# On-the-fly Dynamic Virtual Organizations in a Secured Grid Environment

Sylvain Jeuland, Christine Morin, Yvon Jégou

# On-the-fly Dynamic Virtual Organizations in a Secured Grid Environment

Sylvain Jeuland, Christine Morin, Yvon Jégou

INRIA, Centre Rennes - Bretagne Atlantique, Rennes, France
{Sylvain.Jeuland,Christine.Morin,Yvon.Jegou}@inria.fr

**Abstract.** This paper describes a system for managing dynamic collaborative workflows among business partners working in the engineering field and submitting proprietary applications to the grid. Although grid Virtual Organizations allow users from different administration domains to share resources for job submission, building static Virtual Organizations and managing the security concerns is a burden for partners who wish a short-lived collaboration. We present X-DVOs which are on-the-fly and automatically built dynamic Virtual Organizations. They manage the security configurations on behalf of grid users. In the future X-DVOs will allow business partners to automatically initiate engineering choreographies and orchestrations on geographically dispersed grid and cloud resources.

## 1 Introduction

Engineering collaborations aim allowing users from different institutions to reach an engineering common goal. Partners share resources and perform a workflow where applications are executed and data exchanged among partners. If applications need a lot of computing resources for being executed partners can use the Grid [1] infrastructure which is the combination of computer resources from multiple sites and which provides big processing power.
Virtual Organizations (VOs) are temporary or permanent alliances of enterprises or organizations that come together to share resources, skills, information and core competences in order to better respond to business opportunities or large-scale application processing requirements, and whose cooperation is supported by computer networks. VOs rely on the grid infrastructure which combines computer resources from multiple administrative domains that share a common task, usually to a scientific, technical or business problem that requires processing ressources or large amounts of data.
The XtreemOS grid operating system [2] provides for grids what a traditional operating system offers for a single computer: abstraction from the hardware and secure resource sharing between different users. It thus simplifies the work of users belonging to VOs by giving them the illusion of using a traditional computer while removing the burden of complex resource management issues of a typical grid environment. When a user runs an application on XtreemOS, the

grid operating system automatically finds all resources needed for the execution, configures users credentials on the selected grid resources and starts the application on the allocated resources.
Nevertheless classical Virtual Organizations (like those used in XtreemOS or by VOMS [3]) are not suitable to set up on-the-fly engineering collaborations in a dynamic way since they rely on static registration of institution grid resources or users. Indeed they need a long time to be set up since they require manual steps and are adapted to long-term cooperations.
In this paper we present the X-DVO system, an architecture which enables the creation of on-the-fly Dynamic Virtual Organizations (DVOs) gathering users from different engineering projects. X-DVO relies on XtreemOS and extends its static VO concept. We describes how new users register to a DVO manager, get automatically their security credentials and upload grid resources for sharing. This work about DVOs is a first step to manage automated collaboration engineering workflows where a set of users run their applications sequentially inside the DVO.
In Section 2 we present an example of engineering collaboration scenario. Then the Section 3 describes the static VOs used in XtreemOS and their limitations. The Dynamic VO concept is described in Section 4 before the new DVO functionalities (Section 5). Section 6 shows how the DVO works for the considered engineering collaboration scenario example. Section 8 concludes the paper.

## 2 Engineering Collaboration Scenario

An engineering collaboration scenario gathers partners working together to meet a common goal. They run engineering applications and give the output data to the next partner in the collaboration plan. Computing resources can be shared by partners to build a overlay.
We use a specific use-case to present the Dynamic Virtual Organizations. A formula one (F1) car designer wishes to perform an aerodynamic analysis of his car. He needs to mesh the car and the air and to compute the aerodynamic forces onto the moving car. Since the designer does not possess the meshing and aerodynamism computing expertise and software, he found some partners who do. Then a collaboration is initiated among the F1 car designer, the partner who provide the meshing expertise and the partners who provides the aerodynamism computing expertise.
A collaboration is needed in this scenario because some engineering applications are private and the owner wants executes his program by himself or lend it for a chosen lifetime. Moreover some data must be protected. How the applications, data and resources are shared is decided by the partners. Here is our use-case:

- the F1 designer has his own F1 design software but no resource to execute it,
- the mesh computing partner shares local resources and executes his own mesh program with F1 car data.

