised in June 2004, Experimental OGF document
Short description: The document describes extensions made by the Globus project to the Generic Security Service API described in RFC 2743. These extensions allow export and import of credentials between processes, delegations of credentials at other times than initial context establishment handling of credential extensions such as restrictions. This API is implemented in the Globus tookit's *globus_gssapi_gsi* library. There exist other GSS-API implementations with these extensions as well.	
Relevance to ARC: While there is no direct relevance to ARC, important legacy components such as GridFTP servers and some SRM implementations rely on implementation of this specification, hence ARC's support for it.	
Current ARC conformance status: Wherever needed ARC makes use of the implementation of the specification through the Globus libraries. Support for legacy components (GridFTP servers and some SRM implementations) relying on GSS-API implementations have been integrated into ARC by implementing the GSIMCC component which uses the Globus *globus_gssapi_gsi* library. Whenever support for GSS-API is needed, the GSIMCC can be configured. In order to use the GSS-API and its extensions indirectly via the GSIMCC no code change is required in the ARC components.	
Potential issues: Despite the document, the respective Globus libraries and API are still subject to changes, which may adversely affect ARC functionality.	
Further plans: Follow eventual changes in the library and adjust ARC tools and services accordingly.	

2.1.18 Grid Security Infrastructure Message Specification (GFD.78)

Organisation/Group	OGF, Security Area
Reference	http://www.ggf.org/documents/GFD.78.pdf

Status/Type	Published in May 2006, Informational document.
Short description: The document describes how the "GSI communication channel" is implemented in the Globus Toolkit, provides a description of the mechanism used to secure messages exchanged by the Globus Toolkit pre-web services and the format of the portion of those messages related to security. It captures the message formatting performed by the Grid Security Infrastructure (GSI) GSS-API libraries. It is applicable to developers wishing to interoperate with pre-WS Globus Toolkit services, including the GridFTP server.	
Relevance to ARC: Ideally ARC should not use non-standard protocols. Unless KnowARC wants to re-implement the GridFTP protocol, there is no need for this document to be considered. The specification is relevant in communication with Gridftp servers.	
Current ARC conformance status: The Informational document does not define any specifications but describes a specific implementation by Globus. The content of the Informational document is automatically covered in ARC by using the corresponding Globus libraries (more specifically, it is covered in GSIMCC).	
Potential issues: This is not a standard protocol but an implementation-defined solution. Implementation changes may adversely affect ARC functionality.	
Further plans: Follow eventual changes in the library and adjust ARC tools and services accordingly.	

2.1.19 XML Encryption, XML Signature Syntax and Processing

Organisation/Group	W3C
Reference	http://www.w3.org/TR/xmlenc-core/ http://www.w3.org/TR/xmldsig-core/
Status/Type	The latest version of XML encryption syntax and processing was proposed on December 10th, 2002; the latest version of XML signature syntax and processing (second edition) was proposed on June 10th, 2008. Both are W3C Recommendations
Short description: “XML Signature Syntax and Processing” document specifies XML digital signature processing rules and syntax. XML Signatures provide integrity, message authentication, and/or signer authentication services for data of any type, whether located within the XML that includes the signature or elsewhere.	

"XML Encryption Syntax and Processing" document specifies a process for encrypting data and representing the result in XML. The data may be arbitrary data (including an XML document), an XML element, or XML element content. The result of encrypting data is an XML Encryption element which contains or references the cipher data.
Relevance to ARC: The XML related security implementation in ARC, such as SAML and WS-Security, are based on these two documents. So these two documents are implicitly relevant for ARC.
Current ARC conformance status: The *XMLSec* library[12] is used for XML signature and encryption processing in ARC via implementing a wrapper over the XMLSec library. Therefore, the conformance completely relies on the implementation provided by the XMLSec library.
Potential issues: None.
Further plans: Follow eventual changes in the library and adjust ARC tools and services accordingly.

2.1.20 WS-Security 1.1

Organisation/Group	OASIS
Reference	http://www.oasis-open.org/committees/wss/
Status/Type	Latest version was approved as formal OASIS standard on February, 2006; the errata version was approved on November, 2006.
Short description: WS-Security (Web Services Security) contains specifications on how integrity and confidentiality can be enforced in Web services messaging. WS-Security describes how to attach signatures and encryption headers to SOAP messages. In addition, it describes how to attach security tokens, including binary security tokens such as X.509 certificates and Kerberos tickets, to messages. WS-Security incorporates security features in the header of a SOAP message, working in the application layer. Thus it ensures end-to-end security. WS-Security specification includes the core specification, Username Token Profile, X.509 Token Profile, SAML Token profile, Kerberos Token Profile, Rights Expression Language (REL) Token Profile, and SOAP with Attachments (SWA) Profile.	

[12] http://www.aleksey.com/xmlsec/

Relevance to ARC: WS-Security is relevant for the message level security in ARC. The XML related security implementation in ARC, such as SAML and WS-Security, are based on these two documents. Therefore these two documents are implicitly relevant to ARC.
Current ARC conformance status: The UsernameToken(V1.1), X.509Token(V1.1) and SAMLToken(V1.1) profiles are supported in ARC as plugins. Users can secure SOAP message by simply configuring these plugins into the configuration of service/client. An ARC test attribute authority service is implemented for issuing SAML assertions that are compatible with the SAMLToken(V1.1) profile. The VOMS SAML service can also be configured as the attribute authority.
Potential issues: None.
Further plans: As ARC conforms to the specification, no further work is needed.

