Project no. FP6-033576

XtreemOS

Integrated Project
BUILDING AND PROMOTING A LINUX-BASED OPERATING SYSTEM TO SUPPORT VIRTUAL ORGANIZATIONS FOR NEXT GENERATION GRIDS

Security Requirements for a Grid-based OS

D3.5.2

Due date: November 30th, 2006
Actual date of submission: January 11th, 2007

Start date of project: June 1st 2006
Type: Deliverable
WP number: WP3.5
Task number: T3.5.1

Responsible institution: CCLRC
Editor & and editor's address: Ian Johnson
CCLRC RAL
Chilton OX11 0QX
UK

Version 1.2/ Last edited by Ian Johnson / 11th January 2007

Project co-funded by the European Commission within the Sixth Framework Programme		
Dissemination Level		
PU	Public	✓
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Revision History:

Version	Date	Authors	Sections Affected / Comments
1.0	17/11/2006	CCLRC, SAP, XLAB, ICT, ULM, INRIA	All sections
1.1	30/11/2006	CCLRC, SAP, XLAB, ICT, ULM, INRIA	All sections
1.2	11/01/2007	CCLRC	Front page – date of last edits and submission

Table of Contents

1 Executive Summary

In this document we present the work of WP3.5 and related work packages in obtaining the initial security requirements for XtreemOS and defining the security objectives that will satisfy the requirements.

We acknowledge that XtreemOS faces certain hurdles if it is to be widely adopted in the Grid community. These barriers to adoption include those of distribution and acceptance in the Linux community, the task of developing applications using new Grid APIs, and of securely integrating applications running on XtreemOS into an existing computing environment.

These hurdles are being addressed in the following ways:

- WP2.1 has, as one of its aims, encouraging the Linux development community to accept and use the XtreemOS extensions to Linux.
- WP3.1, Grid APIs is proposing the use of the SAGA [6] Application Programming Interface to allow new and existing developers to develop Grid applications running on top of XtreemOS.
- In WP3.5, we will provide a specification and implementation of a set of security services to allow XtreemOS applications to operate securely.

WP3.5 aims to provide a high level of assurance to potential XtreemOS users that the security services available to them are more than adequate for their purposes.

2 Introduction

The XtreemOS project is concerned with providing a highly-scalable, available and secure platform for grid computing, supporting Virtual Organizations (VOs) that span multiple machines and multiple administrative domains. This platform will take the form of extensions to the open source Linux operating system to natively support VOs, and will be able to run on hardware ranging from ambient devices (such as Personal Digital Assistants or high-end smartphones), to single PCs and clusters of PCs. The XtreemOS system is composed of two parts: XtreemOS foundation, the so-called XtreemOS-F, the modified Linux system embedding VO support mechanisms; and XtreemOS high-level services, the so-called XtreemOS-G, implemented on top of XtreemOS-F and offering a common infrastructure for highly available and scalable service, including services for security, data and application management

This document presents the initial requirements for securing XtreemOS-G. This has been a collective work carried out by all partners involved in WP3.5 (Security Services in VOs) in task T3.5.1, with a strong interaction with partners from other work packages, particularly WP2.1 (VO Support in Linux) and WP4.2 (Applications, Experiments and Evaluations).

2.1 The Importance of Security Services in XtreemOS

Having a secure and reliable system to support Virtual Organizations is crucial to the adoption of XtreemOS. Without such assurances, the system is unlikely to be accepted outside of the existing application and testbed owners in the project consortium.

Work Package 3.5 of XtreemOS, Security Services in Virtual Organizations, has the following overall challenges:

- Production of security services for XtreemOS that will detect and prevent unauthorized actions by users;
- Successful integration of all security services at all levels (application, XtreemOS-G, XtreemOS-F)
- Ensuring that there is no compromise of the security or functionality of the underlying Linux system.

The approach taken to start meeting these challenges is to define milestone decisions that drive the design and implementation of the security services in XtreemOS. The members of WP3.5 have carried out a task, T3.5.1, to define the security objectives and security requirements of a Grid-based OS, which we present in the following sections. This document, D3.5.2, is being used in task T3.5.2, the definition of the initial set of security services for XtreemOS.

2.2 Process followed to create this deliverable

This deliverable represents on of the outputs of the members of Work Package 3.5, coordinated with WP2.1 on definitions of Virtual Organizations and their requirements. WP4.2 provided a substantial input to this task, and the security requirements listed in section 4 are taken from D4.2.1. WP3.2 provided security requirements concerning Highly Available and Scalable Grid Services.

2.3 Document Structure

This document is structured as follows:

Section 1 is an Executive Summary.

Section 2 is an Introduction.

Section 3 considers some fundamental security objectives in maintaining the confidentiality, integrity and availability of a distributed system, and also puts these security considerations within the business environment, setting out the assumptions we are using about the business drivers and technical infrastructure of this environment.

Section 4 presents work derived from the parallel task of WP4.2 which relates to security infrastructure, which have been analysed, consolidated and refined into a common set of security requirements.

Section 5 presents the Virtual Organization Lifecycle and considers the security requirements that are of interest at each stage of the lifecycle which need to be taken into account.

In section 6 we consider the requirements for the use of the concept of Trust and Reputation to monitor and regulate the relationships between entities in the Virtual Organization.

We will provide section 7, Conclusion, with the revised version of this document.

Section 8 provides References.

Annex A presents the requirements identified by WP3.2, Highly Available and Scalable Grid Services. Annex A presents some conclusions as they apply to the specific security concerns of WP3.2

3 Security Objectives and Considerations

This section defines the main security objectives to be taken into account in XtreemOS as well as considerations/assumptions to be considered in the project.

3.1 Basic Security Objectives

3.1.1 Confidentiality

Confidentiality has been defined by the International Organization for Standardization (ISO) as "ensuring that information is accessible only to those authorized to have access" and is one of the cornerstones of Information security. The data being processed in a Grid may be subject to considerable confidentiality constraints, either due to privacy concerns or issues of intellectual property. For instance, the use of Grids implies that confidential data is stored in online accessible databases; therefore access to their interfaces must be carefully controlled, both to allow access only to appropriate users, and also to allow queries and simulations to run over these highly confidential data without that data being compromised or revealed.

3.1.2 Integrity

Integrity refers to assurance that the information is authentic and complete, ensuring that information can be relied upon to be sufficiently accurate for its purpose. In Information Security, the term Integrity is used frequently to represent one of the primary indicators of security (or lack of it). The integrity of data is not only whether the data is 'correct', but whether it can be trusted and relied upon. For example, making copies (say by e-mailing a file) of a sensitive document, threatens both confidentiality and the integrity of the information. This is because, by making one or more copies, the data is then at risk of change or modification.

3.1.3 Availability

Availability refers to the ability to use the information or resource desired. Availability is an important aspect of reliability as well as of system design because an available system is at least as bad as not system at all. The aspect of availability that is relevant to security is that someone may deliberately arrange to deny access to data or to a service by making it unavailable. System designs usually assume a statistical model to analyze expected pattern of use, and mechanisms ensure availability when that statistical model holds.

Attempts to block availability, called *Denial of Service (DoS) attacks*, can be the most difficult to detect, because the analyst must determine if the unusual access patterns are attributed to deliberate manipulation of resources or of environment. Complicating this determination is the nature of statistical models. Even if the model accurately describes the environment, atypical events simply contribute to the nature of the statistics. A deliberate attempt to make a resource unavailable may simply look like, or be, an atypical event. In some open environments, like Grids, it may not even appear atypical.

3.2 Security Considerations

To effectively reason about the security of a system, we need to set down some assumptions and expectations. These refer to the operational properties and conditions of the security system (model, services, architecture and topology), its targets, its users, its administration and its environment. It can then be said that in order for the security system to be effectively secure, these particular properties and conditions must be constantly in place. This section defines an initial list of assumptions and expectations that we identify as the basic properties and conditions of the models, services, architecture and topology of an XtreemOS-supported Grid infrastructure, which supports multiple Virtual Organizations and membership by various administrative domains. These are placed in four categories; (i) the Corporate Assumptions: what minimal contractual agreements and affiliations are assumed in place between organizations and individuals, (ii) the System and Infrastructure Assumptions: the types of distributed security services and trust relationships that are assumed to be available, (iii) the Technical Assumptions: the assumptions of the nodes where XtreemOS is installed and (iv) Administrators and Users: the assumptions concerning the behaviour and capabilities of administrators and users of distributed resources in the Grid infrastructure. For each of these set of assumptions, their impact on security models, services, architecture and topology are referred to.

3.2.1 Corporate Assumptions

A Grid is a network of multiple computational nodes and distributed resources. Each node and resource however has a human or organization as their legal owner and responsible if they fail. Especially in environments where accounting and charging are included, if the nodes or resources behave in a way that cause the quality and performance of the Grid to drop, then there need to be means of compensating and coordinating the transfer of responsibility. There is therefore a need for the humans and organizations in the Grid to have high level agreements concerning conduct, compensation and coordination. While it is also possible for contracts to be specified in the form of machine interpretable service level agreements (SLAs) and protocols, these are beyond the scope of the features of the XtreemOS. Nevertheless, the assumption that the relevant contracts, service level agreements and protocols are specified and enforceable, even if not at the OS level, must hold. Contracts, agreements and protocols will be referred to as agreements collectively. There are three different types of agreements that are assumed in a Grid environment, discussed in the following subsections. These are Grid-Wide, VO-wide and bipartite agreements, indicating a narrowing of scope of the agreements.

3.2.1.1 Grid-Wide Membership Agreements

Grid-wide membership agreements are the base conditions that any member in the Grid must agree to before providing resources in the Grid or receiving access to other resources. Members in the Grid can therefore assume that any other member has agreed to particular conduct, compensation and coordination arrangements. It is assumed that there is a system of doing background checks and proofing credentials before applicant registrants in the Grid are accepted. For example, a node or resource belonging to a known hacker ring or a blacklisted organization would not typically be accepted into a Grid that promises a high level of integrity. There is therefore some expectation of human-level screening and reputation checks of members that join the Grid. This can be potentially semi-automated, given that there is a means of maintaining and proofing lists of trust parameters and membership prerequisites. However, the topic of automated blacklisting and online reputation opens up a set of

possible attacks that are beyond the scope of the mechanisms intended for the OS. It is assumed that there is a governing body in place for the global Grid, such that registration is centrally supported and regulated.