– the aerodynamic computing partner shares local resources and shares his aerodynamic program as a service.

In the following we show how to build a Dynamic Virtual Organization which responds to these requirements : resource and application sharing, data exchanges among partners, job submissions onto grid, security mechanisms and fast DVO creation. Can we set up a simple VO for this collaboration ?

## 3 XtreemOS static Virtual Organizations

We describe here the Virtual Organization management in XtreemOS and its limitations.

### 3.1 VO as a Living Cell

A Virtual Organization gathers different persons and servicesUsers from administration domains submit jobs onto resources managed locally by a resource certification authority. Services manage the VO and the security: user authentication, resource access control and authorization policies.

**Users** are the VO persons who want to run jobs onto some VO resources. They request and obtain credentials to prove their identity, their membership to the VO and they have the right to run jobs onto resources. Users can belong to a group or be given a specific role. These parameters can be used for resource access authorization policies.

**The VO Manager** is the person who manages a VO. Each VO has a VO manager (the VO creator for instance) who accepts or refuses the incoming user membership requests. He can attribute users specific roles or put them into groups to permit them to pass resource access local policies based on roles, groups and VO identifiers.

**Administration Domains** are the foundation of the VOs because they are the institutions where VO users and VO resources come from. The VO manager allows users to be member of the VO since he knows their administation domain provenance and he has a kind of contract with that administation domain. Users are identified by the VO manager by presenting their own home administration domain certificate which prove the appartenance of their institution.

**Credential Distribution Authority (CDA)** Once a user is registered as a VO member he needs to get security credentials delivered by the Credential Distribution Authority. This one provides a VO certificate which weares the signature of the CDA and the Certification Authority (CA) traditionally used in classical public key infrastructures. In the local policy authorization process the resource checks that the VO certificates have been (directly of indirectly) generated by the CDA or the CA.

**Resources** allow the execution of user jobs. They perform authorization with their local policies based on VO identifers, roles, groups and certificate chain (CA to end VO user). If an policy information does not match certificate attributes the user session is not opened and the job not submitted. The policy configuration is static since each local administrator has to add a rule each time a VO is created (VO identifier and possibly role and group parameters).

**Resource Certification Authority (RCA)** The VO manager registers participating resources from different domains thanks to domain RCA servers which are in charge of local administration domain resources. A local resource has to register to the local RCA to enter a VO and to get a VO security certificate which allows it to run jobs.

**VO Policy Service** VOPS is a component which sets up global VO authorization policies (groups of users, user roles). Policies permit to select appropriate resources when a job is submitted. Global policies must be calibrated with local resource policy to avoid that a job is submitted onto a resource which do not run the job because of unauthorized access.

### 3.2 VO Lifecycle

Here we describe how to set up a VO and to register users. Once the VO is created users can submit jobs in the next phase.

**Creating the VO** When partners wish to create a Virtual Organization, they choose someone who connects to the VO management web interface (VOLife) and create the VO. He can add a Resource Certification Authority from an administration domain. This last one are able to register resources in the VO.

**Adding a New User** A user connects the VOLife to request the VO membership. Then the VO manager connects to the VOLife to give membership to the user. Then the user gets its new VO certificate from the VOLife and downloads in its file system volume. The VO certificate is needed for submitting job on VO resource. The file system used is XtreemFS [4], a grid file system federating storage resources in different administration domains accessible from any node.

**Submitting a Job** In XtreemOS the Application Execution Manager selects the available resources for submitting an application according the hardware requirements and the VO policies. When a user submits a job with the JSDL [5]specification file, the AEM checks the JSDL and the users VO certificates. Then it contacts the VOPS to filter resources according to global policies while checking the VO certificates attributes. If it passes, the job is submitted on appropriate hardware resources. If all match, the submission is not blocked by the aforementioned local policies.

