Grid technologies to support B2B collaboration

Nicola Capuano*, Angelo Gaeta
and Francesco Orciuoli

CRMPA – Centre of Research in Pure
and Applied Mathematics, Italy
and
DIIMA – Department of Information Engineering
and Applied Mathematics, Italy
E-mail: ncapuano@unisa.it
E-mail: agaeta@crmpa.unisa.it
E-mail: orciuoli@diima.unisa.it
*Corresponding author

Matteo Gaeta, Sergio Miranda
and Pierluigi Ritrovato

DIIMA – Department of Information Engineering
and Applied Mathematics, Italy
E-mail: gaeta@diima.unisa.it
E-mail: smiranda@diima.unisa.it
E-mail: ritrovato@diima.unisa.it

Abstract: In the context of the European Commission Project BEinGRID (FP6), the authors have defined a set of design patterns to develop software components based on service-oriented grid technologies. Some of these patterns have been used to improve the software components of a service-oriented grid middleware named GRid-based Application Service Provision (GRASP) that the authors have defined, designed and implemented in the frame of a former homonymous European Commission Project (FP5). The main improvement of GRASP due to the application of the BEinGRID design patterns is the support for the creation and life cycle management of Virtual Organisations (VOs).

This paper presents the authors' experience and lessons learnt in adopting the GRASP middleware to set up a Business-to-Business (B2B) federated environment supporting collaboration among enterprises. The concrete case study relates to online gaming applications and the adoption of the software as a service business model to provide gaming applications.

In addition, a set of lessons learnt during the analysis of several Business Experiments (BEs) of the BEinGRID project are reported.

Keywords: service-oriented architecture; SOA; grid; virtual organisation; business-to-business; B2B.

Reference to this paper should be made as follows: Capuano, N., Gaeta, A., Orciuoli, F., Gaeta, M., Miranda, S. and Ritrovato, P. (2009) 'Grid technologies to support B2B collaboration', *Int. J. Knowledge and Learning*, Vol. 5, No. 2, pp.107–121.

Copyright © 2009 Inderscience Enterprises Ltd.

Biographical notes: Nicola Capuano is a Research Assistant at the Department of Information Engineering and Applied Mathematics of the University of Salerno, Italy. His main research interest is artificial intelligence and, among its applications, intelligent tutoring systems and knowledge representation. He collaborates as a Project Manager with the Research Centre in Pure and Applied Mathematics (CRMPA) of Salerno and the Research Centre on Software Technology of the University of Sannio (Benevento) and as an R&D Consultant with MoMA Software and Research s.r.l.

Angelo Gaeta graduated with a Laurea degree (cum laude) in Electronic Engineering from the University of Salerno, Italy, in January 2002 with a thesis concerning the distributed simulation and next-generation architectures to support wide-scale distributed simulation. His current interests are related to investigations in grid and SOA for e-business and e-science. He is the co-author of several scientific papers related to web and grid technologies. He has been involved in research activities in several EU co-funded projects. Currently, he collaborates with the CRMPA and is mainly involved in the VI FP IP project BEinGRID, where he leads the activities of the Work Package (WP) related to VO management.

Francesco Orciuoli obtained his BS (cum laude) in Computer Science from the University of Salerno, Italy, in 1999. In the last seven years, he has been working for the CRMPA on the definition of innovative e-learning solutions and, in particular, on the utilisation of knowledge representation techniques for the provisioning of personalised and context-aware learning experiences deployed upon the web infrastructure. Furthermore, he has been involved in research activities in several EU co-funded projects (*e.g.*, InTraSys, m-Learning, Diogene, ELeGI, *etc.*) as a CRMPA Collaborator. He is the co-author of several scientific papers related to e-learning, web and grid technologies. At present, he is a collaborator (in the Computer Science area) of DIIMA and he is focusing his research activities on the design of software architectures for modular, extensible, scalable, interoperable, open, multichannel, service-oriented and standard-based e-learning platforms.

Matteo Gaeta obtained his degree in Information Sciences from the University of Salerno, Italy, in 1998. At present, he is an Associate Professor of Systems of Information Processing within the Department of Information Engineering and Applied Mathematics (DIIMA) at the Faculty of Engineering of the University of Salerno. He is a member of the Working Group of Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR), acting as an expert for the evaluation and control of projects under Art. 14 and Coordinator of the Assessment Panel of projects submitted to Strategic Programme No. 10. He is the Project Supervisor and Coordinator of numerous R&S and technology transfer projects of the CRMPA. His scientific research activity is focused on various fields: systems, platforms software, distributed architecture and computing grids; artificial intelligence and learning methodologies and technologies; artificial vision and image analysis.