2.1.21 ByteIO Specification v1.0 (GFD.87)

Organisation/Group	OGF, ByteIO WG
Reference	http://www.ogf.org/documents/GFD.87.pdf
Status/Type	Version 1.0 was released as an OGF proposed recommendation on 2007-01-12.
Short description: The ByteIO document describes a set of port types that give users a concise, standard way of interacting with bulk data sources and sinks in the grid. These port types provide the means for treating such data resources as POSIX-like files. The first of the ByteIO port types supports the notion that a data resource is directly accessible and that clients can handle the maintenance of any session state (such as file pointer, buffering, caching, etc.). The other port type presents a more stream-like interface to clients and as such contains implicit session state. In this latter case, data resources with this port type don't represent the bulk data source/sink directly but rather represent the resource of the open stream between the client and the data source/sink. ByteIO can optionally use DIME or MTOM for transfers. The document uses a pseudo-schema for the port types, an explicit WSDL is not given. Accompanying this document there will be a number of Profile Rendering documents which will normatively describe the details.	
Relevance to ARC: The ByteIO Specification defines simple POSIX-like interfaces suitable for random access of bulk data through a web service interface, which is relevant for the simple ARC Storage service. ByteIO makes it possible to implement a self-contained light-weight storage element that has only a single web service interface yet still allows	

getting any part of any hosted files.
Current ARC conformance status: The Chelonia ARC storage system has an independent low-level service component implementing the SOAP rendering of RandomByteIO portype. The implementation offers only the Simple Transfer Mechanism with very limited set of arguments. In particular, only the RandomByteIO interface and not StreamableByteIO is supported. Only the simple transfer mechanism of the three ByteIO bulk transfer mechanisms (simple, DIME and MTOM) is supported. The RandomByteIO interface has 4 methods (read, write, append and truncAppend) out of which only the read and write supported. The 'read' method has the following arguments: start-offset, bytes-per-block, num-block, stride and transfer-mechanism. Chelonia only takes into account the 'transfer-mechanism', the others are simply ignored. If the transfer-mechanism is 'simple', Chelonia puts the whole file into the response (regardless of start-offset, bytes-per-block, num-block or stride), if it is something other than 'simple', a 'non-supported transfer mechanism' message is returned. The 'write' method has these arguments: start-offset, bytes-per-block, stride and transfer-mechanism. Only the 'transfer-mechanism' is taken into account, the rest is just ignored. If it is 'simple', the data is obtained from the message and written it into a new file, if it is anything else, a non-supported transfer mechanism' message is returned. Despite the limited support fir ByteIO detailed above, the Chelonia ByteIO implementation is usable for file transfer purposes and sufficient for our purposes.
Potential issues: None.
Further plans: With a very low priority eventually develop the prototype into a full ByteIO implementation with the Simple Transfer mechanism and optionally add support for the DIME or MTOM transfer mechanisms. We will also provide client-side ByteIO support as part of the ARCLIB data component.

2.2 Specifications with potential for future implementation

The section lists specifications that are of potential interest for implementation by one or more of the ARC components. This group includes mostly immature specifications or those not embraced by the community, thus requiring close watch. Their (partial) implementation with the eventually necessary extensions could be considered in the medium or long term, beyond the lifetime of the KnowARC project.

2.2.1 Extensible Access Control Markup Language (XACML) v2.0

Organisation/Group	OASIS, Access Control TC
Reference	http://www.oasis-open.org/committees/xacml/
Status/Type	Version 2.0 was approved as an OASIS Standard in February 2005. Version 3.0 is being drafted.

Short description:

XACML is a declarative access control policy language implemented in XML, and a processing model describing how to interpret the policies. XACML has been standardized within OASIS. XACML describes both an access control policy language and a request/response language. The policy language is used to express access control policies ("who can do what when"). The request/response language expresses queries about whether a particular access should be allowed (requests) and describes answers to those queries (responses).

The XACML v2.0 specification set comes together with associated profiles, some of the more relevant ones are “Hierarchical resource profile of XACML v2.0”, “Role based access control (RBAC) profile of XACML v2.0” and “SAML 2.0 profile of XACML v2.0”. The latter defines how XACML authorization decision query/response and XACML policy query/response can be put into SAML request and SAML assertion. In the upcoming XACML3.0 specification set, there is an“Administration and Delegation Profile” which can be used by users to delegate certain permission to others.

These profiles bring XACML and SAML closer to each other, they have been designed to complement each other; for example, an XACML policy can specify what a provider should do when it receives a SAML assertion, and XACML-based attributes can be expressed in SAML.

Relevance to ARC:

An access control policy language is a central part of any kind of security system. Most of the grid implementations have been supporting or have plans to support XACML as a policy language; therefore supporting XACML is highly relevant for interoperability purposes as well.

Current ARC conformance status:

: Not supported yet. Currently ARC uses its own XML schema which is implemented as a simplified version of XACML (without <Obligation/> and complicated <Condition/> support).

Potential issues:

As far as we are aware, implementation only exists in Java. Moreover, XACML has a heavy structure and hence it is difficult to use or manage.

