Grid access control models and architectures

Antonios Gouglidis, Ioannis Mavridis
University of Macedonia, Greece

ABSTRACT

In recent years, Grid computing has become the focal point of science and enterprise computer environments. Access control in Grid computing systems is an active research area given the challenges and complex applications. First, a number of concepts and terminology related to the area of Grid access control are provided. Next, an analysis of the Role Based Access Control (RBAC) and Usage Control ABC ($UCON_{ABC}$) models is given, due to their adaption from the Grid computing systems. Additionally, a presentation of well known Grid access control architectures illustrates how the theoretical access control models are implemented into mechanisms. In a comparative review of the examined access control models and mechanisms, their pros and cons are exposed. Apart from the mapping of the access control area in Grid computer systems, the given comparison renders valuable information for further advancement of current approaches.

INTRODUCTION

The Grid is an emergent technology that can be defined as a system able to share resources and provide problem solving in a coordinated manner within dynamic, multi-institutional virtual organizations (Foster, Kesselman, & Tuecke, 2001). This definition depends mostly on the sharing of resources and the collaboration of individual users or groups within the same or among different virtual organizations, in a service oriented approach. The Grid's unique characteristics, such as its highly distributed nature and the heterogeneity of its resources, require the revision of a number of security concepts

Trust, authentication, authorization and access control are some of the security concepts met in Grid systems, as these are identified in the existing literature (Gouglidis & Mavridis, 2009). In this chapter, we will further examine the latter of the aforementioned. Access control is of vital importance in a Grid environment since it is concerned with allowing a user to access a number of Grid resources. An extensive research has been done in the area of access control in collaborative systems (Tolone, Ahn, Pai, & Hong, 2005; Zhang, Nakae, Covington, & Sandhu, 2008). Nonetheless, further examination is demanded. This is mainly due to the partially or weak fulfillment of the access control requirements in Grid systems.

The aim of this document is to provide the reader with a comprehensive report on the access control models and architectures currently used in Grid computing systems. The value of this chapter is the mapping of the Grid access control area, so as to assess the applicability of access control solutions in modern Grid applications. Along with the identification of a number of core Grid access control requirements, a comparative review of access control models and mechanisms determines their pros and cons. The results from the comparison greatly value the applicability and appropriateness of both models and architectures in being used in Grid systems.

The structure of the remainder of this chapter is as follows. The next section provides a prerequisite terminology used in access control, in the context of Grid systems. Furthermore, a number of Grid access control requirements are presented. An analysis of the Role Based Access Control and the Usage Control models follows. In addition, an examination in regard

to the implementation of the theoretical access control models into mechanisms is displayed. A complementary discussion section provides a comparative review of all the examined access control models and mechanisms, respectively. Finally, we present our concluding remarks along with some future thoughts.

BACKGROUND

This section introduces the basic concepts and terminology, related to Grid systems and access control. A presentation of the access control process and the identification of core Grid access control requirements follows.

Terminology and access control concepts

As mentioned in the definition of the Grid, terms such as users, resources and services play an important role. To this effect, we explicitly set the following definitions, mainly based on (Benantar, 2005; Chakrabarti, 2007; Ferraiolo, Kuhn, & Chandramouli, 2003; Foster & Tuecke, 2005; Ravi S. Sandhu, 1994).

A *service* is an implementation of well defined functions that are able to interact with other functions. The *service oriented architecture* (SOA) is comprised of a set of services that can be realized by technologies such as the web services.

A *domain* can be defined as a protected computer environment, consisted of users and resources under an access control policy. The collaboration which can be established among domains leads to the formation of a virtual organization.

A *user* in a Grid environment can be a set of user identifiers or a set of invoked services that can perform on request one or more operations on a set of resources. Furthermore, we identify two types of users. These are the resource requestor and the resource provider. The former type of user acts like a resource access or usage requestor, and the latter type of user acts like a provider of its own sharable resources. All users are restricted by the policies enforced in their participating domains and virtual organization.

A *resource* in a Grid environment can be any sharable hardware or software asset in a domain and upon which an operation can be performed.

Access control's role is to control and limit the actions or operations in the Grid system that are performed by a user on a set of resources. In brief, it enforces the access control policy of the system, and at the same time it prevents the access policy from subversion. Access control in the literature is also referred to as access authorization or simply authorization.

A Grid *access control policy* can be defined as a Grid security requirement that specifies how a user may access a specific resource and when. Such a policy can be enforced in a Grid system through an *access control mechanism.* The latter is responsible for granting or denying a user access upon a resource. Finally, an *access control model* can be defined as an abstract container of a collection of access control mechanism implementations, which is capable of preserving support for the reasoning of the system policies through a conceptual framework. The access control model bridges the existing abstraction gap between the mechanism and the policy in a system.

Grid access control requirements

The identification and definition of Grid access control requirements, namely the access control policy, greatly amplifies the design of a model and the implementation of a mechanism regarding access control. In order to appoint the core access control requirements we use the conceptual categorization for Grid systems proposed in (Gouglidis & Mavridis,

2010). Figure 1 depicts the four layers of the conceptual categorization. A set of core requirements for access control systems that are considered important for the Grid environment, follows. These requirements may vary depending on the use cases that need to be supported by a specific system.

Figure 1. Conceptual categorization layer.

In the initial layer of entropy, we identify two basic requirements. The first is that access control should be enforced among all the collaborative domains. Thus, interoperability among domains should be supported within and among virtual organizations. Although each domain has its own access control system, in order for them to successfully collaborate, a unified access control system should be provided. The second requirement refers to the number of the participating domains or users that can change during the time span of the collaboration. In more detail, during the collaboration it is possible for new domains or users to join, and existing ones to quit. The access control system should be able to be monitored continually and handle such modifications in the structure of the virtual organization.