Sergio Miranda obtained his Laurea degree (cum laude) in Computer Science from the University of Salerno, Italy, in 1996. Since 1995, has been collaborating with the CRMPA at the University of Salerno. He is a Graduated Technician for the University of Salerno (where he is employed in research activities) and he is a Tutor for Computer Programming courses; he is also the Production Director for Modelli Matematici ed Applicazioni (MOMA), a commercial spinoff of the CRMPA. His research activities deal with e-learning systems, databases, web-based applications, knowledge management, neural networks and fuzzy logic.

Pierluigi Ritrovato obtained his BS in Computer Science from the University of Salerno, Italy, in 1992. At present, he is an Assistant Professor of Computer Science at the Faculty of Engineering, the University of Salerno, and responsible for the international research cooperation and technology innovations for the CRMPA. In the last five years, he has been focusing his scientific research on grid technologies and their use for business and, in particular, learning, taking into account the aspects related to the operational management of virtual organisations, services orchestration and security as well as on Distributed Software Engineering. He is the Chair of the Service Orchestration, SLA usage and Workflow – Technical Group for EU FP6 Grid projects. He is the Editor of the book *Towards the Learning Grid: Advances in Human Learning Services*.

1 Introduction

We present in this paper the improvement done to the GRid-based Application Service Provision (GRASP) middleware[1] in order to allow the creation and management of a secure federated environment. The GRASP middleware is the main result of the homonymous FP5 EU project and represents a novel architecture supporting grid-enabled collaboration for the purposes of Application Service Provision (ASP).

Leveraging the convergence of grid and web services technologies (Foster *et al*., 2002a–b), during the GRASP project, the authors anticipated the emergence of new business and scientific computing paradigms that are based on dynamic Virtual Organisations (VOs) (Foster *et al*., 2001). These VOs span across organisational boundaries and enable the enactment of collaborative processes that integrate services, resources and knowledge in order to perform the tasks that the VO partners could not undertake on their own.

In the context of the FP6 EU IP BEinGRID,[2] the authors have analysed a set of concrete case studies, namely Business Experiments (BEs), relating to the adoption of grid technologies in business domains. During this analysis, some concrete requirements have been elicited and design patterns[3] to develop software components[4] based on service-oriented grid technologies have been produced.

In the definition of design patterns, the authors have taken into consideration past experiences such as Gamma *et al*. (1995), Easton *et al*. (2006), Rana and Walker (2002), Fowler (2003) and Alur *et al*. (2001). Relying on these patterns, the GRASP middleware has been improved. This paper focuses mainly on the components and their preliminary evaluation, as well as on the lessons learnt during the analysis of BEs. For more information on the design patterns, interested readers can refer to Gaeta *et al*. (2007).

The rest of the paper is organised as follows. Section 2 briefly describes the GRASP middleware and its key and distinctive features with respect to other middleware. Section 3 presents the improvement of the GRASP middleware due to the application of the BEinGRID design patterns. Section 4 presents the concrete case study and our preliminary results. Section 5 presents the lessons learnt during the analysis of the BEinGRID BEs, that are concrete cases of study of adoption of grid technologies for business sectors. Section 6 draws conclusions and presents the future work.

2 The GRASP middleware

The GRASP project[1] has experimented with the use of grid computing in order to support the operation and evaluate the sustainability of new ASP models and contribute to the evolution from traditional ASP to new paradigms.

Basically, two new ASP models have been investigated in GRASP:

1. the 'federated' ASP model, which can be described as the collaboration of many grid service providers that provide services that can be combined into complex services addressing a customer need that each of them could not achieve themselves
2. the 'many-to-many' model, which is essentially an evolution of the classic 'one-to-many' ASP model, achieved by evolving its foundation from the client server to the service-oriented paradigm: the entity can take the role of either a consumer or a service provider in the context of the same application, depending on the required interactions.

GRASP has explored the use of web services as a means of providing a timely and effective technological basis supporting the evolution of the ASP market towards a sustainable utility computing model.