Further plans:

As of now our own custom policy language is sufficient, nevertheless there is already a detailed plan worked out on how to support XACML in the future:

XACML support will include XACML 2.0 Core and SAML 2.0 profile of XACML v2.0, and XACML 3.0 Administration and Delegation Profile. The “XACML 3.0 Administration and Delegation Profile” will be used to implement authorization delegation in ARC. The SAML profile of XACML 2.0 will be

supported for the policy or authorization decision querying.

XACML support will be functional inside a PDP (policy decision point) called XACMLPDP and act as a plugin which can be configured inside each Message Chain Component (MCC) or Service. When a MCC/Service processes a message, it will call a handler (called SecHandler) that then will compose the XACML request, send request to the XACMLPDP and get the policy decision. The XACML PDP support can also be a separate web service that will act as SOAP access endpoint.

2.2.2 Security Policy Assertion Language (SecPAL) v1.0

Organisation/Group	Microsoft Research
Reference	http://research.microsoft.com/en-us/projects/SecPAL/
Status/Type	Version 1.0 was released on 2007-02-15. Microsoft also provided a .NET implementation.

Short description:

The Security Policy Assertion Language (SecPAL) provides a flexible and robust declarative authorization language developed for large-scale Grid Computing Environments. The specification describes the XML syntax and data encoding conventions required to support an implementation of SecPAL.

SecPAL aims at providing flexible and robust mechanisms for expressing fine-grained trust relationships and constrained delegation of rights across organizational boundaries. SecPal aims to provide a solution where control policy can be authored and reviewed in a manner that is human readable - allowing auditors and non-technical people to understand such policies.

It supports finer-grained control and easy adaptation to different operational models. The core technology is a new declarative language for expressing security policies and other security-critical information.

Relevance to ARC:

An access control policy and assertion language is a central part of any kind of security system. The capabilities of SecPAL look very attractive to ARC.

Current ARC conformance status:

Not supported.

Potential issues:

SecPAL is licensed under a special Microsoft licence.

Further plans:

Investigate the novel technology offered by SecPAL. Study the constrained delegation approach; investigate their human-readable policy language.

2.2.3 A Simple API for Grid Applications (SAGA) (GFD.90)

Organisation/Group	OGF, SAGA-CORE WG
Reference	http://www.ggf.org/documents/GFD.90.pdf
Status/Type	An OGF proposed recommendation document (P-REC) published on 2008-01-15.

Short description:

This document specifies the core components for the Simple API for Grid Applications (SAGA Core API). SAGA is a high-level grid API that directly addresses the needs of grid application developers and aims to provide an simple API that can be used with much less effort compared to the vanilla interfaces of existing grid middleware. SAGA wants to provide a standardized, common interface across various grid middleware systems and their versions.

The SAGA design team identified the following areas of functionality (called SAGA packages): job management including resource discovery, files and logical files, streams, remote procedure calls and auxiliary API (session handle, security context, async. method call, ACLs, attributes, monitoring, error handling).

The design team chose to use SIDL, the Scientific Interface Definition Language, for specifying the API. This provides a programming-language neutral representation of the API, but with well-defined syntax and clear mapping to implementation languages. The abstract SAGA API specification, as provided by this document, is language independent, object oriented, and specified in SIDL. Normative bindings for specific languages, both object oriented and procedural, will be defined in additional documents.

Relevance to ARC:

A general easy-to-use high level Grid API definitely should offer access to ARC-enabled grid resources as well. Therefore making the ARC grid layer's functionality available through such an API is a reasonable goal. Nevertheless SAGA is not suitable to be the main ARC client API. SAGA is both oversimplified and in some other areas overcomplicated.

Current ARC conformance status:

Not supported.

Potential issues:

There are no language bindings yet available, therefore SAGA is still just an abstract API which can't be realised. Furthermore, it seems that SAGA lacks acceptance by Grid middleware developers.

Further plans:

Don't implement the specification yet. Wait for matured language bindings and wider community acceptance. Investigate the SAGA support plans of other middleware providers.

2.2.4 Distributed Resource Management Application API (DRMAA) Specification v1.0 (GFD.22)

Organisation/Group	OGF
Reference	http://www.ogf.org/documents/GFD.22.pdf
Status/Type	Version 1.0 was published on April 2004 as an OGF proposed recommendation. The document since then has received the full OGF recommendation status as a result of four independent implementations (see GFD.103, GFD.104, GFD.105, GFD.117 experience reports). Version 1.0 of GFD.22 was updated in May 2007.

Short description:

The Distributed Resource Management Application API (DRMAA) provides a generalized API to traditional batch systems such as PBS/Torque, Condor, SGE, etc.

The scope of DRMAA is limited to job submission, job monitoring and control, and retrieval of the finished job status. The API is divided in its five logical sections: init/exit, job template handling, job submission, job monitoring and control, and auxiliary routines. DRMAA provides application developers and distributed resource management builders with a programming model that enables the development of distributed applications tightly coupled to an underlying batch system.

The document makes use of an Interface Definition Language (IDL) - like language to specify the API. A subsequent OGF document[13] has recently been released which provides the proper IDL specification for DRMAA v1.0.