Regarding the layer of assets, we identify a dyadic nature regarding the access and sharing of an asset. More specifically, we recognize that the fine-grained sharing of any resource in a Grid system includes a resource requestor and a provider. When user requests access to an asset, access must be granted only if the requestor is a legitimate user and also authorized to access the specified asset. Additionally, resource providers should be able to define quality factors on their shareable resources. The quality factors concern the level of resource usage and can also be characterized as obligations that must be met from a provider when granting access to a resource requestor. For instance, quality factors could apply for setting disk quotas, memory or CPU utilization levels and so on and so forth.

In the management layer, we define a list of requirements that refer to the management of the policies of the individual domains, as well as the virtual organization itself. A first requirement is that each administrative user of a domain should administer the local policies of the domain. Additionally, administrators should run the policies in the collaboration that refer to resources of the administrator's domain. Furthermore, it must be guaranteed that no conflicts should exist among the policies of the individual domains at the level of the virtual organization, where policies are joined. Last but not least, the process of identifying policy violations should be automated, both in intra-domain and inter-domain collaborations.

At the logic layer, we identify the enforcement of the autonomy and security principle (Shafiq, Joshi, Bertino, & Ghafoor, 2005). The autonomy principle refers to the permission of an access under secure interoperation, if it is also permitted within the individual domain. The security principle pertains to the denial of an access under secure interoperation, if it is also denied within the individual domain. Furthermore, the principle of containment (Ravi

Sandhu, 2008) that subsumes the principles of the separation of duties, least privilege and so forth, should be supported in each and among domains. The latter requirement greatly enhances the adoption of Grid technologies in business organizations, where the existence of conflict of interest policies is presumed.

Access control enforcement

In this section, a brief presentation of the reference monitor concept is given. This is mainly done because the application of the reference monitor concept is known to achieve high assurance access control mechanisms. Furthermore, it provides guidelines for the design and implementation of secure computer systems (Ferraiolo, Kuhn, et al., 2003).

The process of access control in any computer system guarantees that any access to the resources of the system conforms to its access control policy. The application of the abstract concept of the reference monitor is capable of providing the requirements that are posed from the access control process. As it can be also seen in Figure 2, the reference monitor operates as an access mediator between the subject's access requests and the system's objects. The accesses comply with the system's security policy. The reference monitor can be informed for the security policy of the computer system from an access control database. Moreover, all the security relevant transactions are kept into an audit file for security and traceability reasons.

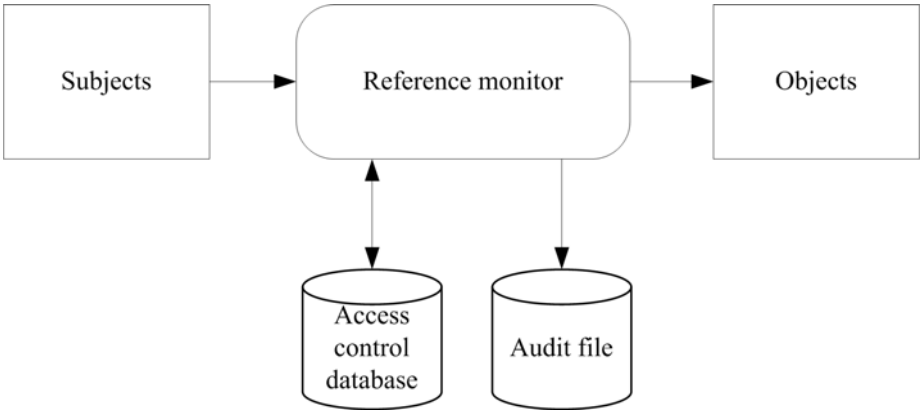

Figure 2. The reference monitor.

The architecture of the reference monitor is the result of the application of three key implementation principles. These principles are the completeness, isolation and verifiability. Completeness requires from the reference monitor to invoke all the subject's references to an object and also to constitute it impossible to bypass it. The isolation principle assures that the reference monitor must be tamper-proof. This means that it must be impossible for an attacker to penetrate the reference monitor in a malicious way. Lastly, the verifiability principle appertains to the checking and validation of the system's security design through the use of software and system engineering techniques.

Nonetheless, the aforementioned reference monitor principles seem to be insufficient, especially in enterprise environments. This is mostly because the main objective of the reference monitor is the enforcement of each system's policy. Yet, it does not interfere with the articulation of a system's security policies. Thus, the principles of flexibility, manageability and scalability are introduced. The first principle assures that the access control policy of an enterprise can be enforced by the existing security system. The next refers to the ease of policy management and the latter requires from the security system to cope with the fluctuations in the number of the participating users and resources in a computer system.

The concept of reference monitor in open systems has been standardized with the X.812 access control framework (ITU-T, 1995). In brief, the main functions in X.812 are the Access Control Decision Function (ADF) and the Access Control Enforcement Function (AEF). The former component is responsible for the making of access control decisions. The decisions are made based on information applied by the access control policy rules, the context in which the access request is made, and the Access Control Decision Information (ADI). ADI is a portion in the Access Control Information (ACI) function, which includes any information used for access control purposes, including contextual information. Lastly, the AEF is responsible for the enforcement of the decision taken from the ADF. Figure 3 illustrates the fundamental access control functions in X.812.

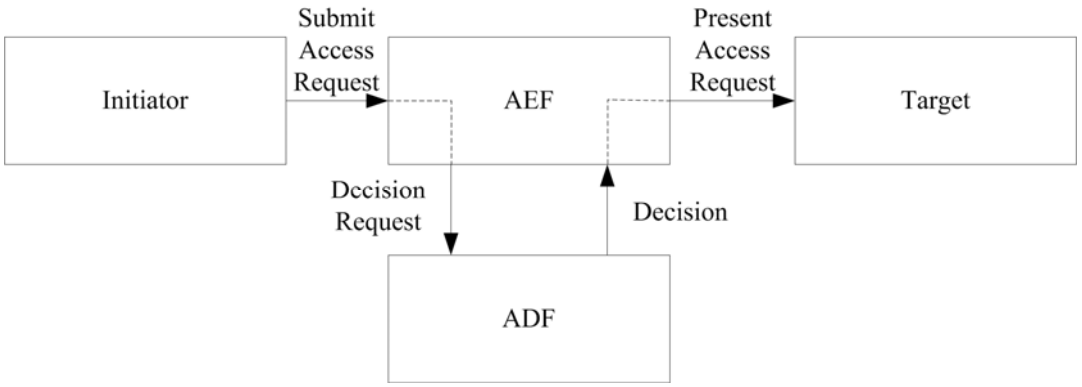

Figure 3. Fundamental access control functions in X.812.