### 3.3 Limitations of XtreemOS VOs

VOs in XtreemOS are long-term and rather static since organizations share resources for a long duration. If partners want to initiate a short-lived collaboration with few job submissions, the setting up of classical VOs could appear very heavy. A new XtreemOS tool is needed with the objective to speed up and to automate the VO creation and operation. We have the three following issues.

**Communication among VO Entities** The setting up of the VO requires communications among various entities. First a potential user has to request membership to the VO manager while using the VOLife web interface. The VO manager has to validate the request. Each step requires that the entities are well coordinated but the whole process may take a long time, espcially if the number of users grows.

**Automated Credential Distribution** Users have to care about getting credentials. So they have to connect to the VOLife web interface to build the VO credential while choosing the VO. Then they store manually the certificate at the appropriate location in their file system volume. this static process is very long and not suitable for short-lived collaborations.

**Not Suitable VO Local Policies** The resource administrators store the VO identifiers in the local policies to restrict access for VO users. Each time a long-lived VO is created, the policies are updated with a new VO rule. This is not suitable for multiple short-lived VOs because administrators are not able to modify very quickly the policy by adding new rules with a new VO identifier.

## 4 X-DVO: XtreemOS Dynamic Virtual Organizations

To solve the problems we discussed above we have proposed the X-DVO system which creates Dynamic Virtual Organizations. DVOs are a new kind of short-lived VOs which are created on-the-fly to meet the engineering collaboration scenario needs.

### 4.1 How to Solve Static VO Problems

The goal of Dynamic VOs is to solve the problems met in the world of static (XtreemOS) VOs.

**Communication among DVO Entities** Instead of connecting to a web server to create VOs and register to a VO, users and the person managing the DVO run daemons on their client machines. These daemons send messages to other peer endpoints. Secure TLS (Transport Security Layer) channels permit to send secured messages and to authenticate the peer and to check some certificate parameters. So the manager daemon can add a user to a newly created DVO by verifying where the user comes from in real time. If the mannager accepts the user, it sends automatically the user data to other appropriate daemons.

**Automated Credential Distribution** Since the DVO entities can communicate in a dynamic way, it is possible to build on-the-fly processes. So the whole user registration process is automated. This is the same for the DVO credential creation and distribution. Infrastructure machines are needed to run some services which help the DVO creation and can be provided by the manager daemon.

**Not Suitable VO Local Policies** As we have seen above we have to remove the VO identifier parameter in local policies because it is not possible to add manually a set of rules per DVO creation. The domain administrators could dedicate some DVO-aware by installing a DVO module which allows a DVO User who has already an opened access (same administration domain) to delegate access rights to the other DVO partners. The DVO identifier must be provided to the local policy without human intervention.

### 4.2 DVO Ecosystem

The DVO ecosystem contains some entities like user daemons (which are run by partners), resources and the manager daemon (which is the daemon run by the partner managing the DVO). Nevertheless entities do not communicate each others with the same way.

**DVO Users** are the daemons run by collaboration partners. A partner daemon which needs to collaborate with the already registered DVO Users begins by register itself to the manager daemon. To become a DVO user the partner daemon needs to provide a home administration domain certificate that the manager daemon accepts. When a partner registers to the DVO he can provide resources from his home admistration domain into the DVO collaboration since he has a local contract with these home resources.
The main DVO user actions are:
- to register to the manager daemon and retrieve their credentiels.
- to provide administration domain resources into the DVO.
- to copy their local engineering data to the shared XtreemFS DVO volume.
- to submit application onto DVO resources. The application can be provided by a partner who has shared the application for all DVO users.

