Computing and Informatics, Vol. 33, 2014, 303–326

SECURITY AND VO MANAGEMENT CAPABILITIES IN A LARGE-SCALE GRID OPERATING SYSTEM

Benjamin Aziz, Ioana Sporea

School of Computing
University of Portsmouth
Portsmouth PO1 3HE
United Kingdom
e-mail: {benjamin.aziz, ioana.sporea}@port.ac.uk

Abstract. This paper presents a number of security and VO management capabilities in a large-scale distributed Grid operating system. The capabilities formed the basis of the design and implementation of a number of security and VO management services in the system. The main aim of the paper is to provide some idea of the various functionality cases that need to be considered when designing similar large-scale systems in the future.

Keywords: Grid computing, security engineering, VO management

Mathematics Subject Classification 2010: 00-General

1 INTRODUCTION

Grid middleware has become an integral part of scientific computing. The notion of Virtual Organization (VO) is used to scale the Grid to large numbers of users and computing nodes. Synergies can be achieved by grouping users, which share OS-specific, application-specific resources and computing nodes [9]. However, their use is yet to become widespread commercially. The complexity of managing VOs and the difficulty assuring user and VO security are two major barriers to such wider use. As a result of the criticality of the management of VO membership and security, it is mandatory to consider such functionality at an early stage of the software development lifecycle to ensure a safe implementation of the system.

This paper presents the main security and VO management capabilities in a large-scale Grid operating system called XtreemOS [6, 11]. These capabilities are used by administrators, users and XtreemOS components to establish, operate, evolve and terminate a Grid and the VOs within. The capabilities define the relevant use cases in XtreemOS and help establish some idea about what may be needed to manage security and VOs in a large-scale distributed operating system.

1.1 Overview of the XtreemOs Operating System

XtreemOS (`www.xtreemos.eu`) [6, 11] was an EU FP6 project, which aimed at building a Grid-based distributed operating system that provided a single abstraction of physical hardware and software services offered by a collection of standalone Linux operating systems to users within a Grid. These operating systems could function collaboratively to support the utilisation of computational and storage resources regardless of the geographical location of their users or machines. A major function of XtreemOS was to hide the complexity of distributed resources dynamically aggregated from large-scale cross-domain resource providers and to ensure the transparency of using such a distributed operating system. Hence, similar to a standalone operating system, once a user is registered with XtreemOS, it should be conceptually the same to utilise resources from any machine that the system is composed of, regardless of whether such resources have been recently added to the system or have been there before.

As illustrated in Figure 1 [6], XtreemOS is composed of two parts: the XtreemOS foundation, called XtreemOS-F, and high level Grid services, called XtreemOS-G. XtreemOS-F is a modified Linux kernel embedding support for Virtual Organisations (VOs), where a VO in XtreemOS is a temporary collaboration among various Grid resource providers and resources for achieving a specific goal. XtreemOS-F provides kernel level process checkpoint/restart functionality. XtreemOS-G comprises several Grid OS distributed services to deal with resource and application management in VOs, and is implemented onto XtreemOS-F.

XtreemOS targets scalable and flexible management of dynamic VOs [5]. XtreemOS Grid spans over multiple administrative domains on different sites, comprising heterogeneous resources that can be shared by the participating organisations. A Grid member can create a VO, for which s/he becomes the VO owner. Any Grid member can request his/her registration in a given VO, subject to the VO owner approval. Resources can be registered in VOs as well. The VO owner defines policies stating permissions and usage rules for VO resources. The Grid administrator also defines policies regulating what a Grid member can do (for example, permission to create a VO). Resource owners in different administrative domains may also define local policies for resource usage. Grid, VO and local policies are enforced by the XtreemOS system.

The Application Execution Management (AEM) [2] services are in charge of discovering, selecting and allocating resources for job execution, as well as starting, controlling and monitoring jobs. Data management in XtreemOS is achieved with

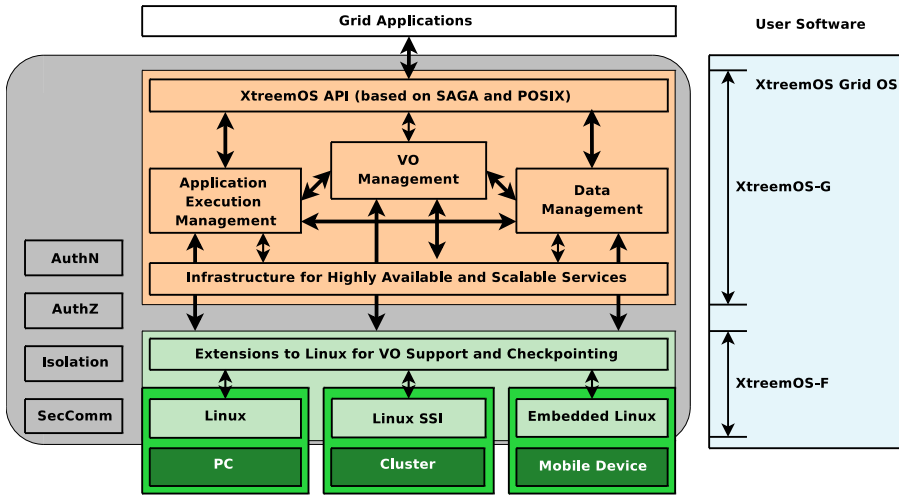

Figure 1. XtreemOS software architecture [6]

the XtreemOS Grid file system – XtreemFS [15]. XtreemFS federates multiple data stores located in different administrative domains and provides secure access to stored files to VO members, whatever their location.

1.2 Virtual Organisation Lifecycle in XtreemOS

The VO lifecycle in XtreemOS consists of a number of phases:

- Grid Management: In this phase, site administrators willing to offer their site resources to the Grid advertise the general site. At this stage, there are no VOs formed yet.
- VO Creation: Once the XtreemOS Grid has been set-up in the previous phase, VOs will be created at this phase and their structure and attributes defined. However, VOs have not yet been populated at this stage.
- VO Evolution: After the creation of a VO, users, core services, resources and policies can be added to and removed from it.
- VO Operation: A fully formed VO is operational, which means that users can start utilising the VO's resources.
- VO Termination: Finally, once the VO has achieved its goal/purpose, it can be terminated by dissolving its membership and terminating any outstanding jobs.