Relevance to ARC:

The ARC computing elements (Grid Manager and A-REX) interface to the local batch systems[14]: the grid layer transfers grid jobs submitted by grid clients to the local batch system, queries the local batch system about resource status and job status, finally it retrieves job output. That kind of batch system interface, called the LRMS backend, is currently implemented on a case-by-case basis, relying on the batch system specific commands of the particular batch system. In principle, a uniform batch system interface could simplify the LRMS backend development and maintenance provided that unified interface offers the full functionality currently being used in the ARC LRMS backends. DRMAA comes as a potential candidate.

Current ARC conformance status:

Not supported.

Potential issues:

DRMAA is rather an LRMS (Local Resource Manager System) API then a distributed resource management system API. DRMAA is overly batch-system

[13] Distributed Resource Management Application API 1.0 - IDL Specification, http://www.ogf.org/documents/GFD.130.pdf

[14] ARC Batch System Back-end Interface Guide, NORDUGRID-TECH-13, http://www.nordugrid.org/documents/Backends.pdf

specific, and so can't be used as a general grid-level batch system interface. It is unaware of grid-level concepts (e.g. Runtime Environments, grid-level data staging, etc.). DRMAA also has no support for data staging or batch system log processing. The status of DRMAA language bindings is unclear, especially with respect to uniform language bindings over the DRMAA implementations of different batch systems.
Further plans: As of Today DRMAA remains rather unpopular among grid middleware developers. The main reason is that the support for DRMAA does not increase interoperability simple because DRMAA is only a "backend" interface. Backends are not those modules where current middleware teams want to be interoperable. Nevertheless with very low priority ARC can consider the usage of the unofficial C language bindings of DRMAA within its LRMS backend modules. ARC's main goal with a potential DRMAA support would be to create a uniform LRMS backend layer which uses the DRMAA interface offered by the underlying batch systems. For this purpose, the capabilities of DRMAA must be thoroughly evaluated. Obviously, there is no sense to migrate the backend layer to DRMAA if some part of the batch system interfacing still has to be done in the conventional manner: using batch system specific commands.

2.2.5 HPC File Staging Profile, Version 1.0 (GFD.135)

Organisation/Group	OGF, HPCP WG
Reference	http://www.ogf.org/documents/GFD.135.pdf
Status/Type	An OGF proposed recommendation document (P-REC) published on 2008-06-28.
Short description: This document profiles the file staging capabilities of the Job Submission Description Language (JSDL) and the BES interface for use by HPC Basic Profile services. It includes clarifications, refinements, interpretations and amplifications of the JSDL and BES specifications in order to promote interoperability. This profile addresses how file staging can be performed by HPC Basic Profile compliant services using the JSDL v1.0 <*DataStaging*> directives. In addition, the profile extends these elements by defining an additional <*Credential*> element. Furthermore, the profile extends the BES state model with staging related substates and defines an optional <*bes-factory:FactoryResourceAttributesDocument*> extension element to advertise supported file transfer protocols in BES.	
Relevance to ARC: Proper description of the file staging operation of a grid job is highly relevant for ARC, so is the advertisement of supported file staging capabilities of a computing service Nevertheless we feel that the file staging profile offers a rather short-sighted, non-general solution which questions its usability.	

Current ARC conformance status: Not supported.
Potential issues: The profile is not general enough to describe the common file staging scenarios of production environment. The BES extension part will be irrelevant with the support of Glue2. The implementation of the required *Credential* element may be problematic. In general, the profile is too much *bes-factory* dependent when it comes to resource property advertisements. Especially, the composability of the profile with the Glue2 and the Nordugrid JSDL/BES extensions can be problematic. The profile's future community acceptance is rather questionable.
Further plans: Monitor the community reaction and once the profile gets broader acceptance consider providing some minimalist support. Further evaluation of the usability of the profile is needed.

2.2.6 JSDL SPMD Application Extension, v1.0 (GFD.115)

Organisation/Group	OGF, JSDL WG
Reference	http://www.ogf.org/documents/GFD.115.pdf
Status/Type	Version 1.0 was released as an OGF proposed recommendation (P-REC) on 2007-08-28
Short description: This document specifies the structure and semantics of a parallel application extension to JSDL v1.0. The parallel extension covers a single-program-multiple-data (SPMD) application. It is limited to a single executable and re-uses a number of elements already defined in the POSIXApplication extension to JSDL v1.0. The document includes the normative XML schema for the extension as well as informative examples.	
Relevance to ARC: The SPMD JSDL extension offers a fine-grained way to describe a parallel job request (running within the same cluster). ARC currently does not require this level of granularity in JSDL.	
Current ARC conformance status: Not supported.	
Potential issues: Parallel job environments and templates come in such a big variety of flavours that a JSDL element can't capture all the necessary details.	
Further plans: Currently it is not planned to be supported. ARC will use *RuntimeEnvironment*	

instead in order to advertise and request parallel environments.