**The DVO Manager** is the daemon run by the person managing the DVO (who is one the collaboration partners). This one is in charge of creating the DVO and registering new incoming DVO users after having checked their identity. The DVO manager has a local DVO database and stores the list of users for each DVO he controls. Then he forwards the DVO user home administration domain certificate to the DVO-Box a service, which is in charge of creating the DVO file system volume and providing DVO certificate. One the DVO user is validated by the DVO-Box the DVO manager warns him.

**Administration Domains (ADs)** are the basic pools of DVO users and resources. The DVO manager can choose only DVO users who come from a list of known administraiton domains. Moreover some DVO users have

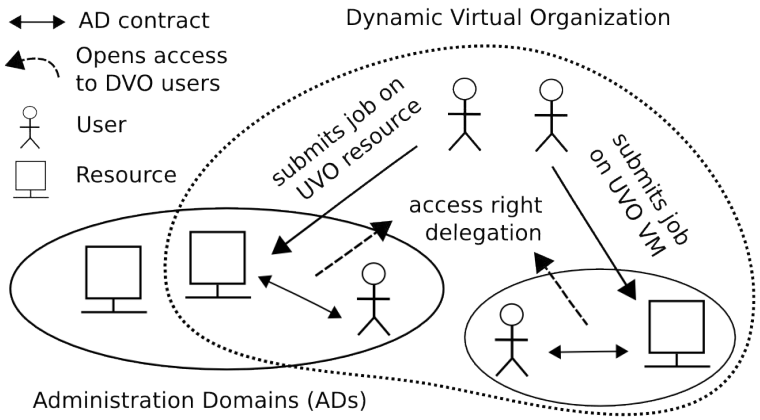


**Fig. 1.** DVO structure and resources: the users provide resource into the DVO and delegates resource accesses to DVO users.

already access to some home administration domain resources: that allows them to put these resources into the DVO with delegation.

**Resources and Access Delegation** The DVO resource overlay is built from the DVO Users administration domain resource pools (see Figure 1). Each DVO User provides (or not) some resources from its own administration domain. The other DVO Users are allowed to submit job onto these provided resources since the DVO User providing them has already access to these AD resources and can delegate the access right to all DVO Users. So DVO resource accesses are done on behalf of the related DVO User.

Some configurations have to be done to allow resources to take part in a DVO. The resource administrator makes the resource DVO-aware by adding a new local policy temporal rule: accept DVO Users from the specific DVO identifier. For instance during the collaboration a user B can submits a job on the resource provided by the user A into the DVO. This resource will check the user B DVO certificate and its current local policies. The accounting is done on behalf of user A.

If a DVO user has an access contract with a resource from the same administration domain, a local policy daemon runs on the resource and stores the DVO identifier which has been provided by the DVO User. When a job is submitted by an other DVO User, its DVO certificate and the DVO identifier are validated by the local policy daemon and the application is submitted. The home administration domain of the submitting DVO User can be additionally checked against authorized administration domain attributes.

### 4.3 Lifecycle

The lifecyle has two main phases: people registers to the DVO in the Formation Phase and applications are executed by DVO Users in the Operation Phase.

**The Formation Phase** is the creation of the DVO. As we mentionned before users register to a DVO manager. This one triggers the operation phase when the conditions for running the collaboration (lists of partners) are filled. We describe the formation phase in more details in the next section.

**In the Operation Phase**, users can send messages to coordinated their actions, application submissions on resources and data exchanges. These exchanges are facilitated by the DVO XtreemFS volume which collects all data. A user A can access data of a user B by accessing in the user B folder. Folder access rights have to be well configured to open the folder to the dedicated users with access control lists for instance. So user A can run an application using user B data on resources shared with user C. Once all users have executed their applications, the workflow is done and the DVO can be dissolved.

## 5 The X-DVO Services

In this section are presented the new services which support the Formation Phase and the Operation Phase. Then we describe the detailed process to generate credentials for a new incoming DVO user (see Figure 2).

### 5.1 Services

The main services are the DVO-Box and the automated DVO-CDA which creates DVO certificates from DVO request certificates.

