A Ranking Based Cross Domain Role Mapping and Authorization Architecture for Grid Computing Systems

T.L.Prasanna Venkatesan, NIT Warangal, prasanna@computer.org
G Geethakumari, NIT Warangal, geethamaruvada@gmail.com
Srikanth Jampala, University of Hyderabad, srikanth_hcu05@yahoo.co.in
Dr Atul Negi, University of Hyderabad, atul.negi@gmail.com

Abstract

Secure interoperability has been a growing concern for multi domain computing systems like grids. Collaboration in such a diverse environment requires integration of all local policies to compose a global access control policy for controlling information and resource. Though the classical Role Based Access Control (RBAC) is an effective access control standard, it does not address the issue of resolving a local role into a global role. Here, we present an architecture based on RBAC standard which can establish role equivalence among the domains by mapping a local domain role to its equivalent global role. We use the approach of weighted ranking for the same. The final authorization decision is made based on the mapped global role ranking.

1.0 Introduction

Grid computing is regarded as an emerging technology of immense potential in both industry and academia. A grid environment supports resource sharing with scalability and heterogeneity. Security is of prime concern while sharing data and computational resources in a grid. Grid security consists of different measures such as naming and authentication, trust and policy management, authorization etc [1]. A grid security procedure starts when a user initiates a resource access. Controlling the access to any resource primarily involves two phases namely authentication and authorization. Existing grid security mechanisms are based on authentication and do not lay emphasis on authorization [3]. We address this problem of authorization in grid system environments. Mandatory authorization schemes were initially used to define access control policies. The evolution of RBAC as a reliable standard for single enterprises motivated researchers to think of ways in which it could be incorporated into grid environment. A grid is often viewed as a logical organization formed of multiple physical organizations or enterprises and hence the integration of RBAC into grid is only a logical extension of the standard RBAC implementation [2]. For a grid system usually formed by multiple domains which are maintained by different companies, organizations or institutions, interoperability is a major issue. The *role*, which is the basis for RBAC, signifies different meanings in different organizational contexts. Here we arrive at a mechanism by which we can map the role of one enterprise into its new semantics in another enterprise.

This paper has been organized as follows. Section 2 gives the related research and also the motivation for our work. In section 3, we propose a cross-domain authorization architecture based on ranking of roles. In section 4, we give an insight into the implementation details and also the possible enhancements. Section 5 summarizes our work.

2.0 Related Research and Motivation

Though there has been considerable work in the area of grid security, the emphasis has been on authentication. Grid access control and authorization are still open research issues, which needs much attention. In this paper, we focus on the issue of Interoperability between Different domains when it comes to the issue of authorization. James B.D.Joshi et.al[7] suggest an integer programming (IP) based approach for secure interoperation involving RBAC policies. But their work does not reflect the distinct characteristics and requirements of grid authorization. Another proposed approach is user-credential based role-mapping [8] where by a user's credentials associated with the role form the basis for role-mapping. We believe, this is a premature and non-standard way of mapping roles, as the fundamental unit of RBAC is a *role* itself and hence cannot use its associated credentials as the sole criteria for role mapping. Liang Chen et.al, [9] have proposed an inter-domain role mapping technique based on the principle of least privilege. They suggest a minimal cardinality for a role across a domain to avoid misuse of access. This again, does not suit dynamic and heterogeneous environments like the grids. Some of the existing grid authorization mechanisms are Permis, Akenti, Shibboleth, VOMS, CAS[4]. Though Permis, Akenti and CAS introduce the concept of roles in a grid environment, they are not role-based implementations like RBAC. Also they lack the flexibility of RBAC and are static in nature.

The absence of a standard role-mapping mechanism to address the grid authorization and access control issues, combined with the fact that the present form of RBAC for single enterprises cannot support grid access control motivated us to develop a new architecture which can truly reflect a multi-domain grid access environment.