Each of these phases is implemented by means of a number of *capabilities*, which define the functionality that the operating system provides in support of the various use cases in each phase of the VO management lifecycle. And since these capabilities

also take the security of the system into account, we have termed them the *security and VO management capabilities in XtreemOS*.

1.3 Structure of the Paper

The rest of the paper is organised as follows. In Section 2 we give an overview of related work. In Section 3 we describe the structure of the XtreemOS security and VO Management services, the trust domains and the various actors involved. We then introduce the security and VO management capabilities in Section 4. Finally, in Section 5 we conclude the paper.

2 RELATED WORK

Grid middleware systems, such as Globus (`www.globus.org`), provide several examples of VO management services. Of these, the two most popular ones are VOMS (Virtual Organization Membership Service) [1] and CAS (Community Authorization Service) [12]. VOMS [1] is an important VO reference implementation because it provides a popular approach for integrating VO information (e.g. user roles in a VO) into node-level enforcement mechanisms. However, managing VOMS effectively is an non-trivial task because authorization decisions are often a result of a joint process between the VOMS server (participating in the form of VOMS credentials) and nodes. Managing such decision making process consistently and coherently can be non-trivial and may potentially have scalability problems for large VOs with a significant number of resources. It also makes it difficult to create new VOs and introduce new resources and members dynamically.

CAS [12], on the other hand, builds on top of the Grid Security Infrastructure (GSI) [8]. It takes control over the policy specification by explicitly spelling out the relationship between VO users and resources. CAS represents a push model of enforcing VO policies because its policy enforcement is decoupled from VO information (e.g. user groups/roles). Therefore, it is comparatively easy to create dynamic VOs using the CAS model. However, it is not always easy to figure out what VO resources users need to use in advance. Overall, VOMS and CAS represent two different ends of the spectrum. VOMS leans to a pull model where access control is done at nodes by pulling policy information on demand (i.e. from the VOMS), while the CAS model uses a push approach by distributing the fine-grain control of community policy.

Generally, the VO management system in XtreemOS (XVOMS) is not compatible with other VO systems in production grids, since it was designed with operating system architectures (rather than existing Grid middleware technologies) in mind. Hence, for example, it is not possible to use VOMS or CAS instead of XVOMS.

The importance of incorporating special use cases for security in the development lifecycle of computing systems has been highlighted in the past by Tenday [14], who proposes the use of misuse cases and obligation use cases in the software development

lifecycle as a means for expressing security concerns at the very beginning of the development process. Similarly, Firesmith [7] defines some examples of such security use cases in the general context of computer systems relevant to integrity, privacy and access control. In the context of Grid systems, Rosado et al. [13] deploy reusable security use cases in defining the requirements for mobile Grid environments.

3 THE SECURITY AND VO MANAGEMENT MODEL IN XTREEMOS

In this section, we provide an overview of the XtreemOS security and VO management model in terms of the trust domains, actors and different services involved.

3.1 Trust Domains

The security and VO management services in XtreemOS are based on three main trust domains. These domains are described as follows:

- The Resource site domain: This includes sites that offer resources to the Grid and any VOs formed out of the Grid.
- The User site domain: This includes sites that provide users of VOs who will submit jobs to the resources included in those VOs.
- The Core site domain: This represents the core site in which the Security and VO Management (and possibly other XtreemOS) services may be running.

From the trust point of view, the Core site represents the root of trust for both the Resource and User sites. In other words, a Core site must have a high level of assurance since it will be running critical system components.

3.2 Actors

Having defined the main trust domains in the previous section, we now introduce the main actors of the security and VO management services.

- The User: this actor is the user of VOs, who is also registered in the Grid within which the VOs are created.
- The VO Administrator: this actor is a previous User who created a VO and became the owner and administrator of that VO. Therefore, the VO Administrator has full authority on managing the VO.
- The Resource Administrator: this is the actor owning the resources offered to VOs. The actor could be either a whole site administrator or the owner of one or more machines belonging to the site.
- The Grid Administrator: this actor is responsible for managing the core XtreemOS security and VO management services.

These actors are shown in Figure 2, which also depicts to which trust domain is each actor expected to operate.

Figure 2. Actors and their trust domains in XtreemOS

3.3 Core Security and VO Management Services

The core XtreemOS security and VO Management services are shown in Figure 3.

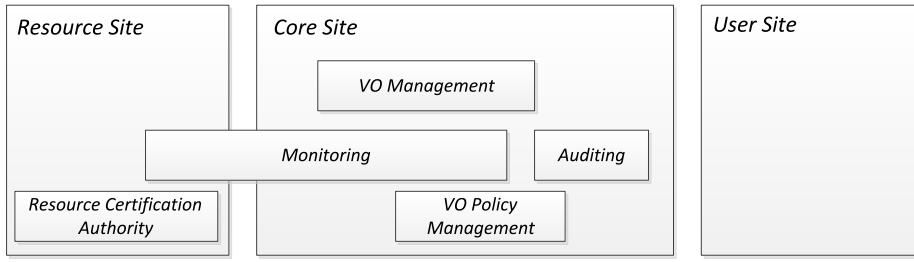

Figure 3. The XtreemOS security and VO management services and their trust domains

These services consist of the XtreemOS VO Management Service (XVOMS), the Resource Certification Authority (RCA) and the VO Policy Service (VOPS). Additionally, monitoring and auditing services can also be deployed across the Core and resource sites to raise the level of assurance as to the general behaviour of the

system and its processes. However, we shall only focus here on the core security and VO management services. Next, we describe each of these services independently.

3.3.1 XtreemOS VO Management Service (XVOMS)

The XVOMS [4] is a VO and trust management service whose architecture is illustrated in Figure 4. XVOMS provides a logical grouping of the infrastructural services needed to manage the entities involved in a VO and ensure a consistent and coherent exploitation of the resources and capabilities inside the VO.