**A Daemon Communicating With the TLS protocol** run for each entity and service (DVO Users, DVO Manager, DVO-Box, DVO-CDA, etc). They listen the peer requests and react depending on them (user registration, credential distribution, etc). Daemons have the advantage to respond very fast to client requests because they do not require a human intervention. For security concerns all communications are protected by the TLS protocol.

**The DVO-Box** interacts with the DVO manager and receives some requests such as:

- It receives a DVO creation request from the DVO manager and creates a DVO with an XtreemFS DVO volume.
- It receives a DVO user membership request and adds him to the DVO while making a user folder inside the XtreemFS DVO volume.
- It sends a message to the DVO manager when a DVO user is well registered.
- It receives DVO User DVO request certificates for signature and forwards them to the Credential Distribution Authority (DVO-CDA).

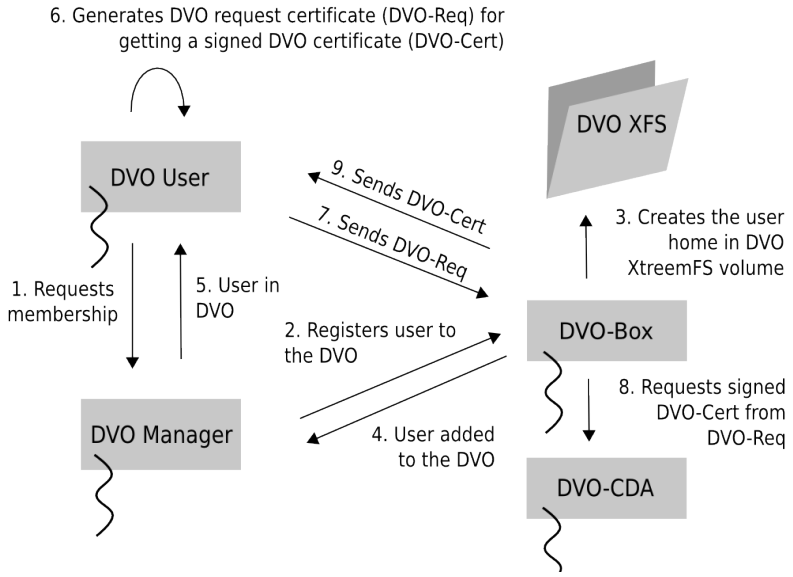


**Fig. 2.** Adding a user to the X-DVO

– It receives signed DVO User DVO certificates from the DVO-CDA and forwards them to the user.

The DVO-Box is a central component in the DVO creation since it creates the common data volume and passes the certificates to validation. Moreover further checks are done when a job is submitted to the Application Execution Manager. The AEM checks with the DVO-Box if a user is a member of a given DVO.

**The Credential Distribution Authority (DVO-CDA)** is now an automated daemon and it has to sign the incoming user DVO certificate when the DVO-Box sends it user DVO request certificates. The DVO-CDA authenticates DVO Users with their home administration domain certificate before the creation of the new DVO certificate.

**A Local Policy Delegation Daemon** is run by an administrator on each resource and it allows the DVO Users of a given DVO to access the resource. The DVO identifier is specified by a DVO User which is in the same administration domain and can access the resource. The local policy delegation daemon can check other attributes like the AD provenance of the DVO User or the access rights the DVO Users needs to run its job. The local policy daemon may block resource access if the needed rights are too important.

### 5.2 Formation Phase and Credential Retrieval

In this subsection we show how a DVO is formed. We describes the different steps and messages occuring among a user, the DVO manager, the DVO-Box and the DVO-CDA (see Figure 2 again). To begin the DVO manager must start on a resource the different XtreemOS services like the Application Execution Manager, the XtreemFS services, etc. Then all users have to get members of the current DVO:

- The user daemon sends his home administration domain certificate to the DVO manager which contacts the DVO-Box. The DVO-Box creates the DVO and the XtreemFS volume (if not yet done) and adds the DVO User to the DVO.
- Once the DVO User is registered, it creates a DVO certificate request with some job requirements (hardware requirements), his attributes (administration domain, needed rights) and sends it to the DVO-Box. The DVO-CDA is requested to build the DVO certificate from the certificate request.
- When the DVO Manager know that all DVO Users have their certificates, the DVO Users transfers their application data to the XtreemFS DVO volume. The Operation Phase is triggered.

