

Supporting Collaborative Virtual Organisations in the Construction Industry via the Grid

Liviu Joita¹, Jaspreet Singh Pahwa¹, Pete Burnap¹, Alex Gray¹, Omer Rana¹,
John Miles²

¹School of Computer Science, Cardiff University, Queen's Buildings, Newport Road, PO Box 916, Cardiff, CF24 3XF, Wales, U.K.

²Cardiff School of Engineering, Cardiff University, Queen's Buildings, The Parade, PO Box 925, Cardiff, CF24 0YF, Wales, UK.

Abstract

A typical AEC industry project involves many individuals and companies forming a consortium for the duration of a project. These projects are usually unique, very complex and involve many participants from a number of organizations acting collaboratively, and forming Virtual Organisations to enable this collaboration. This document is intended to describe the complete design and implementation of a Grid-enabled Product Supplier Catalogue Database (PSCD) application. As part of the Grid-enabling process, a Security Management Service and a Multiple Database Search Service are developed, as well as specialised metadata, to enable PSCD to more effectively utilise Grid middleware such as Globus and Java CoG toolkits.

1. Introduction

In the Architecture/ Engineering/ Construction (A/E/C) industry, large projects are tackled by consortia of companies and individuals, who work collaboratively for the duration of the project. Such projects are complex and the consortia members provide a range of skills to the project from its inception to completion. The planning, implementation and running of these A/E/C industry projects requires the formation of secure virtual organisations (VOs) to enable collaboration between its members by sharing project information and resources. An important feature of the consortia is that they are dynamic in nature and are formed for the lifetime of the project. Members can participate in several consortia at the same time and can join or leave a consortium as the project evolves.

This is a complex situation to support with software and the Grid will be an important (vital) infrastructure in future systems supporting this type of working. The Grid is perceived as providing additional functions to the existing functionalities of the Internet. As well as high speed networks it offers features such as enhanced security infrastructure including single sign-on capability, security between consortia, simple setting up of networks to support VOs, distribution of

computationally intensive jobs across multiple distributed processors and resource information sharing.

This paper describes the complete design, development and implementation of the Grid-enabled solution to the Product Supplier Catalogue Database (PSCD) application, an ActivePlan Solutions Ltd. (www.activeplan.co.uk) software package, which is part of such of VO, which brings together designers, contractors, suppliers and product manufacturers to work collaboratively in multiple consortia in a virtual environment. Collaboration occurs between: product suppliers/manufactures and contractors for procurement of supplies using the distributed Grid-enabled Multiple Database Search Service (MDSS); product specification designers for defining and building industry standards to describe available products; and members of the consortia working on a particular construction project that requires information on the products. Figure 1 gives a conceptual view of the PSCD application and its collaborative aspects using the Grid infrastructure.

In this paper we describe our experience whilst Grid enabling the PSCD application using Grid middleware such as Globus and Java CoG toolkits and the development of software to support these three types of consortium. As

