On the Definition of Access Control Requirements for Grid and Cloud Computing Systems

Antonios Gouglidis and Ioannis Mavridis

Department of Applied Informatics, University of Macedonia,
156 Egnatia Street, 54006, Thessaloniki, Greece
{agougl,mavridis}@uom.gr

Abstract. The emergence of grid and cloud computing systems has introduced new security concepts, so it requires new access control approaches. Traditional systems engineering processes can be enriched with helper approaches that can facilitate the definition of access control requirements in such complex environments. Looking towards a holistic approach on the definition of access control requirements, we propose a four-layer conceptual categorization. In addition, an example is given so that to demonstrate the utilization of the proposed categorization in a grid scenario for defining access control requirements, and evaluate their fulfilment vis-à-vis contemporary employed access control approaches.

Keywords: cloud computing, grid computing, access control, security requirements engineering

1 Introduction

Grids [1] and clouds [2] are two promising computing technologies, which in the recent years have become the focal point of the science communities and the enterprises. However, contemporary implementations are characterised by an intrinsic complexity due to lack of standards, ad-hoc implementations and use of approaches which are not specifically designed for these environments. Access control is such an example. Security system designers need to define access control approaches that can cope with the complexity of these environments. Systems engineering can be used as a process in their development; however, an approach that incorporates the characteristics of these systems is non-existent. Therefore, we identify the need for a holistic approach in access control requirements definition that will enhance and facilitate the process of their identification and consequently, result in new access control models for grid and cloud computing environments.

Concerning access control approaches in current systems, we identify two main categories. The first is the Role-Based Access Control (RBAC) [3] and the second is the Usage Control (UCON) [4], [5]. The latter subsumes the Attribute Based Access Control approach (ABAC) [6]. To the best of our knowledge, there is no standard definition of ABAC [7] and for that we omit to further analyze it. RBAC supports the principles of abstract privileges, least privilege, separation of administrative functions and separation of duties [8]. Recent research in [9] has ventured to enhance RBAC to a next-generation access control approach by introducing the ASCAA principles. ASCAA stands for abstraction, separation, containment, automation and accountability. A grid authorization system that makes use of RBAC is PERMIS [10]. UCON has introduced a number of novelties such as rights that are determined during the access of an operation, mutability of attributes and decision continuity. More characteristics are the support of authorizations, obligations and conditions. Research has been done in [5] for its use in collaborative systems. UCON has been adopted in GridTrust [11]. However, as an attribute based approach, it inherits its complexity and can be error-prone, especially in highly heterogeneous environments [12].

The remainder of this paper is structured as follows: In section 2, a conceptual categorization for grid and cloud computing systems is proposed. A motivating scenario is presented in section 3, to demonstrate the identification of access control requirements based on the proposed categorization and assess their implementation in relation to contemporary approaches. Finally, the paper is concluded in section 4.

2 The Proposed Conceptual Categorization

Current grid systems have been categorized and classified in the existing literature based on different criteria, either qualitative or quantitative. Most of these categorizations are quite vague, in regard to the limits of each category [13]. This makes the definition of access control requirements a difficult process. Moreover, despite the use of generic systems engineering processes, security engineers lack a helper abstract model able to enhance and facilitate the definition of access control requirements. As a solution, a conceptual four-layer categorization that is capable of defining and evaluating security requirements is proposed.

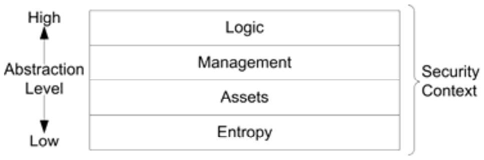

Fig. 1. Conceptual categorization layers.

As depicted in figure 1, the proposed conceptual categorization is based on four abstraction layers: entropy layer, assets layer, management layer and logic layer. The differentiation from generic security engineering approaches is that, in our case, factors that affect the security of the systems are mainly considered in their categorization. Briefly, the conceptual categorization identifies and groups security requirements into discrete layers of different abstraction levels. The abstraction level refers to the ability of a layer to identify requirements in different breadth and depth. The entropy layer identifies requirements from the dispersion of the objects in a system and the assets layer from the type of shared objects within the boundaries of the entropy layer. The next layer defines requirements from policy management and the logic layer incorporates requirements that are not handled by the former layers.