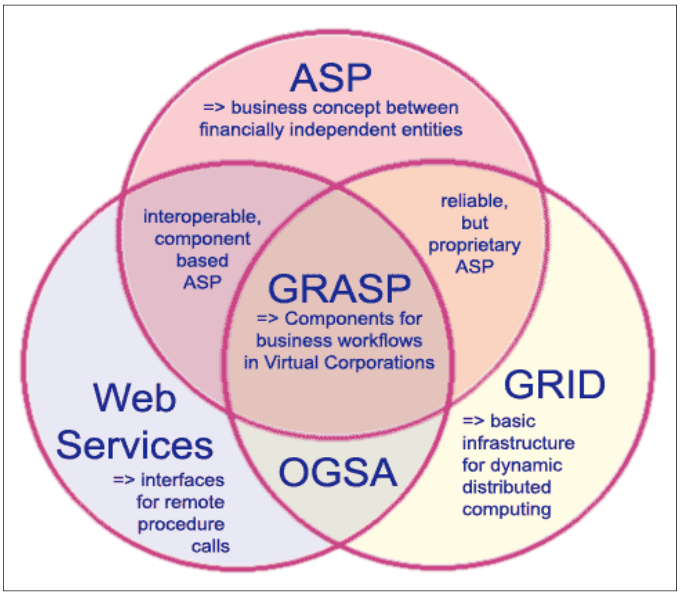
The main results achieved in GRASP are an architectural framework for GRASP, a prototype realisation of this framework in a GRASP platform and 'proof-of-concept' implementations of 'federated' and 'many-to-many' ASP models in multiple domains.

In order to support the ASP models, the GRASP middleware presents the following key services:

- *Orchestration* – One of the most important aspects of the new grid-based ASP models is that no longer a single vendor does not control the whole process. This means that a mechanism is needed that orchestrates the services offered by different vendors and ensures a controlled collaboration. The GRASP orchestration service is based on Business Process Execution Language for Web Services (BPEL4WS) and provides the possibility for a hybrid orchestration for grid services, but also for web services.
- *Service Level Agreement (SLA) monitoring* – The orchestrator can only fulfil its task of controlling the collaboration between the different services if enough information for the decision process is available. The SLA monitoring services monitor, enforce and provide notifications in order to assist the orchestrator in this task.
- *Accounting and billing* – Without accounting and billing, no ASP can be performed. As the services are no longer controlled by one single entity but by many different service providers over time, new ways to collect provided services must be introduced. Especially for the 'many-to-many' model, new solutions must be identified.

Figure 1 presents a graphical view of the three key pillars of GRASP.

Figure 1 The GRASP pillars (see online version for colours)



3 Evolution of GRASP towards supporting B2B collaboration

The GRASP middleware provides a solid foundation for service-to-service interactions, but it lacks the capabilities to support VO formation and life cycle management. In the context of the BEinGRID project, two components have been designed and developed to fulfil this lack.

The following two sections present an overview of these components, namely the VO set-up and application virtualisation.

3.1 VO set-up

The VO set-up is a web service providing functionalities to support the VO life cycle, in particular, the VO identification and formation phases, where members of the VO have to be identified and a circle of trust among them has to be created.

The component allows for the management of VO-related registries and secure federation life cycle. In terms of functionalities, the main ones are the following:

- secure federation life cycle management in a B2B environment
- members, policies and roles management
- selection of VO members on the basis of the capabilities they can offer
- management of VO registries (VO member registry, VO service instance registry).

The design and development of this component has been motivated by the business scenarios analysed in the BEinGRID project where there is the need for a solution allowing to publish/discover enterprises and capabilities and to create a secure federation among autonomous administrative domains.

The VO set-up component presents several benefits and distinctive features:

- adoption of standards; the component is based on enterprises' widely adopted standards such as Web Services Interoperability (WS-I), Universal Description, Discovery and Integration (UDDI), Security Assertion Markup Language (SAML)
- extensibility; it is very easy to add new criteria for partner identification and selection
- ability to associate trust relationships with a business context
- agility in responding to new needs/requirements and improved time-to-market (by the set-up of a VO when a new opportunities arises)
- improved trust in B2B interactions
- dealing with the geographical and organisational distribution of teams.

The high-level architecture of this component is presented below:

- *VO set-up* – this is in charge of finalising the VO creation process. It is mainly a façade interacting with the other components.
- *Registry* – this contains the new members and service instances of the VO.
- *Federation* – this is in charge of creating a federation and managing the identify of the federation members. This component can be designed according to the Secure Federation Design Patterns and other patterns proposed in the security area of the BEinGRID project.[2]

The component is useful in several concrete scenarios where there is a low level of dynamicity in the VO creation process (meaning that the other steps of the creation process, such as agreement negotiations and policy definition, may be performed offline).

In contrast to the current state-of-the-art that is mainly based on the results coming from the efforts of the eScience community (*e.g.*, Virtual Organisation Membership Service (VOMS)),[5] the proposed solutions take into consideration the needs and requirements of the business world. This aspect has a deep impact on the design and implementation of the proposed software components.