3.2.1.2 Virtual Organization (VO-Wide) Membership Agreements

Virtual Organization (VO) Membership agreements are created as a means of isolating a selected subset of members from the overall Grid. VO wide agreements are still dominated by the Grid-wide agreement and cannot override these terms and conditions. Therefore, VOs formed within a particular Grid, using its agreed namespace, services and infrastructure must be compliant with its regulations. It should not be possible for VOs to be formed with participants that have not passed the basic qualification for entry in the Grid infrastructure. However, in some instances, the Grid-wide agreement may be open and public, in a similar manner that the Internet is. Nevertheless, we would assume that this is not often the case within the context of XtreemOS, as a result of the nature of applications identified. The ability to enforce and validate membership in the Grid and in VOs is therefore both an assumption and a functional requirement.

3.2.1.3 Bilateral Contracts, Agreements and Protocols

In addition to the VO-wide agreements between subsets of participants, it is also possible that individuals and organizations that have joined a Grid or VO may have had relationships with each other beforehand. Secondly, they may be competitors or have conflicts of interest, such that their agreements need to be bilateral in order to protect their interests. Bilateral or bipartite agreements are also dominated by the VO-wide agreement within which they occur. If there is an agreement outside of the VO, then it is dominated by the Grid-wide agreement. Otherwise it is not of concern for the Grid context and not treated as an assumption or requirement.

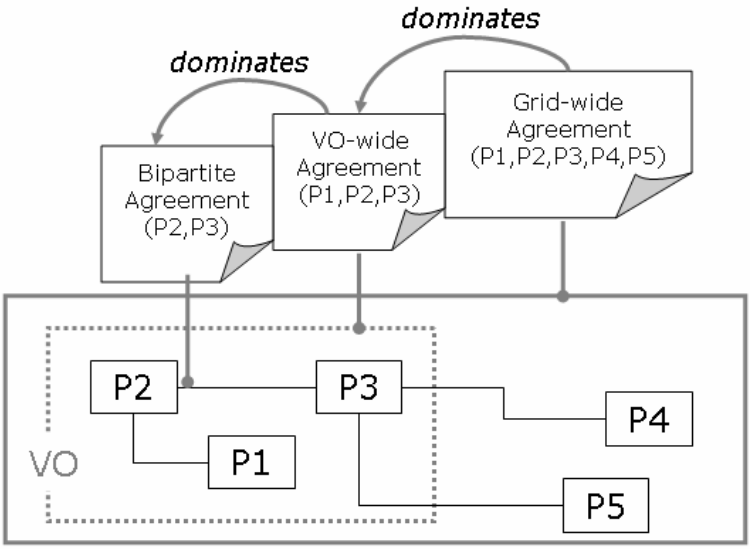

Figure 1. Summary of Corporate Assumptions concerning agreements between participants in a Grid and VOs

The corporate assumptions assert the following:

- There are one or more organizations that are responsible for membership in a Grid infrastructure – they sanction the initial membership of entities in the Grid and provide offline, online or inline validation of their membership
- These authorities are trusted within

- Any VO also has an authority or manager that provides a server for registration and membership in a VO
- Organizations or individuals may be members of multiple VOs, given that they are registered members with the overall Grid infrastructure
- Contracts, agreements and reputations do not need to be understood explicitly by the OS, but it is assumed that there are services that provide proof and assertions of the status of these to OS-level services

3.2.2 System and Infrastructure Assumptions

Certain capabilities and principles are also assumed at the system and infrastructure level of the Grid, which impact on the security requirements to be implemented by XtreemOS. This section presents these by considering: (i) Security models that may be assumed, (ii) basic security principles that should be followed and (iii) general infrastructure services and trust relationships that can be assumed in the environment where XtreemOS nodes are deployed.

3.2.2.1 Security Models

A security model provides a simplified, formal definition of what safety means within a complex system. Within XtreemOS, it is assumed that both Mandatory and Discretionary models of security will define the safety of the supported applications and systems.

Mandatory security is enforced by the system and cannot be altered by users or resources owners. For example, it is assumed that communication between members of the Grid, members of a VO and participants in a bipartite agreement is encrypted with a secret key shared within the boundaries of the agreement, establishing security associations. It must therefore be possible to establish and distribute such keys and associations securely. Secondly, mandatory security assumes that data can be labeled throughout its lifetime. It is assumed that messages can at least be identified as originating from a member of the Grid or from a member of a VO.

Discretionary security allows the owners of resources to specify and alter security policies, changing permissions and parameters of access control to resources, network availability and encryption of stored and transmitted data. It should be possible for participants in the Grid and in VOs to maintain autonomous control over their resources.

3.2.2.2 Security Principles

In addition to the security models, there are some security principles that are fundamental to ensuring a secure operational environment. The principles were first established by Saltzer and Schroeder in 1975 [5], and have been applied and extended in many systems. These principles should also be used within the XtreemOS security mechanisms and assumed of the design of applications. The provision of utilities that test and verify that these principles have been followed is however beyond the scope of XtreemOS. The principles taken from Saltzer and Schroeder are:

1. Principle of economy of mechanism: the security services developed for XtreemOS should be simple (i.e. simplest possible interface, state-machine, parameters and exceptions) and not have a large, complex code-base. This reduces the risk of introducing vulnerability and provides a simple trusted computing base.

2. Principle of fail-safe defaults: the security services should deny access by default, and grant access only when explicit permission exists. This again follows from the two security models (mandatory and discretionary), where the mandatory security model will block any access without the appropriate policy and credential mapping, while the discretionary security model allows these policy and credential mappings to be explicitly specified.

3. Principle of complete mediation: the security services should be capable of checking every access to every object.

4. Principle of open design: the algorithms and protocols used for the security services should be made available to the general public, without attempt to obfuscate their inner workings. We assume that the security services developed in XtreemOS will either draw from openly available specifications, or will endeavor to make our developments publicly available. The non-publication of passwords and cipher keys are not included in the list or information to be made available.

5. Principle of separation of privilege: the security services should grant access based on more than one piece of information. For example, a subject should not be allowed access to a resource or a transmission, just because they are a member of the Grid or a member of the VO. The credentials and keys that have been specified within Grid, VO and bipartite agreements must be dissimilar.

6. Principle of least privilege: the security services should force every process to operate with the minimum privileges needed to perform its task. Subjects or nodes should not gain access to resources unless there is some proof that there is a job pending that requires this access.

7. Principle of least common mechanism: the security services should be shared as little as possible among users. Although this is not always a practical or consistent goal for Grid environments, it should be noted that having multiple users sharing the same security service is a point of weakness in the architecture, for which compensatory mechanisms need to be developed.

8. Principle of psychological acceptability: the security services should be easy to use (at least as easy as not using it). This is in line with the high level goals of XtreemOS, where the provision of Grid and VO management support in the OS should provide transparency to applications.

3.2.2.3 Trust Relationships

Trust in XtreemOS is the degree of belief that a node can have in another node behaving according to agreed protocols. For security services, this includes the response to given events, issuing, validation and revocation of credentials, provision of access, usage of correct keys and algorithms for encryption, and the availability of resources for performing security services. There are three types of providers of security services assumed in the Grid infrastructure, as depicted in various papers:

- Authorities: these are also known as trusted third parties, and are available in the Grid for issuing and validating credentials and attributes of dif-

ferent participants in the Grid. These credentials and attributes are typically concerned with the global identity and reliability of services of other participants.
- Managers: this role is played by nodes that maintain membership databases to VOs.
- Member: this role is played by nodes that seek to be

Following these three operational roles, there are tree types of trust relationships:

- Mandatory trust: between authorities and participants in the Grid. If they want to participate in the Grid, they must trust at least one of prevailing authorities. The authority must also trust participants, having issued them with credentials, until otherwise evidenced
- Transitive trust: when a participant becomes a member of a VO, the
- Discretionary trust: members may choose to deny access to their private resources to other members, without sufficient evidence of their trustworthiness, even if they are members of the same VO

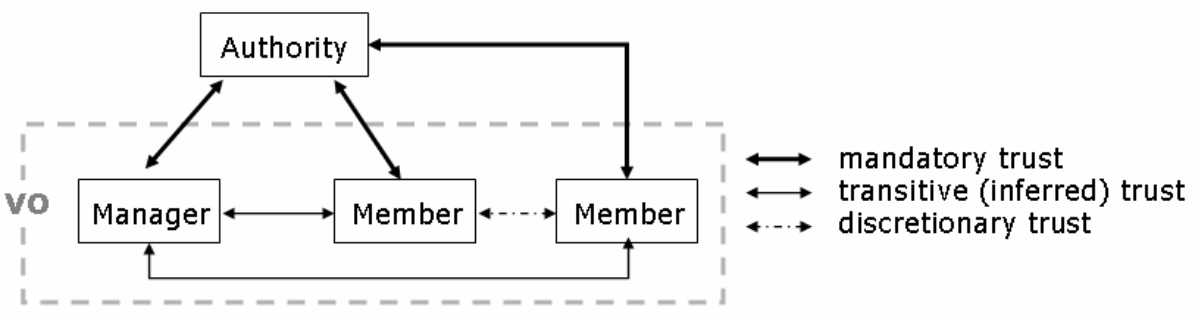

Figure 2. Trust relationships in a Grid

Having these system and infrastructure assumptions the following are asserted:

1. Security services provided by XtreemOS will enforce mandatory security policies identified as basic requirements that span applications
2. Participants can maintain their own local security policies even when in VOs, controlling access to their resources
3. Established security principles are known and observed by the implementers of security services
4. Established trust models and infrastructure services in Grid environments will be used as a basis

3.2.3 Technical Assumptions

This part describes technical properties of individual and collective nodes in the Grid. These follow distributed computing concepts. In general this requires a fine grained access control for subjects in different roles or functions and objects. However in a VO shared resources from different administrative and trusted domains are interconnected. This could make it more difficult to mediate access from subjects to objects.

Therefore it is assumed that a mix of security mechanisms is applied in distributed OS implementing a VO:

- Predefined access control mechanisms mediating access from subjects to objects. These mechanisms work well in predefined environments and prevent unauthorized access.

- Accountability mechanisms which document access from subject to objects in a faithful and integrity protected way. They allow the detection of access to objects which could not be explicitly coded in a machine readable policy.

3.2.3.1 Processors

Data in the VO is processed by the CPUs of the VO resources. It is assumed that the predominant amount of processors is used for general purposes and is not dedicated for security critical tasks. Every CPU can process security relevant data, however they might not all provide trusted security modules.

It is further assumed that a VO resource can have more than one CPU and provides sufficient performance to execute the application, the operating system and to perform necessary encryption.

3.2.3.2 Storage and Memory

Data in the VO can be stored on persistent and temporary storage of VO resources. It is assumed that, unless every VO resource can prove the existence of a standardized reference monitor mediating access from subjects to objects, data can accessed only by its owner. It is assumed that users or services are protecting confidential data with additional cryptographic mechanisms.

