Supporting Collaborative Working of Construction Industry Consortia via the Grid

Pete Burnap, Liviu Joita, Jaspreet Singh Pahwa, Alex Gray, Omer Rana, John Miles
Welsh E-Science Centre, Cardiff University

ABSTRACT

In the AEC (Architecture / Engineering / Construction) 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.
This document is intended to describe the design for Grid-enabling of the Product Supplier Catalogue Database (PSCD) application. As part of the Grid-enabling process, specialised metadata will be developed to enable PSCD to more effectively utilise Grid middleware such as Globus and Java CoG toolkit. We also describe our experience in designing, developing and deploying the application using the Globus Security Interface (GSI).

1. INTRODUCTION

A typical AEC industry project involves many individuals and companies forming a consortium for the duration of a project. Such projects range in size from design and construction of a single building, to the creation of a large national infrastructure such as airports, dams, and highways. These projects are usually unique, very complex and involve many participants from a number of organizations acting collaboratively. The members are geographically dispersed. The consortia include design teams, product suppliers, contractors and inspection teams who must collaborate and conform to predefined scheduling constraints and standards. These participants also work concurrently, thus requiring real time collaboration between the geographically remote participants. A typical consortium member is often providing similar services to multiple projects simultaneously involving different partners. Web based communication technology is beginning to play an increasingly important role in supporting collaboration in AEC projects particularly to enable a project management (or team) to identify the current state of a project, its activities, and the constraints on these activities and their schedules. The planning, implementation and running of these projects is thus a complex task in which the Grid will be an important (vital) infrastructure in the future.

This industry has been examining how network technology can be used to improve the management of these projects. ActivePlan Solutions Ltd (APSL)[1] has produced two software packages to support the collaborative work of such consortia using web-based technology. The functionality of these packages is restricted by the current limitations of network technology such as bandwidth, support for collaborative working, etc.

The ActivePlan application is concerned with the interactive design of Building Environment facilities such as offices, shops, and hospitals. The functionality of the system includes spatial layout planning, looking at proximity relationships between spaces and determining the facilities and products that need to be acquired to create a given facility.

In the Product Supplier Catalogue Database (PSCD) application the database system Product Class Database (PCD) supports the creation of Product Classes which are then subscribed by the suppliers to show available products. The PSCDs at suppliers' end are used for the procurement of goods (e.g. doors, windows) for the construction projects. Both PCD and PSCDs are linked via the XML based Web Services technology [2]. Materials are sourced on a global basis from a large number of potential suppliers. Products are available from a wide number of suppliers, some of whom already have existing product databases so there are problems of linking to these and heterogeneity. Often the procurers have special relationships in terms of costs and supply with various manufacturers so the PCD has to signal to the suppliers the status of the organisation which is undertaking the search. The original database was found to be inadequate for the planned Grid enablement and it has been completely re-designed.

The development of the PCD and PSCD is concerned with making available to members of the project design consortium information about

[1] ActivePlan Solutions Ltd – http://www.activeplan.co.uk

products which can be acquired from external suppliers so that availability, delivery and cost can be taken into account in the planning. PCD is a database that stores information regarding product classes, product categories and product specifications and uses these to facilitate a search across a network. The search engine queries a dynamic selection of relevant supplier databases to extract, in real time, detailed information about the products which the user wishes to acquire, using the Grid infrastructure. Suppliers therefore will have to manage their own databases and also either have to provide an interface between these and the PCD or they can use a provided data structure which is PSCD compliant.

As part of the Grid-enabling process, specialised metadata will be developed to enable ActivePlan and the PSCD application to more effectively utilise Grid middleware such as Globus, with emphasis on the Globus Security Interface (GSI) and the MetaComputing Directory Service (MDS). The current development of OGSA-DAI will also be monitored to determine its role. The goal of OGSA-DAI is to provide uniform service interfaces for data access and integration via the Grid, providing interfaces that allow disparate, heterogeneous data sources and resources to be treated as a single logical resource [10].

This paper will describe the design and implementation of the PSCD database from the point of view of product versioning and product specification releases. It will also define a security framework implementation of the PSCD application using 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 Grid resources on a user's behalf. This facility is incorporated in the Globus and Java CoG toolkit.

The rest of the paper is organized as follows. In section 2, we discuss collaborative support considerations from the points of view of the construction industry, Web security aspects, user management and data management. Section 3 illustrates our design, deployment and integration of a Grid-enabled architecture for a PSCD web-based application. Conclusion with further work follows in Section 4.

