UDDI and WSIL for e-Science

Rob Allan, Dharmesh Chohan, Xiao Dong Wang
UK Grid Support, CLRC e-Science Centre,
Daresbury Laboratory, Daresbury, Warrington WA4 4AD, UK

Mark McKeown
UK Grid Support, Manchester Computing,
University of Manchester, Oxford Road, Manchester M13 9PL, UK

John Colgrave
Architect, IBM WebSphere UDDI Registry,
IBM UK Laboratories, Hursley Laboratory, Winchester SO21 2JN, UK

Matthew Dovey
Oxford e-Science Centre, Oxford University Computing Services,
Wolfson Building, Parks Road, Oxford OX1 3QD, UK

Contact e-Mail: `r.j.allan@dl.ac.uk`

October 14, 2002

Abstract

In this paper we describe how an private UK e-Science UDDI registry or Web Services Inspection document hosted by the Grid Support Centre might be used to register information about e-Science Virtual Organisations and to enable inter-working between them by exposing their contacts and service points. We propose using UDDI and WSIL to provide APIs for information about UK e-Science projects and also show how individual projects might use the same technology. Examples of the latter are the CLRC Integrated e-Science Environment project (IeSE) and EPSRC's MyGrid. These show how UDDI could be used within a single e-Science project for discovery of its own businessEntities and services by high-level components such as applications and portals.

We believe that by providing interfaces to e-Science projects using (proposed) Web services standards, such as UDDI and WSIL, it will facilitate commercial uptake. A partly moderated top-level service will build confidence, allow for testing but still provide the capability to register with the worldwide Universal Business Registry via the publisherAssertion capability as projects become more mature and wish to expose their services to international partners. It nevertheless remains to be seen how the proposed services could be used to enable electronic contract negotiation via the so-called "tModels".

Finally appendices describe UDDI and WSIL implementations and a proposed architecture for accessing Web services through a firewall using a proxy service. Implementations of this architecture will show if the performance is acceptable for a variety of purposes.

This document will from the basis of the UK e-Science UDDI registry policy document (ref Chapter 9 UDDI version 3 Spec).

Keywords: UK e-Science, Grid Architecture, Virtual Organisations Grid projects, User Management, Grid Information Systems

Contents

1 Introduction

In this paper we describe how an private UK e-Science UDDI registry or Web Services Inspection document hosted by the Grid Support Centre might be used to register information about e-Science Virtual Organisations and to enable inter-working between them by exposing their contacts and service points. We propose using UDDI and WSIL to provide APIs for information about UK e-Science projects and also show how individual projects might use the same technology. Examples of the latter are the CLRC Integrated e-Science Environment project (IeSE) and EPSRC's MyGrid. These show how UDDI could be used within a single e-Science project for discovery of its own businessEntities and services by high-level components such as applications and portals.

We believe that by providing interfaces to e-Science projects using (proposed) Web services standards, such as UDDI and WSIL, it will facilitate commercial uptake. A partly moderated top-level service will build confidence, allow for testing but still provide the capability to register with the worldwide Universal Business Registry using the publisherAssertion capability as projects become more mature and wish to expose their services to international partners. It nevertheless remains to be seen how the proposed services could be used to enable electronic contract negotiation via the so-called "tModels" (trading models).

1.1 UDDI: Universal Description, Discovery and Integration

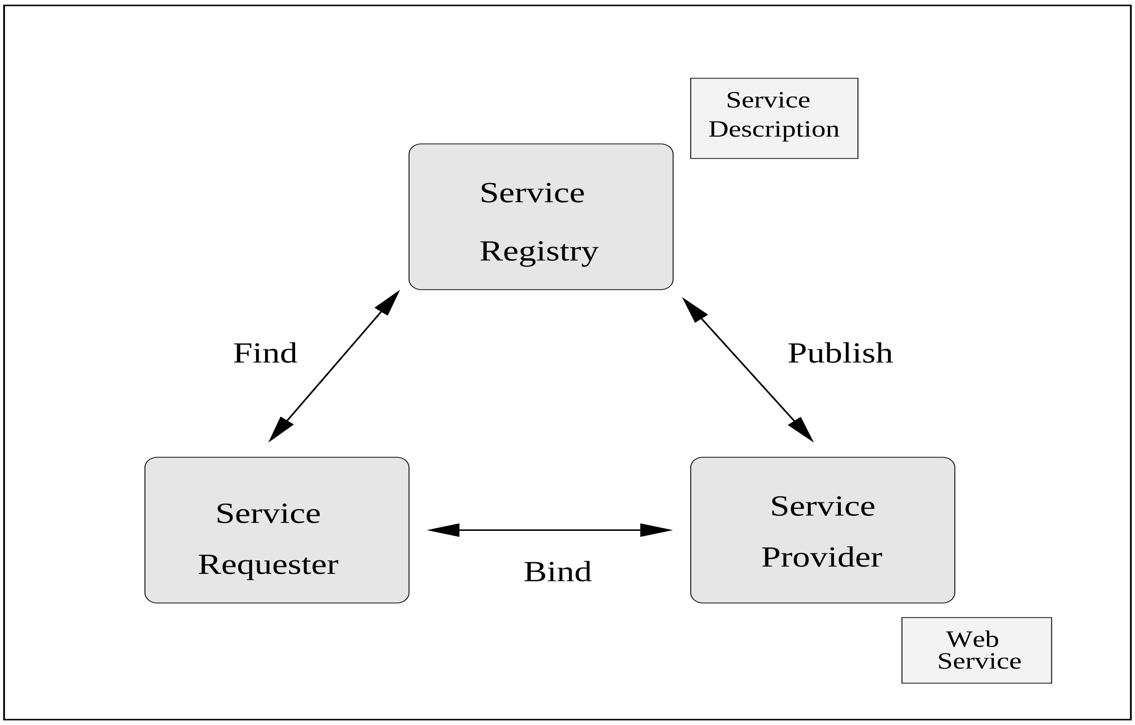

Figure 1: Web Services Architecture

UDDI provides a mechanism for clients to dynamically find other Web services, see Figure 1. Using