It is further assumed that a VO resource provides sufficient memory and storage to execute the applications and the operating system. However some VO resources can use a storage infrastructure with higher reliability, backup and recovery features. They can provide additional information about persistent storage in order to let other applications benefit from these properties.

It is assumed that inter process communication such as shared memory, pipes and semaphores work in a distributed environment with the same properties.

3.2.3.3 Communications

The establishment of a VO requires communication. Even a VO consists of trusted VO resources, they can be locally distributed and the communication has to take place via open networks. Because it is very difficult to detect an eavesdropper it is assumed the network is insecure and confidential data can only be transmitted using security protocols.

This also applies to the integrity of data in transit. Because it is very difficult to detect ‘man in the middle’ attackers in the network (that is, attackers who eavesdrop messages and forward them in a different form, or who replay messages), it is assumed that the network is insecure.

3.2.3.4 Clocks

The time of storing and accessing information is important information. Many operations are based on exact time information and take that data into account. Time in log information is also very important from an accountability point of view to uniquely re-

construct a series of events. It is assumed that on all VO resources a synchronized time service is available.

The synchronization of distributed entities of either a service or a XtreemOS resource is assumed to be available by a dedicated service.

3.2.4 Administrators and Users

There are also minimal assumptions made concerning administrators and users in a VO implemented by XtreemOS.

3.2.4.1 Administrative Capability and Behaviour

Administrators act responsibly and do not misuse their authority and privileged access to resources and information. Administrators should, however, not have read or write access to keys and passwords belonging to users. They may replace or delete users' keys or passwords without the content being revealed.

Other assumptions about administrators include:

- A skilled administrator knows how to organize the rights and roles of users/principals and objects. This covers assigning subjects and resources to groups, specifying policies for groups, assigning privileges to roles.
- An administrator maintains and monitors security associations between VO resources. He audits security logs and detects internal access violations and abnormal network activities and responds appropriately to alerts.
- Any principal holding administrative roles can be identified and the role in which he is acting can be validated.
- There is no overall XtreemOS superuser or administrator; administrators having superuser privileges only in their local domain and over a subset of local resources.
- Administrative privileges may be hierarchical and delegated.
- Administrative privileges may be delegated beyond a local administrative domain, only if there is a VO specified.

3.2.4.2 User Capability and Behaviour

Users cannot achieve super user or administrative access to nodes, unless they are owners of the node. It is assumed that:

- A user of a service or application in traditional computing environment does not need a significant amount of training to use services or applications in XtreemOS.
- A user knows about the responsibilities of different VO resource administrators.
- A user only uses those objects and security mechanisms that affect his role (e.g. single sign on, role based access control). A user does not need to know about the security infrastructure that is required by the distributed character of XtreemOS.

4 Security Requirements for VOs in XtreemOS following from Application Requirements

Here we present the results of one of the major tasks that have provided input to this deliverable, the WP4.2 task on elicitating requirements. This took the form of creating a questionnaire for the application owners to express their requirements in several areas, including those of security. Given these security requirements, the WP4.2 team consolidated them in the tables which follow below.

Traditional security objectives of operating systems apply to a grid based operating system as well. However the achievement of these objectives can get more difficult because of the distributed grid character. A proper function requires a common namespace for users and data as well as the same understanding of rights and policies. Of special interest in a distributed environment are synchronization issues. Inconsistencies can harm confidentiality and integrity of stored data and can also affect accountability. The objectives of concern are stored data as well as communicated data.

- Confidentiality of stored data
- Confidentiality of communicated data
- Integrity of stored data
- Integrity of communicated data
- Identification and authentication of users
- Authorized access to application services
- Guaranteed access to application services by authorized parties
- Accountability of data access and service execution
- Isolation of data per-VO
- Isolation of services per-VO

The methodology applied consisted in taking the empirical security requirements by WP4.2 and refine them in a top-down manner. The more detailed and technically relevant requirements can be used to select security concepts and mechanisms implemented in a security service.

The identified assets during VO operation are:

- User identifiers, user rights
- Data
- Processes

In order to achieve a secure operation of a VO the security objectives must be achieved **at every point in time**. This especially applies to an operating system which has distributed resources.

Note that this is a top-down refinement of requirements of the WP4.2 questionnaire. In addition the requirements of operating systems apply too. The requirement numbers RXX are from deliverable D4.2.1 of WP4.2.

R78 Data stored on resources must only be accessible by users and administrators that are members of a VO with the appropriate access rights

Confidentiality is a fundamental requirement of systems that store, process and exchange sensitive data and information. In a Grid-enabled system the requirement for confidentiality of stored data is to ensure that data can only be accessed and read by services, users and administrators (together known as Principals) that have a the appropriate right to the data. Rights are usually issued on the principle of least privilege to perform the intended task. A principal has a need if the following conditions are true: owner of the data OR registered as a member of a VO with rights to the data AND assigned to a task that requires access to the data. A principal with such properties is referred to as a "valid principal" otherwise we refer to the principal as an "invalid principal".

ASSETS:

This affects all locations where data can be addressed in the VO:

1. Filesystem
   - Data on the local filesystem.
   - Data on a shared or grid filesystem.
2. Shared memory
3. License information

REFINED SECURITY REQUIREMENTS:

a. There is a common namespace for identifying and addressing data.
b. There is a common namespace for users and access is only granted to authorized users. This implies that there is an Identity infrastructure / Single Sign-On
c. There is a common understanding for access rights on all resources in the VO.
d. A reference monitor regulating access is present at every resource.

DEPENDENCIES:

This affects Single Sign-on.
This affects reference monitors to achieve data integrity and controlling access to data.
This affects accountability of the usage of data.

SECURITY SERVICE:

- As this is a basic security requirement a mechanism has to be integrated or added to existing reference monitors for a distributed context.

R79 Confidential data communicated between resources in the same VO must be encrypted using cryptography protocols and keys agreed to within the VO

It is assumed that data is transmitted over insecure channels such as the Internet, such that there is a need for mechanisms that protect messages and responses in transit between computational nodes. More specifically, confidentiality of data is concerned with the protection of message inputs and the corresponding outputs of responses from invalid observers. However, the security policies and mechanisms are now concerned with the properties of the channels over which messages and responses are transmitted.

ASSETS:

Confidential data in this context covers user and process data communicated between VO resources. It also covers data which can influence these in an indirect way or can have negative consequences to services executed by the VO. This also includes data which is necessary to operate the VO itself.

The notion of confidential data here is refined to:

- User and process data transmitted by the OS from one resource to another
- OS specific data (e.g. synchronization information, resource locks)

Note: This covers all data in which a user of a VO is not able to select and apply his own security preferences, i.e. MAC as opposed to DAC.

To achieve a confidential communication the inter-resource communication of the VO has to take place via confidential channels or has to be encrypted.

REFINED SECURITY REQUIREMENTS:

- A protocol to set up a confidential channel should also support invalidation of existing channels
- An encryption scheme which implements a confidential channel should support messages with different message size. It is assumed that small and big messages are exchanged between VO resources and should be transported effectively.
- The sending resource must know all resources and their credentials to which a message has to be sent to.

REFINED REQUIREMENTS:

- Easy management of security associations covering communication confidentiality as well as integrity.
- The used encryption scheme should need as less as possible computational power and increase the network latency as less as possible.

DEPENDENCIES:

This requirement affects VO management procedures.

R80 Loss of integrity of stored data must be preventable and detectable

Storage integrity is typically stated as the ability to prevent illegal changes to data. As data in a VO may be sourced from different participants, who may not be “owners” of the data, it must be possible to guarantee or validate that the data has not been altered by illegal parties. A legal party must be a member of a VO and have the appropriate rights to make changes to data. Data should be hashed and digitally signed by a trusted key stored on the operating system.

ASSETS:

Data stored persistently as well as temporarily at locations other than the users/principal trust domain. This requirement applies also to data belonging a user/principal in his own trust domain but with additional users of other trust domains is granted access to.

REFINED SECURITY REQUIREMENTS:

- There is a common namespace for identifying and addressing data.
- There is a common namespace for users and a user requesting access to an object is known at the resource. This implies the existence of an identity infrastructure and Single Sign On.
- There is a common understanding for access rights on all resources in the VO.
- Integrity of stored data must be ensured by access control/reference monitors with specified properties. Rights are specifying if a principal is granted to e.g. write, append or change data.
- If it can not be ensured that a valid reference monitor controls access to remotely stored data, data should not be transferred to these locations.
- For data with very high requirements according to integrity, digital signature mechanisms should be applied (document fingerprinting and signing).

DEPENDENCIES:

This affects reference monitors for data ensuring confidentiality.
This requires a same namespace for users and rights.
This may affect the filesystem.

SECURITY SERVICE:

- As this is a basic security requirement a mechanism has to be integrated or added to existing reference monitors for a distributed context.

R81 The integrity of data transferred between resources or received from users must be validated before being committed

There is then a need for a OS reference monitor mechanism to capture and validate all incoming and outgoing network traffic. The integrity of communicated data is concerned with ensuring that illegal change is not possible to data in transit. This differs from data in storage as the properties of communication channels tend to be more dynamic, based on the location, operating system and medium used by end point nodes. The operating system must therefore be capable of signing and verifying signatures of data in an end-to-end manner. The term “committed” suggests that a transaction framework is necessary, considering the distributed nature of the resources.

ASSETS:

Data in this context covers user and process data communicated between VO resources. It also covers data which can influence these in an indirect way or can have negative consequences to services executed by the VO. This also includes data which is necessary to operate the VO itself.

The notion of data which is communicated here is refined to:

- User and process data transmitted by the OS from one resource to another
- OS specific data (e.g. synchronization information, resource locks)

Note: This covers all data in which a user/principal of a VO is not able to select and apply his own security preferences.

To ensure the integrity of communication the inter-resource communication of the VO has to be processed in order to add additional information about checksums and authenticity.

REFINED SECURITY REQUIREMENTS:

- Data which have to be transmitted from on resource to another has to be preprocessed by the sender in order to add information about its authenticity and its integrity.
- Data which is received by a resource has the duty to check the integrity and authenticity of received data.
- The receiving resource has the duty to detect replays of an authentic message in order to prevent denial of service attacks to the VO.
- The receiving VO resource must know all qualified senders and their credentials of a message in order to verify a message in transit as authentic.

REFINED RQUIREMENTS:

- Easy management of security associations covering communication confidentiality as well as integrity.

DEPENDENCIES:

This affects the confidential transport of messages between VO resources and VO management procedures.