2. COLLABORATIVE SUPPORT CONSIDERATIONS

2.1 Security considerations

2.1.1 Background of Web-Tier Security issues

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 can access the Web resources. The challenge to build a secure access to Web resources is to allow the integration of the existing product (PSCD) into a single user-friendly system using the capabilities of the GSI. Since the user is remotely accessing the web resources, it must be possible to establish the user's identity with certainty.

2.1.2 Overview of the Grid Security Infrastructure

The GSI is a client-certificate authentication system where all entities (users and resources) are identified by a globally unique name known as a *Distinguished Name (DN)*. Authentication with the GSI is a matter of proving that a user or resource is the entity identified by a DN. Resources then typically have a local configuration for mapping the DN to a local identity (e.g. a file containing DN and username pairs) [1].

The public key infrastructure relies on two keys which can be used in such a way that if one key is used to encrypt a message, the other key must be used to decrypt the message. One key is available publicly (namely *public key*). The other key is private (*private key*). A person can prove that holds the private key simply by encrypting a message. If the message can be decrypted using the public key, the person must have used the private key to encrypt the message.

The main issue in GSI authentication is the *certificate.*

A GSI certificate includes the following information [1]:

- A *subject* name, identifying the person.
- The public key belonging to the subject.

- The identity of a Certificate Authority (CA). The CA has signed the certificate and certifies that the public key and the identity both belong to the subject.
- The digital signature of the CA.

The *subject* name, which is known also as a *distinguished name (DN)* is the identifier of the person. For example, the subject looks like: "C=UK,O=eScience,OU=Cardiff,L=WeSC,CN= firstname lastname".

GSI certificates are encoded in the X.509 certificate format, according with the Internet Engineering Task Force (IETF) data format standard.

Authorization to resources is controlled by a mapping between the user's distinguished name and a local UNIX id via a Grid-map file. The GSI uses the Secure Sockets Layer (SSL - also known, according with the IETF standard, as Transport Layer Security, or TLS) for its mutual authentication protocol. The GSI enhances SSL by providing single sign-on capabilities for users, by generating a proxy certificate. This proxy certificate is a lifetime limited credential that acts on behalf of users and can be used to authenticate to web resources. The advantage of this technology is that a user is not requested to enter a password every time he wants to access a web resource, and he can use his proxy certificate for accessing resources.

2.2 User Management

In any distributed computing environment where many services and computer systems are openly accessible to anybody on the network, security is going to be of crucial importance. Users of an application have to initially log into the system but once they are inside and have access to the services offered within the network there must be restrictions as to who can access which services. This can be done using an authentication credential – a token attached to a user's account upon login which then acts as a virtual 'security pass' with restrictions to certain services as you would expect to have with a security pass to a restricted building. The privileges associated with each token will vary depending on the role assigned to the user in the user login database. For example a system administrator will have access to more important and critical services than a generic user. When considering the implications of consortia within a system it becomes clear that a user may take on several different roles in various consortia – they may be an application developer in one, a project manager in another and a generic client in another, all of which come with different responsibilities and authorities.

The first user management consideration when designing a web application is how to manage the front-end of the web application. This includes logging a user into the system, determining their role in the system based on database information and presenting the user with the appropriate interface for their role. For example, an industry standards product specification designer will be presented with an update and editing interface that has access with update privileges to the central information database; while a generic consortia client will be presented with a search and discover interface that allows only search queries to be executed and returned and products to be procured from suppliers.

It is at the point of procurement that another user management issue arises. In any industry, clients build up a relationship with suppliers over time and ascertain a discount policy in which they get discount based on frequent purchases and bulk orders. This logic must be invoked into an on-line procurement system. Take an example of person P being a single user and also in two consortia – A and B. When dealing with suppliers, person P may get a 2% discount on orders with supplier X, but when they deal with supplier X as a member of consortia A, they get a 5% discount. As a member of consortia B, they may get a 10% discount. So when person P is using the system, suppliers will need to be informed as to which consortia P is in when using the system this time (if any) so that they can use stored logic to work out any discount. A question now has to be answered, where is this information stored? This will be addressed in the subsequent sections.

2.3 Data Management

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 products required.

3. The PSCD

3.1 Security system architecture

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. As was shown in section 2.1.2, the GSI provides the mechanism and security for user to delegate their credentials to the Web resources. This can be done using Globus service or Java CoG toolkit.