2.1 Entropy Layer

Entropy is a layer capable of capturing the abstract characteristics of a system accrued from its distribution. The term entropy refers to the virtual and geographic distribution of a system in association with the factor of time. Current classifications of grid systems are static and based mostly on the geographic distribution of their resources [14] or on their size [15]. The entropy layer uses existing grid distribution characteristics and the incorporated time factor in order to identify changes in the number of participating objects as well as alterations of them over time. Figure 2 depicts the entropy layer classification.

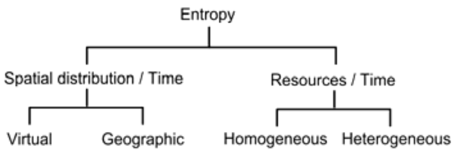

Fig. 2. Entropy layer classification.

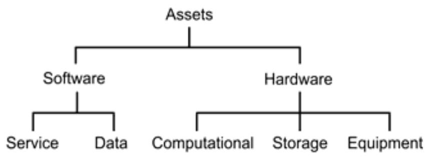

Fig. 3. Assets layer classification.

In order to illustrate the flexibility of this layer in capturing the distribution characteristics of a system, we provide the examples of a volunteer desktop grid project named SETI@home [16] and of a science grid project named EGEE [17]. The data used to plot the graphs in figures 4 and 5 were taken from [18] and [19], respectively. The entropy lines represent the fluctuations in number of the involving objects, in relation to the spatial distribution over time. Issues like the authentication of the distributed objects can be examined under the entropy layer.

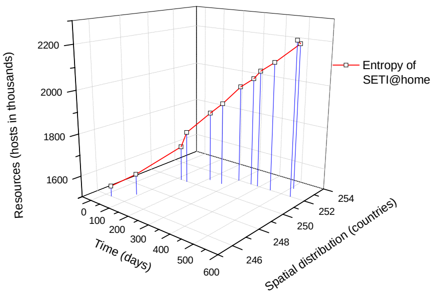

Fig. 4. Entropy of SETI@home.

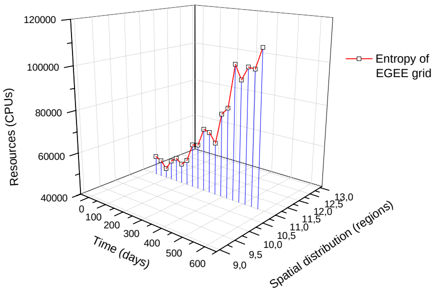

Fig. 5. Entropy of EGEE grid infrastructure.

2.2 Assets Layer

The assets layer, as illustrated in figure 3, is used to wrap all the assets in a system. As an asset we define any sharable object in a system. In a grid system, an asset can either be of software or hardware type. The proposed classification in the assets layer is partially based on the existing literature [15], [20], and [21].

Under the software class, we further divide assets into two subclasses, these of service and data. Services have been used in the grid due to the adoption of service oriented architecture. The provision of fine-grained assets such as data is vital in a grid system. The requirement of sharing information at data-record-level in a database management system among a number of users is an illustrative example [22].

Similarly, we divide the hardware class into three distinct subclasses, those of computational, storage and equipment. Examples of computational assets are the usage of CPU or RAM of a system. Concerning the storage assets we mean the usage of raw storage space for the saving of data. Last but not least, an equipment is an asset that is usually used as an input or output device within a grid system.

2.3 Management Layer

The management layer is used to fulfil the need for capturing the security issues raised from the management of policies among the objects in a system as well as from trust relationships. Figure 6 illustrates the proposed classification.

The distribution level of a system, as defined in the entropy layer, affects the management of its policies. Usually, science grids with a high level of distribution require de-centralized management and vice-versa. Peer-to-peer networks are an example of de-centralized management, too. On the contrary, enterprise applications using cloud computing technologies require centralized management.

