Grid Technologies to Support B2B Collaboration

Nicola Capuano[1,2], Matteo Gaeta[1], Sergio Miranda[1], Francesco Orciuoli[1,2], and Pierluigi Ritrovato[1]

[1] University of Salerno, Department of Information Engineering and Applied Mathematics, via Ponte don Melillo, 84084 Fisciano (SA), Italy
[2] CRMPA, Centro di Ricerca in Matematica Pura ed Applicata, via Ponte don Melillo, 84084 Fisciano (SA), Italy
ncapuano@unisa.it, {gaeta,smiranda,orciuoli, ritrovato}@diima.unisa.it

Abstract. In the context of the European 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 software components of a Service Oriented Grid Middleware, named GRASP, that the authors have defined, designed and implemented in the frame of a former homonymous European Project (FP5). The main improvement of GRASP due to the application of the BEinGRID design patterns is the support for the creation of Virtual Organizations. This paper presents the authors experience and lessons learnt in adopting the GRASP middleware to set up a business to business federated environment supporting collaborations among enterprises. The concrete case study relates to the on line gaming applications and the adoption of the Software as a Service business model to provide the game applications.

Keywords: virtual organization management, SaaS, SOA, grid, design patterns.

1 Introduction

We present in this paper the improvement done to the GRASP middleware [12] in order to allow the creation and management of a federated environment. GRASP middleware is the main result of the homonymous FP5 EU project and represents novel architecture supporting Grid-enabled collaboration for the purposes of Application Service Provision.

Leveraging the convergence of Grid and Web services technologies [1] [2], during the GRASP project authors anticipated the emergence of new business and scientific computing paradigms that are based on dynamic Virtual Organisations (VOs) [11]. These VOs span across organisational boundaries and enable the enactment of collaborative processes that integrate services, resources and knowledge in order to perform tasks that the VO partners could not undertake on their own.

In the context of the FP6 EU IP BEinGRID [10], authors have analysed a set of concrete case of studies relating to the adoption of Grid technologies in business domains. During this analysis, some concrete requirements have been elicited and

M.D. Lytras et al. (Eds.): WSKS 2008, CCIS 19, pp. 388–397, 2008.

design patterns [13] to develop software components [14] based on service oriented Grid technologies have been produced.

In the definition of the design patterns, the authors have taken into consideration past experiences such as [3], [4], [5], [6] and [8]. Relying on these patterns and component, the GRASP middleware has been improved. This paper focuses mainly on the components and on their preliminary evaluation. For more information on the design patterns, interested readers can refer to [9].

The rest of the paper is organized in the following way. The section 2 briefly describe the GRASP middleware, its key and distinctive features with respect to other middleware. The section 3 presents the improvement of the GRASP middleware due to the application of the BEinGRID design patterns. The Section 4 presents the concrete case study, our preliminary results and the lessons learnt. The section 5 draws conclusions and presents future work.

2 The GRASP Middleware

The GRid based Application Service Provision (GRASP) project [12] have experimented the use Grid computing in order to support the operation and evaluate the sustainability of new models of ASP and thus contribute to the evolution from traditional ASP via IBSP to new paradigms.

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

- the “federated” ASP model, which can be described as the collaboration of many (ASP) GridService providers that provide services that can be combined to complex services addressing a customer need that each of them could not achieve themselves, and
- 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 Grid-based Application Service Provision, a prototype realization 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 services.

Orchestration – One of the most important aspects of the new Grid based ASP models is that no longer one single vendor controls the whole process. This means that a mechanism is needed that orchestrate the services offered by different vendors and ensure a controlled collaboration. The GRASP orchestration service is based on BPEL4WS and provide the possibility for a hybrid orchestration for Grid Services but also for Web Services.

SLA Monitoring – The Orchestrator can only fulfil its task in controlling the collaboration between the different services if enough information for the decision process is avail-able. The SLA monitoring services monitor, enforce and provide notifications in order to assist the Orchestrator in this task.

Accounting & Billing – Without Accounting & Billing no Application Service Provision can be performed. As the services are no longer controlled by one single entity but from many different service providers over time new ways on collecting provided services must be introduced. Especially for the "many-to-many" model new solutions must be identified.