2.2.7 OGSA WSRF Basic Profile 1.0 (WSRF-BP) (GFD.72)

Organisation/Group	OGF/OGSA-WG
Reference	http://www.ggf.org/documents/GFD.72.pdf
Status/Type	Version 1.0 published in May 2006.
Short description: The document describes the first and only OGSA profile; it proposes a WS-RF based realization of the OGSA architectural concepts. It defines the proper usage of a group of standards. These are: WS-Addressing 1.0, WS-ResourceProperties 1.2, WS-ResourceLifetime 1.2, WS-BaseFaults 1.2 and WS-BaseNotification 1.3. The profile itself is based on the WS-I Basic Profile 1.1 Even though it refers to WSRF, the basic profile does not include one of the basic WSRF specification, the WS-ServiceGroup 1.2.	
Relevance to ARC: This is the first and so far the only OGSA profile. The document determines a subset of the WSRF framework and specifies how those should be used in conformant manner in order to fit the Open Grid Services Architecture. Some parts of this profile may be relevant for ARC due to interoperability with other OGSA-based middleware solutions.	
Current ARC conformance status: Not supported as a complete profile. Some small subset of the profile related to WSRF and WS-Addressing is fulfilled.	
Potential issues: The profile, in its complete form, has not seen wide adoption. Nevertheless, subsets of its content have been followed by several grid implementations. This indicates that the profile failed to identify the proper scope.	
Further plans: The complete profile will not be supported. Before investing into further support wait for broader community acceptance. Sub-specifications related to WS-Addressing and WS-Resource Properties, WS-BaseFaults will be followed. The current WSRF and WS-Addressing support of ARC will be evaluated against the profile's requirements.	

2.2.8 OGSA Profile Definition v1.0 (GFD.59)

Organisation/Group	OGF/OGSA-WG
Reference	http://www.ggf.org/documents/GFD.59.pdf
Status/Type	Version 1.0 was published January 2006 as an OGF

	Informational document.
Short description: A normative definition of OGSA will be provided as a number of OGSA profiles, modelled along the lines of WS-I Profiles. This informational document outlines how to write normative OGSA Profiles for describing collections of specifications and their interactions. The intention of these Profiles is to describe precisely the requirements imposed upon implementations in order to ensure interoperability. The document also provides objective definitions for the classification of referenced specifications (status and adoption level).	
Relevance to ARC: May be needed if ARC will have to define its own profile e.g. for "job management" built around the ARC extensions of JSDL, BES, HPC-BP and other specifications.	
Current ARC conformance status: Currently not used.	
Potential issues: None	
Further plans: The informational document will be followed in case an OGF profile definition would become necessary for ARC.	

2.2.9 Web Service Reliable Messaging (WS-RM) v1.1

Organisation/Group	OASIS/ IBM, BEA Systems, Microsoft, TIBCO Software as main supporting vendors.
Reference	http://docs.oasis-open.org/ws-rx/wsrm/v1.1/wsrm.html
Status/Type	The original specification was written by BEA Systems, Microsoft, IBM, and Tibco and in March, 2003 and subsequently refined over the next two years. The February 2005 version was submitted to the OASIS in June. The resulting WS-ReliableMessaging 1.1 was approved as an OASIS Standard on June 14th, 2007. The most recent errata are from January 2008.
Short description: The proposal describes a protocol that allows messages to be delivered reliably between distributed applications in the presence of failures of a software component, system, or network. The protocol is described in this specification in an independent manner, allowing it to be implemented using different network transport technologies. To support interoperable Web Services, a SOAP binding is defined within this specification.	
Relevance to ARC: ARC services may need the functionality of Reliable Messaging. Currently there is no	

urgent need for such a feature.
Current ARC conformance status: Not supported.
Potential issues: Fortunately, with the OASIS acceptance of the specification the ambiguous situation around WS-RM was resolved. WS-RM became the chosen standard suppressing the competing WS-Reliability specification.
Further plans: WS-RM is a mature standard and once an ARC component requires reliable messaging, WS-RM will be the candidate framework to implement such a feature.

2.2.10 WS-Notification v1.3 (WSN)

Organisation/Group	OASIS/Web Services Notification (WSN) TC
Reference	http://www.oasis-open.org/committees/wsn
Status/Type	Completed and approved in July 2006
Short description: The purpose of the Web Services Notification (WSN) standard is to define a set of specifications that standardise the way Web Services interact using "Notifications" or "Events". They form the foundation for Event Driven Architectures built using Web Services. They can be thought of as defining "Publish/Subscribe for Web Services". WS-Notification framework includes the WS-BaseNotification, WS-BrokeredNotification and WS-Topics 1.3 specifications. The WS-RF, WS-Notification and WS-Addressing specifications are used as the main building blocks of the WS Distributed Management (WSDM) framework.	
Relevance to ARC: The publish/subscribe functionality of Web Services may be needed in ARC components: a common use-case would be when clients want to subscribe to status change information of services and/or their activities. The WS-N family provides an approved solution for this.	
Current ARC conformance status: Not supported.	
Potential issues: There are several competing standards for the same functionality. Also, the effect of the WSDM/WS-Man reconciliation process is not clear.	
Further plans: The specification needs further investigation. The WSN will be considered as a candidate specification wherever there is a need for notification functionality. Currently notification support is a low priority area of ARC development.	

2.2.11 Secure Addressing Profile 1.0 (GFD.131)

Organisation/Group	OGF, OGSA-WG
Reference	http://www.ogf.org/documents/GFD.131.pdf
Status/Type	OGF proposed recommendation (P-REC) released on 2008-06-13.
Short description: This document defines a profile about how to bind WS-SecurityPolicy specification within WS-Addressing endpoint references, and how such endpoint references can be made to be tamper-evident, in order to securely convey the information about secure communication mechanisms required by endpoints.	
Relevance to ARC: The functionality covered by the specification is currently out of scope of ARC, no immediate use cases are known. Furthermore, since the profile has no existing implementations yet, there is no interoperability requirement either.	
Current ARC conformance status: Not supported.	
Potential issues: There is no existing implementation based on this profile which is used in production environment. The community acceptance is not clear.	
Further plans: The support of this profile is currently not planned. Wait for broader community support.	