The enforcement of several management operations is another factor that needs to be further classified. Here, we identify two classification levels, that of static and dynamic enforcement. By static we refer to operations that can take place before and after the execution of a number of actions performed on an object by a subject. The dissimilarity between static and dynamic enforcement of operations is that, in the latter, the policy enforcement can also take place during the execution of an operation.

The automation level pertains exclusively to the intervention of an administrator to the management routines. Fully automation means that management is done by the system itself [23]. Semi automated systems are those that are partially managed by the system itself and the administrators. However, cases still exist where management automation is completely absent. Such systems are solely administered by humans. Operations, such as problem identification, conflict resolution and revocation of privileges should be considered under the management layer.

Finally, trust management must be taken under consideration in the process of security engineering. The life cycle of trust includes the creation, negotiation and management of it [24] and is considered to be an important part of security.

2.4 Logic Layer

The main concern of the logic layer is the application models and the type of their execution in a system. Based on the definition of grid and cloud computing systems and the requirements identified in the existing literature [13], [25], the classification of the logic layer as depicted in figure 7 is suggested.

The logic layer is split into two classes. The models class helps in the identification of security requirements that can rise from the nature of the application being executed in the grid. We propose a further classification of it into business and science applications. However, in both subclasses similar requirements exist. Usually the support of collaborations, workflows and co-operations fall under science projects. In addition, technologies such as infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS) are enterprise examples, which are usually met in cloud computing systems [2].

Furthermore, a classification of the execution mode of a grid or cloud application into batch and interactive can be made. Science projects usually require a batch-based execution of applications to provide results through the computation of data. In contrast, most business applications require an interactive environment to tackle the highly dynamic enterprise environment.

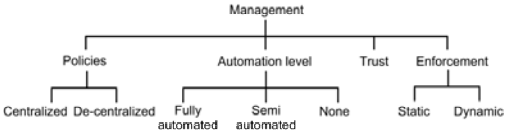

Fig. 6. Management layer classification.

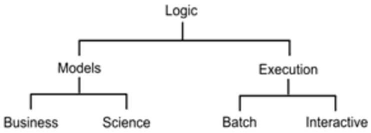

Fig. 7. Logic layer classification.

3 Identifying Access Control Requirements

A generic grid access control scenario, enriched with some of cloud computing characteristics, follows. By applying our proposed conceptual categorization, we demonstrate the process of identifying access control requirements for the scenario.

The operational environment is illustrated in figure 8. The Virtual Organization (VO) is comprised of individually administered domains, which can dynamically join in or quit the collaboration. Users from the participating domains can request on demand usage of grid services. More precisely, the VO is comprised of companies A and B, represented respectively by domains A and B. An Application Service Provider (ASP) is a corporate organization that can share a number of pay-per-use services. A complementary entity provides a computational computing infrastructure (CCI). Users Alice from company A and Bob from company B require collaborating and producing a statistical analysis on a subset of their data. Figure 9 illustrates the information flow between the involving entities, on VO level. Users can request capabilities from their local domain, collaborate with other users, manage their data and request on demand the use of services. Services can be administered and also permitted to use segments of users' data via a delegation mechanism. In turn, a service can submit data segments to the CCI. Services can be provided as composite services, thus requiring automatically re-delegation mechanisms among the involving services. The system may prompt users for parameters completion during an operation, whose life span can vary, depending on the complexity of the computations. At the CCI, the resource owner can alter any access control policy for any resource and user at runtime. For instance, let's assume a policy that permits the execution of the statistical analysis application at the CCI for both Alice and Bob. However, prior to the statistical analysis completion, the resource owner restricts Bob's access with a new policy permitting him to use CPU cycles only when CCI is idle, thus leading to a delay of his computations, until the completion of Alice's job.

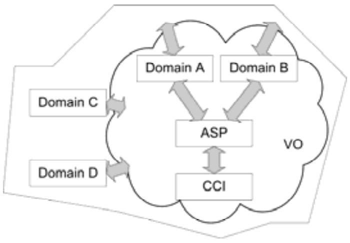

Fig. 8. Operational environment.