Figure 1 shows the system architecture which was designed, deployed and integrated for the applications. The Tomcat server used to handle connections to the applications can support both HTTP and HTTPS connections. In this development phase the communication between client and web server uses only the HTTP connection.

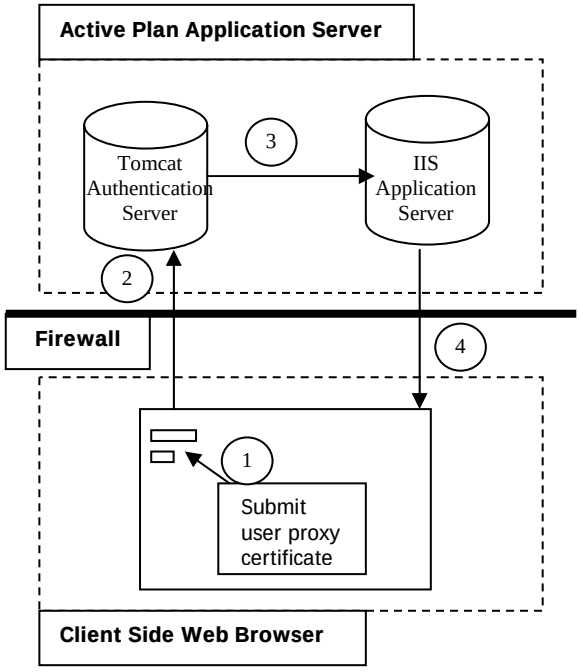

Figure 1 – ActivePlan Secure Login Authentication Architecture

1. First, a user has to have a valid proxy certificate. The user submits their proxy certificate to the Tomcat authentication server (AS) via a web interface using JSP - servlet interface
2. Tomcat AS authenticates the certificate and obtains the local user name for the PSCD application from the Grid-map file
3. Tomcat AS passes the local user name to the IIS server that runs the PSCD system. IIS then matches the user name to its local DB and creates a session for that user
4. User preferences are applied to the 'index' page of the PSCD system and the user is presented with the home page of the application

3.2 User management

In the PSCD application, once the user has been authenticated with the tomcat authentication server using their Globus security certificate and the GSI interface, the user's local user name is extracted from the distinguished name associated with their certificate and forwarded to the Internet Information Services (IIS) Server that runs the web application. This user name is then mapped to a user name in a 'Users' table of the PSCD central database. The user name in the users table has an associated 'Role' column which contains the role of that particular user; once this has been found it can then be used to determine which interface to display at the web application front-end.

A session is created for the user entering the system and information is stored such as user ID, role, interface preferences etc. This session has an inactivity timeout limit of 30 minutes, this means that if the user does not interact with the system for this amount of time, the system logs them out and next time there is a click on a page there will be a reference back to the original login page.

The structure of the web application front end is that of a frameset of four frames – top, bottom, left and right. The top frame displays information about the user currently logged in, namely user's name, company and role within the system as well as a link to log out of the system (see Figure 2).

Figure 2 – Logged user information

The bottom frame contains basic information about the site. The right frame takes up 75% of the width of the page and is the 'main' frame where all the pages of the application are loaded. This is empty to begin with. The left frame occupies the remaining 25% of the width and contains a tree-style navigation menu (see Figure 3). There are several different versions of the menu written into the system, each of which contains links available to the user depending on their role in the system; examples of which include: Project Manager, System Administrator,

Surveyor and Client trees. This is a simple but effective way of only allowing relevant parts of the system to each type of user. The appropriate menu is loaded into the left side frame by performing a case test against the user role session variable.

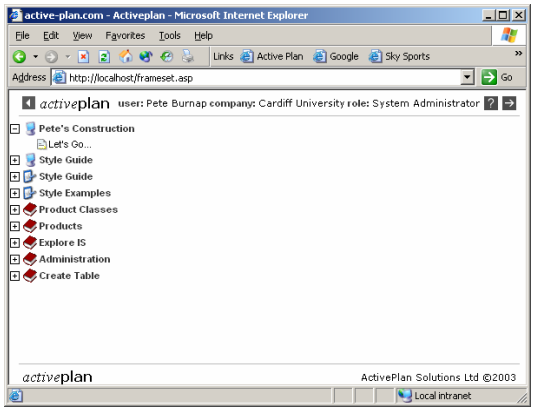

Figure 3 – User navigation menu

The issue of creating system logic for allowing users to mimic real industry relations and obtain discounts etc has not yet been implemented into the PSCD system. However, research into the methodology of this implementation has been undertaken and an initial architecture to illustrate the proposed 'virtual organisation' interaction has been thought out. Figure 4 illustrates ways in which a user may be logged into the system, namely as a virtual organisation consisting of a single person user or as a member of single or possibly multiple consortia - consortia could also feasibly be created within other consortia in an unlimited nested hierarchy