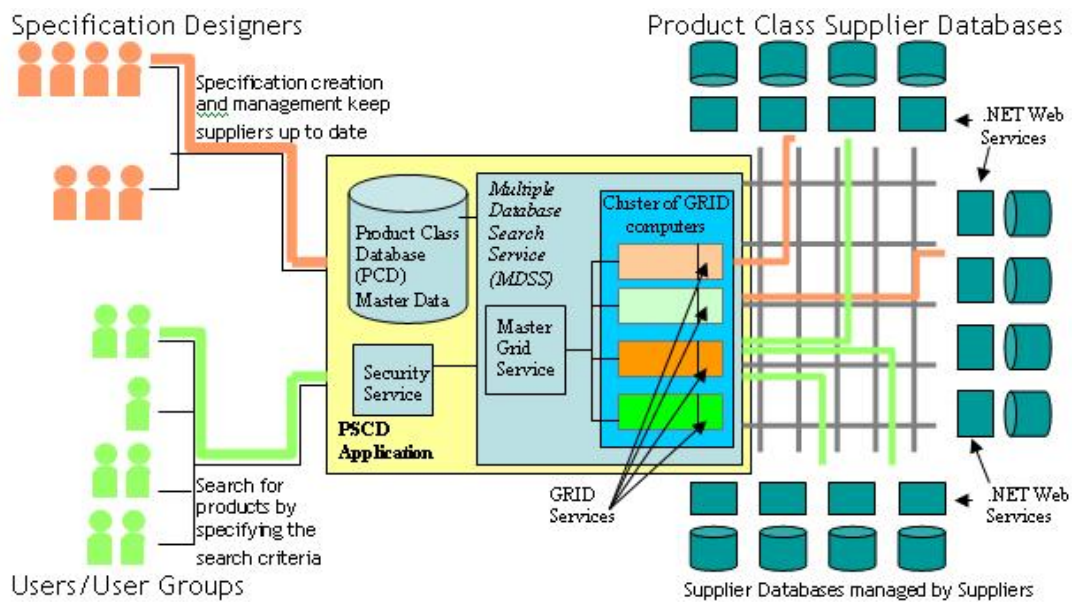


Figure 1 - Grid-enabled PSCD application diagram

the Grid is new and unstable in terms of the software and documentation we have encountered problems during our development phase which are outlined in the paper. The PSCD application is divided into two functional services: Security Management and Multiple Database Search Service.

The Security Management service defines a security framework for the PSCD application using the Globus Security Infrastructure (GSI). GSI is based upon Public Key Infrastructure (PKI) and requires users to have a private key and an X.509 certificate used to authenticate to the Grid services. The important feature of GSI is the single sign-on capability and the ability to perform delegation, known as a proxy, to perform the authentication to the PSCD resources on a user's behalf. The proxy can be enabled on the user's machine or on a designated MyProxy Server machine [3]. This facility is incorporated in the Globus and Java CoG toolkits. The Security Management service also provides the capability of role-based privileges within the VO.

In the construction industry a consortia procures supplies such as building materials, furniture, air management systems, etc from suppliers who specialise in manufacturing or retailing these products. For the members of the consortia to reach a large number of suppliers and to precisely identify what products and services would best suit the project requirements, taking into consideration facts such as materials cost, delivery time, etc, an infrastructure is required to enable the members of the consortia to make a search across a large

number of product suppliers for the desired products. The MDSS system incorporates an infrastructure which we call the Product Class Database (PCD) system. The PCD System functions within the overall PSCD application and serves its data needs. The Security Service plays an important role in preventing members of a VO accessing data of other VOs. The development of the PCD System is concerned with making available to members of the project design consortium information about products which can be acquired from external suppliers so that availability, delivery and cost can be taken into account in the collaborative planning of the project.

The rest of the paper is organized as follows. In section 2, we discuss the PSCD Security Management Service. Section 3 illustrates the Product Class, while Section 4 describes the MDSS System. Conclusion follows in Section 5.

2. The PSCD Security Management Service

Security aspects rely on keeping important and sensitive information in the hands of authorized users. There are four important issues to deal with: authentication – being able to verify the identities of the parties involved; authorization – limiting access to resources to selected users or programs; confidentiality – ensuring that only the parties involved can understand the communication; integrity – being able to verify that the content of the communication is not changed during transmission [1].

Without a strong authentication, an unauthorized user, as a single user or as part of a VO, can access proprietary web resources containing information of other VOs. The Security Management Service developed for the PSCD application is based on the GSI [1], a client-certificate authentication system, where users are identified by a globally unique name known as Distinguished Name (DN). A format example of a DN (also known as a subject name) is: "C=UK, O=eScience, OU=Cardiff, L=WeSC, CN=firstname lastname".

Currently, Web browsers and Web servers do not support the concept of delegation. This means the creation of a lifetime limited private key and a certificate pair, known as a proxy, which can be used to authenticate to Web resources. The GSI provides the security mechanism for a user to delegate its credentials to the Web resources. This can be done using either Globus or Java CoG toolkit facilities. The challenge to build a secure access to PSCD web resources is to require the integration of the Security Management Service of the PSCD application into a single user-friendly service using the capabilities of GSI. Since the user accesses PSCD web resources remotely, it must be possible to establish the user's identity, the user's role and the VO the user belongs to with certainty. A Tomcat Web server is used to host the Security Management Service Module and to handle the connections to the PSCD application via the HTTPS protocol, as well as a MyProxy credential repository to host users' proxies. The entire architecture, design and implementation of the Security Management Service are presented in [2].