SECURITY SERVICE:

- Security protocol ensuring confidentiality and integrity (protocols IPSEC, TLS)

R82 It should be possible for a user to use a single method of authentication (i.e. single sign-on) to gain authorized access to resources in a VO

Identification and authentication are again fundamental requirements for security, as integrity and confidentiality are difficult without the capability to identify and authenticate principals. Identification ensures that different principals (e.g. a source or receiver of a message) are repeatedly distinguishable from each other, while authentication associates attributes used to identify a principal with a unique root attribute such as a legal name or public key. It must be possible for all resources in a VO to identify and authenticate users requesting access to data. Users of resources should not have to be bothered with changing the way they interact due to changes in the hosting of the resource. One example is cross-domain single-sign-on (SSO), which requires an agreement of how tickets and attributes are encoded and verified, which assert that users have been authenticated and possess the appropriate authorizations to perform actions in the VO.

ASSETS:

User identifier within a common namespace for user identities. Authentication credentials/secrets.

REFINED SECURITY REQUIREMENTS:

- Provide unique user identifiers in a VO.
- Provide unique user identifiers among different VOs.
- Support roles? Provide rights and authorizations?
- It mustn't be possible for a user to change his user identifier without a valid authentication.

REFINED REQUIREMENTS:

- Support to consolidate existing user identifiers from different trust domains.
- Allows to interact with existing system user authentication frameworks in Kerrighed/Linux

DEPENDENCIES:

This affects rights and authorization management.
This affects reference monitors enforcing access control (e.g. confidentiality & integrity).

R83 It must be possible to transfer and validate authorizations to virtualized resources when the host is changed

Authorization requirements precede confidentiality and integrity requirements and depend on identification and authentication. Authorization is the requirement that principals can only access data that they must use in order to perform tasks, or, in the case of multilevel security systems, that they have the requisite clearance in the system. The indication of a principal's rights to perform a task is usually indicated using a token, ticket or credentials, which are different forms of associating an identity with a specific right. This follows from R3.5.5, as the authorizations should also be consistent across the domains providing resources. However, the challenge is still making sure that the local administrators of hosts do not have to breach the private policies enforced by their operating systems.

ASSETS:

Assets for this requirement are authorizations regulating access for users/principals and processes. In the user context it can cover rights and roles. In the context of a process it covers the rights and roles on behalf the process is acting for as well as license information necessary for the specific process to run.

REFINED SECURITY REQUIREMENTS:

- The amount of granted credentials must not leave or create credentials which are not in accordance to the VO policy.
- Migrated credentials from resource A to resource B have to be invalidated on resource A.
- Authorizations/credentials in migration must be consistent with authorizations/rights in the authorization database.
- Unique credentials or credentials with limited amount in quantity (e.g. software licenses) must be migrated with the corresponding process.
- Non-migratable credentials (e.g. license dongle) have to be virtualized to the migrated platform or marked as non migratable in order to prevent a migrated process from being able not to continue.

DEPENDENCIES:

This affects software license management.
This affects authorization management and user authentication.

R84 It must be possible to validate membership in VOs and ensure access to resources given proven membership and rights

Authorization and guaranteed access are two different requirements although enforced by interdependent security mechanisms/ services. That is, a principal may have been provided with a token, ticket or credential but the appropriate access control policy or service interface is not available at the time of request. The locally evaluated rules that determine if a party is authorized or not (beyond the possession of a token, ticket or credential), must also be agreed to across the set of resource providers. It may not be possible to implement this in the OS, but there need to be "hooks" to higher level services that can perform such evaluations.

ASSETS:

VO membership identity and authenticity.

REFINED SECURITY REQUIREMENTS:

- A VO resource must be able to verify the authenticity of user identities.
- VO user identity must be unique.
- In case of delegation there must be a unique chain leading to a unique VO user.

DEPENDENCIES:

This affects user and rights management, and the use of Single Sign On.

R85 It must be possible for administrators to record usage (by whom and when) of resources without users being able to deny (repudiate) such usage

Accountability is the ability to enforce and prove that a principal has performed an action on a given resource at a given time. The requirements for accountability are typically a secure audit service with the ability to timestamp messages. This is for the purposes of non-repudiation, should there be a case where it must be proven that a principal has indeed performed an action, as well as billing. It should also be possible to record within which VO the resource was used.

ASSETS:

All events triggered by a user/principal in a direct or indirect way are treated as an asset for the security goal accountability. This covers access and usage of:

- Filesystem (file identifier and action: read, write, append, delete, etc)
- Process management (process identifier and action: start, termination, halt, … of processes)
- Communication (communication identifier and action: read, write, etc)
- Application specific events and invocation of methods

It can also be necessary to log events which are triggered or at least tried to be triggered by non VO users to detect e.g. Denial of Service (DoS) attacks on the network.

REFINED SECURITY REQUIREMENTS:

- Reference monitors must monitor the usage of resources and log them to a logging service. The reference monitors have to acquire as much data as necessary to perform the intended audit (security, billing, etc).
- All reference monitors in a VO must be in accordance to an agreed on specification.
- The integrity of the evidence/log must be ensured. This require the use of write only memory or a secure logging protocol providing this characteristic.
- The database/file of usage logs must be accessible to qualified users or a security administrator in order to detect misbehaving users/principals or processes.

DEPENDENCIES:

This affects reference monitors controlling access to files, quotas, network and application specific reference monitors.
This affects VO management procedures to do resource specific usage and utilization logging.

SECURITY SERVICE:

- Trustworthy logging service in filesystem, network, application, process management reference monitor.
- Trustworthy collector storing log data to a write only memory.

R86 Isolation of VO users - It must be possible to maintain users for different VOs separately

As users may be involved in multiple VOs, it is then necessary to separate their user data and have a means of determining for which VOs are they currently working in, when accessing data.

Isolation of a user having duties in VO_1 and in VO_2 can be interpreted by the security goal confidentiality and integrity at the same time.

ASSETS:
User identifier and user data.

REFINED SECURITY REQUIREMENTS:

- Differentiation between a valid user/principal and a valid user in a VO.
- Mapping of valid users to valid VO users, e.g. by a role based access control scheme
- User/principal identifier of a VO_1 is invalid in another VO, e.g. VO_2
- Invalid users/principals must not have any access to resources in a VO.

DEPENDENCIES:
This affects SSO.
This affects confidentiality and integrity of user data.
This affects migration of authorizations.

SECURITY SERVICE:

- Using different, non overlapping name spaces in XtreemOS.
- Multiple virtualized instances of XtreemOS per VO resource (via, XEN, VMware).

R87 Isolation of data per-VO – data stored on the same resource for different VOs must show non-interference

The isolation of data per-VO enables data to be separated between different groups and contracts. Data belonging to a VO should be logically isolated only for that VO, such that changes made in one VO, although referring to the same data element, should not be possible. There is an absolute need for having automated enforcement of policies. This is however not that surprising, as confidentiality is a fundamental security requirement and most organizations with sensitive data would have already invested time and money in acquiring, developing and integrating mechanisms to enforce confidentiality policies.. However, there is a mix of implementation dependent on the OS-Layer and integrating at Application Layer, which suggests that there still needs to be a consolidating framework that allows reuse of OS security services as well as application layer libraries and security modules. A multilayered architecture for security services are therefore foreseen, which however means that the integration points between layers must also be analyzed and secured.

Isolation of a user, storage areas and processes having duties in VO_1 and in VO_2 can be interpreted by the security goal confidentiality and integrity at the same time.

ASSETS:
Data of the VO covering user/principal data, executables, applications, libraries, security modules and configurations.

REFINED SECURITY REQUIREMENTS:

- Users/principals in VO_1 can not induce changes in VO_2 unless services in both VOs communicate via defined interfaces.
- Users/principals in VO_1 can not initiate actions which derogate the security, trustworthiness or the service level of another VO, e.g. VO_2
- User/principal identifier of a VO_1 is invalid in another VO, e.g. VO_2
- Invalid users/principals must not have any access to resources in a VO.
- Identifiers for files, memory, processes, shared memory, pipes of VO_1 are invalid in VO_2 and vice versa

DEPENDENCIES:
This affects SSO.
This affects confidentiality and integrity of user data.
This affects migration of authorizations.

SECURITY SERVICE:

- Using different, non overlapping name spaces in XtreemOS.
- Multiple virtualized instances of XtreemOS per VO resource (via, XEN, VMware).

R88 Isolation of services per-VO - the secure access to virtualized resources and services must be customized for each VO

In addition to isolation of data, services must also be isolated per VO. That is, different instances of the same VO will have different security requirements and must not provide means of illegal information flow. It must not be possible for parties in different VOs to recognize that they are sharing resources nor to gain knowledge of what other parties are doing with those resources. If one of two virtualized services on the same physical resources fails, this should not interfere with the other.

Isolation of a user, storage areas and processes having duties in VO_1 and in VO_2 can be interpreted by the security goal confidentiality and integrity at the same time.

ASSETS:
Data of the services and processes providing services within a VO.

REFINED SECURITY REQUIREMENTS:

- Processes (also ones providing services) in VO_1 can not induce changes in VO_2 unless services in both VOs communicate via defined interfaces.
- Processes (also ones providing services) in VO_1 can not initiate actions which derogate the security, trustworthiness or the service level of another VO, e.g. VO_2
- Identifiers for processes, shared memory, pipes of services of VO_1 are invalid in VO_2 and vice versa

DEPENDENCIES:
This affects SSO.
This affects confidentiality and integrity of user data.
This affects migration of authorizations.

SECURITY SERVICE:

- Using different, non overlapping name spaces in XtreemOS.
- Multiple virtualized instances of XtreemOS per VO resource (via, XEN, VMware).

R89 The reuse and realization of established security standards and utilities is suggested

It must be possible to reuse and realize established standards for authentication and authorization in the OS – e.g. PKI (public key infrastructure), PAM (pluggable authentication modules) and SSH (Secure Shell).

This is an economic requirement to already used and well known IT security services and infrastructures. Since setting up security infrastructure can be very time consuming it should be possible to reuse existing ones.

ASSETS:

The assets of this requirement are the subjects, objects and rights and architecture covered by the mentioned items.

REFINED REQUIREMENTS:

- It should be possible to authenticate trust relations and user/item identities, (mostly by the use digital certificates anchored at a root of trust) in an already established framework
- The framework may/must?? support hierarchies
- The framework may support different representations of proofs of identity

DEPENDENCIES:

This affects user identity management/SSO.

SERVICE:

- Use of existing security services and test cases.

R90 Linking of Trust Management services with OS mechanisms

There is a need to link mechanisms implemented at the OS layer to higher level reputation and third party trust management services, which influence access control decisions.

ASSETS:

Identities/identifiers of security relevant services.

REFINED SECURITY REQUIREMENTS:

- Authentication of components/services according to a common accepted specification is necessary.
- Authentication of trust relationships depending on code or identity authentication.
- It should be possible to authenticate components and security services by their issuer and thereby authenticate a trust relationship with the issuer.
- It should be possible to authenticate components and security services by its code in order to authenticate a trust relationship with the issuer or to determine, if several resources use identical components with the same properties.