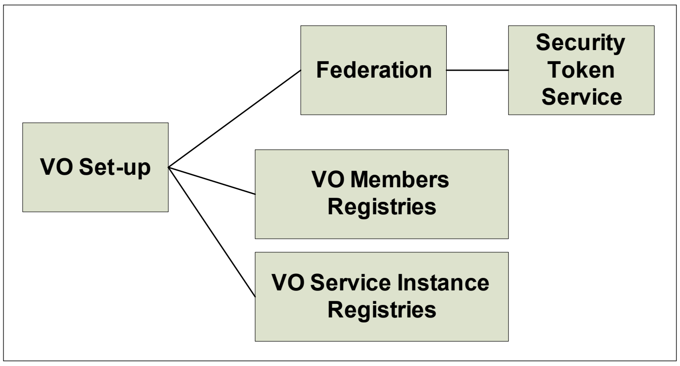
While in fact, most of the eScience solutions propose and implement coarse-grained models to address issues related to the membership and management of resources in a VO (for instance, allowing access to whole resources for job submission), in our case, we have requirements that foresee a fine-grained approach (for instance, allowing access to specific capabilities offered by a service provider).

The main benefits relate to the federation part of the component. These allow, on the one hand, managing the life cycle of circles of trust between providers and, therefore, the life cycle management of federations of trust realms and, on the other hand, managing the life cycle of the identities and privileges of users and resources within such federations of trust realms.

The obvious benefits include:

- facilitating the creation of communities of identity providers that enable identity brokerage and management by supporting open standards such as Liberty Alliance, SAML and WS-Federation and, therefore, giving rise to new means of revenue generation
- enabling the customer to choose the identity provider that is more appropriate for a specific collaboration instead of being locked into what is incorporated in their Service-Oriented Architecture (SOA) platform by some middleware vendor or departing from expensive product integration projects that give them identity provision and federation (at a very high cost) for the specific application at hand.

Figure 2 The VO set-up (see online version for colours)



3.2 Application virtualisation

This component provides a way to integrate and expose application capabilities through a single access point that is configured to manage their execution and forward requests to the application capabilities.

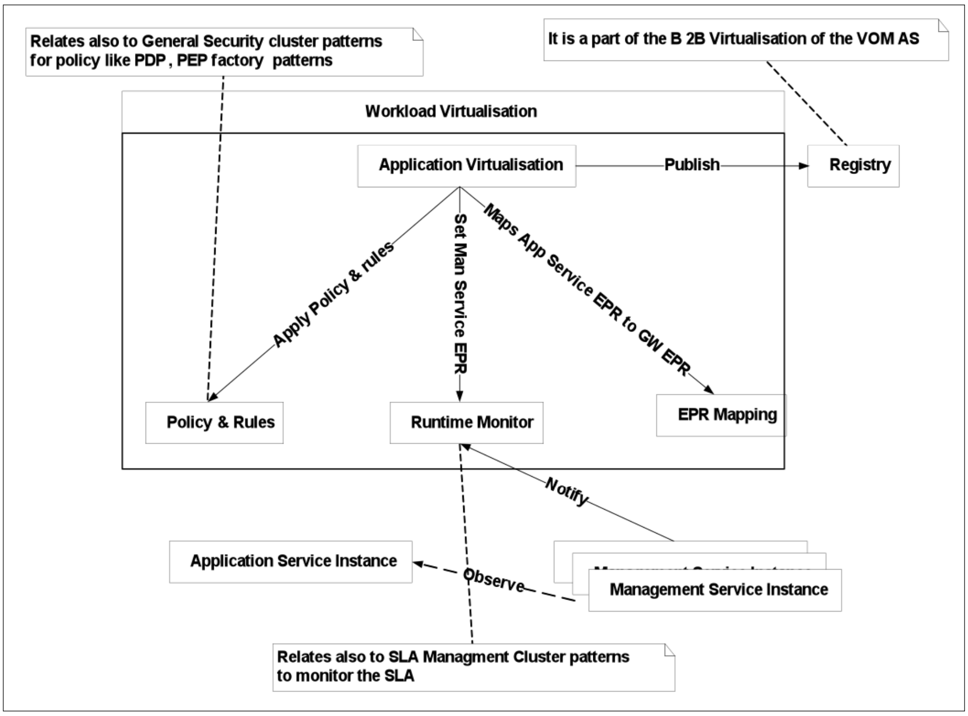
In a VO, in fact, there can be the need to expose application capabilities, for direct usage or for composition, as network-hosted services in order to avoid the direct and unmanaged access of VO members to VO resources.

It is appropriate to use this component when there is the need to:

- decouple service access logic from the rest of the application
- hide the complexities of accessing a service from the application
- have a single point providing common management
- avoid direct access to resources.