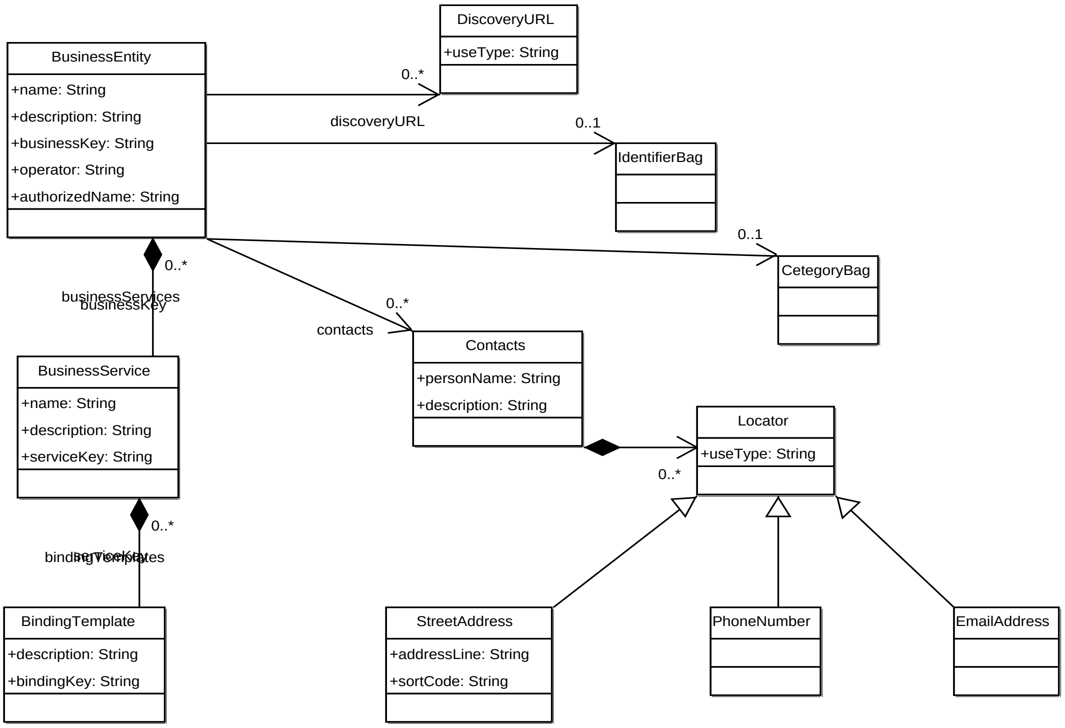

Figure 2: Simplified UDDI Schema

a UDDI interface, applications such as portals may contain clients which can dynamically connect to services provided by other applications. A UDDI registry is similar to a CORBA trader, or it can be thought of as a DNS service for applications. A UDDI registry has two kinds of clients: applications that want to publish a service (and its usage interfaces); and clients that want to obtain services of a certain kind and bind programmatically to them. UDDI is typically layered over SOAP and assumes thatrequests and responses are UDDI objects sent around in SOAP messages. An introduction to Web Services, based on XML, SOAP and WSDL is given in a separate document [1].

Table 1 gives a guide to the information content and uses of a UDDI registry, and Figure 2 gives a simplified view of the relationship between some of the elements of the UDDI schema. This is based on the UML class diagram of Dave Carlson `http://www.XMLModeling.com`, but we have omitted references to the tModels and other details.

There are plans for UDDI to support more complex logic, including support for hierarchical business organisations. UDDI has fairly broad support since IBM,Ariba, and Microsoft are driving it, but it is not yet an open standard. We will return to this when discussing links to the UBR below.

1.1.1 Potential Uses of UDDI

In an article for the IBM DeveloperWorks e-Zine, Steve Graham discusses the private uses for UDDI and classified possible types of service [15]. Under his classification scheme we envisage e-Science UDDIs to be useful as both an *e-Marketplace UDDI* and as a *Testbed UDDI*.

Information	**Operations**	**Detailed information** (supported by lower-level API)
White pages: Information such as the name, address, telephone number, and other contact information of a given application	**Publish:** How the provider of a Web service registers itself via the Publish API.	**Application information:** Contained in a `businessEntity` object, which in turn contains information about services, categories, contacts, URLs, and other things necessary to interact with a given application.
Yellow pages: Information that categorises applications. This is based on existing (non-electronic) standards	**Find:** How an application finds a particular Web service via the Inquire API.	**Service information:** Describes a group of Web services. These are contained in a `businessService` object
Green pages: Technical information about the Web services provided by a given application.	**Get:** How an application connects to, and interacts with, a Web service after it's been found via the Inquire API.	**Binding information:** The technical details necessary to invoke a Web service. This includes URLs, information about method names, argument types, and so on. The `bindingTemplate` object represents this data.
		Service Specification Detail: This is metadata about the various specifications implemented by a given Web service. These are called `tModels` in the UDDI specification

Table 1: UDDI

e-Marketplace UDDI

This private UDDI node is hosted by an e-Marketplace, a standards body, or a consortium of organisations that participate and compete in the industry (e.g. the UK e-Science Programme). In this UDDI node, all the inquiry APIs (that is, publish and find) may be deployed for access over the Internet by any of the member organisations (in our case the Grid Support Centre).