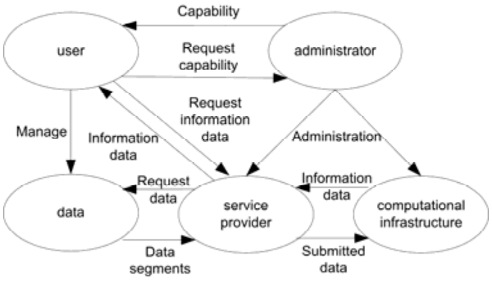

Fig. 9. Flow of information in a VO.

Entropy requirements: The virtual distribution level of the system is low since there is only one formatted VO. On the other hand, the geographic distribution level that depends on the number of the participating domains can be high, which additionally entails heterogeneity issues. In order for the access control system to limit access to participating objects, it must be able to successfully authenticate them, since domains may make use of different authentication protocols. Furthermore, since the VO formation is not static, the access control system must continually observe all kinds of modifications.

As far as this scenario is concerned, UCON can cope with the complexity of the entropy layer. This is due to the support of attribute repositories that can be dispersed across the domains. The use of attributes also overcomes the heterogeneity issues. UCON is flexible enough to deal with the dynamic changes in the number of participants during the collaboration. On the contrary, RBAC handles better centralized architectures where participants are known a priori. Therefore, RBAC appears to be inappropriate for the current scenario and layer.

Assets requirements: Access control must be enforced on different types of assets. The scenario considers fine-grained access control on data, since it requires sending for computation only segments of users' data. The ASP provides a number of services and the CCI a number of hardware resources. Access control for both service and hardware level can be characterized as coarse-grained, since the scenario describes only permission, denial and restriction of access upon them. Thus, the access control model must be able to enforce fine-grained access control on data and coarse-grained on services and hardware resources, respectively.

UCON can handle fine-grained access control because of attributes. RBAC is rather coarse-grained compared to the former approach when it comes to assets definition. Assets, in RBAC, are grouped under roles and in order to become more granular, the assignments must be split into more. However, the use of context variables in known RBAC variations [26] overcomes such limitations. Once again, the UCON approach is preferred, since it supports natively fine-grained access control, and because it is easier to modify in order to support course-grained access control than for RBAC to support fine-grained access control.

Management requirements: In this scenario, a number of services uses segments of users' data and submits them at the CCI. This requires a delegation mechanism. Thus, the access control system must be able to support delegation of access rights from grid users to the ASP and CCI. A security issue is that of delegated rights revocation. We assume that delegated rights must be revoked after the completion of a job or on demand by the user. The former requirement demands from the access control system an automation level and the latter to apply changes dynamically. Furthermore, trust relationships must exist between the involving parties. In another use case, a user from an unknown domain may request to use a service. The access control system must be in position to decide whether to deny or provide limited access to the user. Policy conflict resolution must also be examined when composite services exist. This is required due to the inheritance of authorization rights amongst services.

Delegation of rights and trust relationships are supported by both access control approaches. Policy conflict resolution can be cumbersome for UCON, and easier for RBAC. In this case, a sensible choice would be the selection of RBAC, since it supports improved administrative capabilities compared to UCON. Revocation of user assignments, hierarchies and temporal constraints are some of RBAC's virtues making it superior in comparison to UCON.

Logic requirements: During Bob's collaboration with Alice, his access at the CCI has been restricted by the resource owner. This requires an access control system that must support dynamic collaborations. Occurring interactions between the user and the application require from the access control system to support them as well. More requirements are the support of stateful sessions due to long lived transactions and decomposition of composed services.

UCON is the only approach capable of supporting interactive environments via continuity of decisions and mutable attributes. Moreover, the use of obligations can handle well a number of business requirements. However, topics like service decomposition are left intact from all access control approaches.

4 Conclusions

Classic systems engineering processes have been used in the definition of access control requirements for grid and cloud computing systems. In many cases, this led to the adoption of existent or modified access control approaches. Furthermore, contemporary implementations seem to be inadequate in fulfilling the new security requirements set by these systems. Stemmed from the need to design new access control approaches and contemplating a helper holistic approach in defining security requirements, we recommended a four-layer conceptual categorization for grid and cloud computing systems. Its layered scheme is able to enhance and facilitate the process of defining access control requirements. We anticipate the proposed conceptual categorization to serve as a foundation in defining access control requirements and thus resulting in new access control models for grid and cloud computing systems.