ACCESS CONTROL MODELS

During the last decades various access control policies have been introduced, namely the Mandatory Access Control policies (MAC), the Discretionary Access Control policies (DAC) and the Role Based Access Control policies (RBAC). Each one of them serves specific security requirements in different working environments. As mentioned in the definition of the access control policy, a number of access control models are required and were developed in order for the policies to be represented by formal methods. Research on the MAC, DAC and RBAC has proven that an access control model, which can express the role based access control policies is also capable of enforcing both MAC and DAC policies (Ferraiolo, Kuhn, et al., 2003). It is noteworthy that an attempt started along with the advancement of RBAC for the design of a series of Attribute Based Access Control models (ABAC). The ABAC model was mainly introduced to overcome a number of RBAC's shortcomings (Yuan & Tong, 2005) and has also been proven capable of enforcing MAC, DAC and RBAC policies (Park & Sandhu, 2004). For the aforementioned reasons, we will present the standard for the role based access control (American National Standard Institute, 2004), and Usage Control (Park & Sandhu, 2004; R. Sandhu & Park, 2003; Zhang, et al., 2008) in the rest of this section. Both RBAC's and UCON's characteristics are able to tackle the complexity posed from Grid systems at a satisfactory level.

Role based access control (RBAC)

The RBAC access control model has received considerable attention from researchers, mainly due to its abstraction and generalization. It is abstract because it includes only properties that are relevant to security, and it is general since it supports various designs that can all be interpreted as valid ones. More of RBAC's virtues are the support of a significant number of principles, namely the least privilege, separation of administrative functions and separation of duties (R. S. Sandhu, Coyne, Feinstein, & Youman, 1996). Following in the section, RBAC's standard model will be put forward. This model consists of four different components and each one of them assigns to RBAC a number of functionalities. These components are the

core RBAC, the hierarchical RBAC, the static separation of duty relations and the dynamic separation of duty relations.

As it is illustrated in Figure 4, the core RBAC model is composed of five static elements. These elements are the users, roles, and permissions, with the latter being composed of operations applied on objects. The relationship among the elements of the core model is straightforward. Roles are assigned to users and permissions are assigned to roles. The type of relation between users and roles and between roles and permissions is many-to-many. This means that one user can be assigned many roles and that many users can be assigned one role. The same applies for the role to permission assignment as well. Declaration of negative permissions is not supported in RBAC. This indirect assignment of users to permissions greatly enhances the administration in RBAC. Revocation of assignments can also be easily done. Moreover, we identify two distinct phases in RBAC. The first is the design and the second the run-time phase. During the design phase, a system administrator can define a number of assignments between the elements in the computer system. At the run-time phase, the assignments in the system are enforced by the model as it is specified by the security policy of the system, which was prescribed during the design phase.

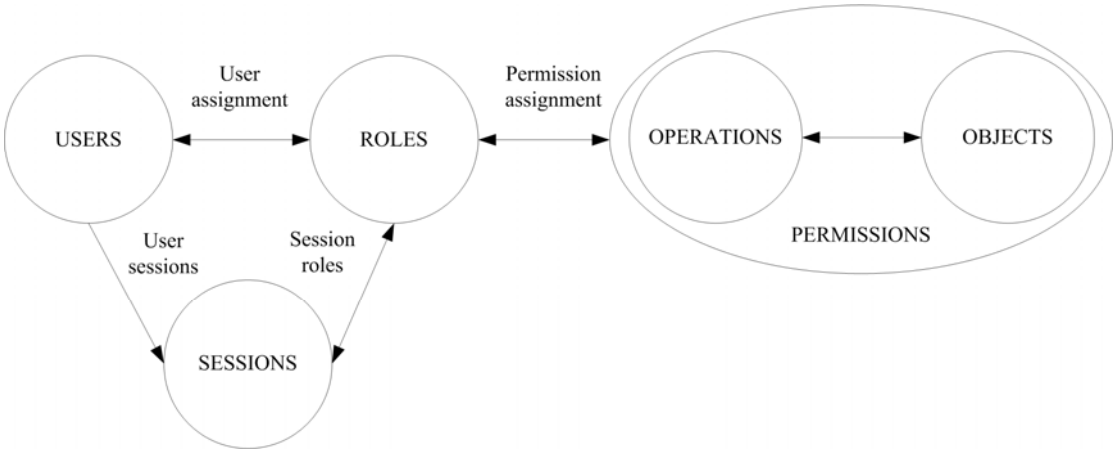

Figure 4. The core RBAC model.

The run-time phase that was previously mentioned can be supported in RBAC through the concept of the session. The latter distinguishes RBAC from other group based mechanisms and adds great features and functionality to the RBAC model. During a session, roles for a subset of users are allowed to be activated. This means that a user could be assigned various roles during the design phase, but these roles do not need to be activated always or simultaneously, preserving at the same time the principle of least privilege. Without the support of the notion of sessions this would not be possible to achieve. It is also feasible to enforce a number of constraints during a session. We will further discuss the support of constraints in RBAC latter in this section. However, although the sessions strengthen RBAC, there has been an argument concerning the existence of sessions that proposes their replacement from a separate component in the core RBAC model (Li, Byun, & Bertino, 2007). The argument continues regarding the number of activated roles during a session. It is proposed that it should be possible for core RBAC to further support the activation of single roles during a session, as a requirement of some systems.