Figure 3 shows a high-level diagram of the component. In the figure, there are notes evidencing that some components relate to other patterns produced in the BEinGRID project. This means that, for instance, the policy and rules component can be designed according to the patterns proposed by the security area.

Figure 3 Application virtualisation



The application virtualisation component follows the Façade pattern by Gamma *et al.* (1995) and it is in charge of invoking the other classes of the system in order to execute the virtualisation process.

The application virtualisation, runtime monitor and management service can iterate (in some way) the Observer pattern. Management service instances notify the runtime monitor with the updates of some parameters and the runtime monitor can notify violation to the application virtualisation.

If the application virtualisation component is also the gateway when a request to access a service arrives, the application virtualisation can operate according to the Chain of Responsibility (Gamma *et al.*, 1995) and pass the request along a chain of handlers. The building blocks are described below:

- *Application virtualisation* – this implements the virtualisation process' steps. It delegates requests to the appropriate subsystem objects. It returns to the client a reference to access the created application instance through the gateway. It can be configured to be the gateway and when a request to access a service arrives, it can pass the request along a chain of handlers.
- *End Point Reference (EPR) mapping* – this is in charge of mapping the EPR of the created application service instances to the EPR of the gateway. It is also in charge of activating and deactivating the application service instances if, respectively, the process creation and termination of service instances succeed.

- *Policy and rules* – this is in charge of applying the policies and rules associated to the application service instance. It can work according to the policy patterns identified by the General Security cluster of the BEinGRID project.[2]
- *Runtime monitor* – this is in charge of collecting the management information of the application service instances. It evaluates the execution of the application service instances again the parameters of the contracts associated to it. It notifies eventual violations. It can work according to the patterns to monitor and evaluate the SLA identified by the SLA management cluster of the BEinGRID project.[2]
- *Registry* – this is a registry of the created application instances.
- *Management service* – this notifies the runtime monitor of the change of the status of some parameters of the application service instance.

The obvious benefits include cost reduction, operational management risk mitigation through outsourcing and the reduction of time-to-market timescales.

In addition, application virtualisation offers to the customer (*i.e.*, the application service provider) the ability to select among competing offerings of infrastructure services such as identity, access and policy management service providers. Such a choice offers to the customer, on the one hand, the potential of avoiding to be locked into investing in proprietary SOA solutions that fit one market sector but are not good for another sector and, on the other hand, the prospect of getting better value for money by increasing the competition between vendors offering enabling enterprise solutions as a service in an Independent Software Vendor (ISV) fashion though 'pay-per-use' or 'pay-as-you-grow' models.

4 Case study: online gaming application

Within the context of the BEinGRID project, the evolution of the GRASP middleware has been adopted in the pilot focusing on the design and development of a Virtual Hosting Environment (VHE) for online gaming, namely the BE09.[6]

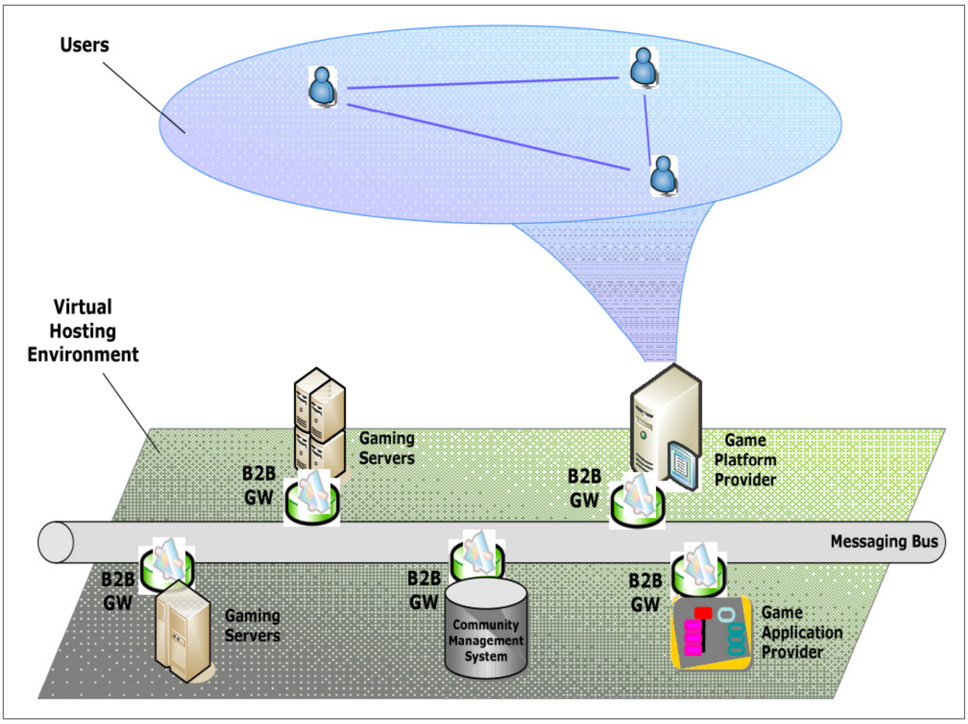
The virtualisation of hosting environments refers to the federation of a set of distributed hosting environments for the execution of an application and the possibility to provide a single access point (*e.g.*, the VHE gateway) to this set of federated hosting environments.