2.2.12 Secure Communication Profile 1.0 (GFD.132)

Organisation/Group	OGF, OGSA-WG
Reference	http://www.ogf.org/documents/GFD.132.pdf
Status/Type	OGF proposed recommendation (P-REC) released on 2008-06-13.
Short description: This document defines an interoperability profile for the secure communication with Web services. This profile has three primary purposes: refining commonly-used security mechanisms profiled within the WS-I Basic Security Profile 1.0 in order to	

provide transport-level security as well as message-level security; profiling the WS-Security Policy v1.2[15] language to accommodate the inclusion of versioning timestamps and actual security tokens within policy documents; defining normative, reference-able, composable policy documents identifying commonly-used security mechanisms.
Relevance to ARC: The functionality covered by the specification is currently out of scope of ARC, no immediate use cases are known. Furthermore, since the profile has no existing implementations yet, there is no interoperability requirement either.
Current ARC conformance status: Not supported.
Potential issues: There is no existing implementation based on this profile which is used in production environment. The community acceptance is not clear.
Further plans: The support of this profile is currently not planned. Wait for broader community support.

2.2.13 OGSA Basic Security Profile 2.0 (GFD.138)

Organisation/Group	OGF, OGSA-WG
Reference	http://www.ogf.org/documents/GFD.138.pdf
Status/Type	OGF proposed recommendation (P-REC) released on 2008-07-28. This profile obsoletes GFD.86 and GFD.99.
Short description: This document links two OGF profiles on Secure Addressing and Secure Communication with some extensibility points for Grid scenario. OGSA services are expected to use such a profile for each infrastructure capability. This profile can be composed with other basic profiles. In particular this profile satisfies the security requirements of the WSRF Basic Profile 1.0 and can be composed with it. This document obsoletes OGSA Basic Security Profile 1.0 -- Core (GFD.86) and OGSA Security Profile 1.0 – Secure Channel (GFD.99).	
Relevance to ARC: The two underlying profiles are currently not relevant, which implies that this specification is not relevant either. Furthermore, the profile has no existing implementations yet, there is no interoperability requirement either.	
Current ARC conformance status:	

[15] http://docs.oasis-open.org/ws-sx/ws-securitypolicy/v1.2/ws-securitypolicy.html

Not supported.
Potential issues: The profile is closely coupled to the OGSA WSRF-BP (2.2.7). There is no existing implementation based on this profile which is used in production environment. The community acceptance is not clear.
Further plans: The support of this profile is currently not planned. Wait for broader community support.

2.2.14 The Blocks Extensible Exchange Protocol Core (BEEP) (RFC3080)

Organisation/Group	IETF
Reference	http://rfc.net/rfc3080.html
Status/Type	IETF RFC
Short description: This memo describes a generic application protocol kernel for connection-oriented, asynchronous interactions called the BEEP (Blocks Extensible Exchange Protocol) core. At BEEP's core is a framing mechanism that permits simultaneous and independent exchanges of messages between peers. Messages are arbitrary MIME content, but are usually textual e.g. XML. All exchanges occur in the context of a channel – a binding to a well-defined aspect of the application, such as transport security, user authentication, or data exchange. Each channel has an associated "profile" that defines the syntax and semantics of the messages exchanged.	
Relevance to ARC: BEEP can be used to test flexibility of the core components, whether one can easily change from HTTP/SOAP to a more asynchronous protocol.	
Current ARC conformance status: Not Supported.	
Potential issues: None	
Further plans: Very low priority. ARC might provide a mechanism in the core development to extend the communication layer with BEEP.	

2.3 Specifications of no immediate relevance

During the standards evaluation process we tried to analyse at least all the OGF proposed recommendation documents. The specifications listed within this section were found to have very little or no relevance to ARC development because

- they addressed out-of-scope areas
- they were made obsolete by some other specification,
- they were not embraced by the community,
- they did not present a normative specification,
- or simply because of their abandoned or rather preliminary status.

The list of the surveyed and unrelated specifications is kept below for reference[16]:

1. The Open Grid Services Architecture (OGSA), v1.5 (GFD.80)
2. OGSA Data Architecture (GFD.121)
3. OGSA EMS Architecture Scenarios, Version 1.0 (GFD.106)
4. Authorization Glossary (GFD.42)
5. Conceptual Grid Authorization Framework and Classification (GFD.38)
6. Open Grid Services Architecture Glossary of Terms v1.6 (GFD.120)
7. Defining the Grid: A Roadmap for OGSA Standards v1.1 (GFD.123)
8. Site Requirements for Grid Authentication, Authorization and Accounting (GFD.32)
9. OGSA Basic Security Profile 1.0 – Core (GFD.86)
10. OGSA Security Profile 1.0 - Secure Channel (GFD.99)
11. Report for the GGF 16 BoF for Grid Developers and Deployers Leveraging Shibboleth (GFD.79)
12. CA-based Trust Issues for Grid Authentication and Identity Delegation (GFD.17)
13. A Requirements Analysis for a Simple API for Grid Applications (GFD.71)
14. Web Services Agreement Specification (WS-Agreement) (GFD.107)
15. WS-Naming Specification (GFD.109)
16. Web Services Distributed Management (WSDM) v1.1
17. WS-Management (WS-Man) v1.0.0a
18. Application Contents Service Specification v1.0 (GFD.73)
19. Configuration Description, Deployment, and Lifecycle Management (CDDLM) Deployment API (GFD.69)
20. Configuration Description, Deployment, and Lifecycle Management (CDDLM) Component Model v1.0 (GFD.65)