The hierarchical RBAC provides the model with a great enhancement in regard to the administration of its policies. Role inheritance provides more flexibility in the management of the policies in an organization. Permissions that are assigned to a role can easily be inherited to another role, without the need to reassign the same permissions to the latter role, too. For instance, let's assume two roles R1 and R2 and two permission sets P_{R1} = (P1, P2) and P_{R2} = (P3, P4), which are initially assigned to roles R1 and R2, respectively. If role R1 inherits role

R2, it means that all of R2's permissions are available via R1. The available permissions to role R1 are expressed by the union of permissions on sets P_{R1} and P_{R2}. When hierarchies are represented in graphs, the immediate inheritance relation is shown as →. The head of the arrow or the arc defines both the permissions and user membership inheritance. For the previously mentioned example, we have R1 → R2. User membership refers to the assignment of users to roles in a hierarchy. In such a case, users are authorized to access all the permissions assigned to roles either directly or via inheritance relationships. Yet, another functionality that is provided in the hierarchical RBAC is the support of both general and limited role hierarchies. General hierarchies comprise the most common cases in role inheritance, and they are depicted as partial order sets. However, in more restrictive environments there might be the requirement for the support of limited hierarchies. This involves usually the existence of either a single immediate ascendant or descendant role in the hierarchy tree structure.

Another virtue of RBAC is the support of constraints. The two components that can enforce constraints are the static and dynamic separation of duty relationships. The main objective in both types of constraints is to preserve the security of the system and prevent it from being compromised. Usually they are used to deliver business requirements to the security system that incorporates an enterprise's logic. Static separation of duty relationships copes with the enforcement of conflict of interest policies. For example, let R1 and R2 be two conflicting roles, and user U1 assigned to role R1. By enforcing a static separation of duty constraint between roles R1 and R2, RBAC prohibits the assignment of user U1 with role R2, since the two roles are conflicting. These types of constraints are defined and enforced in RBAC during the design phase. In the presence of a role hierarchy, the static separation of duties constrains are enforced in the same way for all the directly assigned and inherited roles. Dynamic separation of duty relationships handles conflict of interest policies in the context of a session. In this case, the user is actively logged into the system and a set of the user's assigned roles is activated. These constraints are described during the design time, as it happens with the static separation of duty relationships. However, they are applied during the run-time, in the context of a session, and they prevent the simultaneous activation of two or more conflicting roles. In case of role hierarchies, the same as in static separation of duty relationships applies with the difference that they are enforced only on the activated user's roles.

Lastly, one of its greatest virtues is the role based administration of RBAC. It can be said that RBAC is divided into user space and administrator space. The former includes user and the latter administrative roles, permissions and operations, respectively. Once again, the principle of least privileged is maintained. In the literature various models have been proposed, each one providing a different approach in the role based administration of RBAC (Crampton, 2002; Ferraiolo, Chandramouli, Ahn, & Gavrila, 2003; Oh & Sandhu, 2002; R. Sandhu, Bhamidipati, & Munawer, 1999).

Usage control (UCON)

Attribute based access control (ABAC) has lately gained a lot of attention due to the development of internet based distributed systems. However, in contrast to RBAC, attribute based access control has not been standardized yet. The latter type of access control models can provide access decisions on resources based on the requestor's owned attributes. The advantage of this approach is that it is possible to provide access to users in a collaborative environment without the need for them to be known by the resource a priori. In this section, we will present in brief the $UCON_{ABC}$ model (Park & Sandhu, 2004) as a representative attribute based access control model, which is based on a modern conceptual framework. The UCON conceptual framework encompasses traditional access control, trust management and digital rights management for the protection of digital resources. Nonetheless, functionalities such as administration and delegation are still absent.

UCON has introduced a number of novelties compared to both RBAC and other ABAC models, like its support for mutable attributes and continuity of access decision. Research has also been done regarding its usage in collaborative systems (Zhang, et al., 2008). Figure 5 illustrates the $UCON_{ABC}$ model, which consists of eight components, viz. subjects, subject attributes, objects, object attributes, rights, authorizations, obligations and conditions. The notion of subjects and objects as well as the association with their attributes is straightforward. A subject can be an entity in a system and its definition, as well as its representation, is given by a number of properties or capabilities in the associated subject's attributes. For instance, role hierarchies similar to RBAC can be formed through the use of subject attributes. In regard to objects, they also represent a set of entities in a system. Each object can be associated with object attributes. Subjects can hold rights on objects. Through these rights, a subject can be granted access or usage of an object. This type of attributes can serve, for example, in the classification of the associated objects, by representing classes, security labels and so on and so forth. It is worth mentioning that both subject and object attributes can be mutable. This means that the values of the attributes can be modified as a result of access. When an attribute is characterized as immutable, its value can be modified only by an administrative action and not by its user's activity.

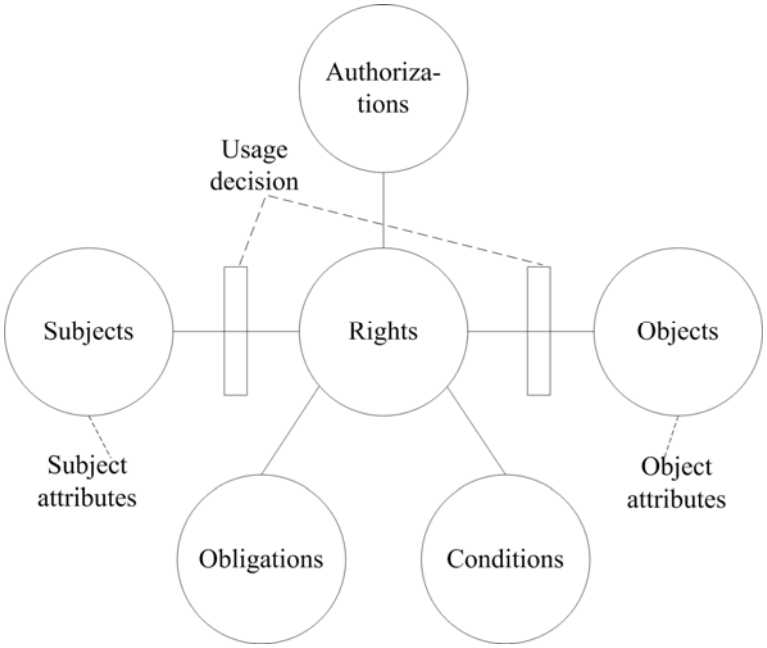

Figure 5. The $UCON_{ABC}$ model.