DEPENDENCIES:

This affects reference monitors for confidentiality and integrity.
This affects user identity management/SSO.

SECURITY SERVICE:

- Authentication mechanism for software or components
- Attestation mechanism for certain properties, e.g. based on code signing and Trusted Computing

R91 Semi-automation of administration and configuration of security infrastructure is necessary

It must be possible to set up and configure an XtreemOS security infrastructure in less than 1 working day, and, in worse case, less than 10.

ASSETS:

VO management information.
User identities and their VO relationship.

REFINED REQUIREMENTS:

- VO resources must be identified and associated which each other within the requested timeframe.
- User identities of different resources must be consolidated within the requested timeframe.
- Resources in the VO must be inventoried within the requested timeframe.
- Access controls have to be set up and checked within the requested timeframe.
- Management consoles must have usable and intuitive interfaces.
- Automated process for the distribution of security configurations.

DEPENDENCIES:

This affects user identity management/SSO.
This affects VO management.

SERVICE:

- Usable and intuitive remote management console

R92 Semi-automation of adaptation and reconfiguration of the security infrastructure is necessary

It must be possible to make adaptations to the infrastructure in less than 1 working day and, in worse case, less than 5. This therefore implies a high degree of automation or a very simple set of guideline for flexible modifications to the infrastructure

ASSETS:

VO management information.
User identities and their VO relationship.

REFINED REQUIREMENTS:

- Management consoles must have usable and intuitive interfaces.
- Automated process for the distribution of security configurations.

DEPENDENCIES:

This affects user identity management/SSO.
This affects VO management.

SERVICE:

- Usable and intuitive remote management console.

R93 Multiple bundles or configuration for XtreemOS cryptography have to be considered to support different CPU performance constraints

In some cases partners have indicated that they expect a solution that consumes <0.5% of the CPU, while others have indicated as much as <50%.

The computational load on a CPU of a resource for cryptography purposes depends on hone hand mainly by the executed applications and services. On the other hand it depends also on the used cryptography mechanisms and protocols. The so called “OS-noise” may already consume a distinct amount of CPU usage. Overall this is would be “nice-to-have” but can not be guaranteed in every application case. Cryptography accelerators could be used to offload all cryptography operations from the main CPU.

REFINED REQUIREMENTS:

- Cryptography accelerators must be available for the underlying hardware of a resource
- The OS of a resource must support cryptography accelerators
- All software stacks providing cryptography operations have to support the cryptography accelerator in order to offload these tasks from the main CPU
- State of the art cryptography mechanisms have to be used in case no accelerator is available
- Cryptography mechanisms implemented either in software or in hardware have to be compatible

DEPENDENCIES:

This affects confidentiality and integrity of communication between VO resources.
This affects cryptographic procedures and libraries in general.
This may affect encrypted filesystems.

SECURITY SERVICE:

- Single interface for all cryptography mechanism to make use of cryptography accelerators.

R94 A standard security assessment criteria and profile should be followed for evaluating the XtreemOS

Either we will need to extend existing metrics and evaluation criteria for security architectures, or we can adopt one such as Common Criteria, and first define a Protection Profile according to their specification.

ASSETS:

Properties, security properties, management, maintenance and documentation of XtreemOS covered by a criteria catalog.

REFINED REQUIREMENTS:

- Set of evaluation criteria in general.
- Application specific evaluation catalog depending on the application which is executed on XtreemOS.

DEPENDENCIES:

This affects mechanisms implemented in XtreemOS and necessary procedures to set it up, maintain and manage it. This may also affect the documentation.

5 Security Requirements for VO Management

5.1 Definition of Virtual Organization

A VO can be seen as a temporary or permanent coalition of geographically dispersed entities (individuals, groups, organizational units or entire organizations) that pool resources, capabilities and information to achieve common objectives. There usually will be legal or contractual arrangements between the entities. The resources can be physical equipment such as computing or other facilities, or other capabilities such as knowledge, information or data.

Virtual Organizations can provide services themselves and thus participate as a single entity in the formation of further Virtual Organizations. This enables the creation of structures with multiple layers of value-added service provision.

Key components of a VO are:

- an owner/administrator of the VO
- a set of participating users in different participating domains.
- a set of participating resources in different participating domains
- a set of roles which users/resources can play in the VO.
- a set of rules/policies on resource availability and access control.
- an (renewable) expiry time of the VO.

For the purposes of XtreemOS, we don't model VO Goal or Workflow, though XtreemOS tools should allow these to be supported at the application level. This will typically require enforcement of policies, event notification of the completion of processes, and monitoring of exceptional events, such as jobs still executing at VO expiration. Similarly, we would not expect kernel support of contractual arrangements, but require monitoring and enforcement of policies which can be derived from contracts.

A VO and its implementation by an operating system can reside in several stages of VO lifecycle: **VO Identification, VO Formation, VO Operation, VO Evolution**, and **VO Dissolution**. In each stage a set of security threats to the overall system exist.

Since the overall security of a VO depends on its weakest component, the mechanisms for managing a VO have to ensure sufficient security properties presented at a particular VO node. Defects of traditional security mechanism can result in a lack of confidentiality, integrity, accountability and availability, e.g. by an invited VO node which does not adhere to a certain policy.

5.2 VO Lifecycle

We start by revising the security aspects in a VO lifecycle.

5.2.1 VO Identification

VO Identification includes two steps: identify VO candidates and name them. The identification of VO candidates should only be known to those selected, which do not know the identity of other candidates. Negotiation is needed for VO formation, which includes actual Quality of Service (QoS) parameters, availability of the service, "willingness" of the candidate to participate, etc. If the intended formation fails, the identification will be delayed to another time.

To name VO candidates, each VO should have a set of policies and its members should have unique identities. At runtime, the mapping between virtualized and physical resource should be guaranteed for integrity. QoS and Service-Level Agreements (SLA) should be accounted to classify VO nodes at different levels of service and trustworthiness.

5.2.2 VO Formation

This is the second phase in VO lifecycle. In this stage, the VO is created and configured according to the anticipated roles of members. Only authorized users can copy, merge, or split VOs, and such operations must be done in a single transaction. It must be possible to apply fine-gained access control on multilevel information in VOs. VOs must provide WS-I and WS-security compliant interfaces.

5.2.3 VO Operation

VO Operation comes after the VO identification. Members must be identified for effectively logging and auditing. The VO should be able to classify the resources to different access control level for effective management. Communication confidentiality, operation and context integrity should be maintained when accessed across VOs. The VO should be able to access the runtime information (status) of resources to allow authorized users to know which resources can be used. VO operations and activities should be accountable and auditable

5.2.4 VO Evolution

VO Evolution takes place when the VO is altered during its lifespan, by a change in the participating entities in the VO or in their conditions of use. In this phase, members can be added and linked into a VO by authorization. VO users can be classified at different levels with associated operation rights. This may occur during the operation of the VO which should have minimal influence the operations in current session, or may require a suspension of the VO for a renegotiation of the terms of the VOs operation. No one should be able to change the configuration of the VO apart from a delegated administrator.

5.2.5 VO Dissolution

When a VO is dissolved, non-persistent information should be deleted, credentials reclaimed, and users and resource providers should be notified. The action of deleting a VO should take place after all activity finished, and it must be deleted in an

atomic transaction. The resource usage of VO nodes should be made available for accounting purposes.

5.3 General Security Objectives for VO Management

GSR1 VO must provide WS-I, WS-security, WS-trust and WS-Federation compliant interface

VO interfaces should be compliant with existing WS-I and security related standards for web services, which are widely accepted in service computing field. Compliance with existing standards is important to gain maximum interoperability with other grid platforms.

Importance: **Obligatory**

GSR2 Operations in VO must be performed in atomic transactions

In all phases of VO lifecycle, many operations will occur and the operations should not be disrupted during execution. The atomicity of operations must be enforced. This means all operations must either be performed completely, or if the operation fails, have no side-effects. This is the general objective to ensure security and Quality of Service. In identification, listing candidates should not be disrupted; in formation, creation should not be disrupted; in operation, deleting a node should not be disrupted; in dissolution, clearing up should be done completely. Not requiring atomic VO operations would make the secure operation of a VO much more difficult to ensure.

Importance: **Obligatory**

5.4 Requirements for VO Identification

The requirements of security of VO Identification follow the basic objectives as below:

- Select VO candidates (from Section 3.1.2 of D3.5.1, Security State of the Art in Security for OS and Grids)
- Identification of VO nodes
- Identification of VO node properties - Authentication/Attestation of VO node properties - Secure/faithful identification of node properties is necessary for security relevant components (reference/log monitors) guaranteeing its security properties, e.g. confidentiality, integrity and accountability.
- Classification of qualified VO nodes

GSR3 The selection of VO candidates should be kept confidential and if requirements not satisfied, the formation will be delayed

All the selected candidates only know itself is selected and don't know the other selected ones before they get the membership. Other users and resources outside of the candidate set can not know any information of the set. This information is confidential

The selection of VO candidates includes two stages: discovery and negotiation. Discovery is to discover VO candidates according to specified requirements. The requirements to identify candidates for VO contain service descriptions, security grades, trust & reputation ratings, resource types, availability, policy and SLA. After discover a set of candidates, negotiation of actual Quality of Service (QoS) parameters, availability of the service, "willingness" of the candidate to participate, etc, is needed for VO formation.

At the stage of negotiation, if the specified requirement not satisfied, take QoS, SLA as example, either the requirements may be reduced or the actual formation may be delayed to be re-launched at a more suitable time.
Importance: **Obligatory**

GSR4 Each VO should have unique identity, as well as members involved in it

On the level of VO, each VO should be identified by a globally unique identity. And in the internal of VO, its member should also be uniquely identified.
Importance: **Obligatory**

GSR5 A VO and its members should be identified by relative name or address

The identity should be a relative name or address that reflects the hierarchy membership of VO.

It must be possible to use both virtual name and physical name to reference VO and its members. This would allow the application to use virtual names to ensure loose coupling with real resources, gaining flexibility, reliability, fault-tolerance, etc. Because the same resource can be added to different VOs with different policies, it should have a different virtual name in each VO. (e.g. /VO1/node2/resource3)

When an application refers to virtual name of resources, these virtual names related to VO help to keep the application from being too tied to specific VOs. If a resource is identified by, e.g., *node2/resource3*, the application using this relative name can run in any VO with the VO's help binding the relative name to the real resource dynamically. The VO implements the name mapping and binding between different VOs.

However, if a resource is named by an absolute path such as */VO1/node2/resource3*, it cannot be used in other VOs when VO1 is dissolved, even if the real *resource3* is still available.