[16] For a detailed description of a particular specification please refer to the previous versions of this deliverable

21. CDDLM Configuration Description Language (CDL) Specification v1.0 (GFD.85)
22. Resource Usage Service (RUS) based on WS-I Basic Profile 1.0 (OGF artf3387)
23. Information Dissemination in the Grid Environment - Base Specifications (GFD.110)
24. Resource Namespace Service Specification (GFD.101)
25. ByteIO OGSA WSRF Basic Profile Rendering v1.0 (GFD.88)
26. OGSA-DMI Functional Specification 1.0 (OGF draft)
27. Web Services Data Access and Integration - The Core (WS-DAI) v1.0 (GFD.74)
28. Web Services Data Access and Integration - The XML Realization (WS-DAIX) v1.0 (GFD.75)
29. Web Services Data Access and Integration - The Relational Realisation (WS-DAIR) Specification (WS-DAIR) v1.0 (GFD.76)
30. WSDM/WS-Man Reconciliation
31. WS-ResourceTransfer (WS-RT) v1.0
32. A GridRPC Model and API for End-User Applications (GFD.52)
33. OGSA Resource Selection Services: Specification (OGF doc14791)

3 Summary

This document concludes three years of intensive middleware standardization work by KnowARC. An important goal set by the project from the very beginning was to promote community Grid standards as the mean to establish Grid as a ubiquitous service provided by very heterogeneous resources.

This was a challenging task, some standards may change rapidly, while the others may become stale. At the same time, it may take years to formalize a new standard – for example, the SRM specification took 7 years to be written.

This complexity is reflected in Table 1, which presents an overview of KnowARC's work on standards since 2006. One can see that a number of key standards (GLUE, GridFTP, JSDL, OGSA-BES, SRM, WS-I BP, WS-I BSP and X.509) remained highly relevant throughout the course of the project and are supported in KnowARC products. Some new specifications have emerged, and on many occasions, the importance of older standards was recognized with time – this is particularly true for security-related standards. Some other specifications, most notably, OGSA, gradually lost their relevance.

The GLUE standard is of special interest, as it was developed with significant contributions from KnowARC, though the project also contributed to standards such as HPC-BP, OGSA-BES and JSDL through public comments and experience documents.

By mid-2009, ARC conforms to 21 community standards, with plans to support 14 more in future. Figure 1 shows an overview of the evolution of KnowARC's assessment of various standards. It can be clearly seen that the number of specifications supported or planned to be supported by KnowARC steadily increases, both through adoption of new emerging standards and through re-assessing relevance of older ones.

	Specification	2006	2008	2009
1	GLUE (GFD.147)	++	++	++
2	GridFTPv2 (GFD.47)	++	++	++
3	JSDL (GFD.56, GFD.136)	++	++	++
4	OGSA-BES (GFD.108)	++	++	++
5	SRM (GFD.129)	++	++	++
6	WS-I Basic Profile	++	++	++
7	WS-I BSP	++	++	++
8	X.509 (RFC3820)	++	++	++
9	UR (GFD.98)	+	++	++
10	WS-Addressing	+	++	++
11	WSRF	+	++	++
12	JSDL HPC Extension (GFD.111)	o	++	++
13	OGSA HPC BP (GFD.114)	o	++	++
14	XPATH		++	++
15	SAML	+	+	++
16	ByteIO (GFD.87)		+	++
17	GSS-API Extensions (GFD.24)	+	o	++
18	MyProxy (GFD.54)	+	o	++
19	GSI Message Specification (GFD.78)	o	o	++
20	WS-Security			++
21	XML Encryption and Signature			++
22	OGSA WSRF BP (GFD.72)	++	++	+
23	XACML	+	+	+
24	DRMAA (GFD.22)		+	+
25	HPC File Staging Profile (GFD.135)		+	+
26	SAGA (GFD.90)		+	+
27	WSN	+	o	+
28	BEEP (RFC3080)	o	o	+
29	OGSA Profile Definition (GFD.59)	o	o	+
30	SecPAL	o	o	+
31	WS-RM	o	o	+
32	JSDL SPMD extension (GFD.115)		o	+
33	OGSA BSP (GFD.138)		o	+
34	Secure Addressing Profile (GFD.131)		o	+
35	Secure Communication Profile (GFD.132)		o	+