The entries in this species of UDDI relate to a particular industry or narrow range of related industries. Such a membership process allows the entries to be pre-filtered to include only legitimate businesses participating in the industry. The membership process can also restrict who is allowed to invoke find operations against the UDDI node. For example, a steel manufacturers' association may choose to host a UDDI node, allowing its members to register their business and Web services.

An e-Marketplace UDDI is a target-rich environment for finding Web services metadata for doing business within a particular e-Marketplace or industry. The e-Marketplace UDDI would be the logical place to find industry-specific custom taxonomies (standard product coding hierarchies, specialisations of North American Industry Classification System [NAICS] categories, and so on) as well as standard Web service interface definitions (*tModels*) for common business processes in the industry. For example, vendors would be able to determine which types of purchase order placements are typically made in the steel industry by examining the tModels that are contained in the e-Marketplace's UDDI. As industries agree on standard Web services interface definitions, Web services adoption will accelerate.

This style of private UDDI node allows an e-Marketplace to provide value-adds like quality of service monitoring of the partners' Web services response times and "Better Business Bureau" style industry self-monitoring of its members' Web services business practices.

Testbed UDDI

This type of private UDDI node allows for testing of applications. The testing can be for both requestor applications and provider Web services.

If *acme.com* provides Web service access to their purchase-order placement-system, *acme.com* uses this type of UDDI node to make sure of two things: that the UDDI entries for their Web service are accurate and that applications can use the Web service information to generate proxies from the UDDI entry to access the Web service.

If *acme.com* builds applications to consume Web services, it tests the applications' ability to cope with different variants of Web service description with entries from the testbed UDDI. *Acme.com* will clearly want to run some trials against any new UDDI entry discovered in the operator cloud [1], or an e-Marketplace UDDI or partner's transactional Web presence UDDI. Entries will typically be copied here first. A battery of tests are then run to make sure the application can use the information found in the entry. Only after testing, will the entry will be promoted to the vetted partner's UDDI, or the enterprise application integration UDDI node.

1.1.2 UDDI and Search Engines

There is no near-term plan for UDDI to directly support full-featured discovery e.g. geography-limited searches or bidding and contract negotiation supported by vendors like eLance. Smart search engines are therefore needed to work with data extracted from UDDI registries. Indeed developers expect UDDI to be the basis for higher level services.

Siddiqui [17] describes the steps that might be required of a smart search engine to discover detailed information about how to interact with a business that has published its details via UDDI. This is included in the methods in the UDDI inquiry API and briefly encompasses:

1. find a list of businesses based on some search criteria such as products, product categories, keywords, etc. and obtain their businessInfo structures;

2. find a list of services that any individual business offers;

3. find a list of tModels satisfying specific criteria such as business categorisation. Users can search

[1] A group of operators maintaining mirros of a Universal Business Registry - a public UDDI service. Information registered with any operator will eventually propagate to the others.

for specific “fingerprints” using this method. A tModel will for instance tell you which kind of transaction a business can perform, which may be domain specific;

4. list all binding templates (and therefor access points) for a specific business. A binding template may, via its access point, reference a WSDL document or a fingerprint;
5. get the details of a particular binding template. Get methods may in fact be applied to any of the above-mentioned structures to get the details following an abbreviated find method;

Clearly complex search operations may be built up from combinations of the above.

`it was not totally clear from this how the identifier and category bags come into play...?`

1.1.3 Access Control

Although not mandated in the UDDI version 3 specification it is envisioned that the various implementors of UDDI registry software will allow some form of access control for UDDI registry providers using x509 certificates. The type of control that a UDDI registry provider will have depends on the software implementation that she uses, for example the control offered by IBM Websphere through x509 certificates allows the provider to define a set of users who can publish in the UDDI registry and another set of users who can search the UDDI registry.

For the UK e-Science registry hosted by the GSC it is envisioned that only UK e-Science certificate holders can publish to the registry, this will help prevent bogus or malicious entries being published in the registry.

A VO may have certain policies regarding who may know what services they provide,broadly these policies are: anyone can know what services they provide, only UK e-Science certificate holders can know what services are provided, only UK e-Science certificate holders and a select few others can know what services are provided and finally a VO may only want a few select users to know what services it provides. Implementing these policies for the UK e-Science community may be difficult for the GSC if the level of access control offered by the implementation used can only be applied to publish or find operations, and not to individual entries in the UDDI registry.

One possible solution to the above scenario using an implementation that only allows access control to publish/find operations is to limit find operations to UK e-Science certificate holders. If a VO wants their entries to be world readable then they can export them to the UBR, if the UK e-Science UDDI registry key space is taken from the UBR then this should be a simple operation for the user.

If the VO has more restrictive demands than this as regards who can know about the services they offer then they should create their own UDDI registry over which they would have total control. If they get their initial UDDI registry key space from the UBR then they should be able to export parts of their registry to the UK e-Science UDDI registry or to the UBR. VOs who have these kind of restrictions will probably be in collaboration with industrial partners who may already have a private UDDI registry behind a firewall.

1.1.4 Support for XML Digital Signatures

Under UDDI Version 3 specification publishers to a UDDI registry can sign their entries using XML-Signature Syntax and Processing (see http://www.w3.org/TR/xmlsig-core?). Since it is envisioned that only UK e-Science certificate holders will be allowed to publish entries in the UK e-Science UDDI registry publishers should not need to sign their entries. If however they plan to export their entries to other UDDI registries then they should consider signing their entries in the UK e-Science UDDI registry.