The PSCD application offers to the user two methods of being authenticated and authorized: (1) a proxy certificate which has been previously enabled on his/her local machine or (2) a proxy certificate that resides on the MyProxy credential repository [3] located at the PSCD application site. The user has to provide the username and password that protect his/her proxy certificate on the MyProxy credential repository.

The user's certificate verification plays an important role in preventing unauthorized users access to PSCD resources. The Security Management Service has to deal with the following important checks in order to verify a certificate:

- to verify that a third party (the Certificate Authority) which the PSCD application trusts issued the chain's first certificate.
- working down the chain of the proxy certificate, for every certificate in the chain, to

verify that the certificate's subject is the issuer of the next certificate in the chain.

- for every certificate in the chain, to verify that the certificate is valid at the current time.

- for every certificate in the chain, to verify that the certificate is not in the CRL (Certificate Revocation List) of the Certificate Authority (CA).

The PSCD application is using as a third party, for testing purposes, the UK eScience Certificate Authority and allows users with the certificates issued by this CA to access the application resources

3. The Product Class

The system works by utilising data structures named Product Classes – which are created by a team of industry knowledgeable specification designers. These classes are made up of a collection of assigned attributes; specification objects, specification tables, specification groups, specification lists and other previously released classes. The type of attribute used depends on the complexity of the representation of a specification type within a database. The data related to these classes is stored on the server side in what is known as the Product Class Database (PCD). An example of such a Product Class could be an air conditioning fan coil unit. It would have specifications such as Chassis, Fans, Motors and Filters, which all have their own sub-specifications – for example a Fan would have criteria such as air flow rate, cooling duty and electrical supply, which again have a unit of measurement as a specification. The Product Class in itself forms a tree structure of attributes that is represented in the system as a data structure in a database (Figure 2).

There is collaboration between the Specification Designers when designing the Product Classes and specification types. A Product Class can be defined as a template made up of a number of different specification types. Product Classes aid product suppliers to populate their databases with their own products in a structured, standardised manner.

Before Product Classes can be released, they need to be reviewed and accepted by each member of the Product Class design team. A future development in the system is going to be a reviewing method that allows each newly created or new version of a Product Class to be tagged with changes and comments before it is accepted. Once the class has been accepted by all members of the team it can be released for use by others. Over time products evolve and often have new attributes assigned to their

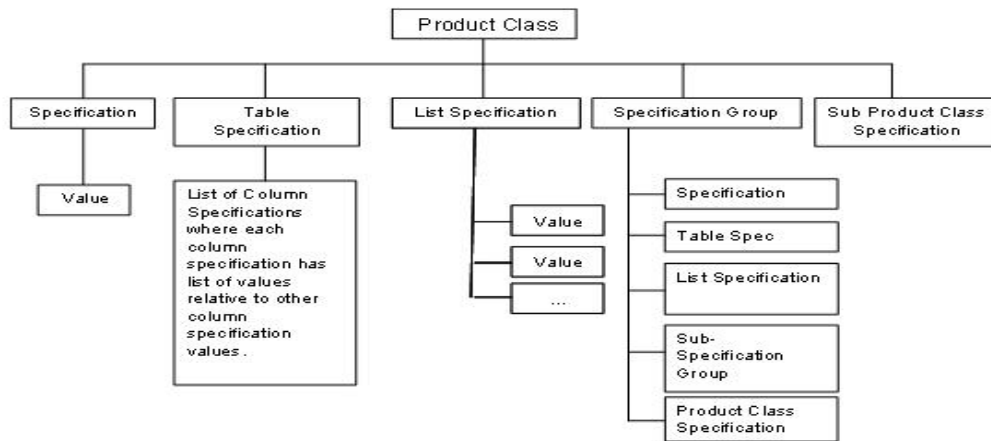


Figure 2 – The Product Class Data Structure

definition. As this process takes place the application must keep track of changes so as to maintain consistency of standardization. As mentioned earlier, newly created or updated Product Classes have to be tagged and reviewed before they are released. When a new version of an existing Product Class is created, it is given a new unique individual ID and added to the list of classes as a transient non-released class. When it has been accepted by all members of the design team it will be released and all suppliers that subscribe to the old version of this class will be notified of the new version release so that they can download the current version and update their databases with the latest product information.