Up to now, a presentation of the most common components of the $UCON_{ABC}$ model was given. However, its novelties in access control are accrued mostly from the rest of its components. The rights component represents a number of privileges that can be held and exercised from a subject to an object. In a similar way to RBAC's roles, the UCON conceptual framework supports hierarchies among rights. It is also notable that rights are not set a priori, but they are determined during the access. The access decision is given from a usage function by considering the following factors of subject and object attributes, authorizations, obligations and conditions. Authorizations in UCON are functional predicates, whose evaluation is used for taking decisions, namely if access to a subject is granted to an object. In a same manner with the usage function, the evaluation of the authorizations is based on subject and object attributes, requested rights and a set of authorization rules. Authorizations can be characterized as pre-authorizations or ongoing-authorizations. The pre prefix refers timely before the requested right and the ongoing prefix during the time span of access.

Furthermore, obligations in UCON are used to capture the requirements that must be met from a subject requesting the usage of an object. They are also expressed as functional predicates and, as already mentioned, they are used in the evaluation of access both in the usage function as well as with authorizations. Obligations are also divided into pre-obligations and ongoing-obligations. The former is used usually for the retrieval of history information and the latter to check if the requested requirement is fulfilled during the time span of access. Last but not least, conditions in UCON are used to capture factors that are accrued from the environment of the system. The semantic differential between conditions and other variables, namely authorization and obligation, is that the former cannot be mutable, since there is no direct semantic association with subjects.

GRID ACCESS CONTROL MECHANISMS

As mentioned above, the terms of authorization and access control are used interchangeably. Nonetheless, the former definition is most commonly used in Grid systems. In this section, we will further analyze some of the access control mechanisms implemented in existing Grid middleware. A clustering of a number of implemented authorization infrastructures by the capabilities they support is provided in (Schlager, Sojer, Muschall, & Pernul, 2006). The access control architecture used in the majority of them is based on an attribute based approach. The main components in this architecture are the attribute authority (AA), the policy enforcement point (PEP), the policy decision point (PDP) and the policy authority (PA). This architecture is based on the access control framework recommended in (ITU-T, 1995). In X.812 the policy enforcement and decision point are referred to as access control enforcement functions (AEF) and access control decision function (ADF), respectively. The attribute authority is responsible for the generation and management of the subject, object and environment attributes. It is also responsible for the association of attributes with their owning elements as well as the provision and discovery of the attributes. The policy enforcement point requests and enforces access decisions coming from the policy decision point, which have to do with subject to object authorizations. The policy decision point is responsible for evaluating the system's policies and for decision taking. The decision for the granting or denial of access is passed to the policy enforcement point. Lastly, the policy authority is responsible for the creation and management of the authorization policies.

Furthermore, the Grid authorization systems are also characterized by the way the authorization of a user to a resource is achieved (Chakrabarti, 2007). There are two different models used in the currently implemented Grid authorizations systems. These are the push and the pull models. Most systems support either the former or the latter model. However, there are Grid authorization systems that support both of them. In the push model, a certificate generator usually creates certificates based on the user's credentials. Each one of the certificates is pushed on an access controller so as to grant or deny access to the resource, based on the validity of the certificate. On the contrary, when the pull model is used by the authorization system, a minimum number of user credentials is provided to the access controller. In turn, it is the controller's responsibility to check the validity of the user based on the policies of the system. The push model is considered to be more scalable than the pull model. Nonetheless, the push model lacks usability, something in which the pull model is better, since users do not have to obtain the certificate from the certificate generator. Moreover, the responsibility of granting access to a user is passed to the access controller.

Last but not least, the Grid authorization systems can be categorized as virtual organization level systems and resource level systems (Chakrabarti, 2007). The former refers to systems where a centralized authorization system handles the provision of credentials to the users, in order for them to access the resources. In opposition to the virtual organization level, systems that allow the users to access the resources based on the credentials presented by the users are characterized as resource level ones. It is worth mentioning that as noted in (Chakrabarti, 2007) the virtual organization and the resource level authorization systems cope with different

aspects of the grid authorization. The first category of systems provides a consolidated authorization service for the virtual organization and the second category of systems implement the decision to authorize resource access. As a consequence, they complement each other and can provide a holistic authorization solution if combined.

Community authorization service (CAS)

The community authorization service (CAS) (Pearlman, Welch, Foster, Kesselman, & Tuecke, 2002) is a virtual organization level authorization service developed by the Globus team. Its main objective is to cope with the flexibility, scalability and policy hierarchy issues, which primarily exist in Grid's security infrastructure (GSI) and GridMap, since the latter provides only a one-to-one mapping between global user names and local ones. CAS is capable of allowing the resource owners to grant access on portions of their resources to the virtual organization by letting the community determine who can use this allocation. CAS manages to overcome the limitations existing in GridMap by introducing a CAS server that operates as a trusted intermediary between the users of the virtual organization and the resources. The CAS server is capable of managing all the policies that control the access to the resources of a community. It contains information about the users, resources, certificate attributes, servers as well as policy statements. According to CAS, a user has to contact the CAS server at any request to access a resource in a community. This requires from the user to be authenticated by providing the user's own proxy credential. The identity and the rights that the user holds in the virtual organizations are established by using its local database. In turn, the server issues a signed policy assertion with the user's identity and rights in the target virtual organization. The policy assertion is then embedded in a new proxy certificate generated by the CAS client. The new proxy certificate is used on the resource of the virtual organization to authenticate the user and to grant access to the resource based on the embedded policy assertion. The certificates that are used in CAS are X.509 extensions. The proxy credentials that authenticate the user on the CAS server have much longer span of life that the proxy certificates.