For information on XML Digital Signatures for UDDI see Appendix I of the UDDI version 3 Specification [13].

1.2 Web Services Inspection

Service discovery defines a process for locating service providers and retrieving service description documents and is a key component of the overall Web services model. Web Services Inspection Language, (WS-Inspection) is designed around an XML-based model for the building and aggregating references to existing Web service descriptions, which are exposed using standard Web server technology. The WS-Inspection specification does not define a service description language. WS-Inspection documents are fairly easy to write and maintain.

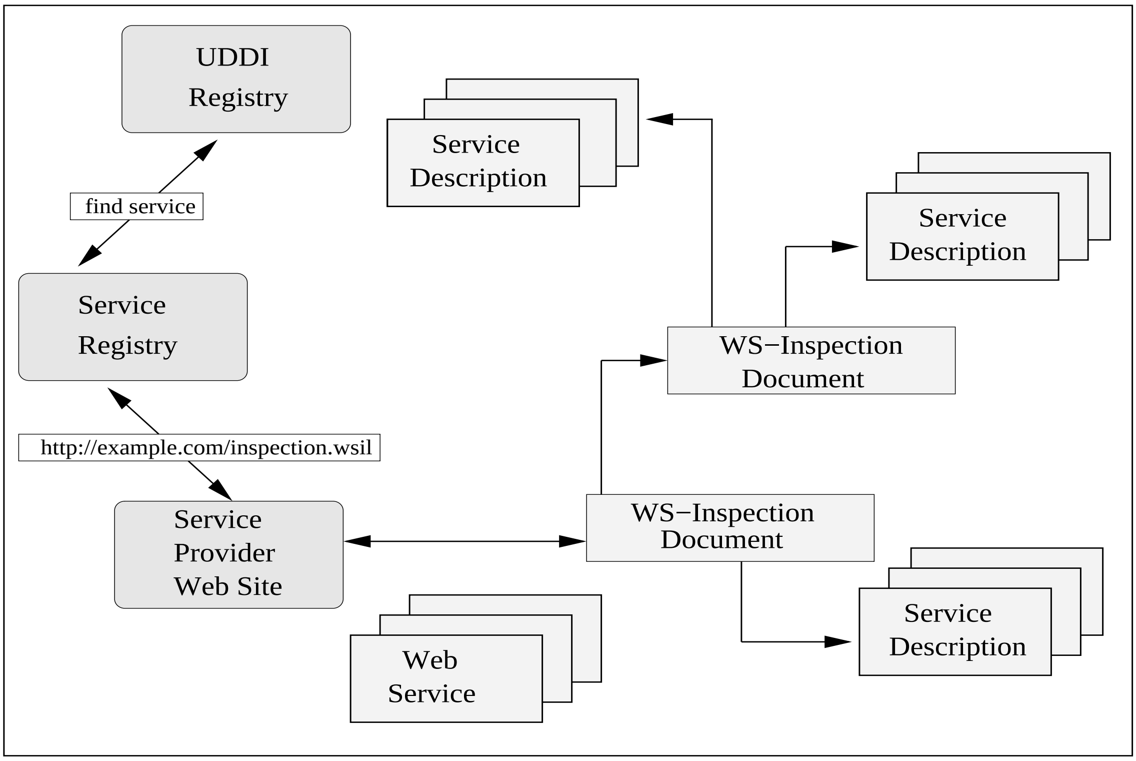

Figure 3: WS-Inspection Overview

Figure 3 shows how a user would retrieve a service description from a service provider using WS-

Inspection. The user retrieves a WS-Inspection document from the service provider's Web site, in accordance with the WSIL specification the document is called *inspection.wsil* and should be in the root directory of the service provider's Web site. The WSIL document can contain: pointers to WSDL documents that describe the services; pointers to UDDI entries that describe the business entity and/or the services provided; or pointers to other WS-Inspection documents.

WS-Inspection is useful when a user knows the service provider he wishes to use and only requires a description of the services provided. It avoids one of the current difficulties with UDDI, namely that entries in UDDI registries are not moderated and a user can not be sure that an advertised service actually belongs to the service provider he thinks it does. For example anyone could place an entry for the UK Grid Support Centre in the UDDI Business Registry but a user cannot check that the entry actually belongs to the GSC. [2]

Within a WSIL document, a single service can have more than one reference to a service description. For example, a single Web service might be described with both a WSDL file and within UDDI. References to these two service descriptions should be put into a WSIL document allowing the user of the document to pick and choose from the available descriptions and to access those which they are able to understand.

One important feature of the WS-Inspection specification is the ability to link a WS-Inspection document to one or more different WS-Inspection documents. This feature can be used to manage service description references by grouping them into different documents, for example a VO could provide two sets of services, A and B, supported by two groups, each group would provide a WSIL document that describes the services they support whilst a top top level document for the VO would hold two entries which would point to each group's WS-Inspection document. A hierarchy of WSIL documents can thus be built using these individual documents.

```
<?xml version="1.0">
<inspection xmlns="http://schemas.xmlsoap.org/ws/2001/10/inspection/">
  <service>
    <description referenceNamespace="http://schemas.xmlsoap.org/wsdl/"
           location="http://example.com/exampleservice.wsdl"   />
  </service>
  <service>
    <description referenceNamespace="urn:uddi-org:api">
       <wsiluddi:serviceDescription location="http://example.com/uddi/inquiry">
         <wsiluddi:serviceKey>
            52947BB0-BC46-11D5-A432-0004AC49CC1E</wsiluddi:serviceKey>
      </wsiluddi:serviceDescription>
    </description>
  </service>
  <link referenceNamespace="http://schema.xmlsoap.org/ws/2001/10/inspection"
    location="http://example.com/tools/toolservices.wsil"/>
</inspection>
```

[2] The use of a "private" UDDI registry proposed here partly gets around this problem by the use of moderation and certificates, but does not fully eliminate it.