3.1 Product Class Subscription

As illustrated in Figure 3 the Product Classes are subscribed to by product suppliers/manufacturers that supply those particular products. By subscribing to a class, they are provided with the industry defined data

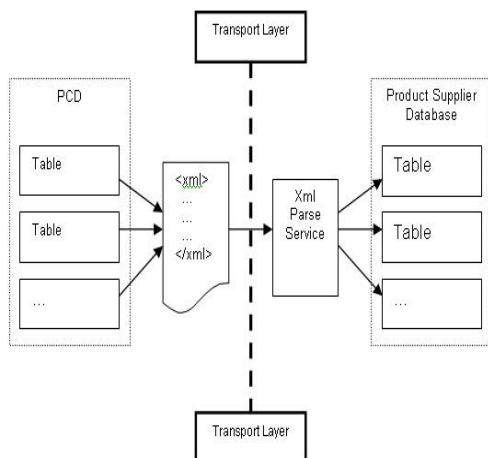


Figure 3 - Product Class subscription architecture

structure for that class, allowing the supplier/manufacturer to populate their own internally maintained databases (PSCDs) with details about their own products that fall under the category of the subscribed class.

```

<?xml version="1.0" standalone="yes" ?>
- <NewDataSet>
  - <classDetails>
    <IDProdClass>1033</IDProdClass>
    ....
  </classDetails>
  - <specDetails>
    <IDSpec>1001</IDSpec>
    <SpecName>Length</SpecName>
    ....
  </specDetails>
  - <specTableDetails>
    <IDTableVerDef>1106</IDTableVerDef>
    <IDTableSpec>1094</IDTableSpec>
    ....
  </specTableDetails>
  - <specGroupDetails>
    <IDSpecGroupDef>1066</IDSpecGroupDef>
    ....
  </specGroupDetails>
  - <specGroupSGADetails>
    <IDSpecGroupDef>1065</IDSpecGroupDef>
    <IDSubSpecGroupDef>1066</IDSubSpecGroupDef>
    <IDSpecGroup>1016</IDSpecGroup>
    ....
  </specGroupSGADetails>

```

Figure 4 – XML code segment of Product Class subscription

To facilitate this, there needs to be a user interface that allows a product supplier to view all the details of the available Product Classes and select the one they wish to subscribe to.

The application comprises a two way information retrieval system. First the product supplier has to subscribe to the Product Classes; this involves pulling and merging data from several different tables within the main PCD database into a single XML file for transportation via the Internet to the supplier side (Figure 4). The XML file is then parsed at the supplier side and split into chunks to be put back into the relevant tables of the PSCD. Once

the Product Class has been subscribed to, the product supplier will be included in a user search when looking for a product that falls under that class.

4. The Multiple Database Search Service System

It is necessary to support data access from external data sources that are autonomously managed by the organisations. Incorporation of data sources is required in the Grid environment so that the data retrieved from a large number of disparate database systems can be processed. Hence, data may come from external database systems which are not part of the Grid environment and the applications in the Grid environment may rely heavily on these external sources to satisfy the data needs of VOs in the Grid environment. In such a set-up where the Grid services depend on external data sources data access operations by the Grid services can be of significant importance when compared to other data operations such as data insert and update.

The MDSS System retrieves data from a large number of autonomously managed Supplier Database (SD) Systems belonging to individual organisations. The MDSS System is based on the Open Grid Services Architecture (OGSA) [4] model and provides a Grid service solution for processing a large amount of data. The System is an effort to fill the gaps identified when enabling large scale data access across a large number of autonomous databases providing data. It is based on the concept of information sharing. There are situations when commercial organisations share information with a large user base. Information shared by the organisations refers to products and services produced or supplied by them. Often organisations constantly seek new channels to reach wider users for their products. The MDSS System is designed to use the Grid Technology in a commercial environment for processing, analysing large scale information and sharing it in real time with those who are in need of such information. In particular, the MDSS System in its initial releases is being designed to aid the members of a consortium to search a large number of SD Systems for products to be used in construction projects. However there is a considerable scope to implement the MDSS System in other application domains as well.