### 5.3 Operation Phase and Job Submission

The DVO Users submit their job and exchange data in the Operation Phase (see Figure 3). When an application is executed a message is sent from a DVO User to the next one.

## 6 How it Works in Engineering Collaboration Scenarii

In this section we come back to our use-case and show how to build the DVO. Before the DVO creation the three partners decide that the F1 designer manages the DVO. So this one run the DVO Manager daemon. The endpoint reference of the manager daemon is known by all.

**The F1 designer** has his own F1 design software but does not provide resource to execute it. The F1 designer runs the DVO User daemon with his home administration domain certificate as a parameter. This one contacts the DVO Manager which adds the F1 DVO User to the DVO. At the end F1 DVO User credentials are stored at the right place in its folder in the XtreemFS DVO volume. Almost no human intervention was needed for getting the DVO credentials and putting them the right place.

**The mesh computing partner** wants to share local resources and executes his own mesh program with F1 car data. As the F1 designer partner does, the mesh computing partner runs DVO user and retrieves the DVO credentials. Moreover a new resource parameter has been added to DVO User command: the resource IP address. The resource Local Policy Delegation Daemon has

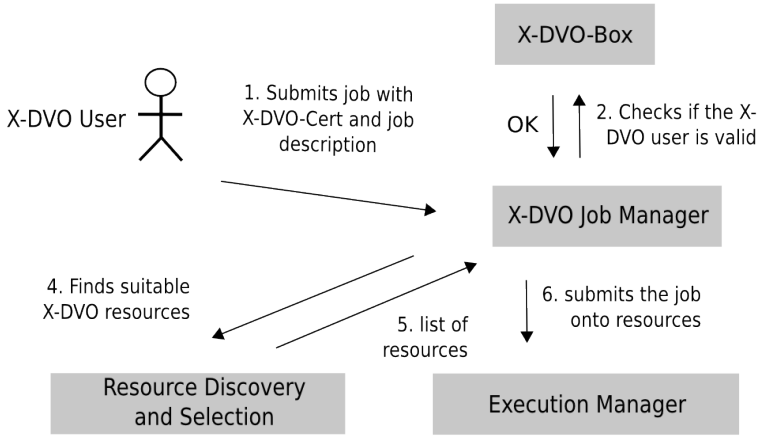


**Fig. 3.** The user submits a job on the X-DVO resources.

been contacted by the DVO User and this daemon will authorize all DVO user accesses since it has a permanent contract (the home administration domain certificate is used) with the mesh designer DVO User. The resource registration information was sent by to the DVO Manager. Almost no human intervention was needed for getting the DVO credentials, putting them the right place and for providing a resource into the DVO.

**The aerodynamic computing partner** shares local resources and shares his aerodynamic program as a service. This user lets the aerodynamic programm in the home administration domain resource and delegates the access right to the F1 designer resource. This partner runs the DVO User command. So he gets the credentials and provide the resource to the DVO without additional intervention.

**Problems solved** All the human interventions (needed in VO creation) have been removed in the DVO creation process. Setting up an on-the-fly DVO is now simple and user-friendly.

**Limitations** Some information have to be exchanged among the partners before the Formation Phase. For instance the mesh partner must know he has to use the F1 designer partner data to run the meshing service. Moreover the aerodynamic computing partner must know he has to lend the software to the F1 designer partner. Information are stored in one document called the X-CDL which is read by DVO Users.

## 7 Related Work

TrustCoM [6] developped a framework for trust, security and contract management in dynamic Virtual Organisations. It enables the secure enactment of collaborative business processes in the web and grid service fields whereas DVOs focus on collaborative grid engineering applications and the building of DVO resource overlay onto the XtreemOS Linux grid operating system.