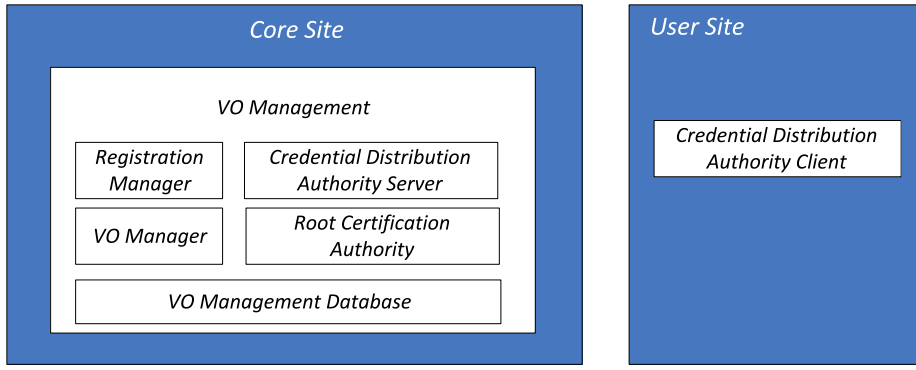

Figure 4. Structure of the XtreemOS VO management service

The XVOMS system consists of the following components:

The Root Certification Authority. This is a manual service that creates the XtreemOS trust anchor, the *root certificate*, and uses it to certify the identity of core services within XtreemOS. This service can be performed offline to avoid compromise of the root private key. The certification of core services can optionally be performed by the CDA service, described next.

The Credential Distribution Authority Server. This component, also referred to as the CDA server, is responsible for distributing XtreemOS identity certificates (XOS-Cert) to users. XOS-Cert are X509-based certificates [10]. The CDA server may optionally be configured to also provide service certificates, certifying the identity of XtreemOS core services.

The Credential Distribution Authority Client. This is a client-side program that interfaces with the CDA server, in the case when it is not possible to use the VO Web Frontend interface.

The Registration Manager. This component is responsible for managing the initial registration of users and RCAs with the XtreemOS system.

The VO Manager. This component controls the lifecycle of the VO as described in Section 1.2. There is one instance of a VO Manager component running

per each live VO, and its functions are controlled by the user who is the VO's Administrator

The XVOMS Database. This is the main database in XVOMS in which all the information regarding the user and RCA registrations, VO membership and lifecycle is stored.

The VO Web Frontend. This is a Web-based interface to the functionality offered by XVOMS.

3.3.2 Resource Certification Authority (RCA)

The RCA [4] is a certification authority at the level of administrative sites offering resources to XtreemOS VOs. The RCA consists of the following components, as shown in Figure 5. The RCA is responsible for bootstrapping trust in the individual resource domains. This trust is used by other XtreemOS components (e.g. AEM services) to be able to submit jobs to resources.

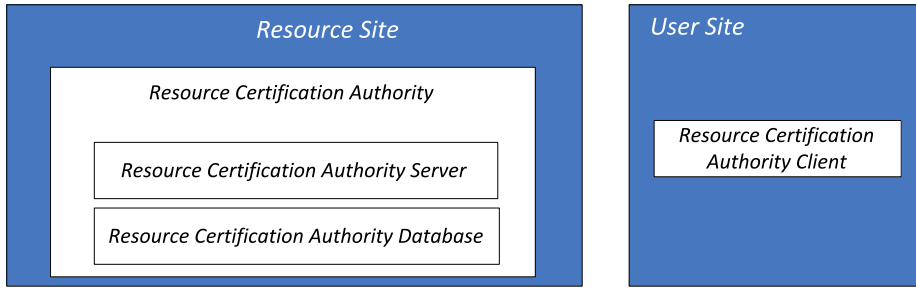

Figure 5. Structure of the XtreemOS RCA service

The following paragraphs give an overview of the main logical components of the RCA.

The RCA server. This is the main component, which provides the functionality of the RCA. The server is responsible for issuing certificates to resources.

The RCA client. The RCA client is a client-side program that can interact with the RCA server.

The RCA Web Frontend. The RCA Web Frontend is a Web-based interface to the RCA server, which is an alternative to the RCA client program.

The RCA database. The RCA database stores the state of resources in each administrative domain. This state could indicate that a resource is unregistered with the Grid, registered with the Grid and if so, whether it is currently offered to any VOs in the Grid. The main interface to the RCA database is through the RCA server functionality.

3.3.3 VO Policy Service (VOPS)

The VOPS [4] is used to manage and enforce VO policies. The service consists of the following components, as shown in Figure 6.

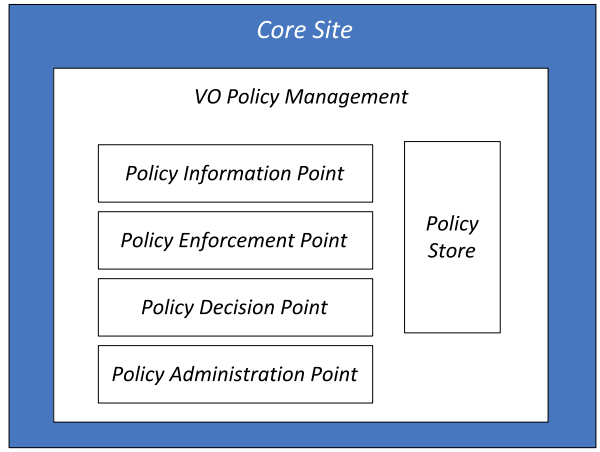

Figure 6. Structure of the XtreemOS VO policy service

VOPS is composed of the following components.

The Policy Enforcement Point. The Policy Enforcement Point (PEP) is where the users' requests are intercepted in order for these requests to be checked and appropriate decisions enforced on the requests. The user requests may carry user credentials regarding their attributes.

The Policy Decision Point. The Policy Decision Point (PDP) is the component which enforces the security policies on user requests. The PDP contains the logic that is computed against the policies and the users requests.

The Policy Information Point. The Policy Information Point (PIP) is a component of VOPS which queries information about the request arriving from a user, additional user credentials and information about the context of the request and the system.

The Policy Administration Point. The Policy Administration Point (PAP) allows the site or resource administrator to add, delete and update policies in the policy store.

The Policy Store. The Policy Store (PS) is a database containing all the policies related to the different resources.