The sample WSIL document above contains two references to different service descriptions, and a single reference to another WSIL document. The first <service> element contains only one service description and it is a reference to a WSDL document. The second <service> element also contains only one service description reference, to a business service entry in a UDDI registry. The UDDI service key identifies one unique business service. The <link> element is used to reference a collection of service descriptions, in this case using a WSIL document.

HTML pages can contain links to WSIL documents using a META tag carrying the value of the serviceInspection in its name attribute.

The WS-Inspection specification also defines a convention for finding WSIL documents, they can either be found through links in other documents as described above or through a fixed name: *inspection.wsil.* A WSIL document with this name should be placed at common entry points of a business's Web site. For example, if the company *example.com* has a common entry point `http://example.com` then the location of their WSIL document would be `http://example.com/inspection.wsil`. If they offer services through `http://example.com/financial` and `http://example.com/shipping` then they may also put *inspection.wsil* files accessible under each of those URIs as well. It is possible that the *inspection.wsil* file at the root level, `http://example.com` may only contain pointers to other WS-Inspection files.

The *inspection.wsil* document at the root level of the service provider's Web site can be created dynamically, for example the IBM WS-Inspection toolkit uses a Java servelet that will search for WSDL service description documents within the toolkit directory structure whenever a request is made for an *inspection.wsil* document.

The Grid Support Centre recommends that we create a "top level" WSIL document for all the UK e-Science projects. The document should contain links to each project's WSIL document (if it exists) and/ or reference to the project's business entity UDDI entry (described more fully in Section 2 below). If a project wanted to be included in the UK e-Science WSIL document they would send e-mail to the GSC with a pointer to their WSIL document and/ or a reference to their business entity UDDI entry. The document would be located at the GSC web site, `http://www.grid-support.ac.uk/inspection.wsil`, and possibly mirrored to other relevant Web sites.

For futher details on WSIL check [7, 8, 9, 10].

1.3 LDAP: Lightweight Directory Access Protocol

Currently LDAP is being used to publicise and discover information about Grid resources, in particular computer systems. It is not expected that UDDI will replace this, as LDAP is a hierarchical directory service implemented in this case to respond to a dynamic situation. It is also closely integrated with Globus at present (the MDS component). UDDI is implemented as a single entity, possibly with mirror sites, and is more typically static in nature. This use of LDAP is fully described in [14].

1.4 Conclusion

Clearly things in the Grid and Web services world are changing rapidly, and discovery is no exception. However WSIL and UDDI currently provide stable working platforms for VO-related information which has a relatively long lifetime. We have therefore chosen them for our present purposes.

2 UDDI for UK e-Science

We propose that the UK Grid Support Centre provide a UDDI registry for electronic access to information about UK e-Science projects, or rather Virtual Organisations (VO). A VO is a collection of people collaborating in a well-defined research area, using a set of resources and (usually known via an MDS GIIS server) and with a clearly defined organisation structure to their joint project. We are therefore able to name the project that the VO is engaged in (aka the VO name), define the responsibilities of its primary individuals, describe and categorise its businesses and, where appropriate provide links to its services and other information sources.

The remit of the GSC is to provide the infrastructure for deploying and using a Computational Grid in the UK. The GSC thus already maintains a hierarchical LDAP-based MDS information service which provides a view of all compute resources registered with Globus GRIS and GIIS servers in the hierarchy. UDDI provides a complementary mechanism, in line with the move towards a Grid Service-based environment (e.g. OGSA and GT3), to publish via a programmatic interface information about Virtual Organisations. We envisage much of the data to be provided via the databases at the National e-Science Centre so that GSC does not need to duplicate this function. UDDI also provides a publication mechanism for VOs to register with the directory, and scripts may need to be provided to periodically update the databases based on these electronic registrations.

`Discussion of the Universal Business Registry (UBR)` [3]`. Allow projects to export to public business registry. Need UBR business key.`

2.1 UDDI Schema and Project Information

As described above, it is possible, with only small changes to the original meaning, to adopt the usual UDDI e-Business schema. Some notes on this are now provided. Further references can be found via `http://www.uddi.org`. Some of the elements discussed are shown in Figure 2.

businessEntity: Full name of the VO (and/ or the project).

description: A short verbal description of the VO including the project description. Should contain enough information to enable a party interested in a particular area of science to decide if this is a useful VO to contact.

discoveryURL: URL of a Web server maintained by the VO which will provide further information on the project and its organisation. The useType elements which are contained in the discov-

[3] The UBR is maintained by IBM, Microsoft and Ariba see `http://www.ubr.com`

eryURL enable us to reference any information that can be acquired using the HTTP GET protocol, such as Web URL, WSIL URL or other businessEntity.