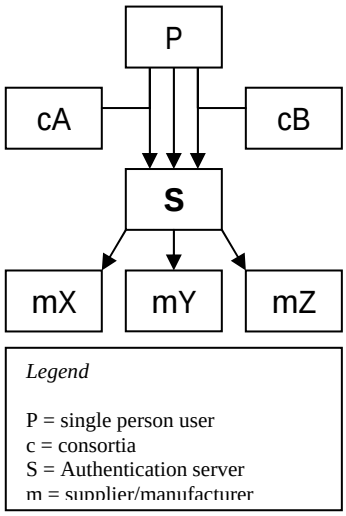

Figure4 – Certificate Authentication Architecture

Person P in figure 4 could log into the server on their own or be a part of consortia A or B – they cannot enter the system as a combination of these options. The user must log in as a single user *or* a member of a single consortium. Changing virtual organisations would involve logging out and back in again.

3.3 Data Management

The PSCD will support the creation of the products by the suppliers who subscribe to relevant Product Classes. Product Classes are discussed in Section 3.3.1. The PSCDs could be searched using the XML based Web Services technology [2]. For searching PSCDs, a search criteria is required to be built specifying all or some of the specifications that a product should have and the range within which the product specification values should lie. For example, whilst procuring fan coil units for a particular construction project the consortia could be interested in specifications such as its air flow rate, cooling capacity, dimension, weight, electrical supply, maintenance cost, warranty, bulk purchase price etc. Therefore a search must specify the range within which specification values should lie.

The PSCD system architecture is based on the service oriented Open Grid Services Architecture (OGSA) model [3]. The operations within the PSCD application are defined as Grid services (portTypes) having Web Services Description Language (WSDL) interfaces [6]. These grid services can be invoked and bound to other grid services or clients like any other web service using the Simple Object Address (SOAP) protocol [4].

3.3.1 The Product Class

A mechanism is required to create standard product definitions to be used by the product suppliers when storing products in the PSCD. We call these product definitions product classes[2]. With time the product suppliers tend to evolve their products by adding new features into them. The product classes support new and evolving products by providing versioning support. A product class can be used by a large number of product suppliers to create and list their products in the PSCD.

The creation of product classes is the first fundamental task required before creating

[2] The terms "Product Class" and "Product Class Definition" are used interchangeably in this paper. However, both the terms mean one and same thing.

products in the PSCD. A Product Class can be defined as an entity made up of a number of specifications. The specifications can be of several types and are created as part of creating a Product Class. The specifications correspond to the pre-defined specification types and provide the mechanism for defining the properties a product can have. So far five different types of specification types have been identified. Some of the specification types can be further decomposed into a number of sub specification types as shown in Figure 5. Defining a product class requires defining its specifications and constraints. For example, a window product class can have specifications such as width, height, wood type, panel shape, glazing configuration etc.

As a part of the PSCD application, the development of a prototype database centric tool called the Product Class Database (PCD) occurred. It enables its users (the independent industry specification designers) to create the Product Classes. PCD enables specification designers to create new product class definitions or new versions of existing product class definitions. It is expected that industry standards would be developed and standard criteria established for product classes and their evolution.

3.3.2 Versioning of Product Classes

New products or a new range of existing products are introduced by suppliers on a regular basis as they enhance features and functionality. These changes to products cannot be defined within the scope of the existing product class definitions which cannot support these extra features. The product classes need to evolve with the evolution of the products. For this versioning of product classes is required so that new versions of existing product class definitions can be created. The new version of product class definitions allows the product suppliers to create new products with enhanced features.

Versioning of product class definitions requires versioning of its specifications as product class definitions are made up of a number of different types of specifications. However, only those specifications that support the extra features need versioning. This allows reuse of existing specifications and new versions are only required when creating new product classes. This facilitates rapid creation of new product classes. The prototype Product Class Database is still undergoing development and full discussion of product classes and specification types is beyond the scope of this paper.

A product class is a template for suppliers to create products. It is envisaged that suppliers will subscribe to product classes that correspond to the products they supply. Product Classes will be used to create products in the PSCD by supplying values for the specifications defined for the product class. In this way products can be rapidly created once their product classes are available.