4 SECURITY AND VO MANAGEMENT CAPABILITIES

This section presents the different capabilities associated with the security and VO management functionality in XtreemOS. These capabilities are organised around the XtreemOS VO lifecycle of Section 1.2. The capabilities represent "what" the system can do in terms of its services outlined in Section 3.3. All of these capabilities were implemented as API using Java language and they can be run through command line interface, and in some cases through a Web interface (e.g. Capabilities 4.4.1, 4.4.2).

4.1 Grid Management Capabilities

The Grid management capabilities include the registration and removal of users and RCAs with the Grid, the registration of local resources with RCAs, the set-up of Root Certificate Authority (RCA) and running the various security and VO management services. These capabilities are depicted in Figure 7 and below we describe each of these in more detail.

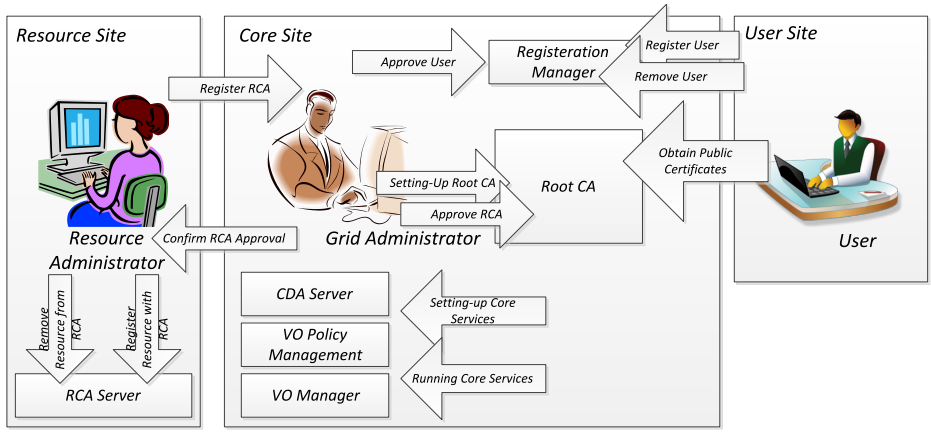

Figure 7. The grid management capabilities

One can divide the grid management capabilities into the following two broad categories: *setting-up the Grid infrastructure* and *populating the Grid*. The first set of capabilities is related to the actual set-up of a new XtreemOS-based Grid infrastructure. This is the very first step in the VO lifecycle.

4.1.1 Configuring and Creating the Root CA

This capability is concerned with the creation and configuration of the root of trust in a XtreemOS Grid, i.e. the Root CA. The capability is performed by the Grid Administrator, who is responsible for generating the private key for the Root CA

and the public key certificate, which is later used for signing any Certificate Signing Requests (CSRs) from other services. In the ideal case, the machine (node) running the Root CA must be of high assurance and not networked, to minimise security vulnerabilities. The public key certificate itself however is placed on a networked core machine ready and available for public distribution when needed. Since this is an early step in configuring the Grid, it must be executed before any of the core services are configured and started running.

4.1.2 Creating the XVOMS Database

The second capability is related to the creation of the XVOMS database, which will later hold all the information related to the XtreemOS VOs as well as the Grid membership. Again, this capability is performed by the Grid Administrator, who initialises and sets the database ready for use. This also includes setting up a password for the database, which could be the same password assigned to the Grid Administrator (root login). Similar to the previous capability, this is an early step in configuring the Grid and therefore it must be executed before any of the other Core Services are configured and started running.

4.1.3 Setting-Up and Running Core Services

Once the Root CA and the XVOMS database components have been set-up and initialised, it is now possible to set-up and run the Core Services in a XtreemOS Grid. The actors responsible for this capability are the Grid Administrator as well as the Resource Administrator for any organisations willing to join the Grid and provide resources. Once this capability has been executed, the Core Services of the XtreemOS system will be up and running ready for users. This also implies that their security credentials have been created. The Root CA certificate and the CDA (Credential Distribution Authority) certificate are considered public certificates and therefore are placed on a networked node ready for distribution. Additionally, the Root CA certificate should be made available for downloading from the home page of the VO Web Frontend.

4.1.4 Obtaining Public Certificates

This capability is initiated by a User who wishes to obtain the public key certificate(s) of one or more of the Core Services from the Root CA. As a precondition, the User is expected to have the public key certificate of the trusted certification authorities installed on his/her system and providing a way for checking the trust and security of the communications with the Root CA. Once the capability is successfully executed, the User will have obtained the public key certificates of the requested Core Services in the XtreemOS system.

4.1.5 Processing Certificate Requests

This capability is performed by the Grid Administrator in order to generate certificates verifying the identity of a Core Service when requested by a User as per the previous capability. A CSR (Certificate Signing Request) is converted into a public key certificate, which is sent to the originator of the request.

The next set of capabilities are related to the addition of users and resources to an already set-up XtreemOS Grid.

4.1.6 Registering Users

The first capability for populating the Grid aims at allowing Users to register to the Grid. In concrete terms, this means that a User can request to have a Grid account upon providing their new account details (user name, password, real name, organisation and email address.) The capability is executed on the Registration Manager (part of XVOMS) by the User via the VO Web Frontend interface. A precondition is that the XVOMS (XtreemOS VO Management Service) database must already have been set-up and configured ready for receiving information on Grid membership. Once the capability has been successfully executed, the User will have an account on the database and he/she will end up sharing a password with the Registration Manager. The success of this capability is dependant on the completion of the next capability on approving Users.

4.1.7 Approving Users

This capability is applied by the Grid Administrator, who will react upon receiving a request to join the Grid from some User as per the previous capability. The Grid Administrator will make sure that the request itself is valid (e.g. the organisation to which the User belongs is admissible to the Grid). It also implies that the Grid Administrator can contact the User by email or telephone to request further details before approving their request. If the Grid Administrator is satisfied with the request and the associated information about the User, he/she will then approve the request leading to a successful completion of this capability as well as the previous one.

4.1.8 Removing Users

The Grid Administrator can also remove an already registered User from the XtreemOS Grid. As a result, the User will no longer be capable of logging in to the Grid, joining VOs or submitting any computational jobs. Another form of this capability is that the User himself decides to leave the Grid by either sending a request to the Grid Administrator or by executing the relevant commands on the VO Web Frontend.