References

1. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the grid - enabling scalable virtual organizations. International Journal of Supercomputer Applications 15 (2001)
2. Foster, I., Zhao, Y., Raicu, I., Lu, S.: Cloud computing and grid computing 360-degree compared. Grid Computing Environments Workshop, 2008. GCE '08 1--10 (2008)
3. Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R.: Proposed NIST standard for role-based access control. ACM Trans. Inf. Syst. Secur. 4(3) 224--274 (2001)
4. Sandhu, R., Park, J.: Usage control: A vision for next generation access control. Computer Network Security 17--31 (2003)
5. Zhang, X., Nakae, M., Covington, M.J., Sandhu, R.: Toward a usage-based security framework for collaborative computing systems. ACM Trans. Inf. Syst. Secur. 11(1) 1--36 (2008)
6. Yuan, E., Tong, J.: Attributed Based Access Control (ABAC) for Web Services. Proceedings of the IEEE ICWS, IEEE Computer Society, 561--569 (2005)
7. Busch, S., Muschall, B., Pernul, G., Priebe, T.: Authrule: A generic rule-based authorization module, DBSec, Springer (2006)
8. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. IEEE Computer 29(2) 38--47 (1996)
9. Sandhu R., Bhamidipati V.: The ASCAA principles for next-generation role-based access control. Availability, Reliability and Security, 2008. ARES 08 xxvii--xxxii (2008)
10. Chadwick, D.W., Otenko, A., Ball, E.: Role-based access control with X.509 attribute certificates. IEEE Internet Computing 7(2) 62--69 (2003)
11. GridTrust: Gridtrust, http://www.gridtrust.eu/gridtrust (2009)
12. Priebe, T., Dobmeier, W., Kamprath, N.: Supporting attribute-based access control with ontologies. In: ARES '06: Proceedings of the First International Conference on Availability, Reliability and Security, Washington, DC, USA, IEEE Computer Society 465--472 (2006)
13. Alexander Kipp, S.W., Lutz Schubert, R.P., Horst Schwichtenberg, C.T., Karanastasis, E.: A new approach for classifying grids. Technical report, BEinGRID (2008)
14. Gridipedia: Types of grid, http://www.gridipedia.eu/types-of-grids.html (2009)
15. Kurdi, H., Li, M., Al-Raweshidy, H.: A classification of emerging and traditional grid systems. IEEE Distributed Systems Online 9(3) 1 (2008)
16. SETI@home: http://setiathome.ssl.berkeley.edu/ (2009)
17. EGEE: Enabling grids for e-science (EGEE), http://eu-egee.org/ (2009)
18. BOINC: Boinc all projects statistics - distributed computing statistics, http://www.allprojectstats.com/ (2009)
19. Gridmap: Gridmap visualizing the “state” of the grid, http://gridmap.cern.ch/gm (2009)
20. Green, D.: Grid technology. The future of the internet? The future of it?, https://ludit.kuleuven.be/nieuws/pdf/grid.pdf (2002)
21. Krauter K., Buyya R., M.M.: A taxonomy and survey of grid resource management systems for distributed computing. Softw. Pract. Exper. 32(2) 135--164 (2002)
22. Broadfoot, P.J., Martin, A.P.: A critical survey of grid security requirements and technologies. Technical Report RR-03-15, Oxford University Computing Laboratory (2003)
23. Kephart, J.: Research challenges of autonomic computing. Software Engineering, ICSE 2005. Proceedings. 15--22 (2005)
24. Chakrabarti, A.: Grid Computing Security, Managing Trust in the Grid. Springer Berlin Heidelberg (2007)
25. Altmann, J., Veit, D.: Grid Economics and Business Models, 4th International Workshop, GECON 2007, Rennes, France, August 28, 2007, Proceedings. Volume 4685 of Lecture Notes in Computer Science, Springer (2007)
26. Tolone, W., Ahn, G.J., Pai, T., Hong, S.P.: Access control in collaborative systems. ACM Comput. Surv. 37(1) 29--41 (2005)