4.1 Conceptual view of the MDSS System

The Grid enabled MDSS System is based on an architecture where the autonomous behaviour of

the database systems external to the Grid environment is sacrosanct. The architecture of the MDSS System is described in section 4.2. The autonomy of these database systems must be maintained, as the individual database systems are owned by organisations that may not want to provide database operations such as data definition, data insert and data update to the outside world but are willing to supply data through local operations on their database. By providing data access operations to external applications operating in a Grid environment, the data owners contribute to the processing of their data by the Grid application. This benefits both parties. The MDSS System provides the value added service of searching and processing large amount of data belonging to a large number of individual organisations prepared to supply products. The processed data refers to a saleable item and this information is made available to a large number of potential customers. On the other hand individual organisations will benefit from such Grid applications (for example the PSCD application and particularly its MDSS System component can help find new markets or increase their market share for the products and services they sell). It is this market opportunity that means suppliers conform to the product class descriptions. It also means that products have standard names agreed across the industry and there is no need for ontologies as products will not be recognised if suppliers use non standard names.

The MDSS System operates in a B2B set-up and accesses services and resources of participating organisations, and provides the middleman support to consortium for procurement of products and services. The MDSS System is being designed to hide all the complexities that come when creating a Grid infrastructure within itself. This eliminates the need for the supplier to implement a complex Grid infrastructure as the provision of a simple XML based Web Service interface hooks their system to the MDSS System.

Figure 5 shows the conceptual view of the MDSS System when it interacts with a number of external SD systems owned by individual organisations to retrieve appropriate data which is analysed, processed and finally dispatched to its requester. A requester can be a single user, a single organisation or a member within a VO. This is a situation where the suppliers are supporting local services in their database to provide information about their products in a standard format.

The MDSS System is designed and implemented to support database search operations for products required by a consortium where product information is supplied using Product Class templates. It provides modules to distribute a search across a number of machines in a Grid network and modules to aggregate the data retrieved from a large number of database systems. It is designed to operate in an environment where it has no control over the data at the individual sites in the distributed environment.

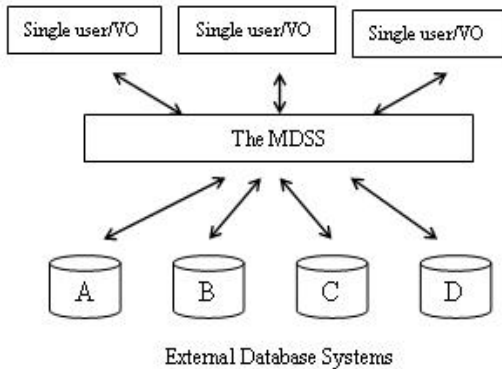


Figure 5 - The conceptual view of the MDSS System

Data resources are managed by data owners and the only access permission required is granted to the MDSS System. In the MDSS System only the data access mechanism is Grid enabled and not the participating data sources. In such a set-up the MDSS System acts as a central Grid service that can be used by a large user base to access information in the information provider's databases without the need to register with each information provider (SD System). The MDSS System identifies appropriate data sources to be searched in real time. This identification is based on the metadata and the search criteria submitted by the user.

4.2 Grid enabled MDSS System Achitecture

MDSS searches for products based on the criteria submitted by a contractor or a VO. It is anticipated that the VOs will use the MDSS System as part of the PSCD application for searching large numbers of supplier databases for the products required in the project. The prototype MDSS System is presently deployed on machines in the local area network and machines from the Welsh e-Science [7]. A database search job is divided into a number of equal parts which are allocated to these machines to search for product information in the SD Systems which are also set up in the

local area network. The architecture is shown in Figure 6. Searching for products requires the searching of all the supplier databases that have subscribed to the corresponding product classes. The database search is divided into two parts. In the first part the VO specifies the search criteria in the PSCD Application. This is then analysed to identify the suppliers that meet the requirements of the VO. The search criteria serve as the underpinning means for the dynamic creation of an XML document which contains the list of all the SD Systems that need to be searched in response to a user's request. For example a VO may be interested in searching for air conditioning units. Therefore only those SD Systems that have subscribed to the air conditioning unit product class need to be searched. Also at this stage the Grid resources that are available to deploy the search are identified. There can be more than one grid machine running the MDSS software in the local area or wide area network. Searching for product information from amongst a large number of SD Systems typically takes place using a cluster of Grid machines which work collaboratively and invoke supplier databases to retrieve product information in the form of XML documents. The Grid resources that are available for undertaking the search are dynamically identified and XML document is created on the fly containing a list of Grid Service Handles (GSH) which are permanent network pointers to Grid service instance [5] - in this case the MDSS instances.