In Shibboleth [7] the user wishing to access a resource is redirected to the WAYF service which asks the user to choose his home identity provider (the IdP is trusted by the resource provider). This last one requires a user login/password authentication and sends a user authenticated SAML [8] assertion to the resource. Since the resource provider trusts the IdP the user is authenticated. The SAML assertion contains the IdP attribute authority that the resource provider contacts gor getting the user attributes. Then authorization is performed on the resource.

DVOs do not need the WAYF service or IdP services because resource providers trust the DVO Users which register them to the DVO. Indeed DVO Users and there related resources are in the same adminstration domain. So the other DVO Users submit applications on resources on behalf of the corresponding DVO User.

## 8 Conclusion

In this article we present the X-DVO architecture and we show how to build automatically on-the-fly Dynamic Virtual Organizations gathering a set of users onto grid resources from different administration domains. Thanks to secured communication daemons partners are able to define a common goal and set up a short-lived collaboration in an easy manner.

DVO partners provide home administration domain resources to the DVO resource overlay. They delegate to all DVO partners the access of these resources which do not authorized external users from others DVOs.

Our work about Dynamic Virtual Organizations permits users from different administration domains to share resources and exchange data in a collaborative way. The future work would focus on the establishement of dynamic and automatic workflows: this would integrate the setting up of on-the-fly DVO lifecyle and the automation of job submissions and data exchanges. Depending on the collaborative nature of the workflow, we plan to implement orchestrations and choreographies in the framework of Dynamic Virtual Organizations.

Moreover it will be interesting to include cloud resources [9] along grid resources which introduces the problems of resource heterogeneity and subscontracting: a cloud provider could outsource the DVO resource providing to an external provider and we would have point to point contracts to manage. The XtreemOS project could be used for these experiments.

## References


[1] Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications **15**(3) (2001) 200–222

[2] Morin, C.: Xtreemos: A grid operating system making your computer ready for participating in virtual organizations. In: ISORC '07: Proceedings of the 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing, Washington, DC, USA, IEEE Computer Society (2007) 393–402

[3] Alfieri, R., Cecchini, R., Ciaschini, V., dell'Agnello, L., Frohner, A., Lrentey, K., Spataro, F.: From gridmap-file to voms: managing authorization in a grid environment. Future Gener. Comput. Syst. **21**(4) (2005) 549–558

[4] Hupfeld, F., Cortes, T., Kolbeck, B., Stender, J., Focht, E., Hess, M., Malo, J., Marti, J., Cesario, E.: The xtreemfs architecture—a case for object-based file systems in grids. Concurr. Comput. : Pract. Exper. **20**(17) (2008) 2049–2060

[5] Rodero, I., Guim, F., Corbalan, J., Labarta, J.: How the jsdl can exploit the parallelism? In: CCGRID '06: Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid, Washington, DC, USA, IEEE Computer Society (2006) 275–282

[6] Deitos, R., Kerschbaum, F., Robinson, P.: A comprehensive security architecture for dynamic, web service based virtual organizations for businesses. In: SWS '06: Proceedings of the 3rd ACM workshop on Secure web services, New York, NY, USA, ACM (2006) 103–104

[7] Sinnott, R., Jiang, J., Watt, J., Ajayi, O.: Shibboleth-based access to and usage of grid resources. In: GRID '06: Proceedings of the 7th IEEE/ACM International Conference on Grid Computing, Washington, DC, USA, IEEE Computer Society (2006) 136–143

[8] Saklikar, S., Saha, S.: Next steps for security assertion markup language (saml). In: SWS '07: Proceedings of the 2007 ACM workshop on Secure web services, New York, NY, USA, ACM (2007) 52–65

[9] Morin, C., Gallard, J., Jégou, Y., Riteau, P.: Clouds, a New Playground for the XtreemOS Grid Operating System. Technical Report RR-6824, INRIA (February 2009)