In a typical scenario, a number of host providers offer hosting resources to the application providers for deploying and running their applications, which are then 'virtualised' with the use of middleware services for managing the nonfunctional aspects of the application and are transparently exposed to the end user via a single VHE.

Consider the case depicted in Figure 4, which is being addressed in the pilot as an example of a business model enabled with a VHE.

In this scenario, the gaming application provider deploys its gaming application onto two different execution environments (gaming servers) owned by different host providers. The gaming platform provider, who wants to offer the game to an end user, discovers gaming servers and creates business relationships with them and with a separate service provider who offers a system for community management (of gaming clans, tournaments, advanced statistics). Through the use of the VHE, these various services are offered transparently to an end user, including the gaming platform provider's ability to perform load balancing and server selection based on the defined SLAs.

Figure 4 A VHE-enabled gaming scenario (see online version for colours)



The VHE developed in this BE consists of a network of B2B service gateways integrated with common capabilities for B2B trust federation, identity management, access control, SLA management and accounting and monitoring, as well as application service and resource virtualisation. The B2B gateway functionality is complemented by a federated messaging bus and community management services that facilitate the establishment of B2B collaborations (*e.g.*, in the form of VOs).

The scenario presented in Figure 4 is clearly a B2B collaborative scenario which foresees the federation of several service and gaming providers. In the scenario above, we have assessed some of the developed design patterns and components.

With respect to the VO set-up, the purpose of our preliminary tests has been to assess the following functionalities required in the VO formation phase:

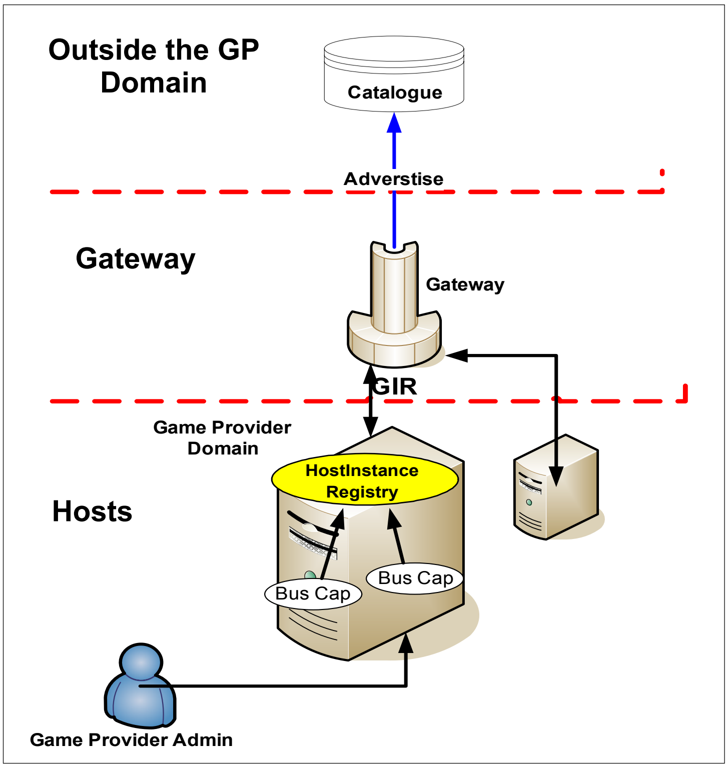
- discovery of potential members on the basis of the capabilities they can offer to the VO
- invite potential members to join the VO
- start the federation process (identity management)
- publish VO members after their acceptance of the invitation.

Before doing the test, our scenario presumes that each potential member of the VO has advertised its capabilities to the rest of the world. This is done, in our scenario, offline by the gaming provider administrator. The gaming provider presents a single point of access and a two-level hierarchy of registries to publish its business capabilities. On each hosts of the provider domain there is a host instance registry, and on the provider gateway there is a gateway instance registry.