contacts: An unordered list of VO member contacts for the project together with a description of their responsibilities. Contact details (locators) may include phoneNumber, emailAddress, streetAddress, etc. as character strings. Since in UDDI v3.0 multiple names can be associated with a contact we recommend that at least the proper name and Distinguished Name (from the person's e-Science X509 certificate) be registered. The responsibilities (roles) of the contacts could include:

recommended	principal investigator PI project manager PM
suggested	commercial manager administrator architect technical authority chair work package manager steering group chair programme board chair Web or publicity manager work package member

businessServices: Links to services provided by the project. Links may be URLs to service descriptions or URLs to WSDL documents.

tModel: The "trading model" (tModel in the language of UDDI) is an important element, or set of elements. It identifies how a potential collaborator could negotiate a contract with the registered business service provider. This represents an enormous potential area of research for e-Science as yet unexplored.

identifierBag: The identifierBag contains a list of unique identifiers for the business, e.g. the NAICS reference etc. For the e-Science programme it is perhaps useful to have the original grant references from EPSRC, NERC, BBSRC and other funding agencies.

categoryBag: Using the ETC type, an unordered list of English keywords which can be used to identify the kind of project being described. Could be used by search engines if published in an appropriate format. We will not discuss this further here, but to make the UDDI useful as an information system we wish to consider the use of ontologies in defining the keywords and their associations. Several e-Science projects are already working in this area of knowledge management.

The following section considers taxonomies in more detail.

2.2 Taxonomy and the UK e-Science Registry

Taxonomies and identifier systems play an important role within UDDI. It is through categorisation and identification that businesses are able to find each other and the services that meet their needs. UDDI facilities the publishing of taxonomies.

A provider of a taxonomy or identifier system can allow unrestricted references or it may choose to validate references. Taxonomy and identifier systems that allow unrestricted references are called unchecked. Conversely, a taxonomy or identifier system that requires references to it to be validated is called checked. A checked taxonomy or identifier system must have an associated validation service that performs value checking each time an attempt is made to save data containing a reference to the taxonomy or identifier system.

Publishing an unchecked taxonomy in a UDDI registry is straight forward, the user need only publish a tModel for it with the UDDI operator. Publishing a checked taxonomy in a UDDI registry is more complicated because the UDDI node operator must check the publishers validation service works correctly before it can be published within the registry. As the GSC will be the UK e-Science UDDI registry operator, VOs wishing to publish checked taxonomies in the UK e-Science UDDI registry should contact the GSC to choreograph the steps required. An outline of the procedure for publishing checked taxonomies can be found in *Providing A Taxonomy for Use in UDDI Version 2* [16].

2.3 UDDI publisherAssertion Structure

The recommended practice for a business with many large sub-divisions that wants to register in a UDDI registry is for each division to register as a separate businessEntity. For example Microsoft might have separate businessEntity entries for their Office divisions, OS divisions etc. This maps well to the structure of the UK e-Science community, rather than have a single businessEntity entry in a UDDI registry which attempts to describe all the services provided by the all UK e-Science projects, each project would have a separate businessEntity which they maintain. Some large geographically dispersed projects might take this further and have multiple businessEntity entries in the UDDI registry representing the various groups working within the project (see example in Section 3 below). It would be useful to make the relationship between UK e-Science projects visible within the UDDI registries, this can be accomplished with the publisherAssertion structure which allows businesses to publish relationships between businessEntitys. To eliminate the possibility that one publisher claims a relationship to another that is not reciprocated, both publishers must publish identical assertions for the relationship to become visible. The allowed relationships are peer-peer (the two businessEntity structures are related as peers) or parent-child (one businessEntity structure is in some sense the parent to the other).

The GSC recommends that in any UDDI registrations that a UK e-Science project might want to make, for example with the UBR, a businessEntity structure is created which will act as the parent of all the UK e-Science project businessEntitys. This *UK e-Science* businessEntity would have limited information and services and should be maintained by the GSC. When a UK e-Science project registers a businessEntity in the UDDI registry they would publish a publisherAssertion that they are a "child" of *UK e-Science*, they should then contact the GSC who can then complete the assertion at which point the relationship will become visible in the UDDI registry.

If this procedure is followed it will allow users to find all the registered UK e-Science projects within a UDDI registry very easily, by calling *find_relatedBusiness* with the UK e-Science businessKey. It is possible that the *UK e-Science* businessEntity could have the same businessKey across different UDDI registries depending on how the registries are initially set up (see Chapter 8 of the UDDI Version 3 Specification [13]). It also addresses the issue of spurious or malicious entries within a UDDI registry, currently no-one moderates entries in the public UDDI allowing people to register any businessEntity