The following picture present a graphical view of the three key pillars of GRASP.

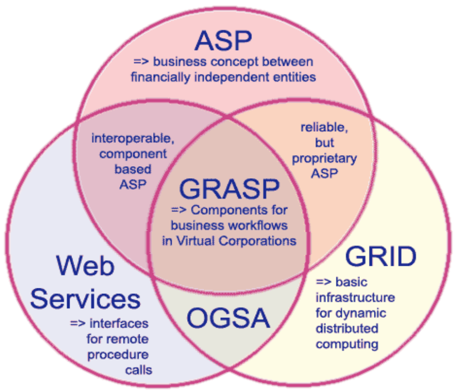

Fig. 1. GRASP Pillars

3 Evolution of GRASP towards Supporting B2B Collaboration

The GRASP middleware provides a solid foundation for Services-to-Services interactions but it lacks of capabilities to support the VO formation and lifecycle 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 the Application Virtualisation

3.1 VO Set-Up

This component is required to set up relevant information of a B2B collaboration allowing the finalization of the VO Creation process.

During the VO Creation process, in general, there is the need to perform operations like configuration of the infrastructure, instantiation and orchestration of the application service, assignment and set up of resources and activation of services, notification of the involved members, and manifestation of the new VO.

The high level architecture of this component is presented below:

- **VO Set-up:** This is in charge to finalise the VO Creation process. It is mainly a façade interacting with the other components
- **Registry:** This contains new members and service instances of the VO
- **Federation:** This is in charge to create a federation and to manage the identify of the federation members. This component can be designed according to the Secure Federation Design Patterns and other patterns proposed by the security area of the BEinGRID project [10].

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 off-line). It is worth to mention that the Federation subcomponent of the VO Set-up is the most challenging one.

In contrast to the current state of the art, that is mainly based of results coming from the efforts of the eScience community (e.g. VOMS), the solutions proposed take into consideration the needs and requirements of the business world. This aspect has a deep impact in 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 membership and management of resources in a Virtual Organisation (for instance, allowing the access to a whole resources for job submission) in our case we have requirements that foresees a fine-grained approach (for instance, allowing the access to specific capabilities offered by a Service Provider).

The main benefits, as evidenced at the end of the previous section, relates 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 federation of trust realms, and on the other hand, managing the life-cycled of 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 SOA platform by some middleware vendor or instead of departing in expensive product integration projects that give them identity provision and federation, at a very high cost, for the specific application at hand.

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 the execution of the exposed capabilities 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 direct and unmanaged access of VO members to VO resources.

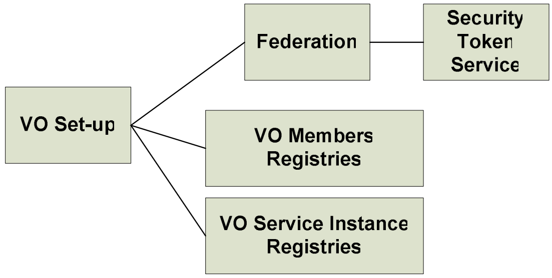

Fig. 2. VO Set-Up

It is appropr\iate to apply 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 picture there are notes evidencing that some components relates to other patterns produced in the BE-inGRID project (of the VOM and/or other clusters). This means that, for instance, the Policy&Rules component can be designed according to the patterns proposed by the Security Cluster.

The Application Virtualization component follows the Façade pattern by Gamma et al [3] and it is in charge to invoke the other classes of the system in order to execute the virtualization process.

The Application Virtualization, the Runtime Monitor and the 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 Virtualization.

If the Application Virtualization component is also the Gateway, when a request for accessing a service arrives, the Application Virtualization can operate according to the Chain of Responsibility [3] and pass the request along a chain of handlers. The building blocks are described below.

- **Application Virtualization:** This implements the virtualization process steps. It delegates requests to appropriate subsystem objects. It returns to client a reference to access the created application instance through the Gateway. It can be configured to be the Gateway and when a request for accessing a service arrives, it can pass the request along a chain of handlers.
- **End Point Reference Mapping:** This is in charge to map the End Point Reference (EPR) of the created application service instances to the EPR of the Gateway. It is also in charge to Activate and Deactivate the application service instances if, respectively, the process creation and termination of service instances succeed.
- **Policy & Rules:** This is in charge to apply 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 [10].