3.3.3 Grid Enabled Multiple Database Search Architecture

Members of the consortium are part of a Virtual Organisation (VO) interested in the procurement of supplies for the construction project being undertaken. The service oriented Grid enabled Multiple Database Search Service (MDSS) based on the OGSA model is also utilised to investigate how a large number of PSCDs can be searched for the desired products by a VO. The project aims to develop the MDSS to search for desired products by building search criteria. The MDSS architecture is shown in Figure 6. Searching for products requires searching all PSCDs of suppliers that have subscribed to the product class definitions corresponding to the products being searched.

The database search is divided into two parts. In the first part the VOs gain access to the MDSS instance using the Grid Service Handle (GSH) which is a permanent network pointer to a particular Grid service instance [5]. The GSH is resolved to the Grid Service Reference (GSR) using the handle resolution service that implements HandleResolver PortType [6].

In the second part, the grid service instance (MDSS) is invoked by the VO to make a search across PSCDs. For this the client creates a proxy for the MDSS instance using its service description which is a Web Services Description Language (WSDL) document defining the bindings, messages, type definitions, etc required to invoke the service [6]. The MDSS instance first identifies the PSCDs that need to be searched based on the search criteria submitted. For example a VO may be interested in searching for air conditioning units only. Therefore only PSCDs subscribing to the air conditioning unit product class need to be searched. For this, a list of required PSCDs is retrieved from the Product Class Database and then the XML based SOAP message is sent to each PSCD requesting appropriate data. The Product Class Database

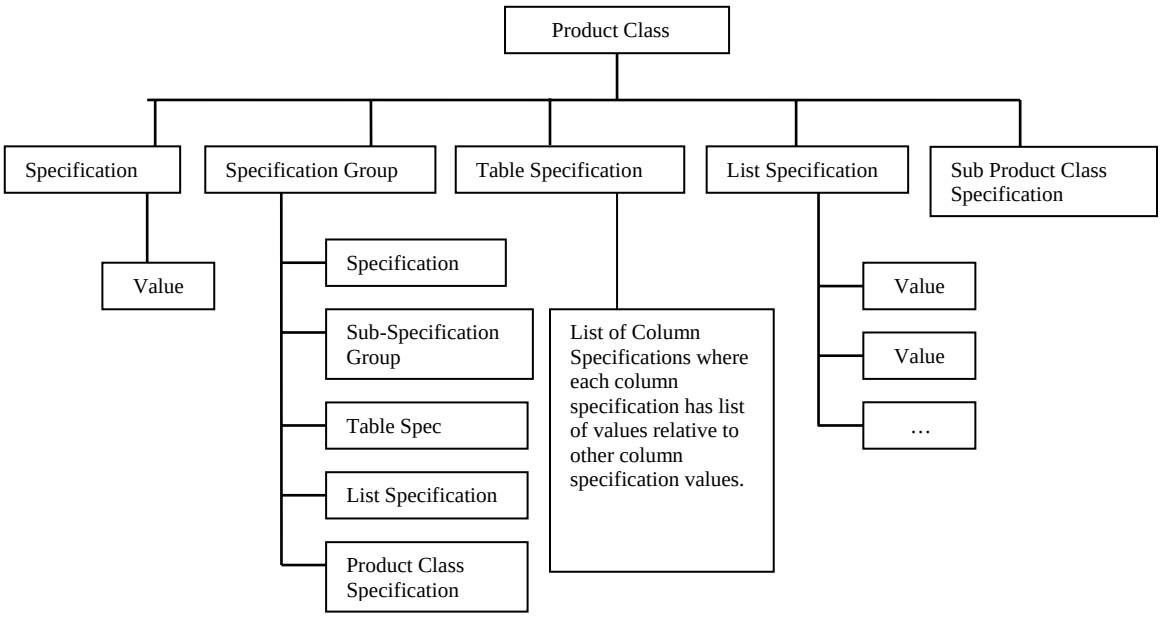

Figure 5 -The Product Class and its various specification types

also holds information about the product classes used by particular product suppliers. The PSCDs at the suppliers' end provide the XML based Web Services interface for the operations that can be performed on the remote data source [2].