4.1.9 Registering RCA

This is the first capability related to the population of the Grid with RCAs representing resource administrative domains. The capability allows a Resource Administrator to request the join of a RCA to a Grid from the Grid Administrator. When successfully completed, the capability will allow the Resource Administrator and the Registration Manager to share a password for managing the RCA's account in XVOMS. However, its success is pretty much dependant on the following capability.

4.1.10 Approving RCA

This capability will allow the Grid Administrator to a request submitted via the previous capability for joining a RCA to the Grid. Once this is achieved, the RCA will obtain an account on XVOMS for its membership in the Grid and future VOs. Similarly, the Root CA is informed of the decision in order to link the RCA to its chain of trust.

4.1.11 Confirming RCA Approval

This capability is for the Grid Administrator to inform the Resource Administrator of the decision of joining the RCA to the Grid. This is necessary as the Resource Administrator can now start applying the following two capabilities related to the offering of resources to the Grid and its future VOs.

4.1.12 Registering Resources with the RCA

Since the RCA is the main gateway for local resources (machines, nodes, services) to join the Grid, this capability is an essential one that allows the Resource Administrator to register the local resources with the RCA. As a result, the resource details will be recorded on the RCA and the resource will be issued with an identity certificate that it can use to identify itself to other Grid resources or users.

4.1.13 Removing Resources from the RCA

Finally, the last capability in the Grid management capabilities is related to the removal of any resources wishing to exit the XtreemOS Grid. As a precondition, it must be the case that the resource is already registered through the previous capability, and that there are no pending jobs related to the Grid users currently running on the resource. Once the resource has been removed from the RCA list of Grid membership, any subsequent requests for its identity certificate or any other machine attributes certificates will fail. Upon successful completion of this capability, the resource is considered to be outside of the XtreemOS Grid.

4.2 VO Creation Capabilities

The VO creation capabilities facilitate the setting-up of VOs, their attributes and their policies. These capabilities are not concerned with populating VOs with users and resources, which is considered to be part of the VO evolution phase discussed later. Figure 8 represents the VO creation capabilities discussed next.

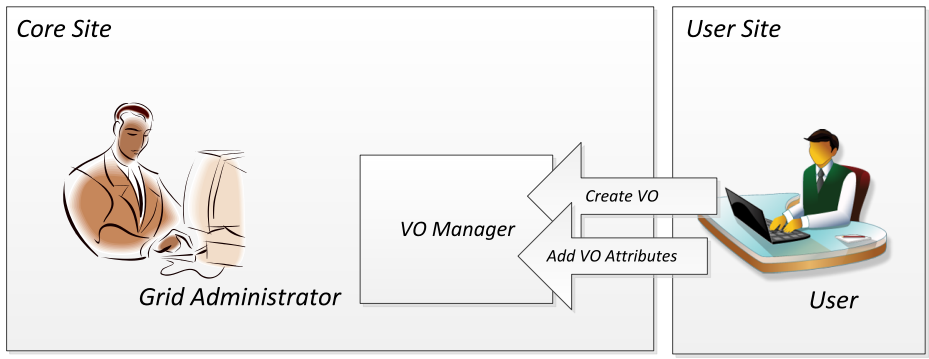

Figure 8. The VO creation capabilities

4.2.1 Creating VOs

The first capability is to allow a User to create a VO (Virtual Organization) using either the command line interface of the VO Manager (part of the XVOMS service) or the VO Web Frontend. The VO must not have existed already and the success of completing this capability will result in the creation of the new VO. The new VO is identified by means of a unique identifier, which is bound to the User as the owner and the administrator of the VO. The VO is also provided with a name for the readability of its users. The details of the new VO will be stored on the XVOMS database.

4.2.2 Adding VO Attributes

The second and last capability in this part of the VO lifecycle is related to the definition of the VO attributes for new VOs created during the previous capability. These attributes define the "structure of the VO" in terms of its groups and the roles within those groups. This capability can only be performed by the owner of the VO.

4.3 VO Evolution Capabilities

The next phase in the lifecycle of a XtreemOS VO is the evolution phase. VO evolution capabilities facilitate the management of users, resources and policies within VOs. The following sections describe the various capabilities grouped by users, resources and policy management.

User Management. The first set of VO evolution capabilities is related to the management of users as shown in Figure 9. User management includes the following specific capabilities.

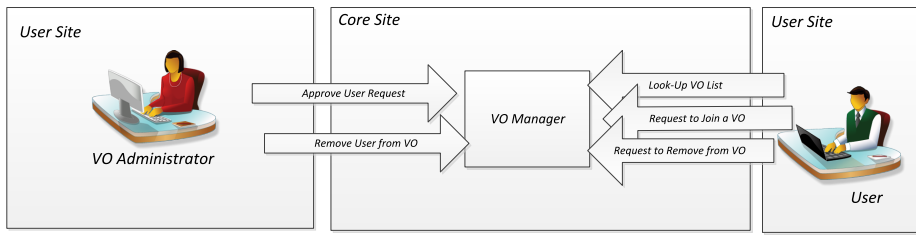

Figure 9. The VO evolution capabilities – user management

4.3.1 User Look-up VO List

The first capability allows a User registered to the Grid to list the set of VOs currently available in that Grid. The request to obtain this list is submitted to the VO Manager component of XVOMS. Upon receipt of the list of available VOs, the User can then proceed to the next capability.

4.3.2 Requesting to Join VOs

This capability allows a User to request from the VO Manager (part of the XVOMS) to join a specific VO (which is in the list of available VOs obtained in the previous capability). This request is submitted via the VO Web Frontend and it will be stored in a queue of similar requests ready for the next capability.

4.3.3 Approving User Requests

Upon receipt of a VO joining request from some User, the VO owner (also some other User) will react by either approving or rejecting the request. If approved, the requesting User will be added to the list of members of the VO (in the XVOMS database). If not, the User is informed of the rejection outcome.

4.3.4 Requesting to Remove from VOs

This capability allows a User, who is currently a member of a VO, to request to be removed from that VO. The request is also submitted via the VO Web Frontend. The request will then be queued for approval by the VO owner in the next capability.