Virtual organization membership service (VOMS)

The virtual organization membership service (VOMS) (Alfieri et al., 2003) is also a virtual organization level authorization service developed for the European Data Grid (EDG) that solves the same problems as CAS does but in EDG. The VOMS system operates as a front-end on top of a database and it consists of four components, viz. the user server, user client, administration server and administration client. The user server receives requests from a client and returns information regarding the user. The user client contacts the server by presenting the certificate of a user or proxy to the latter and receives a list of groups, roles and capabilities of the user. The administration server is responsible for accepting the client's request and updating the database. Lastly, the administration client is used by the administrators of the virtual organization for administrative issues like the addition of new users, the creation of new groups and so on and so forth. According to VOMS, a bidirectional authentication of the server and the client occurs. During the authentication process, a safe communication channel is instantiated between them. In turn, the client can send a request to the server. When the server receives the request from the client, the request is checked for its integrity and if no problem exists, the server sends a pseudo-certificate to the user. The client also checks the pseudo-certificate for its integrity. The user can now create a proxy certificate based on the received pseudo-certificate and present it to the resources to gain access on them. A user in VOMS is allowed to be a member of many virtual organizations and also to receive credentials from multiple VOMS systems.

GridMap

GridMap is the simplest and most widely used resource level authorization service. It is rather static and lacks scalability. GridMap is implemented as a file, which holds a list of authenticated distinguished names of the Grid users and their mapping with the equivalent

account names of the local users. The policies that describe the access restrictions are kept in each local resource. The access control is also left to the local systems, so when a user requests access to a resource, the decision to grant or deny the access permission is based on the information present in the local access control mechanism and the local GridMap file.

Akenti

Akenti (Thompson, Essiari, & Mudumbai, 2003) is a resource level authorization system that was created to cope with environments that consist of highly distributed resources and their use by multiple stakeholders. A stakeholder is defined as someone who controls access on a resource. Akenti consists of a resource gateway that operates as a policy enforcement point and of resources, which are accessed via the resource gateway. It makes use of X.509 certificates for the authentication of the users who request access to a resource. The communication between the user and the resource gateway is accomplished through secure SSL/TLS channels. When a user requests access to a resource, access is determined by the combined policy on the resource. These policies can be created by different and unrelated stakeholders and are expressed with signed certificates. The resource gateway can ask from the Akenti server the privileges that a user has on a resource. The Akenti server operates as a policy decision point. In turn, the server retrieves all the relevant certificates, checks their validity and sends a response back to the resource gateway. The latter enforces the operation indicated by the policy decision point. This architecture gives Akenti the ability to restrict access to resources based on predefined access control policies, without requiring the existence of a central administrative authority.

Privilege and Role Management Infrastructure Standards Validation Project (PERMIS)

PERMIS is a role based X.509 privilege management infrastructure and resource level authorization system (D. Chadwick, 2005; D. W. Chadwick, Otenko, & Ball, 2003) that supports the hierarchical RBAC model. The main components that constitute PERMIS are the PERMIS authorization enforcement point, the authorization decision point, the authorization policy and the privilege allocator. The first two components are responsible for the user authentication and decision making, respectively. The authorization decision point can retrieve policies and attribute certificates from LDAP servers and base its decision on the retrieved information. The descriptions of the policies are specified by the authorization policy. The content of the policies specifies who has access on which resource and under what conditions. The privilege allocator is responsible for the allocation of privileges to the users. The privileges are attribute certificates that include role to user associations. Additionally, a delegation issuing service provides the users with the ability to delegate a subset of their privileges to another user of their domain. When a user requests use of a resource, the authorization enforcement point authenticates the user. In turn, the enforcement point passes the user's distinguished name to the decision point. The latter retrieves information relevant to the user from an LDAP server. After performing the validation of the policies, the roles that are embedded in the attribute certificates are transferred as an object to the user. The user is authenticated in every attempt to access a resource. This results in the transfer of the object, which keeps the roles of the user embedded, from the enforcement to the decision point, so as to grant or deny access.

Usage based authorization framework

An attempt to apply a usage based authorization framework in Grid systems is presented in (Zhang, Nakae, Covington, & Sandhu, 2006). Subject and object attributes are used for the definition of usage control policies, and conditions provide context based authorization for the support of ad-hoc collaborations. Continuity of decision and mutable attributes are also supported. Yet, obligations are not supported. In the current state, the management of attributes is centralized. Nonetheless, in case of a distributed attribute repository, a lot of

complexity is added, since the system must keep all the multiple copies of the attributes consistent. The main components of the framework's architecture include a policy decision point and a policy enforcement point. The attributes and the identity certificates of users can be stored in attribute and identity authorities, respectively. When access is requested, the decision point makes the control decision based on the collected attributes and is enforced by the enforcement point. A notable feature is its support of a hybrid model that uses both the pull and push models to cope with the different types of attributes. Immutable attributes in the usage based authorization framework are pushed to the policy decision point by the requesting subject. On the contrary, when it comes to immutable attributes, they are pulled from the attribute repositories.

DISCUSSION

In this section, the access control models and architectures described in this chapter are compared. The comparison is attempted with respect to the conceptual categorization for Grid systems, proposed in (Gouglidis & Mavridis, 2010) with a view to specify a number of deficiencies in the examined models and architectures. The criteria used throughout the comparison are based on the requirements that were defined and the evaluation is based on the level of fulfillment of the requirements by the access control models and architectures, respectively.

Comparing the access control models

Table 1 illustrates the evaluation of the RBAC and $UCON_{ABC}$ models with respect to the entropy, assets, management and logic layers of the conceptual categorization.

Concerning the entropy layer, the requirements that were defined, demand both the support of access control among different domains and the dynamic joining of new ones. The proposed standard RBAC model, as already seen, handles better centralized architectures and is rather weak in inter-domain collaborations. Such functionality is absent from the standard model. However, research in (Shafiq, et al., 2005) has proven that RBAC can also be applied in multi-domain environments where distributed multiple organizations inter-operate. Yet, RBAC requires that all user domains must be known a priori, in order to access an object. On the contrary, the $UCON_{ABC}$ model, due to its support of attributes, can cope better with highly distributed environments. Furthermore, one of UCON's features is that it is possible to provide access to users in a collaborative environment without the need for them to be known by the resource a priori.