The user's visibility of the resource name space may be restricted, depending on the user's access rights.

Importance: **Obligatory**

GSR6 User's access to resource is implicitly controlled by VO and mutual authentication should be necessary

The access to resource should comply with the policy and SLA of VO. When users access VO, nodes and resources, both subjects and objects of the access should be authenticated to verify their identities.

Importance: **Obligatory**

GSR7 At runtime, the mapping between virtualized resource and physical resource should be guaranteed for integrity

At runtime, the mapping between virtualized resource and physical resource should be stable, and the policy should not change in a single session. The session is an interaction for user to control his applications or to finish his jobs interactively. In a session, user can run a batch application, or multiple applications. All local resources allocated during a session – local UID/GIDs, temporary files – are released at the end of the session. A session can be terminated on user request, when all processes in the session are terminated, or when the user credentials are invalid.

Importance: **Obligatory**

GSR8 Resources should have a single entry point

Resources should be accessed at a single entry point that helps to simplify the security mechanism.

For example, a web service and its related data can be accessed by an URL or a local file path, resulting in two entry points to access it. More than one entry point makes the design of VO security mechanism more complex.

Importance: **Obligatory**

GSR9 The VO administrator should be able to classify nodes according to service and trustworthiness level

Depending on the task a VO should perform, potential nodes should be selected by the level of service they can perform. This is necessary to avoid situations in which bad service quality affects the whole system throughput.

Depending on confidentiality requirements of data, potential nodes should be selected by means of their trustworthiness.

Importance: **Optional**

5.5 Requirements for VO Formation

At the end of the identification phase the candidate set is formed. During VO formation, someone will create a VO and add selected candidates to it. After adding proper access control policies and making all needed configuration decision, all this configuration information must be securely distributed to all partners. At each local site, these VO level policies need to be mapped onto local policies.

During the formation stage, these objectives must be satisfied:

- Creation of VO can be carried by any authenticated grid user.
- VO creator will be the VO administrator, and will serve as the only role that can authorize rights to other VO members.
- Fine-gained access control on multilevel information in VOs.
- Mutual authentication between The VO and the candidates.
- Define policy and check availability according to the QoS, SLA and reputation of users and resources
- Confidential and integrity of data transferred between VO and participating partners.

GSR10 Creation of VO can be carried by any authenticated grid user

VO Formation should be easy to carry out so that VO nodes have a simple way to share resources and work collaboratively. Allowing any authenticated grid user to create VOs without some others' approval can reduce the complexity of VO formation procedure and make VO creation and operation autonomous.

The challenge of such free and autonomous VO creation policy is how to trust the VOs created by individuals. It increases the importance of QoS, SLA, and reputation of VOs themselves, users and resources.

Importance: **Obligatory**

GSR11 VO creator will become the VO administrator, and will serve as the only role that can authorize rights to other VO members

The VO creator becomes the VO administrator and owns all rights in the VO automatically. That means the VO creator has full control on the created VO. Then the VO creator, who also has the role of VO administrator now, can customize membership, policy, and configuration of VO autonomously.

To reduce the management cost of the administrator, the administrator should be allowed to authorize VO operation rights to VO members, excluding the 'grant' right itself. Such delegation means that the users who get these operation rights directly from administrator are not allowed to grant these rights to other users again. Only the administrator of VO can grant operation rights of a specific VO to others. Imposing such limitation on the grant right is in order to reduce the complexity and risk of authorization.

Importance: **Obligatory**

GSR12 It must be possible to enforce fine-gained access control on multi-level information in VOs

There is multi-level information, both about VOs themselves and the entities they contain, such as the VO's identity, statistical information, VO membership information, and other VO properties. Entities in VOs, for example users, have identity and other detailed information, such as profile. VO data, for example, files, has file name, file content and other information. A VO must be able to authorize and enforce fine-gained access control on all this multilevel information.

There is a great need for fine-gained access control policies. For example, the VO administrators might want to ensure that any authenticated users can search VOs by properties to get VOs' identifications and statistics information, but aren't allowed to know their membership. For files in VOs, the VO administrators may want to authorize some users to list the filenames, but prevent them reading or writing file content. With such policies, XtreemOS can allow VO creators to search, view and check needed nodes and user information during VO formation without disclosing sensitive information.

Modification of access control policies must be possible in a semi-automated way. A VO should be able to detect policy conflicts automatically.
Importance: **Obligatory**

GSR13 It must be possible to support mutual authentication between VO and the candidates during VO formation

During formation, VO must authenticate candidates before adding them to VO to avoid spoofing of identity. The candidates also need to authenticate the VO to avoid being added to a malicious VO and, via spoofing, exposing sensitive data.
Importance: **Obligatory**

GSR14 It must be possible to get the QoS, SLA and reputation of users and resources in order to define policies and check the availability

During VO formation, there will be users and resources from many candidates added to the VO. The QoS, SLA and reputation of users and resources is needed to decide whether or not to add them to the VO and to define proper policies for incorporating them. The QoS and SLA of resources is important to grant the availability of the resources at the VO formation point.

Reputation is an important factor of trust. It must be generated by the system and the users or resources can not modify their own reputation. The reputation of principals can be divided into two parts, that related to a VO and that independent of VOs.
Importance: **Obligatory**

GSR15 The confidentiality and integrity of configuration information transferred between a VO and its partners must be ensured

During VO formation, the configuration information of the VO must be distributed to partners. This information must be encrypted to keep it confidential. The integrity of the communicated data must be validated to avoid illegal modification. Loss of integrity of the communication data must be prevented and detectable.

Importance: **Obligatory**

5.6 Requirements for VO Operation

VO operation is the third phase of VO lifecycle. In this phase, many activities will take place between VO members. Traditional security objectives have certainly some effects on grid, but only have been restricted in user's authentication and authorization. Besides these, data consistency (mentioned in D3.4.1) is also an important issue in this phase, and other objectives such as logging user activity and auditing user operations are necessary to enhance VO security. Security objectives can be illuminated if the operations are considered in three areas: operations in VO, operations between VOs, and operations on resources. Confidentiality, integrity, availability, and accountability can be demonstrated in three kinds of operations.

- Authentication and Authorization
- Availability of candidates when members joining in VO
- Confidentiality of runtime information and communication data
- Integrity of operations in same VO, between VO, and to resources
- Isolation and confidentiality of private data
- Effective security management

GSR16 The candidates should be available before being added and linked to a VO after authorization

Availability is necessary in expanding candidate members of VO. When the administrator authorizes membership to the candidates, the candidates should be available before the addition operation begins.

If the availability is not satisfied, unintended effects will occur in VO operation.
Importance: **Obligatory**

GSR17 It is necessary to adopt mutual authentication when candidates are added/deleted from a VO

Authentication requirements precede not only VO administrator operating the candidate (e.g. users, node, and resources) but also the candidates responding to the VO administrator. It is rational for the VO administrator to authenticate which candidates are the one he wants. However, when candidates decide to become a member of a VO, they must know which VO they should link in.

To trust each other, mutual authentication needs to be applied. A candidate will not be willing to be a member of a non-trusted VO. If a candidate enters in a non-trusted VO, there is potential for illegal access to harm the candidate.
Importance: **Optional**

GSR18 It should be possible to classify VO users to different level, and operation rights must be authorized to different level of VO user

It is very important to classify VO users at different levels, such as anonymous user, approved user, and authorized user. To administrate a VO, an administrator should have the rights (add, delete, check, change, etc), which can be authorized to other users.

This requirement is necessary to enable security enforcement. Granting rights must be executed by an administrator. Furthermore, these rights can not be passed on from one user to other user, except the administrator.

Importance: **Optional**

GSR19 It should be able to classify the resources of node to different access control level for effective management

Different access control level classified in resources authorizes user different access control rights. The resources should be divided into

- The opening resources: They have not any local access limitation, and access control is totally decided by VO policy.
- The limited resources: They are controlled by a local Discretionary Access Control mechanism and also controlled by VO policy. How to smooth the collision depends on their agreement.

It is necessary for security enforcement to classify the access rights to resources, similar to the division of user to owner, group and other.

Importance: **Obligatory**

GSR20 It should be able to access the runtime information of resources to allow authorized users to find which resources can be used

Authorized users should be able to get the list of usable resources (including the VOs, nodes, etc). The information about resources such as the node should only be visible to the authorized user.

Isolation is needed to implement the security. By isolating the user vision of accessible resources, access control will be enhanced.

Importance: **Optional**

GSR21 Confidential data communicated across VOs must be encrypted using cryptography protocols

Communication across VOs is necessary, and all the data must be encrypted to the same level as the communication between resources in the same VO.

Communication across VOs is necessary to permit operations such as negotiating service level agreements etc. Data and access context during communication should be kept confidential in user accessing resources across VO.

Importance: **Obligatory**

GSR22 Only the administrator can copy, merge, or split a VO, and such operations must be done in a single transaction

The copying, merging or splitting if VOs will change the number of VOs, and can only be performed by the VO administrator.

Another security requirement for these operations is that they must be done in a single transaction. Consider the following situation: if someone copies a VO, and all the users and data information has been fully copied but only part of the policies has been copied, then the copy operation is not atomic. If we can't detect and enforce the atomicity automatically, then due to the missing access control policies, there might be some malicious attackers who can access unauthorized information, and some users may violate these policies in unintended behaviours.

Importance: **Obligatory**

GSR23 Integrity of operations across VOs must be maintained

The SLA of each resource should be satisfied, when the resources are under control of user's operations. Users operate the data or deal with commands between VOs, and these operations should be completed. It is very trouble that the communication between VOs fails in the course of user's operating. Some strategy such as reconnecting and retransmission is recommended.

User's operations between VOs should be atomic.

Importance: **Obligatory**

GSR24 Confidentiality and integrity of context must be maintained when VO users access the VO resources

Confidentiality of context is a fundamental requirement of systems that provide the condition to judge the user running state and environment. In XtreemOS, a grid user from a different VO who accesses the common resources in the common node will have a different context. The information in the context must be confidential and protected from illegal modification, and only the administrator has the access right to the information.

The context should include the follow aspects:

- User/process profile
- Node/resource information
- VO scope

Importance: **Obligatory**

GSR25 Only VO administrator can change the configuration of VO, and the change should not influence the operations in the current session

The information in VO configuration is very necessary for administrator to maintain its confidentiality and integrity. It will be very dangerous for user accessing malignantly. All the confidential information of VO should be accessed by VO owner or VO administrator.

VO administrator changes the VO configuration for adding/deleting a member, reallocating the policy, etc, but the dynamic changing configuration should not influent the SLA of each current resources. For example, when user is running a job (maybe a long-time job) in a session, the policy of VO changed suddenly. The change would not take into effect before the end of current session.