4.3.5 Removing Users from VOs

Once a request for removal from a VO is received, the VO owner will act to approve the request. Hence, upon successful completion of this capability, the requesting User is removed from the list of members of the VO in the XVOMS database.

Resource Management. The second set of capabilities is related to the management of resources during the VO evolution phase, as shown in Figure 10.

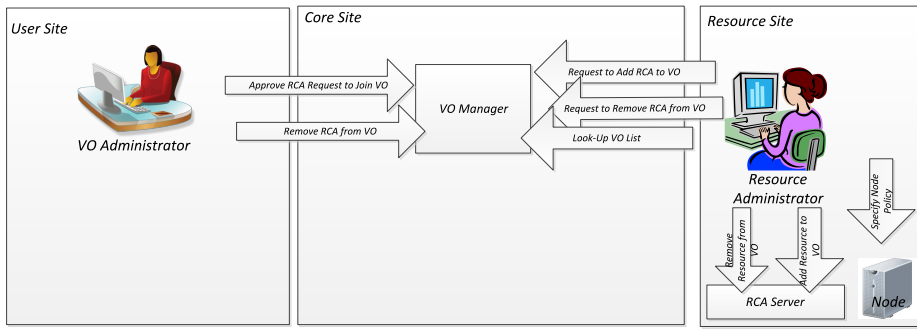

Figure 10. The VO evolution capabilities – resource management

4.3.6 Manager Look-up VO List

The first capability, similar to the case of users management above, allows a Resource Administrator whose RCA is registered to the Grid to look-up a list of the VOs currently available in that Grid. The request to obtain this list is submitted to the VO Manager component of XVOMS. Upon receipt of the list of available VOs, the Resource Administrator can then proceed to the next capability.

4.3.7 Requesting to Add RCA to VOs

After looking-up the list of available VOs, the Resource Administrator will decide on which VO to join the RCA. This is achieved by first submitting a request via the VO Web Frontend or the command line interface to the VO Manager system to join the selected VO. The preconditions to this capability are that the VO must exist and the RCA of the Resource Administrator must have been registered in the

Grid. Once this capability is completed, the request is queued waiting for the VO owner's approval.

4.3.8 Approving RCA Requests to Join VOs

The VO owner can view the list of pending requests from RCAs (their administrators) to join his/her VO with the local VO Manager's database. If approved by the owner, the RCA is then formally part of the VO and can start adding its resources to the VO.

4.3.9 Adding Resources to VOs

With this capability, a Resource Administrator can add their resource to a VO by interacting via some interface with the RCA Server API. The resource added must be capable of obtaining the VO machine attribute certificate issued by the RCA server, which itself is a member of the VO. Once this capability is successfully completed, Users can start utilising the resource in the operational phase of the VO lifecycle.

4.3.10 Removing Resources from VOs

Similar to removing users from a VO, a Resource Administrator can simply remove a resource from a VO, which is currently registered with that VO. As a precondition, the resource should not currently be running any VO-relevant computations or storing any VO-relevant data. Once this capability is executed, it will not be possible any more to obtain the VO machine attribute certificate of the resource and it will not be possible to utilise it in the context of the VO.

4.3.11 Requesting to Remove RCA from VOs

It is possible that at some stage the Resource Administrator may request from the VO Manager system to remove the RCA from a particular VO, either because the VO is no longer relevant or because the RCA cannot afford any more to offer resources to the VO Users. This request is then queued for approval by the VO owner.

4.3.12 Removing RCA from VOs

During this capability the VO owner will check the queue of pending requests by RCAs wishing to be removed from VOs. For any such RCAs, the VO owner may approve the request. This will result in the termination of the membership of the RCA to the specific VO. As a direct consequence, any resources belonging to the domain of the RCA cannot any more join the specific VO.

4.3.13 Specifying Node Policies

The final capability when managing resources is to specify individual node policies. A node policy determines the conditions and rules under which the node can be accessed and used by the VO Users. This capability is executed by the Resource Administrator, and as a result, the existing policy on the node is updated with the new VO policy.

Policy Management. The final set of capabilities is related to the management of VO Policies as shown in Figure 11.

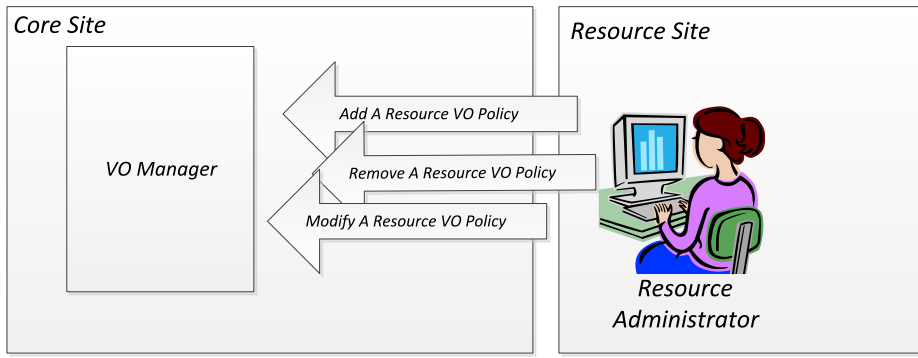

Figure 11. The VO evolution capabilities – policy management

4.3.14 Adding Resource VO Policies

The first capability is executed by the Resource Administrator on the VO Manager component of XVOMS to add a VO policy that will be recorded with the VO Policy Service (VOPS). This policy is at the VO level (enforced at the VO level) and is directly relevant to the resource managed by the Resource Administrator.

4.3.15 Removing Resource VO Policies

This capability allows the Resource Administrator to remove a resource VO policy from the VO Manager and the VOPS.

4.3.16 Modifying Resource VO Policies

This capability allows the Resource Administrator to modify a resource VO policy currently stored at the VO Manager and the VOPS.

4.4 VO Operation Capabilities

The previous sets of capabilities were concerned with setting-up the Grid and creating VOs, at this stage, the VO moves to its operational phase. This phase is represented by two capabilities, one for the Users and one for the RCAs.

VO Operation – User Capabilities. The first VO operation capability is related to the manner in which Users can operate within a XtreemOS VO. This is shown in Figure 12.