Access control models	**Conceptual categorization layers**			
	Entropy	**Assets**	**Management**	**Logic**
RBAC	Low / Medium	Low / Medium	Medium / High	Medium
$UCON_{ABC}$	High	Medium	Low	Medium

Table 1. Comparisons between the different access control models.

In regard to the layer of assets, we mentioned that fine-grained access to resources should be supported. Additionally it should support obligations from the side of the resource provider. RBAC usually provides more course-grained access control to resources in contrast to $UCON_{ABC}$. Research has also been done in RBAC to extend it and to support finer-grained access control through the use of context (Tolone, et al., 2005). Obligations are supported in

$UCON_{ABC}$, but not in the notion demanded by the requirements. The notion of obligations is completely absent in RBAC.

RBAC supports improved administrative capabilities on the level of a domain in comparison to $UCON_{ABC}$. In more detail, RBAC can also provide management in a role-based fashion (Ferraiolo, Kuhn, et al., 2003). However, a number of issues arise when it comes to inter-domain management of policies, and solutions are provided in existing literature (Shafiq, et al., 2005). In contrast to RBAC, $UCON_{ABC}$ lacks administration.

Finally, the fulfillment of requirements in the logic layer is fairly the same in both access control models. Nonetheless, RBAC supports the principles of separation of duties and least privilege better.

Comparing the access control mechanisms

Table 2 depicts the evaluation of the access control mechanisms with respect to the entropy, assets, management and logic layers of the conceptual categorization, while Table 3 illustrates a summary of the comparison. Besides the specified requirements, in our evaluation, we consider a list of extra parameters as stated in (Chakrabarti, 2007). This is due to the adaption of an attributed based approach with strong resemblance by the authorization systems, thus making their evaluation more difficult.

Access control mechanisms	**Conceptual categorization layers**													
	Entropy			**Assets**		**Management**					**Logic**			
CAS	+	+	+	-	-	-	+	-	O	-	-	O	O	-
VOMS	+	+	+	+	-	-	+	-	O	-	-	O	O	O
GridMap	-	O	-	-	-	O	-	+	O	-	+	O	O	-
Akenti	O	O	-	+	-	+	+	+	O	O	+	O	O	-
PERMIS	+	+	+	-	-	+	+	+	O	O	+	O	O	+
Usage based authorization	+	+	+	-	O	+	+	-	-	O	+	O	O	+
	Interoperability	User scalability	Mechanism scalability	Multiple stakeholders	Obligations	Revocation	Administrative overhead	Decentralized management	Ease of management	Automation	Usability	Autonomy	Security	Containment

+: Parameter is supported.
-: Parameter is not supported.
O: Partially or weak support of the parameter.

Table 2. Comparisons among the different access control mechanisms.

The parameters of interoperability, user and mechanism scalability were taken into account in the layer of entropy. Besides the GridMap authorization system, the rest of them handle interoperability well. This is mainly due to the support of standard protocols, namely the SAML and XACML. The support of attributes helps in the fulfillment of the requirements we have defined for the entropy layer. User scalability is affected by two factors. These are the authorization model in use and the type of policy management. Usually systems that support a push based model and a centralized management of policies are less complex. In overall, GridMap exhibits the worst performance in the entropy layer, while CAS, VOMS, PERMIS and Usage based authorization the best.

Regarding the evaluation of the authorization systems for the layer of assets, we examined their ability to permit multiple users to control access on the same resource. As depicted in Table 2, VOMS and PERMIS are able to support multiple stakeholders on a resource. In regard to the parameter of obligations, only the Usage based authorization system supports it. Yet, obligations are from the side of the user and not from the resource provider.

The evaluation of the management of policies is based on multiple parameters, namely the administrative overhead, revocation of attributes, decentralized management, ease of management and automation. As we already mentioned, ABAC approaches lack management. Nevertheless, they provide support of decentralized management and require low administrative overhead in most implementations. Automation of procedures is absent or weakly supported. Lastly, revocation of privileges is present mostly in resource level solutions, and encounter problems in the rest of them.

The principles defined as requirements in the logic layer, in conjunction with the usability of the system, serves as evaluation parameters for the last layer. The principles of autonomy and security are fairly supported by all the examined systems. Nonetheless, the principle of containment is present in PERMIS and Usage based authorization, due to the support of RBAC. Lastly, the usability of a system is affected from either the push or pull model in use.

Access control mechanisms	**Conceptual categorization layers**			
	Entropy	**Assets**	**Management**	**Logic**
CAS	High	Low	Low	Low
VOMS	High	Medium	Low	Low
GridMap	Medium	Low	Low	Low / Medium
Akenti	Medium / High	Medium	Medium	Low / Medium
PERMIS	High	Low	Medium	Medium
Usage based authorization	High	Medium	Low / Medium	Medium

Table 3. Summary of the comparisons among the different access control mechanisms.

CONCLUSION

This chapter introduced and explained in detail the problem of access control in Grid computer environments, including associated concepts and requirements. Access control models and authorization systems in the Grid context are of vital importance due to their distributed nature. This is why we outlined two of the most prominent access control models for collaborative systems. Through the synopsis of both the RBAC and $UCON_{ABC}$ models, we identified their unique and of primal importance characteristics. In addition, a summary of well known Grid authorization system was given. This helped clarifying how the theoretical access control models are turned into access control mechanisms for the Grid systems. A first comparison of the RBAC with the $UCON_{ABC}$ model, has shown that neither of them can tackle the difficulties raised from the defined Grid access control requirements flawlessly. Based on the results of the foregoing comparison, it was expected for the Grid authorization mechanisms to have the same level of applicability in Grid environments. Indeed, the hypothesis has proven right, indicating that the examined mechanisms cannot handle well the defined requirements and parameters in all the layers of the conceptual categorization. Based on the results stemmed from our research, we believe that the design and implementation of proper access control models for the Grid systems is needed. Current access control models are not specifically designed to tackle the requirements of Grid systems. By applying the conceptual categorization for the Grid systems, we illustrated how to identify a list of core requirements and how to use it as a comparison tool. In result, we expect the applied methodology to serve as a foundation for defining access control requirements in Grid computing systems and moreover, to result in improved or new access control models and mechanisms.