- **Runtime Monitor:** This is in charge to collect 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 & evaluate the SLA identified by the SLA Management cluster of the BEinGRID project [10].
- **Registry:** 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.

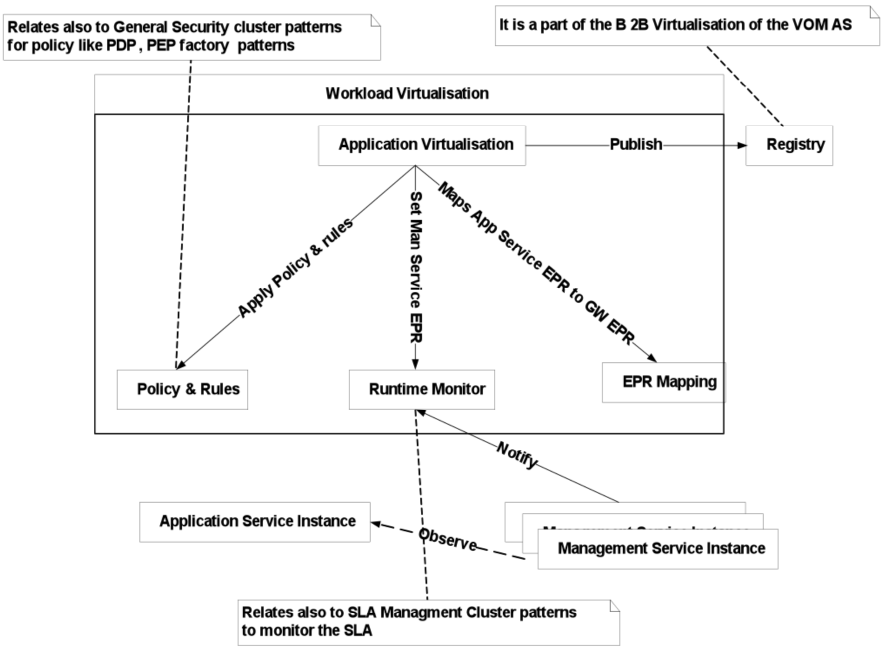

Fig. 3. Application Virtualization

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

In addition the 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 providers, Access & 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 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 ISV fashion though “pay-per-use” or “pay-as-you-grow” models.

4 Case Study: On Line 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 on-line gaming [15].

Virtualization of Hosting Environments refers to the federation of a set of distributed hosting environments for 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 “virtualized” with the use of middleware services for managing non-functional 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 the VHE.

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

The VHE developed in this business experiment consists of a network of B2B service gateways integrated with common capabilities for B2B trust federation, identity management, access control, SLA management, 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 Virtual Organisations).

The scenario presented above is clearly a B2B collaborative scenario which foresee the federation of several Service and Game providers. Currently, we are finalising the development of this scenario but we have done preliminary tests in order to validate the capabilities of the VO set-up and Application Virtualisation 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 member to join the VO,
- Start the federation process (Identity Mgm),
- 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 to the rest of the world its capabilities. This is done, in our scenario, off-line by the Game Pro\vider Administrator. The Game Provider presents a single point of access and a two level hierarchy of registries to publish its business capabilities. One on each hosts of the provider domain there is an Host Instance Registry. One on the provider gateway, there is a Gateway Instance Registry.