Importance: **Obligatory**

GSR26 It must be able to identify the members for effective accounting and auditing

Each VO member should be allocated a unique identity. The identity must be able to uniquely locate the real member. For example, some important jobs must be accounted and audited. If the identity does not uniquely bind to a member, accounting and auditing will not be possible.

The security requirement is presented based on the reasons: some mechanism, such as the account pool of the Globus Toolkit, performs user ID mapping to identify the user, and this does not guarantee that the VO ID binds uniquely to a distinct user.

Importance: **Obligatory**

GSR27 VO operations and activities should be accountable and auditable

For security enforcement, user operations should be logged to enable auditing. It is very useful to find malicious users who are performing illegal operations. Furthermore, VO administrators can discover runtime exceptions by checking the log.

In most operating systems, logging has become an indispensable way to trace system errors.

Importance: **Obligatory**

GSR28 It must be possible to trace user operations and evaluate the resource QoS for effective monitoring of the VO

Logging and auditing user operation is required to trace user activities, showing misused resources and malicious users. The VO administrator must monitor what the VO members are doing, identifying illegal users and detecting illegal operations as soon as possible.

Another important function is evaluating the resource QoS to effectively implement resource sharing and collaborative problem solving.

Importance: **Obligatory**

5.7 Requirements for VO Evolution

By VO Evolution, we understand the process of changing the formation of a Virtual Organization during its lifecycle. This might be due to any of the following reasons:

- The unavailability of a particular participating resource, organization or user;
- The need to introduce a new a particular participating resource, organization or user.
- The need to change policies or roles in the operation of the VO, in response to
    - violations of access control policies by users in the course of the VO Operation
    - non-fulfillment of quality of service policies (SLAs) of resources in the course of the VO Operation
    - Changes in the reputation of users and services recorded across VOs.

In this section, we assume that during the normal operation of virtual organization, there is:

- a constant set of users, resources, policies and organizations
- a monitoring process recording the actions and performance of users, resources, and organizations with respect to a set of QoS and access control policies.
- A reputation service keeping track of the performance of users, resources, and organizations over different VOs.

We also assume that there is a VO administrator (possibly more than one). The VO administrator is either the VO owner, or a user with a designated role of administrator, delegated by the owner.

This would lead to the following requirements for VO Evolution:

GSR29 Add User to an operating VO
Need to be able to add a user with appropriate roles to a VO. There needs to be a mechanism to notify the new user's domain that they are now participating in a new VO, and to issue the appropriate attribute certificates.
Importance: **Obligatory**

GSR30 Remove User from an operating VO
Need to be able to remove a User from a VO, revoking all attribute certificates in the VO and all roles. This will need to be notified and propagated to all participating domains.
The administrator of the VO should take this action.
Importance: **Obligatory**

GSR31 Add Resource
Need to be able to add Resources to VO, with appropriate QoS and Access control policies in the VO. Will need the notification of the participating domains of the new resource and the issuing of appropriate attribute certificates. The administrator of the VO should take this action.
Importance: **Obligatory**

GSR32 Remove Resource
Need to be able to remove a resource from a VO, revoking all attribute certificates in the VO and all policies. This will need to be notified and propagated to all participating domains. The administrator of the VO should take this action.
Importance: **Obligatory**

GSR33 Change Role
The role of a particular user in the VO is changed to a new role. The administrator of the VO should take this action, except in the case of the administrator role, which only the owner of the VO should be allowed to assign.
Importance: **Obligatory**

GSR34 Change Policy
The policy of a particular resource in the VO is changed. The administrator of the VO should take this action.
Importance: **Obligatory**

GSR35 VO Suspend
Need to be able to suspend a VO, complete, checkpoint or abort current processes, and prevent the execution of new processes under that VO. The administrator of the VO should be able to suspend the VO.
Importance: **Optional**

GSR36 VO Resume
Need to be able to suspend a VO, complete, checkpoint or abort current processes, and prevent the execution of new processes under that VO. The administrator of the VO should be able to suspend the VO
Importance: **Optional**

5.8 Requirements for VO Dissolution

- VO node invalidation - Node can be single machine or cluster.
- Software licensing (RWP XXX 2.8, 3.5.6) - License invalidation on a dissolved node has to take place. Otherwise a licensed application can continue without a valid license and violate the integrity of granted software licenses.
- Data and credential deletion - Data and credentials stored on a dissolved VO node but belonging to services of the former VO cluster has to be deleted and invalidated.
- Online or offline notification - active process, user and inactive process that live on VO need to know VO dissolution.

GSR37 Information of the VO should be deleted when VO is dissolved

To avoid confidential information leakage, the following content should be deleted when VO is dissolved.

1. VO identification. If we want to trace the history, the VO identity may be marked as deleted but not be deleted actually. For example, if someone wants to trace one's reputation in all entered VOs even when some of them have dissolved, this mark may be a hint.
2. Information of VO members, e.g. user credential, node, policy repository.

If one real resource (node or cluster) can be linked into multiple VOs, in order to show resource decreasing in VO, the reference counter reside in resource side should be decreased automatically. The action of deleting a VO must not be breakable and should be completed in one transaction.

Importance: **Obligatory**

GSR38 The user and resource provider should be notified when VO is dissolved

Applications built upon virtualized resource that are provided by a VO may gain access rights from the VO at runtime, hence they must be notified when the VO is dissolved. Multiple notification approaches are needed, e.g. signal and mail corresponding to online and offline mode. The application holders then can reconfigure the resource binding by backup VO or other VOs.

Importance: **Obligatory**

GSR39 Reclaiming credentials

Application or software running on real resources may maintain some kinds of credentials obtained from a VO. Credentials may include an access ticket or token, or licenses issued to the application holder. These credentials can be reused illegally If not revoked.

Importance: **Obligatory**

GSR40 The action of deleting the VO should be done only after all activity finished

VO should provide kind of soft-shutdown mechanism to cope with the dissolution. The action of deleting VO must not affect the active user, process and operation that currently using this VO's function. The active process that consumes resource in VO may keep running until met some stop criteria. At the same time, hard-shutdown mechanism is also needed for controlling long term running process and malicious user.

Importance: **Obligatory**

GSR41 The usage of VO nodes (or other resources) should be logged before the VO is dissolved

The consumption of resources in VO should be billed to the correct user, especially when the VO is dissolved. When all the active processes finish, the billing information should be persistently stored for auditing use later. The VO being dissolved has accumulated rich information detailing the reputation of resources accessed. To avoid the loss of such valuable information, it should be possible to export such records to persistent storage.

Importance: **Optional**

6 Requirements for Trust Management

In general, the purpose of security mechanisms is to provide protection against malicious parties. Traditional security mechanisms typically protect resources from malicious users by restricting access to only authorized users. However, in many situations within distributed applications one has to protect oneself from those who offer resources so that the problem is in fact reversed. For instance, a resource providing information can act deceitfully by providing false or misleading information, and traditional security mechanisms are unable to protect against this type of threat. Trust systems can provide protection against such threats. Since trust has such a great relevance in Grids, we have included an analysis of trust requirements.

6.1 Federation of Trusted Domains

Federation of trusted domains means that multiple trusted domains agree to interact with each other with federated identities, attributes and pseudonyms. Such federation constitutes a trust relationship across multiple participating domains so that users of a secure domain can access services in other secure domains. Federations of trusted domains are created in the VO formation stage and evolve in the VO operation stage. The creation, evolution and dissolution of a federation for multiple trusted domains must be done with considerations of security objectives in the VO's lifecycle.

To achieve the basic security objectives defined in section 2.2, the federation of trusted domains must be secured to protect the trust relationships among multiple participating domains. The core of such a requirement is that the risk caused by trusting others for each partner in the federation should be minimized. This involves two aspects. First, collaborations among all partners should be done with guaranteed confidentiality and integrity. Second, the federation should enable each partner to be aware of the reputation, QoS and security states of other partners so that the trust risk can be re-evaluated and the corresponding actions can be performed.

Considering the above two security aspects, the federation of trusted domains should be implemented and controlled under the following constraints.

- The creation of federation of trusted domains needs secure communications for the dissemination of configuration information and the negotiations among participating domains.
- During the operation of the VO, service performance provided by partners in the federation must be monitored and recorded to provide evidence for constructing the reputation of service provides.
- The federation of trusted domains must provide notification mechanisms so that any violation event and security threat detected by a partner locally can be notified to other partners in the federation.
- Partners of a federation must support reconfiguration of locally enforced security policies to adapt to changes and violation events received from other partners.

GSR42 Secure communications must be employed for the dissemination of configuration information in order to create federation of trusted domains at the VO formation stage

VO formation dynamically creates a federation of trusted domains. Global configuration information related to the VO policies will be disseminated to each participating domain for the creation of the federation. This dissemination must be protected with guaranteed confidentiality and integrity by employing secure communications
Importance: **Obligatory**

GSR43 Secure communications must be employed for the negotiations among multiple partners in order to create federation of trusted domains at the VO formation stage

Negotiations among multiple partners may be involved when creating federation of trusted domains at the VO formation state, which must be protected with guaranteed confidentiality and integrity by employing secure communications.
Importance: **Obligatory**

GSR44 It must be possible to monitor and record the service performance provided by partners in the federation of trusted domains during the VO operation stage

To reduce the risk of participating in a federation of trusted domains, each partner must be able to re-evaluate the reputations and QoS of other partners. To satisfy such requirement, service performance provided by partners in the federation must be monitored and recorded to provide evidence for constructing the reputation of service provides.
Importance: **Obligatory**

GSR45 Notification mechanisms for the federation of trusted domains must be available so that any partner is able to receive security-related events detected by other partners

To guarantee that the trust relationship among multiple partners in the federation of trusted domains remains secure, any partner should be able to obtain information on the runtime security status of other partners and to notify its own security status to others. To achieve this, any violation event and security threat detected by a partner locally must be notified to other partners in the federation. Notification mechanisms are required to support such interaction.
Importance: **Obligatory**

GSR46 Any partner of a federation of trusted domains must be able to re-configure and re-enforce local security policies during the VO operation stage

It is possible that the trust relationship among multiple partners in the federation becomes uncertain because of a violation event which happened to some partner. As the response to this, partners of a federation must support reconfiguration of locally enforced security policies to adapt to changes and violation events received from other partners.
Importance: **Obligatory**

6.2 Quality of Service in VO Formation, Monitoring, Policy

When forming a VO we search for appropriate services within organizations that could execute our workflow. Besides their appropriateness (i.e., their ability to execute our tasks) we are also interested in their reputation – can we trust them to meet the required performance (in terms of execution times, availability, etc.) constraints? To this end we employ the SLAs, i.e., the required Quality of Service.