The gaming provider has a business relationship with an entity providing a general catalogue and, if it wants to, the gaming provider advertises its capabilities to the rest of the world via the catalogue.

Figure 5 presents the situation.

Figure 5 Game advertising (see online version for colours)



Apart from the registry management part of the VO set-up component that involves traditional publish/update operations, our tests have been mainly focused on the secure B2B federation of VO members.

From a methodological point of view, we have assessed the added value of the proposed model for the federation of administrative domains with respect to the current state-of-the-art. The proposed model allows, for example, a single administrative domain to federate just a specific capability. This allows a more fine-grained approach to resources and services federation that is more suitable for business applications with respect to the models proposed in the eScience community.

With respect to the application virtualisation component, the executed tests have had the purpose of assessing both the virtualisation process and the graceful shutdown.

The virtualisation process is executed when there is the need to configure the VO's underlying infrastructure. The process involves the creation of the business service that a member has promised to offer, the configuration of the management services (*e.g.*, security services, SLA services, accounting) and, eventually, the integration and exposition via the gateway of the created instances.

In reverse, the graceful shutdown is executed in the VO dissolution phase when the resources have to be released and the bindings of the VO member need to be removed. The process involves invoking the VO set-up to remove service instance entries from the service instance registry, clean up and destroy the management services, clean up the gateway (*e.g.*, remove its internal mapping between virtual and real EPRs) and, eventually, destroy the business service instance.

5 Lessons learnt about the adoption of grid technologies and VO management in e-business

In addition to the design patterns and software components, another relevant result achieved in the BEinGRID project relates to the lessons learnt by the authors about the adoption of VO management in e-business. The lessons that we report in this section have been learnt during the analysis of and interactions with the BEs of the BEinGRID project.

During our analysis, we have understood that despite a strong research interest in the fields of grid technologies and VO management, the use of VO for *ad hoc* dynamic collaboration (also referred to as dynamic VO management) is still immature in terms of interest and adoption in e-business.

Moreover, we observed that a current pattern is trying to reuse in e-business the existing research infrastructures mainly developed in e-science contexts (such as the one developed in the Enabling Grids for E-sciencE (EGEE) project.[7] In the cases above, the objective is to rely on an already existing VO and solutions for VO management for computational and data resource sharing. We have seen that the current reuse of these existing infrastructures and solutions is successful only in specific business cases that foresee as 'mission critical' the execution of simulations, analyses of large datasets and, in general, present High Performance Computing (HPC)-like features.

In many other business cases for which, for example, *ad hoc* dynamic collaboration is required or that foresee the provision of services as applications, this approach should be avoided since, at least in its current stage, the solutions developed for eScience research infrastructures do not offer the capabilities required by these cases such as, for example, a rich trust management model, the ability to separate collaboration contexts and react to contextual changes, *etc*.

These cases should consider the reuse or building on top of the results and findings of projects that developed VO models based on requirements from businesses such as TrustCom,[8] Akogrimo,[9] BREIN[10] or BEinGRID.

Lastly, with respect to the capabilities for VO management provided by grid middleware, we concluded that none of the most adopted grid middleware (such as Globus Toolkit,[11] gLite,[12] Unicore[13]) offers all the capabilities required for VO management. According to our experience, this limitation has a significant negative impact on the adoption of VOs in e-business. Still, the current implementations of the most adopted grid middleware encourage the adoption of the VO paradigm mainly for computational resource sharing without paying adequate attention to the B2B collaboration aspects that underpin a commercially viable use of the VO paradigm in any business context.

In contrast, we observed the emergence of complementary technologies such as, for example, web services-based federated identity management and Web 2.0, based on models that place humans and knowledge at the centre of the process. Our experience in the VO management activities of the BEinGRID project has been that it was useful to investigate such emerging technologies in order to complement and improve the shortcomings of the most adopted grid middleware.

A final consideration is about security for VO and semantics, two challenging areas proposed in the BEinGRID project. These areas attracted a lower interest than initially expected. The two main reasons appear to be that they focused on problems that: (a) have been considered by the BEs too difficult to address in the timescales of their activities (about 18 months) or that (b) they were considered not mission critical for meeting the successful BE demonstration requirement, but more important for the subsequent exploitation of the BE results.