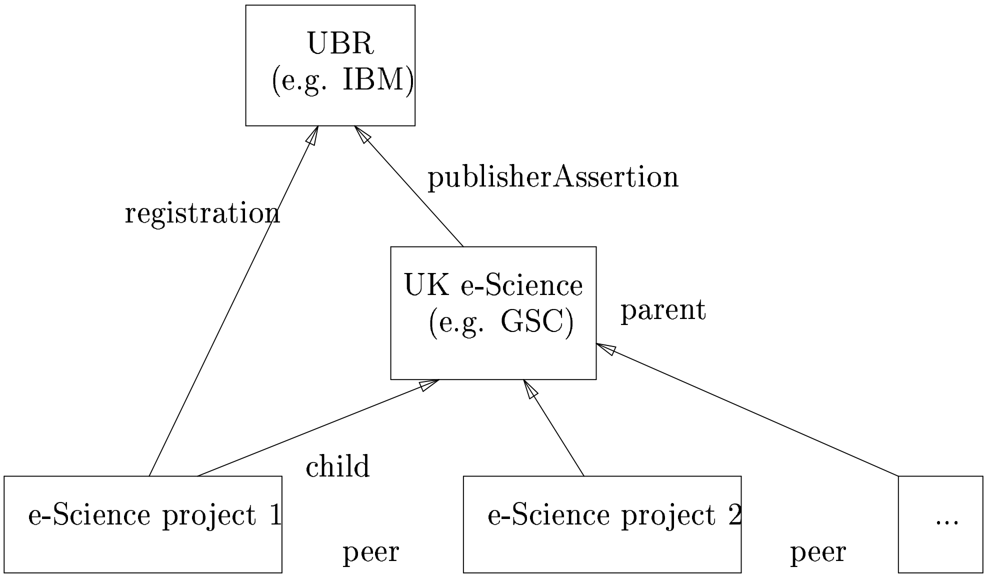

Figure 4: UBR registration architecture

and business services they want. If a user knows that an businessEntity in a UDDI registry is related to the *UK e-Science* businessEntity then they can be confident of the services provided by that businessEntity.

There should be no need for an explicit *UK e-Science* businessEntity within any private UDDI registry hosted by the GSC since all businessEntity structures would be UK e-Science projects by default.

Further information on the publisherAssertion structure can be found in Section 3.7, Section 11.1.4 and Appendix A of the UDDI Version 3.0 Specification [13].

2.4 Sharing Data between the UK e-Science UDDI registry and other UDDI Registries

Before UDDI version 3 a publisher could not import or export data between UDDI registries. Under version 1 and 2 of the UDDI specification only nodes within a UDDI registry could generate keys for entries, a publisher could not pre-assign the key of a UDDI entry thus preventing duplicate keys appearing in a registry. Under version 3 of the specification publishers can assign the key for an entry however for this capability to be useful care must be taken to avoid key collision between registries. The recommended way to prevent such collision is to establish a root registry that acts as authority for key spaces. In the case of the UK e-Science UDDI registry the GSC will get the key space for the registry when the implementation moves to version 3 from the UBR. This will guarantee that publishers in the UK e-Science UDDI registry will be able to export their entries to the UBR or any other UDDI registry that has taken its key-space from the UBR. Conversely they will be able to import data from other UDDI registries that have taken their key-space from the UBR into the UK e-Science UDDI registry. For further details on publishing across multiple registries see Chapter 8, The UDDI version 3 Specification [13].

2.5 Reliability Issues

Web services software platforms including UDDI are described in Appendix ??.

`Here we consider issues connected with running a public service for e-Science, backups, mirrors, UBR node guidelines...`

3 UDDI for an Individual e-Science Project

In this section we outline how UDDI might be used to provide information about the organisation and services of a given e-Science project. To illustrate this we have chosen the CLRC Integrated e-Science Environment (IeSE) project [11].

IeSE is a "federated" set of services which can be (re-)used in many combinations to support experimental or computational projects in science and engineering, and combinations of these. Our first way to access the services was through portals customised to particular needs, e.g. HPCPortal and DataPortal. As more services are added to the system, and older services are replaced by better versions, service discovery becomes an issue.

At the present time we have arbitrarily classified IeSE services as "HPC Services", "Data Services" and "Visualisation Services". These arise from the three prototype components of IeSE and are will define its businessEntities. Other parts of the schema might then be used as follows.

3.1 UDDI Schema and Service Information

As described above, it is possible, with only small changes to the original meaning, to adopt the usual UDDI e-Business schema. Some notes on this are now provided. Further information about UDDI, references and implementations can be found via `http://www.uddi.org`.

businessEntity: Sub-project, workpackage, component or "view" of the services which have a particular meaning or reflect their common usage. A sub-project might be a group of service developers (producers) or consumers. The mechanism for asserting that sub-projects are related was discussed in Section 2.3 above.

description: Description or intention of the sub-project or view.

discoveryURL: More information about the sub-project using that view.

contacts: An unordered list of contacts for the sub-project with their responsibilities. Contact details may include: phone; e-mail; snail-mail; URL etc.

services: Links to services provided for the sub-project. Links may be URLs to services descriptions or URLs to WSDL documents. Services may belong to more than one sub-project, but form part of the complete set of services offered by the VO via its private UDDI, e.g. the IeSE project.

tModel: The "trading model" (tModel in the language of UDDI) is an important element, or set of elements. It identified how a potential customer could negotiate a contract with the registered business service provider.

identifierBag:

categoryBag: Using the ETC type, an unordered list of English keywords which can be used to identify the kind of project being described. Could be used by search engines if published in an appropriate format.

4 Conclusions

We have outlined about how UDDI might be used as a programmatic Web services information directory for e-Science projects and for services within a single project providing its community with multiple "views" of sub-projects.

UDDI will probably need to be supplemented with a "top-level" WSIL document accessible from the UK Grid Support Centre Web site. The UDDI and WS-Inspection sources can be cross referenced.

A private UDDI for the projects of the UK e-Science Programme offers the chance to test the publication of Web services and provide some degree of assurance that the published services will be acceptable to the wider community. Services may also be published to the Universal Business Registry and the publisherAssertion mechanism of UDDI v3.0 can be used to guarantee that they do belong to the programme.