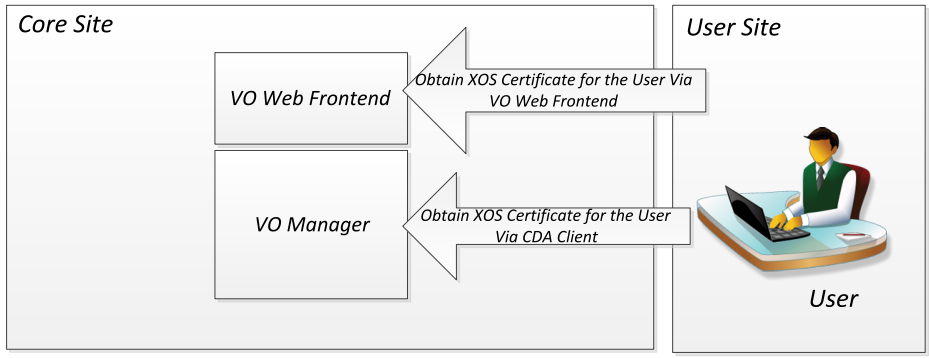

Figure 12. The VO operation capabilities – users

4.4.1 Obtaining XOS Certificates for the User

The first capability allows a user to obtain a XtreemOS VO certificate from the CDA server in order to be able to submit jobs to the VO resources. From this point onwards, the User is verifiably identifiable and can hence start secure communications with the VO resources. The users can obtain XOS certificates via the VO Web Frontend, or via the CDA client.

VO Operation – RCA Capabilities. The second capability is related to the operational phase of the VO RCAs. This is shown in Figure 13.

4.4.2 Obtaining Attribute Certificates for the Resource Administrator

The second capability here is to allow a Resource Administrator to obtain an attribute certificate for the RCA in his/her administrative domain. From this point onwards, any resources connected to the RCA and part of the VO can start accepting jobs from the VO Users. The Resource Administrator can obtain Attribute Certificates via the RCA Web Frontend, or via console commands.

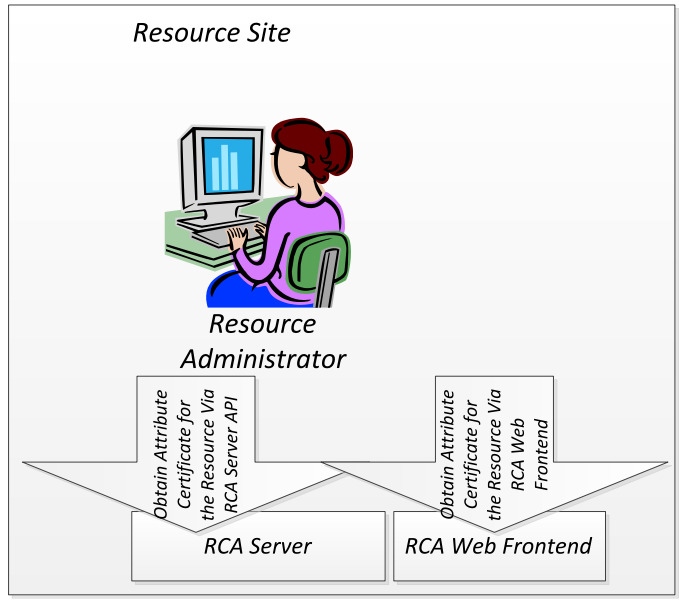

Figure 13. The VO operation capabilities – RCAs

4.5 VO Termination Capabilities

The final set of capabilities in the XtreemOS VO lifecycle are related to the termination of VOs, as shown in Figure 14. Since this is the last phase of the lifecycle, all of these capabilities focus on the deletion of the various elements of a VO and the cleaning-up of its final state.

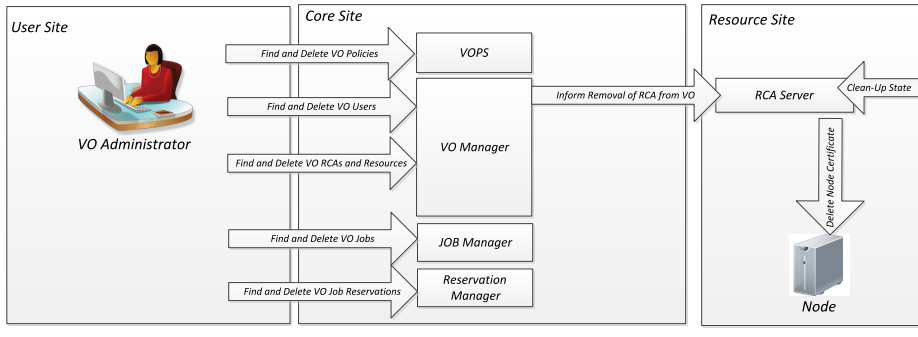

Figure 14. The VO termination capabilities

4.5.1 Finding and Deleting VO Policies

The first capability is related to the location and deletion of any VO policies that have been included on the VOPS server and that form part of the terminated VO. This capability is carried out by the VO owner, who is responsible for terminating his/her VO. Once this capability is carried out successfully, there will be no more VO policies related to the terminated VO.

4.5.2 Finding and Deleting VO Users

The VO owner executes this capability to locate and delete the membership of all the Users currently belonging to the terminated VO. The VO Manager component of XVOMS will be updated accordingly.

4.5.3 Finding and Deleting VO RCAs

The VO owner executes this capability to locate and delete the membership of all the RCAs currently belonging to the terminated VO. The VO Manager component of XVOMS will be updated accordingly.

4.5.4 Informing RCA of Removal from VO

Once the previous capability is successfully completed, the VO Manager will also inform the relevant RCAs of the decision to remove them from the terminated VO. As a result, the RCA will no longer offer any resources to Users of the terminated VO or any other activities.

4.5.5 Clean-up State

Once a RCA has been informed of the termination of a VO to which it belongs, it will execute this capability to clean-up its internal state. This will include removing the VO from its record of active VOs.

4.5.6 Deleting Attribute Certificates

In addition to cleaning-up its own internal state, the RCA will also be required to revoke (delete) any attribute certificates belonging to its resources that certify the membership of those resources in the VO. As a result of executing this capability, the relevant resources will no longer have any activities in the terminated VO.