36	OGSA (GFD.80)	++	++	-
37	OGSA Glossary (GFD.120)	+	++	-
38	Authorisation Framework (GFD.38)	+	+	-
39	Authorisation Glossary (GFD.42)	+	+	-
40	OGSA Roadmap (GFD.123)	+	o	-
41	AAA Site Requirements (GFD.32)	o	o	-
42	OGSA RSSS (OGF doc13767)	o	o	-
43	WSDM/WS-Man Reconciliation	o	o	-
44	WS-RT	o	o	-
45	GridRPC (GFD.52)	-	o	-
46	OGSA Data Architecture (GFD.121)		o	-
47	OGSA EMS Scenarios (GFD.106)		o	-
48	OGSA-SBP SC (GFD.99)	++	-	-
49	ACS (GFD.73)	+	-	-
50	RUS Service (OGF artf3387)	+	-	-
51	CDDLM CDL (GFD.85)	o	-	-
52	Shibboleth BoF report (GFD.79)	o	-	-
53	Simple API requirements (GFD.71)	o	-	-
54	CA-based Trust Issues (GFD.17)	-	-	-
55	CCDLM API (GFD.69)	-	-	-
56	CCDLM Component Model (GFD.65)	-	-	-
57	OGSA-SBP Core (GFD.86)	-	-	-
58	WS-Agreement (GFD.107)	-	-	-
59	WSDM	-	-	-
60	WS-Man	-	-	-
61	WS-Naming (GFD.109)	-	-	-
62	ByteIO WSF BP rendering (GFD.88)		-	-
63	Information Dissemination (GFD.110)		-	-
64	OGSA-DMI (OGF draft)		-	-
65	Resource Namespace Service (GFD.101)		-	-
66	WS-DAI (GFD.74)		-	-
67	WS-DAIR (GFD.76)		-	-
68	WS-DAIX (GFD.75)		-	-

Table 1: Grid standards and their assessed relevance to KnowARC as defined in Different years. The labels are: "++" – highly relevant and (implemented in 2009); "+" – potentially relevant; "o" – of no immediate relevance; "-" – irrelevant (merged with "o" in 2009). Absence of labels indicates that a standard was not reviewed in the respective period.

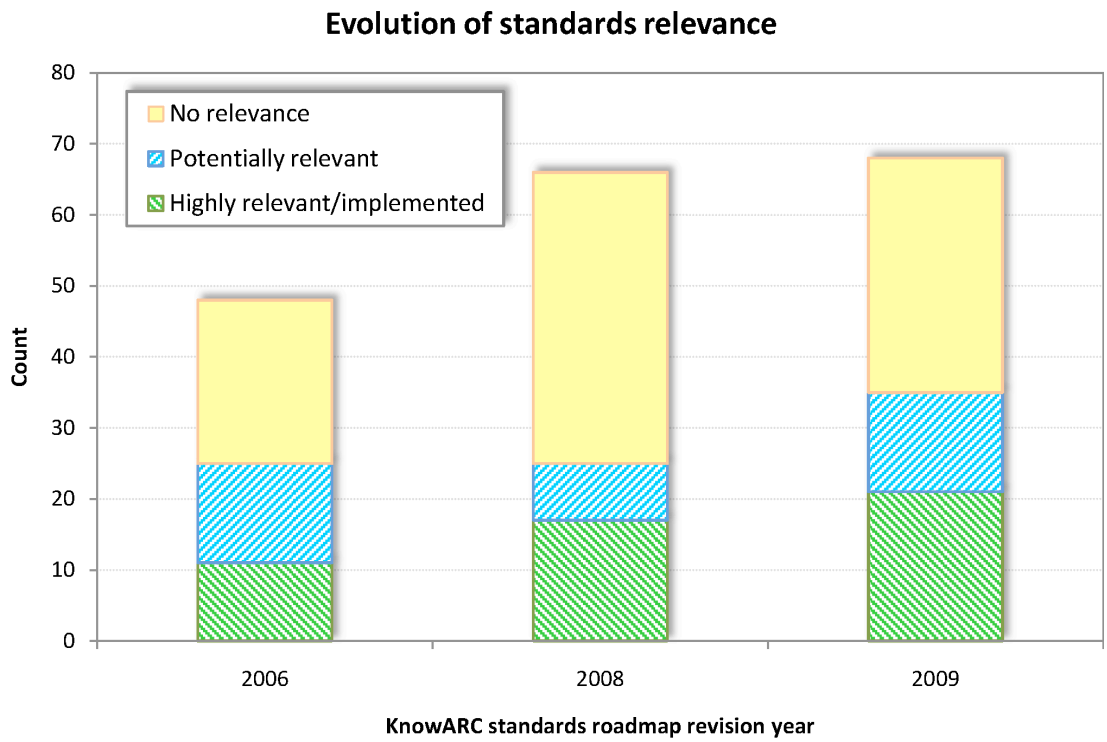

Figure 1: Evolution of Grid standards relevance as assessed by KnowARC in 2006, 2008 and 2009.

Both quantitative and qualitative analysis shows that KnowARC made a major contribution to improving ARC standards conformance, and a quite notable contribution to the worldwide Grid standardization process. This lays a solid foundation for the on-going interoperability processes and future development and usage of ARC.

The project team hopes that other actors in the Grid community will take a similar approach to standards in future, and that proposed bodies such as the European Grid Initiative will enforce the need for standards-driven and standards-compliant development as a key requirement for the success and increased uptake of Grid computing.

4 Acknowledgements

This document is based on D3.3-1 – KnowARC Standards Conformance Roadmap. The deliverable was prepared by Mattias Ellert, Aleksandr Konstantinov, Balazs Konya (editor), Zsombor Nagy, Weizhong Qiang, Gabor Roczei, Oxana Smirnova, and Ferenc Szalai.

This version was prepared with assistance from Owen Appleton, Balazs Konya, Oxana Smirnova and Peter Stefan.