In the second part, two XML documents: a list of suppliers in the form of supplier web service URL and a list of GSHs is submitted to the Master Grid Service (MGS) which divides the total work to be done into equal proportions and allocates each proportion to individual Grid machine running an MDSS instance. The collaboration between the Grid machines helps to perform database search jobs faster as the work load is equally distributed among them. A single MDSS instance in the Grid cluster can be required to search a number of SD Systems. Product information retrieved from the SD Systems by the MDSS instance is aggregated to form one XML document which is submitted to the Master Grid Service. The MGS also performs the service of collating results submitted by the individual MDSS instances to form a complete picture containing information on all the products. The resultant XML document is finally submitted to the user who propagated the search in the PSCD Application. The MGS accesses MDSS instances using the GSH of the instances. The GSH is resolved to

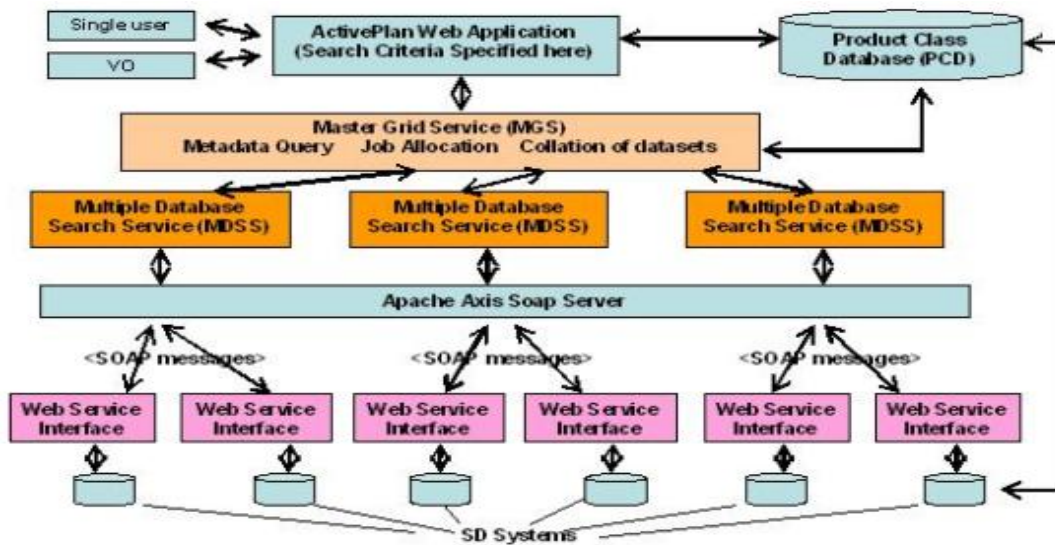


Figure 6 - The Grid Enabled MDSS System Architecture

the Grid Service Reference (GSR) by the handle resolution service that implements HandleResolver PortType [6]. The MGS creates a single proxy for all MDSS instances using its service description which is a Web Services Description Language (WSDL) document defining the bindings, messages, type definitions, etc required to invoke a service [6]. The proxy is created once and is used to gain access to any number of MDSS instances. Communication between the MGS, MDSSs and SD Systems takes place via XML based SOAP messages (see Figure 6).

MGS distributes search jobs to the Grid cluster running MDSS instances that perform the actual work of invoking SD Systems and requesting appropriate data. SOAP messages are created by the MDSS instances using an Apache Axis SOAP Server [8]. The SD Systems provide an XML based Web Service interface for the operations that can be performed by the MDSS [9]. Instances of MDSS are created by the MDSS Factory implementing the Factory PortType. They are registered with and receive a GSH from the handle resolution service [6]. The SD Systems subscribe to product classes created in the PCD System.

The MDSS System provides a data access mechanism only. It does not provide other database operations such as data definition, data insert, data update, data delete, etc as these are not needed for this application. There is scope to incorporate other database operations in MDSS as a part of future research. MDSS investigates the issues that need to be addressed when implementing OGSA based grid services to enable a search paradigm where data is

retrieved from a large number of autonomously managed databases. Since it implements this search paradigm, it has a different set of basic operations than those provided by the OGSA-DAI [10], and the Spitfire approaches [11], [12], [13] which have different objectives.