Table 1 Lessons learnt

The main lesson learnt	Despite its potentialities, dynamic VO has a 'hole' in terms of interest and adoption in e-business.
About the reuse of already existing VO infrastructure and solutions	Reuse of already existing VO infrastructures and solutions (mainly developed in big eScience projects) in e-business is useful just in some specific cases.
	Ad hoc dynamic collaboration requires capabilities, such as trust among members and contract negotiation, that are currently not provided by the abovementioned solutions.
About security for VO and semantics	Security for VO and semantics are considered either too challenging or not mission critical.
About the most adopted grid middleware	The most adopted grid middleware do not offer all the functionalities of VO management.
	Today, the current implementation of the most adopted grid middleware encourages VO adoption mainly in large-scale computations.

Our concern is that many BEs may have made a mistake by avoiding to tackle these issues, especially those related to security for VO that, if not addressed in the early stages of development, may lead to unnecessarily complex fixes or require the substantial reengineering of inadequately informed design decisions and, in some cases, may also hinder the commercial exploitation of an otherwise good solution.

On the basis of the presented considerations, we can report the following lessons learnt.

6 Conclusions and future works

In this paper, we have presented the improvement of the GRASP middleware in order to allow B2B collaboration in the context of dynamic VOs. The presented software components can be reused in several common contexts where there is the need to federate different administrative domains and, as evidenced, can be composed in order to address complex issues, such as the creation and management of the VHE previously described.

As evidenced in the previous section, our preliminary results allow us to assess the benefits of the virtualisation process and the graceful shutdown, as well as of the federation model.

The full validation of these components is coming as part of the other activities of the BEinGRID project. Currently, in fact, it is started as a second wave of BEs having the purpose of selecting some of the developed components and adopting them in their architecture for validation and evaluation purposes.

Acknowledgement

This document has been produced in the context of the BEinGRID project (http://www.beingrid.eu). The BEinGRID project is part of the European Community's Sixth Framework Programme for research and is, as such, co-funded by the European Commission. A sincere thanks goes to all the organisations involved in the BEinGRID Business Experiment 09. The European Commission has no liability in respect of this document, which merely represents the authors' views.

References

Alur, D., Crupi, J. and Malks, D. (2001) *Core J2EE Patterns: Best Practices and Design Strategies*, Prentice Hall.

Easton, J., *et al*. (2006) 'Patterns: emerging patterns for enterprise grids', IBM Redbook, http://www.redbooks.ibm.com/abstracts/sg246682.html.

Foster, I., Kesselman, C., Nick, J.M. and Tuecke, S. (2002a) 'Grid services for distributed systems integration', *IEEE Computer*, Vol. 35, No. 6.

Foster, I., Kesselman, C., Nick, J.M. and Tuecke, S. (2002b) 'The physiology of the grid: an open grid services architecture for distributed systems integration', *Open Grid Services Infrastructure WG, Global Grid Forum*, 22 June.

Foster, I., Kesselman, C. and Tuecke, S. (2001) 'The anatomy of the grid: enabling scalable virtual organizations', *International Journal of Supercomputer Applications*, Vol. 15, No. 3, pp.200–222.

Fowler, M. (2003) *Patterns of Enterprise Application Architecture*, Addison-Wesley.

Gaeta, A., Gaeta, M., Smith, A., Djordjevic, I., Dimitrakos, T., Colombo, M. and Miranda, S. (2007) 'Design patterns for secure virtual organization management architecture', *The First International Workshop on Security, Trust and Privacy in Grid Systems*, 17–21 September, Nice, France, *The Proceeding of the Third International Conference on Security and Privacy in Communication Networks (SecureComm2007)*, IEEE, pp.207–216, DOI 10.1109/SECCOM.2007.4550334.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1995) *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, ISBN: 0201633612.

Rana, O.F. and Walker, D.W. (2002) *Service Design Patterns for Computational Grids, Patterns and Skeletons for Parallel and Distributed Computing*, edited by F.A. Rabhi and S. Gorlatch, Springer-Verlag, ISBN: 1-85233-506-9.

Notes

1 GRASP project: http://eu-grasp.net/english/default.htm.
2 BEinGRID project: http://www.beingrid.eu.
3 VO Management Design Patterns: http://www.gridipedia.eu/258.html.
4 VO Management Software Components: http://www.gridipedia.eu/259.html.
5 VOMS: http://edg-wp2.web.cern.ch/edg-wp2/security/voms/voms.html.
6 BEinGRID BE09: http://www.beingrid.eu/index.php?id=be9.
7 EGEE project: http://public.eu-egee.org/.
8 TrustCom project: http://www.eu-trustcom.com/.
9 Akogrimo project: http://www.mobilegrids.org/.
10 BREIN project: http://www.eu-brein.com/.
11 The Globus Toolkit: http://www.globus.org/.
12 gLite middleware: http://glite.web.cern.ch/.
13 Unicore middleware: http://www.unicore.eu/.