3.0 Cross Domain Authorization Architecture

The present RBAC standard uses *role* as the basic unit of authorization [5]. The standard incorporates features such as role hierarchy, static and dynamic separation of duties and so on. In an RBAC environment, a user will be assigned roles based on his responsibilities in the organization. For example, in a University domain, the potential roles could be Professor, Associate Professor and so on. For an industrial domain the roles could be CEO, General Manager, Manager and so on. Therefore, the semantics of roles in a given domain will not have relevance in another domain. Thus, the role in an organization has to be mapped to its corresponding meaning in another if cross domain resource sharing is to be made possible. We address this issue with a ranking based weighted role approach. Our architecture enables mapping of a local role to a global ranking. We consider a nested and hierarchical domain architecture reflecting the real life grid scenario. The roles in a particular domain follow a local role hierarchy. The cross domain architecture shown in Fig 1 consist of the following components.

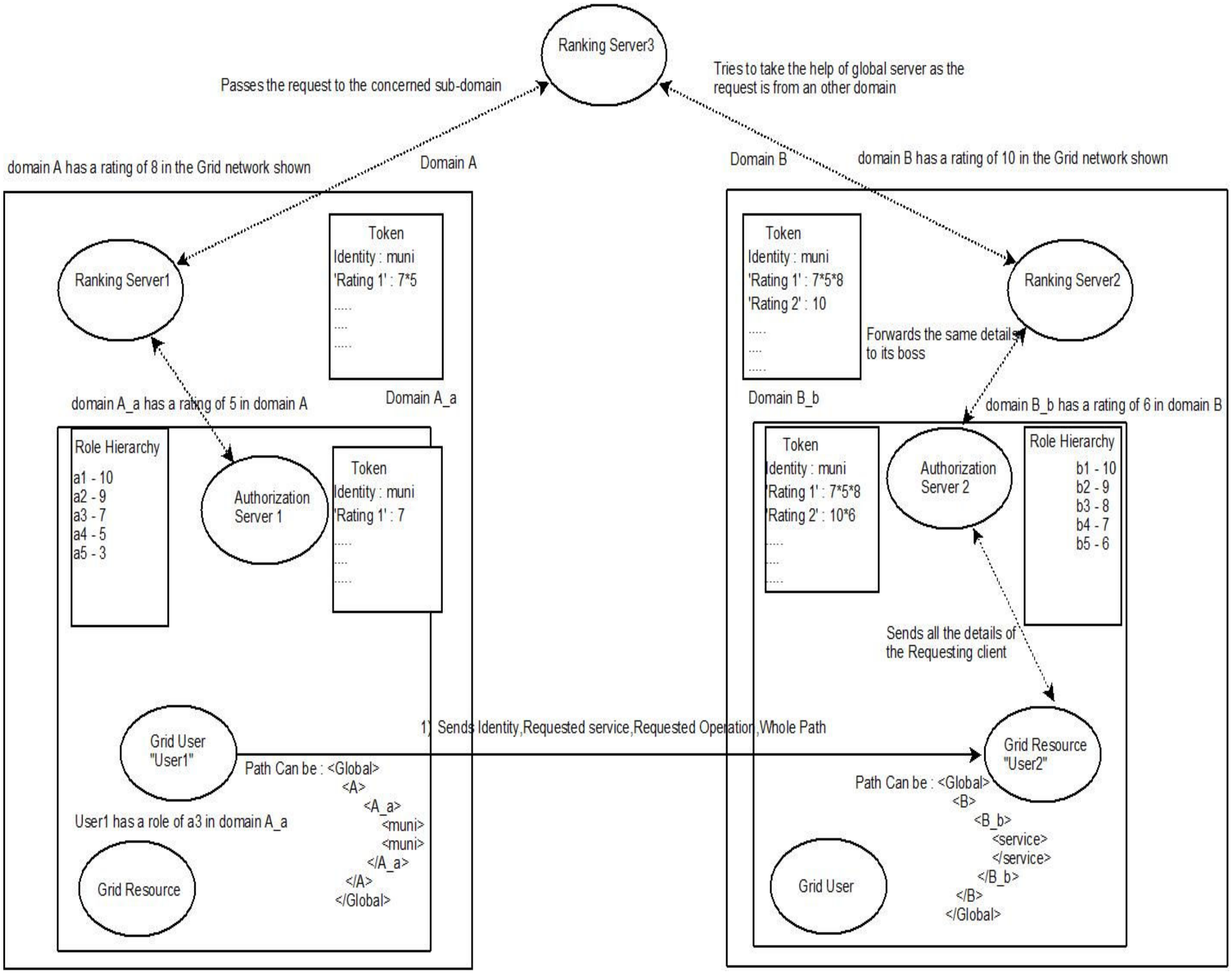


Fig 1 – Cross Domain Role Mapping Architecture

- At the organizational level we consider two Domains A and B
- Domains A and B consist of sub-Domains A_a and B_b
- Further Domains A_a and B_b have grid nodes as their constituents
- Authorization Server1 (AS1) is the local Authorization server for grid nodes from Domain A_a
- Authorization Server2 (AS2) plays a similar role in Domain B_b
- Ranking servers RS1 and RS2 for the two respective Domains A and B store the rating of the sub-domains

The whole grid environment is separated into different domains and sub domains as shown. The sub domains in every domain are given ranks on a scale of 10. The roles in a local domain are also ranked on the scale of 10 based up their hierarchy. The role in a local domain gets translated to a global ranking based on the value of its own ranking and also the rank of its ancestor domains up the tree. Here, we give a generic algorithm for authorization procedure and role mapping

```
Algorithm : Authorize
Input : User Credentials
Output : Grant/Deny a Token
If(user is from a different domain)
    call Rolemap(credentials)
    find minimum rated role to access resource in the domain
    find the normalized global rating of that role GLR
    retrieve the Normalized global rating of user from certificate GLU
    if GLU>=GLR
        return accept
    else
        return deny
else
    retrieve user roles, rating and other credentials from database