REFERENCES

Alfieri, R., Cecchini, R., Ciaschini, V., dell'Agnello, L., Frohner, Gianoli, A., et al. (2003). *VOMS, an Authorization System for Virtual Organizations.* Paper presented at the European Across Grids Conference.

American National Standard Institute, I. (2004). Role Based Access Control, *ANSI INCITS 359-2004* (pp. 56).

Benantar, M. (2005). *Access Control Systems: Security, Identity Management and Trust Models*: Springer-Verlag New York, Inc.

Chadwick, D. (2005). Authorisation in Grid Computing. *Information Security Technical Report, 10*(1), 33-40.

Chadwick, D. W., Otenko, A., & Ball, E. (2003). Role-Based Access Control With X.509 Attribute Certificates. *IEEE Internet Computing, 7*(2), 62-69.

Chakrabarti, A. (2007). *Grid Computing Security*: Springer-Verlag New York, Inc.

Crampton, J. (2002). *Administrative scope and role hierarchy operations*. Paper presented at the Proceedings of the seventh ACM symposium on Access control models and technologies.

Ferraiolo, D. F., Chandramouli, R., Ahn, G.-J., & Gavrila, S. I. (2003). *The role control center: features and case studies*. Paper presented at the Proceedings of the eighth ACM symposium on Access control models and technologies.

Ferraiolo, D. F., Kuhn, D. R., & Chandramouli, R. (2003). *Role-Based Access Control*: Artech House, Inc.

Foster, I., Kesselman, C., & Tuecke, S. (2001). The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Jounral of Supercomputer Applications, 15*(3).

Foster, I., & Tuecke, S. (2005). Describing the Elephant: The Different Faces of IT as Service. *Queue, 3*(6), 26--29.

Gouglidis, A., & Mavridis, I. (2009). A Foundation for Defining Security Requirements in Grid Computing. *Informatics, Panhellenic Conference on, 0*, 180-184.

Gouglidis, A., & Mavridis, I. (2010). On the Definition of Access Control Requirements for Grid and Cloud Computing Systems *Networks for Grid Applications* (Vol. 25, pp. 19-26): Springer Berlin Heidelberg.

ITU-T. (1995). X.812 Recommendation, *Data Networks and Open System Communications Security - Information Technology – Open Systems Interconnection – Security Frameworks for Open Systems: Access Control Framework* (pp. 44): ITU.

Li, N., Byun, J.-W., & Bertino, E. (2007). A Critique of the ANSI Standard on Role-Based Access Control. *IEEE Security and Privacy, 5*(6), 41-49.

Oh, S., & Sandhu, R. (2002). *A model for role administration using organization structure*. Paper presented at the Proceedings of the seventh ACM symposium on Access control models and technologies.

Park, J., & Sandhu, R. (2004). The UCON ABC usage control model. *ACM Trans. Inf. Syst. Secur., 7*(1), 128-174.

Pearlman, L., Welch, V., Foster, I., Kesselman, C., & Tuecke, S. (2002). *A Community Authorization Service for Group Collaboration*. Paper presented at the Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY'02).

Ravi S. Sandhu, P. S. (1994). Access Control: Principles and Practice. *IEEE Communications Magazine, 32*(9), 40-49.

Ravi Sandhu, V. B. (2008). *The ASCAA Principles for Next-Generation Role-Based Access Control.* Paper presented at the Proc. 3rd International Conference on Availability, Reliability and Security (ARES), Barcelona, Spain.

Sandhu, R., Bhamidipati, V., & Munawer, Q. (1999). The ARBAC97 model for role-based administration of roles. *ACM Trans. Inf. Syst. Secur., 2*(1), 105-135.

Sandhu, R., & Park, J. (2003). Usage Control: A Vision for Next Generation Access Control *Computer Network Security* (Vol. 2776, pp. 17-31): Springer Berlin / Heidelberg.

Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-Based Access Control Models. *IEEE Computer, 29*(2), 38--47.

Schlager, C., Sojer, M., Muschall, B., & Pernul, G. (2006). Attribute-Based Authentication and Authorisation Infrastructures for E-Commerce Providers *E-Commerce and Web Technologies* (Vol. 4082, pp. 132-141): Springer Berlin / Heidelberg.

Shafiq, B., Joshi, J. B. D., Bertino, E., & Ghafoor, A. (2005). Secure Interoperation in a Multidomain Environment Employing RBAC Policies. *IEEE Trans. on Knowl. and Data Eng., 17*(11), 1557-1577.

Thompson, M. R., Essiari, A., & Mudumbai, S. (2003). Certificate-based authorization policy in a PKI environment. *ACM Trans. Inf. Syst. Secur., 6*(4), 566-588.

Tolone, W., Ahn, G.-J., Pai, T., & Hong, S.-P. (2005). Access control in collaborative systems. *ACM Comput. Surv., 37*(1), 29--41.

Yuan, E., & Tong, J. (2005). *Attributed Based Access Control (ABAC) for Web Services*. Paper presented at the Proceedings of the IEEE International Conference on Web Services.

Zhang, X., Nakae, M., Covington, M. J., & Sandhu, R. (2006). *A usage-based authorization framework for collaborative computing systems*. Paper presented at the Proceedings of the eleventh ACM symposium on Access control models and technologies.

Zhang, X., Nakae, M., Covington, M. J., & Sandhu, R. (2008). Toward a Usage-Based Security Framework for Collaborative Computing Systems. *ACM Trans. Inf. Syst. Secur., 11*(1), 1--36.