To ensure that the Quality of Service is a) met, b) that the organizations are not advertising services with misleading properties and c) that the users of the services are not exceeding the agreed QoS, we need to implement security measures which prevent tampering with them.

This is done via ensuring a secure way of storing:

- The properties of the available services - these properties are set by the organization that owns the service.
- The reputation of the organizations/services is updated securely. They do not have access to the reputation data about their services.
- The monitoring process is secure. Any tampering with this process must be detectable.
- The services' policy and the corresponding services (e.g., Policy Enforcement Point) must be secured.

The above described measures enforce the availability of the real data which is to be used in finding the appropriate services for the new VO. When running a VO, the tampering within the services policy is also prevented, hence preventing the users to misuse them by enforcing their policies. These security measures also prevent widespread breaches with use of rogue services which only collect sensitive data.

Another view on Quality of Service in VO Formation, Monitoring and Policy is to look at the trustworthiness of different organizations. When forming a new VO, all the participating organizations need to be trusted – their reputation must be above certain level. When these are sufficient and with security measures, defined above, we can be reasonably certain that our requirements will be met and that our resources won't be misused.

When using complex systems that heavily rely on the authentication, authorization and delegation, we must always assume that these security systems may break. It is possible that they are under DoS attack or that the node (nodes) broke down. In these cases the agreed-upon QoS will obviously fail, hence we need a forensic tool for matching which service consequently failed and notify the corresponding users. The same applies when services don't have proper authorizations.

6.3 Reputation and Trust Management

Until recently, reputation based models are barely considered for classical grid systems. As one of the requirements of the next grid generation is to provide dynamic aggregation of resources, provided as services between businesses and virtual organizations, new architectures and detailed mechanisms for bringing together arbitrary resources are required. These architectures should federate security and trust, as ones of the most significant issues [Ahsant et al. 2006]. Reputation is a novel approach for building trusted environments with enhanced dynamics and self-organization.

Reputation is what is generally said or believed about a person's or thing's character or standing [Josang et al. 2006]. They argue that reputation is a mean of building trust, as one can trust another based on a good reputation. Therefore, reputation is a measure of trustworthiness, in the sense of reliability. According to [Abdul-Rahman & Hailes 2000], a reputation is an expectation about an agent behaviour based on information about or observations of its past behaviour. This last definition emphasizes the two main sources for building the reputation of an entity: the past experience and the collected referral information.

According to [Yu & Singh 2002], the challenges that a reputation management system should address are the following:

- how the system rates one entity based on the past transaction history,
- how an agent finds the right witnesses in order to select the referral agents with respect to a partner in a transaction and
- how the agent systematically incorporates the testimonies of those witnesses.

These are the basic requirements for a reputation management system without regard of the further application of the system in a specific area (e.g. in grids). We can notice that reputation is acquired:

- directly from past transactions of a node; we say that we have built a *direct trust* with regard to an entity.
- from the social network, through a 3rd party entity, this is the *indirect trust* regarding an entity

As studies in reputation management proved that systems using both types of trust are more reliable, we should require the combined usage of direct and indirect reputation-based trust. Therefore, two distinct components should compose the core of the reputation system: the direct trust component and the indirect trust one.

With regard to a reputation-based trust management in grids and virtual organizations, two requirements are of a special interest:

- *SLA or QoS negotiation*: reputation models will be directly applied for negotiation of SLA or QoS between 2 parties like a service consumer and producer. The items to be negotiated and after the execution of contract, how each party fulfilled the agreements on the specific items part of a SLA should be directly incorporated in the direct trust component.
- *Trust aggregation*: The model should allow aggregating trust on an organizational basis. This property is of great importance in the context of VO formation and operation as allows one:

  - to obtain the trust and reputation for a VO based on the individual trust on its members
  - to infer the trust or reputation for an individual based on the trust and reputation of organizations the individual belongs to.
  - to infer the trust or reputation for an individual or for a VO with respect to some specific item to be included in a SLA

Usage of a reputation-based trust system has implications on several other components of the whole XtreemOS architecture, out of the scope of the security concerns. In what follows we list the points of the system that will be affected by the usage of a reputation-based trust management scheme:

- The reputation system will require the existence of a *monitoring service* for the grid that will supply with events about how an entity performed during the VO operation phase
- VO identification and VO formation mechanisms will be strongly affected as they should use the information supplied by the reputation management component
- Resource allocation and task scheduling will be affected to incorporate reputation information
- To make the entities aware about their role in the VO and to facilitate the participation of entities to the social network by providing feedback for 3rd parties, an incentive-based mechanism should be designed. The incentive mechanism should:
  - Reward entities that delivered the agreed QoS for a contract
  - Reward entities that participated with useful 3rd party information in the social network
  - Penalize entities that failed to deliver the agreed QoS
  - Penalize entities that cheated with wrong feedback in the social network

  Using an incentive-based approach the reputation management system can act as a method for sabotage tolerance in the grid.

Depending on the type and the architecture that will be selected for the reputation system, other technical issues are under concern.

Therefore, if a *centralized* reputation system will be adopted, the drawback of a single point of failure will exist. The entity which will act as reputation manager needs to have

- large bandwidth to support increased communication overhead
- large storage to be able to deal with a lot of observations and reputation information

If a *decentralized* reputation mechanism is to be adopted, some specific P2P properties are of concern:

- the replication of the reputation data on the entities forming the grid
- the distributed mechanism for data retrieval
- security issues concerned with data protection at the levels of the entities

In an environment with multiple VOs operating at the same time that are composed of entities from different physical organizations, a small degree of decentralization at least is required.

7 Requirements from WP3.2, Highly Available and Scalable Grid Services

7.1 Overview of Distributed Server Architecture

A distributed server is a collection of nodes running a grid process. In the understanding of WP3.2 a grid process consists of one or more operating system processes running on each participating node. All of those processes cooperate in order to reach the grid process' goals. Just as for a single physical server, a distributed server offers an interface to which clients can send requests to the server and get responses. Besides these user interfaces, an administration interface exists that provides a means of controlling the server and the grid processes running on it.

In order to allow the structuring of groups such as a distributed server, WP3.2 will provide a global grid infrastructure based on epidemic protocols. Each node participating in the grid must run these protocols. To start a grid process, the owner first has to allocate nodes. Therefore he specifies the number of nodes and properties that the nodes are to fulfill, and sends both these pieces of information to the global grid infrastructure.

The concept of a distributed server is supplemented by a way of increasing the fault-tolerance of critical parts of the grid process. This is done by grouping a small number of nodes into a ‘virtual node’ and running OS processes with identical behaviour on them. This architecture has security demands that concern multiple layers, so we will have a closer look at each of them in the following sections. Note, WP3.2 has not yet considered the existence of virtual organizations. This may be solved by having explicit properties for node allocation. Moreover WP3.2 is aware of the fact that epidemic algorithms are vulnerable to byzantine attacks. However, securing against such attacks is not seen as a task for WP3.5 as this vulnerability is due to inherent characteristics of the algorithms themselves.

7.2 Security at the Grid Level

The grid level is the set of all nodes contributing to the grid. At this level it is necessary to assure that only authorized nodes take part in the grid and no others. Hence, a node authentication mechanism must be available. Furthermore it should be possible to not only find out about node properties, but also to verify them. A node offering 500 MB of disk space, but holding not more than 100 MB, must not be tolerated. Generally speaking a user should be able to specify as many properties as possible for node allocation and the system must be able to enforce them or at least make them checkable. In the ideal case the number of properties is unlimited.

7.3 Security at the Application Level

The application-level comprises all nodes taking part in running an application, i.e. all nodes forming part of a distributed server. Seeing a grid process as some sort of logical process makes obvious which security demands occur. Grid processes are not allowed to influence each other without the system knowing about it. Thus communication between OS process a, part of grid process A, and OS process b, part of grid process B (and consequently belonging to a different distributed server), must not happen in an arbitrary way, but have to pass the public application interface of B. The underlying system must enforce this, no matter which nodes both OS processes are running on.

Communication among OS processes of the same distributed server may happen in arbitrary ways. This does not hold for nodes participating in a virtual node. They must be accessed by a provided communication infrastructure to avoid inconsistencies among the replicas. Furthermore the aforementioned administrator interface must not be accessed by non-authorized persons. The semantics of this, however, heavily depends on the whole security architecture defining e.g. grid process ownership. User interfaces should also offer the possibility to define access restrictions and authorization mechanisms.

7.4 Security at the Host Level

On the level of an individual node it is important to control the resources used by OS processes, as those must not exceed the resource quotas allocated for them. Regarding communication confidentiality and integrity are absolutely necessary, especially for intra-server communication.

7.5 Conclusions from WP3.2

Putting the demands mentioned so far on an abstract level, one might see a general need for input-output control for each OS process. This can be compared to the Java security mechanism that forces every operation affecting something outside the virtual machine to be approved by the system-wide security-manager. However, WP3.2 security demands go further than that of Java, as Java only tries to protect the environment against a potentially malicious application, but there is an additionally need to protect the application against a potentially malicious environment (consisting of other nodes and applications). Thus, in the case of communication - on all levels, between all entities - at least integrity, privacy and availability of messages will have to be provided.

8 Summary

This document has presented a set of objectives that XtreemOS, a secure Grid operating system providing native Virtual Organization support, should provide. The security requirements from the requirements-gathering exercise in WP4.2, and from WP3.2, Highly Available and Scalable Grid Services, show some concrete requirements to meet. Defining the set of security services that will meet these targets is the task of T3.5.2, which is using the findings of this document.

9 References

1. Abdul-Rahman A. and Hailes S, Supporting trust in Virtual Communities, *in proceedings of the 33rd Hawaii International Conference on System Sciences, (HICSS 2000), (4-7 January, 2000, Hawaii, USA),* IEEE Computer Society, 2000, vol 6, p. 6007

2. Ahsant M., Surridge M., Leonard T., Krishna A. and Mulmo O., Dynamic Trust Federation in Grids, in *Proceedings of the 4th International Trust Management Conference (iTrust 2006), (16-19 May, Pisa, Italy)*, LNCS 3986, Springer, 2006, 3-18

3. Josang A., Ismail R. and Boyd C., A Survey of Trust and Reputation Systems for Online Service Provisioning, submitted to *Decision Support Systems,* 2006

4. Yu B. and Singh M., An Evidential Model of Distributed Reputation Management, in *Proceedings of the 1st Joint International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2002), (Bologna, Italy)*, ACM Press, 2002, 294-301

5. Saltzer, J. and Schroeder M, The Protection of Information in Computer Systems, in Communications of the ACM 17, 7 July 1974

6. The Simple API for Grid Applications - http://wiki.cct.lsu.edu/saga/