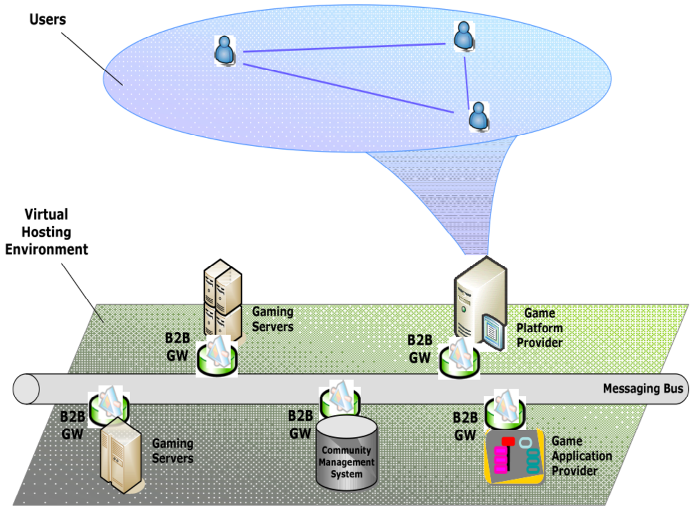

Fig. 4. VHE-enabled Gaming Scenario

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

The following picture presents the situation. A part 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 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 more suitable for business applications with respect to the models proposed in the eScience community.

With respect to the Application Virtualisation component, the tests executed 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 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 Gateway of the created instances.

By the 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 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.

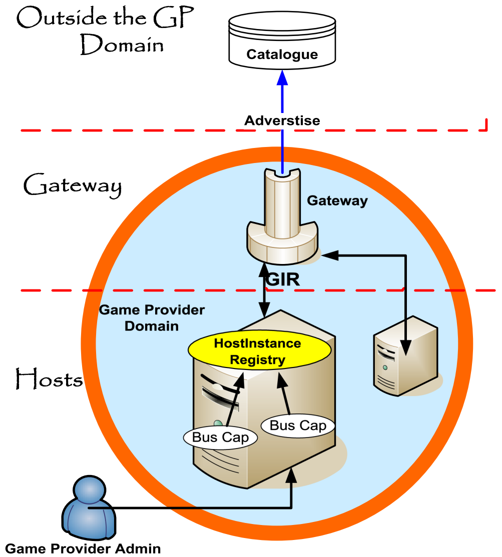

Fig. 5. Game advertising

5 Conclusions and Future Work

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 software components presented can be reused in several common contexts where there is the need to federate different administrative domain 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 allows us to assess the benefits of virtualisation process and the graceful shutdown as well as of the federation model.

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

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 Program for research and is as such co-funded by the European Commission. A sincere thank goes to all the organisations involved in the BEinGRID Business Experiment 09. The European Commission has no liability in respect of this document, which is merely representing the authors' views.

References

1. Foster, I., Kesselman, C., Nick, J.M., Tuecke, S.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Globus Alliance Technical Report (2002)
2. Foster, I., Kesselman, C., Nick, J.M., Tuecke, S.: Grid Services for distributed systems. IEEE Computer (2002)
3. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1995) ISBN 0201633612
4. Easton et al.: Patterns: Emerging Patterns for Enterprise Grids, IBM Redbook, http://www.redbooks.ibm.com/abstracts/sg246682.html
5. Rana, O.F., Walker, D.W.: Service Design Patterns for Computational Grids. In: Rabhi, F.A., Gorlatch, S. (eds.) Patterns and Skeletons for Parallel and Distributed Computing. Springer, Heidelberg (2002)
6. Fowler, M.: Patterns of Enterprise Application Architecture. Addison-Wesley, Reading (2003)
7. Buschmann, F., et al.: Pattern-Oriented Software Architecture. John Wiley & Sons Ltd., Chichester (1996)
8. Alur, D., Crupi, J., Malks, D.: Core J2EE Patterns: Best Practices and Design Strategies. Prentice Hall, Englewood Cliffs (2001)
9. Gaeta, A., Gaeta, M., Smith, A., Djordjevic, I., Dimitrakos, T., Colombo, M., Miranda, S.: Design patterns for Secure Virtual Organization Management Architecture. In: Proceeding of the First International Workshop on Security, Trust and Privacy in Grid Systems, Nice, France, September 17 (2007)
10. BEinGRID project website, http://www.beingrid.eu
11. Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of Supercomputer Applications 15(3), 200–222 (2001)
12. GRASP Project, http://eu-grasp.net/english/default.htm
13. VO Management Design Patterns, http://www.gridipedia.eu/258.html
14. VO Management Software Components, http://www.gridipedia.eu/259.html
15. BEinGRID BE09, http://www.beingrid.eu/index.php?id=be9