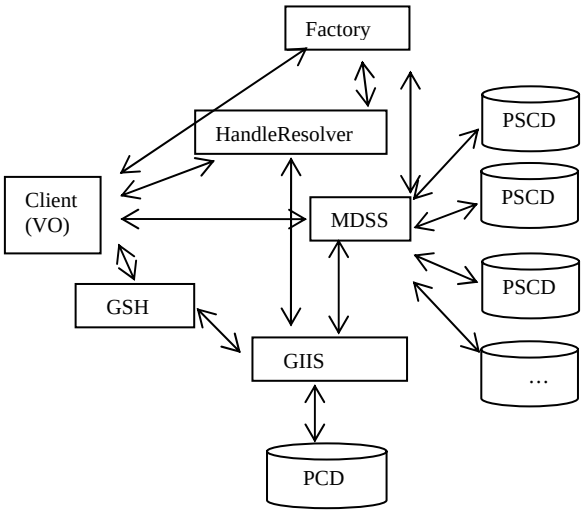

Figure 6 - Grid enabled Multiple Database Search Architecture

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 PSCDs are registered with PCD which is then registered with the Grid Index Information Server (GIIS) (Figure 6) using the Grid Resource Registration Protocol (GRRP) [7] - a soft state registration protocol for registering resources with the GIIS. The Grid enabled Multiple Database Search part of the PSCD system, is still under development and significant changes to its architecture are likely to occur in the future.

3.3.4 Data Access and Integration and the MDSS

There is a widespread awareness within the research community of the need to bring database management systems within the fold of the grid infrastructure [8, 9]. We intend to follow closely and take advantage of the research activities being undertaken in this area. Based on this research we are also currently investigating the issues that need to be addressed for making all autonomously managed PSCDs one virtual database resource and implementing a search using distributed query processing.

The prototype PSCD is being developed to run on a chosen database management system initially. However there is considerable scope for its implementation across heterogeneous data resources. In this respect it is envisaged that work being undertaken as a part of The Open Grid Services Architecture - Data Access and Integration (OGSA-DAI) project will be of potential benefit to the PSCD System [10]. We aim to investigate the applicability of the data federation and distributed query processing capabilities of OGSA-DAI to implement MDSS within a VO.

4. CONCLUSION AND FURTHER WORK

PSCD includes a login interface using the security infrastructure based on GSI mechanism provided by the Grid middleware and Java CoG toolkit. This mechanism is implemented over a HTTP connection. The secure connection (HTTPS) will be implemented in future.

The user management system needs to incorporate logic to calculate discount from manufacturers based on the identity of a user acting as a single user or as a member of a consortia. The storage location for this metadata must also be determined; this is an area that requires careful consideration. Will it reside on the central database or be stored by each supplier in the PSCD?

The PSCD system aims to enable the creation of Product Classes and subsequent subscription by product suppliers who use them for the creation of products. This involves the development of the PCD and PSCD database systems and the Grid enabled MDSS based on OGSA. All these components will be integrated as part of the bigger picture. Prototypes for some parts of the system have been developed and are currently undergoing testing (for example PCD) while others are at the design phase (for example MDSS and PSCD). We aim to build prototypes for the rest of the system in the coming year.

5. REFERENCES

[1] www.globus.org/security

[2] Graham, S., Simeonov, S., Boubez, T., Davis, D., Daniels, G., Nakamura, Y., Neyama, R., *Building Web Services with Java: Making sense of XML, SOAP, WSDL, and UDD.* Sams Publishing 2002.

[3] 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

[4] *Simple Object Access Protocol (SOAP) Version 1.2*, W3C Recommendation 24 June 2003, World Wide Web Consortium. http://www.w3.org/TR/2003/REC-soap12-part1-20030624/#intro

[5] *Open Grid Services Infrastructure (OGSI) Version 1.0 (draft)*, 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] I. Foster, C. Kesselman, S. Tuecke. *The Anatomy of the Grid: Enabling Scalable Virtual Organizations.* International J. Supercomputer Applications, *15(3), 2001*. http://www.globus.org/research/papers.html

[8] Paul Watson, *Databases and the Grid* Version 3.1. UK e-Science Programme Technical Report Series Number UKeS-2002-01, National e-Science Centre. http://www.cs.man.ac.uk/grid-db/documents.html

[9] Norman Paton, Malcolm Atkinson, Vijay Dialani, Dave Pearson, Tony Storey and Paul Watson, *Database Access and Integration Services on the Grid*. UK e-Science Programme Technical Report Series Number UKeS-2002-03, National e-Science Centre,1 February 2002. http://www.cs.man.ac.uk/grid-db/documents.html

[10] *Open Grid Services Architecture – Data Access and Integration (OGSA-DA)*, OGSA-DAI Product Overview, UK e-Science Project (http://www.nesc.ac.uk/projects/). Available at http://www.ogsadai.org/docs/docs.php