References

[1] Rob Allan, Dharmesh Chohan, Xiao Dong Wang and Daniel Hanlon *An Introduction to Web Services and related Technology for building an e-Science Grid.* (CLRC e-Science Centre, 2002) draft

[2] I. Foster, C. Kesselman, J.M. Nick and S. Tuecke *The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration* `http://www.globus.org/research/papers/ogsa.pdf`

S. Tuecke, K. Cjajkowski, I. Foster, J. Frey, S. Graham and C. Kesselman *Grid Service Specification* (2002) `http://www.globus.org/research/papers/gsspec.pdf`

[3] S. Graham, S. Simeonov, T. Boubez, G. Daniels, Y. Nakamura and R. Neyama *Building Web Services with Java: Making Sense of XML, SOAP, WSDL and UDDI* (Sams, 2001)

[4] J. Hodges *Introduction to Directories and LDAP* (June 1997) `http://www.stanford.edu/~hodges/talks/mactivity.ldap.97/index2.html`

[5] L. Zeilenga, R.L. Morgan et al. *LDAP v3 revision* `http://www.ietf.org/html.charters/ldapbis-charter.html`

[6] *UDDI Specifications* `http://www.uddi.org`

[7] Peter Brittenham *Web Services Inspection Language Specification* IBM `http://www.ibm.com/developerworks/webservices/library/ws-wsilspec.html`

[8] Peter Brittenham *An overview of the Web Services Inspection Language* IBM `http://www.ibm.com/developerworks/webservices/library/ws-wsilover`

[9] Peter Brittenham *The WS-Inspection and UDDI Relationship* IBM `http://www.ibm.com/developerworks/webservices/library/ws-wsiluddi.html`

[10] Peter Brittenham *WSIL: Do we need another Web Services Specification?* IBM `http://www.webservicesarchitect.com/content/articles/modi01.asp`

[11] R.J. Allan, D.J. Hanlon, K. Kleese - van Dam, S. Sufi, L. Sastry, D.R.S. Boyd and B.M. Matthews *An Integrated e-Science Environment for CLRC: Grid Portals, HPC Computation, Data Management and Advanced Visualisation* Project Overview (CLRC, 2002), `http://www.dl.ac.uk/TCSC/UKHEC/GridWG/`

[12] WSIL `http://www.apache.org/~dug/wsil4j-v1.zip`

[13] *UDDI Version 3.0 Specification* `http://www.uddi.org`

[14] K. Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman *Grid Information Services for Distributed Resource Sharing* in Proc. Symposium on High-Performance Distributed Computing (HPDC-10) (IEEE Press, August 2001)

See also *Hierarchical MDS* paper under MDS 2.2 installation instructions. `http://www.globus.org/mds`

[15] Steve Graham *The role of private UDDI nodes in Web services, Part 1: Six species of UDDI* IBM Developer Works (17/5/2001) `http://www-106.ibm.com/developerworks/library/ws-rpu1.html?n-ws-5171`

[16] *Providing A Taxonomy for Use in UDDI Version 2*

[17] B. Siddiqui *Using UDDI as a Search Engine* (24/10/2001) `???`

Using SOAP as a UDDI as a Search Engine

A Architecture for Secure Access to Web Services

This proposal outlines the setting up of a UDDI registry for the UK e-Science community. The UDDI registry could be hosted on a Linux machine (say *esc5.dl.ac.uk*). Access would be via a Web browser with a specific URL, e.g. `http://www.grid-support.ac.uk/uddi` which could be directed to a server, say *esc.dl.ac.uk*. We can access the actual service through a firewall using a proxy Web service which has named methods equivalent to the actual Web service but simply re-directs the communication

This architecture is a generic example of how Grid information servers could be made accessible via a Web proxy or “gateway” machine.

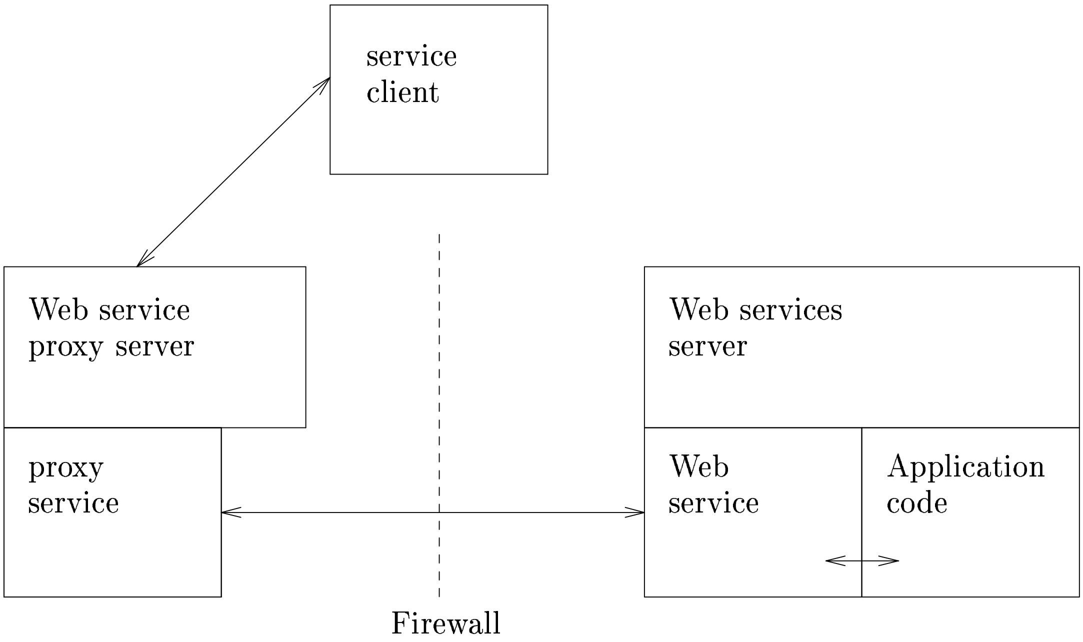

Figure 5: Proxy services through a firewall