The MDSS System is designed to integrate with the PCD System in the PSCD Application. The PCD System enables the creation of product classes that are subscribed by the product suppliers in their SD Systems. These product classes can only be deployed in database schemas that can supply data structures which are identical to the schema of the PCD System. Therefore it is a requirement of the PCD System that all the resource provider database schemas should be able to provide identical data structures. A standard SD System is being designed to be identical in structure to the PCD System. Since the MDSS System collaborates with the PCD System to perform user tasks, it is optimised to search only those database systems which are registered with the PCD System and provide data conforming to the required schema. As part of the current research we are also investigating how the architecture of the MDSS System can be extended to support data search operations in the Grid environment in general supporting different database schemas.

5. Conclusion

A Grid based system for the procurement of products has been described. Although the system is currently aimed at the AEC industry it could be applied universally to any complex

project. The usage of the Grid provides a secure access facility, a vital feature if suppliers are to have confidence in the system, and provides a high speed, distributed search for products. The PSCD application includes a Security Management Service based on GSI. Two scenarios have been developed: first, users use their local proxy certificate, and second, users use a username/password pair in order to be authenticated and authorized to access the server resources. This mechanism is implemented over an HTTPS connection.

A recent extension of the PCD has been to incorporate a peer review system for the creation of Product Classes. Before a new Product Class version is released, it has to be reviewed and accepted by all members of the specification design team; this will involve dynamic tagging and editing of Product Class information within the PCD. The data management aspect of the PSCD application involves creation of Product Classes, subscription to these by the product suppliers, creation of products based on Product Classes, and Grid enabled search by a VO or individual contractors for the required products. The prototype PCD System and the MDSS are developed to support data management.

6. References

1. [Grid Security Infrastructure, 2004. <http://www-unix.globus.org/security/>]
2. Liviu Joita, Omer Rana, Pete Burnap, Jaspreet Singh Pahwa, Alex Gray, John Miles - "A Grid-Enabled Security Framework for Collaborative Virtual Organisations", 5th IFIP Working Conference on Virtual Enterprises (PRO-VE) - alongside the IFIP World Computer Congress, Toulouse, France, 22-27 August 2004 (to appear)
3. MyProxy Online Credential Repository, 2004. <http://grid.ncsa.uiuc.edu/myproxy/>
4. I. Foster, C. Kesselman, J. Nick, S. Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002, <http://www.globus.org/research/papers.html>
5. Open Grid Services Infrastructure (OGSI) Version 1.0 (draft) (2003). Global Grid Form Open Grid Services Infrastructure WG 5 April 2003. https://forge.gridforum.org/docman2/ViewProperties.php?group_id=43&document_content_id=176
6. Web Services Description Language (WSDL) Version 1.2, W3C Working Draft 3 March 2003, World Wide Web Consortium. <http://www.w3.org/TR/2003/WD-wsdl12-20030303>
7. Welsh e-Science Centre. <http://www.wesc.ac.uk/>
8. Apache Axis User's Guide (2004). <http://ws.apache.org/axis/java/user-guide.html>
9. Graham, S., Simeonov, S., Boubez, T., Davis, D., Daniels, G., Nakamura, Y. and Neyama, R. (2002). Building Web Services with Java: Making sense of XML, SOAP, WSDL, and UDDI. Sams Publishing 2002.
10. Antonioletti, M. and Jackson, M. (2004). OGSA-DAI Product Overview. <http://www.ogsadai.org.uk/docs/current/OGSA-DAI-USER-UG-PRODUCT-OVERVIEW.pdf>
11. Description of Spitfire (2003). <http://edg-wp2.web.cern.ch/edg-wp2/spitfire/index.html>
12. DataGrid WP2. (2003) <http://cern.ch/grid-data-management>
13. Bell, W., Bosio, D., Hoschek, W., Kunszt, P., McCance, G. and Silander, M. (2002). Project Spitfire - Towards Grid Web Service Databases. Technical report, Global Grid Forum Informational Document, GGF5, Edinburgh, Scotland, July 2002.