4.5.7 Finding and Deleting VO Jobs

This capability is related to the location and deletion of all the current active jobs in the terminated VO. The VO owner executes this capability on a different XtreemOS service, called the Job Manager (part of the AEM services [2]).

4.5.8 Finding and Deleting VO Job Reservations

Finally, XtreemOS also facilitates the reservation of resource by prospective Users in a VO in advance of the actual job allocation and execution. Therefore, any such job reservations must also be removed from the VO during its termination phase. The capability, like the previous one, will also be executed by the VO owner on the Reservation Manager component (part of the AEM services [2]).

5 DISCUSSION AND CONCLUSION

The Grid operating system XtreemOS was developed with the aim of achieving Grid/OS integration and thus allowing maximum transparency for Grid applications as well as Grid-unaware system tools and applications to be used without being modified or even recompiled.

In this paper we presented the set of VO management and security capabilities underlying the XtreemOS. One of the main aims of defining these capabilities was to capture the VO management and security functionality requirements at an early stage of the software development cycle in XtreemOS.

The design of these capabilities was laid out with several criteria in mind:

Easy use and management. The high degree of automation in these capabilities by means of the use of various system components and the availability of API (through command line and Web-based GUI) facilitates an easier management task for the Grid and VO administrators.

Dynamicity of VOs. One of the main requirements behind the XtreemOS usage scenarios was to allow for a high degree of dynamicity in the structure of VOs. This implies the ability to change the composition of VOs during application runtime, e.g. if certain computing resources fail. In such circumstances, the unavailable resources need to be automatically substituted by alternative resources including also a migration of the affected running application components.

Integrated security and trust management. The incorporation of the various security-related (e.g. access control policy administration) and trust-related (e.g. the public-key infrastructure) requirements in these capabilities meant that security and trust formed an essential part of the early software development lifecycle, rather than as an after-thought addition to the system.

The above criteria and others were derived from a range of 14 applications [3] that formed the usage scenarios for the XtreemOS operating system. The resulting capabilities were implemented in terms of the XtreemOS VO management and security services outlined in Section 3.3.

REFERENCES

[1] Ceccanti, A.: The VOMS Architecture. 2008. `https://twiki.cnaf.infn.it/cgi-bin/twiki/view/VOMS/WebArchitecture`.

[2] XtreemOS Consortium: Design of the Architecture for Application Execution Management in XtreemOS. In XtreemOS public deliverables – D3.3.2. Work Package 3.3, May 2007.

[3] XtreemOS Consortium: Requirements Capture and Use Case Scenarios. Work Package 4.2, January 2007.

[4] XtreemOS Consortium: Third Prototype Implementation of Security and VO Management Services. In XtreemOS public deliverables – D3.5.16, Work Package 3.5, May 2010.

[5] Coppola, M.—Jégou, Y.—Matthews, B.—Morin, Ch.—Prieto, L. P.—Sánchez, O. D.—Yang, E.—Yu, H.: Virtual Organization Support within a Grid-Wide Operating System. IEEE Internet Computing, Vol. 12, 2008, No. 2, pp. 20–28.

[6] Cortes, T.—Franke, C.—Jégou, Y.—Kielmann, Th.—Laforenza, D.—Matthews, B.—Morin, Ch.—Prieto, L. P.—Reinefeld, A.: XtreemOS: A Vision for a Grid Operating System. XtreemOS Technical Report # 4, May 2008.

[7] Firesmith, D. G.: Security Use Cases. Journal of Object Technology, Vol. 2, 2003, No. 3, pp. 53–64.

[8] Foster, I.—Kesselman, C.—Tsudik, G.—Tuecke, S.: A Security Architecture for Computation Grids. In Conference on Computer and Communication Security, 2001.

[9] Foster, I.—Kesselman, C.—Tuecke, S.: The Anatomy of the Grid Enabling Scalable Virtual Organizations. In International J. Supercomputer Applications, 2001.

[10] Housley, R.—Polk, W.—Ford, W.—Solo, D.: RFC 3280 – Internet x.509 Public key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. April 2002.

[11] Morin, Ch.: XtreemOS: A Grid Operating System Making your Computer Ready for Participating in Virtual Organizations. In Proceedings of the Tenth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2007), pp. 393–402. IEEE Computer Society, 2007.

[12] Pearlman, L.—Welch, V.—Foster, I.—Kesselman, C.—Tuecke, S.: A Community Authorization Service for Group Collaboration. In Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY '02), POLICY '02, pp. 50, Washington, DC, USA, 2002, IEEE Computer Society.

[13] Rosado, D. G.—Fernandez-Medina, E.—Lopez, J.: Reusable Security Use Cases for Mobile Grid Environments. In Software Engineering for Secure Systems, 2009, SESS '09. ICSE Workshop, pp. 1–8, may 2009.

[14] Kabasele Tenday, J.-M.: Using Special Use Cases for Security in the Software Development Life Cycle. In Proceedings of the 11th international conference on Information security applications, WISA '10, pp. 122–134, Berlin, Heidelberg, 2011, Springer-Verlag.

[15] XtreemOS Consortium: Requirement Documentation and Architecture for XtreemFS. In Felix Hupfeld, editor, XtreemOS public deliverables – D3.4.1, Work Package 3.4, November 2006.

Benjamin Aziz is a Senior Lecturer in computer security at the School of Computing, University of Portsmouth. He holds Ph. D. degree in formal verification of computer security from Dublin City University (2003) and has research experience in the field of computer and information security spanning 15 years where he worked in the past at Rutherford Appleton Laboratory and Imperial College London, and has published more than 70 articles and book chapters in areas related to the security of large-scale systems, formal security, requirements engineering and digital forensics. He is on board program committees for several conferences and working groups, such as ERCIM's FMICS, STM, Cloud Security Alliance and IFIP WG11.3.

Ioana Sporea obtained her Ph. D. from the University of Surrey in 2012 in the area of supervised learning algorithms for networks of spiking neurons and developing new learning procedures for feed-forward networks by combining different forms of plasticity with a well-known learning paradigm of gradient descent. She has been working as a Research Assistant since 2012 at the School of Computing, University of Portsmouth, under the supervision of Dr. Benjamin Aziz in the area of computer security and digital forensics.