    if(resource is from other domain)
        create token T containing the user details
        return T
    else
        find minimum rated role MR to access resource in the domain
        if(MR > users role rating) return accept
        else return deny
```

```
Algorithm : Rolemap
Input : User Credentials / Token
Output : Token containing updated values
if(user is from different domain or sub-Domain)
    Token T = call Rolemap(credentials)
    Add rating of the sub-Domain to the Global rating of the resource in the Token
    return T
if(user is from same Domain)
    Token T = call Authorize(Credentials)
    Add Rating of Sub – Domain to T
    Return T
```

The role-mapping architecture is a weighted tree and we arrive at the globally mapped role by comparing the global rank of a role with respect to its first common ancestor as shown below. The grid user is granted/denied access to the requested resource through the following procedure.

1. The user from Domain A_a sends his identity, path, the requested resource and also requested operation to Domain B_b
2. The user in domain b forwards the details to its Authorization Server(AS2) and awaits a deny or grant
3. The Authorization Server AS2 executes the algorithm *Authorize* as shown above
4. The credentials are passed up the hierarchy for role mapping as the user is from a different domain

5. The credentials reach AS1 where the attributes like the user's role, rating etc are retrieved and a Token is created.
6. The Token follows the same path in reverse and at every stage, the rating of the domain gets weighted
7. AS2 gets the final version of the Token. It normalizes the user rating to 7*5*8 / 1000 = 0.280
8. AS2 finds the minimal rating of a role needed to access the resource which is 5. So the Normalized rating of the resource is 10*8*5 / 1000 = 0.400
9. AS2 can integrate this ranking comparison with other local policies to either deny or grant access to the user

4.0 Implementation Details and Future work

The Authorization Servers (AS1 and AS2) mentioned in Fig 1, work based on the architecture shown in Fig 2. It follows the algorithm shown as above. The policy Enforcement Point (PEP) takes the credentials of the user and creates a request for authorization in the XACML format[6]. This request is forwarded to the Policy Decision Point (PDP). The PDP checks the policies and sends back a reply in the same XACML format. PEP acts based on this reply and sends either *deny* or *grant* to the request. We have implemented this architecture for a single domain grid enterprise with indirect authorization (delegation) mechanisms.

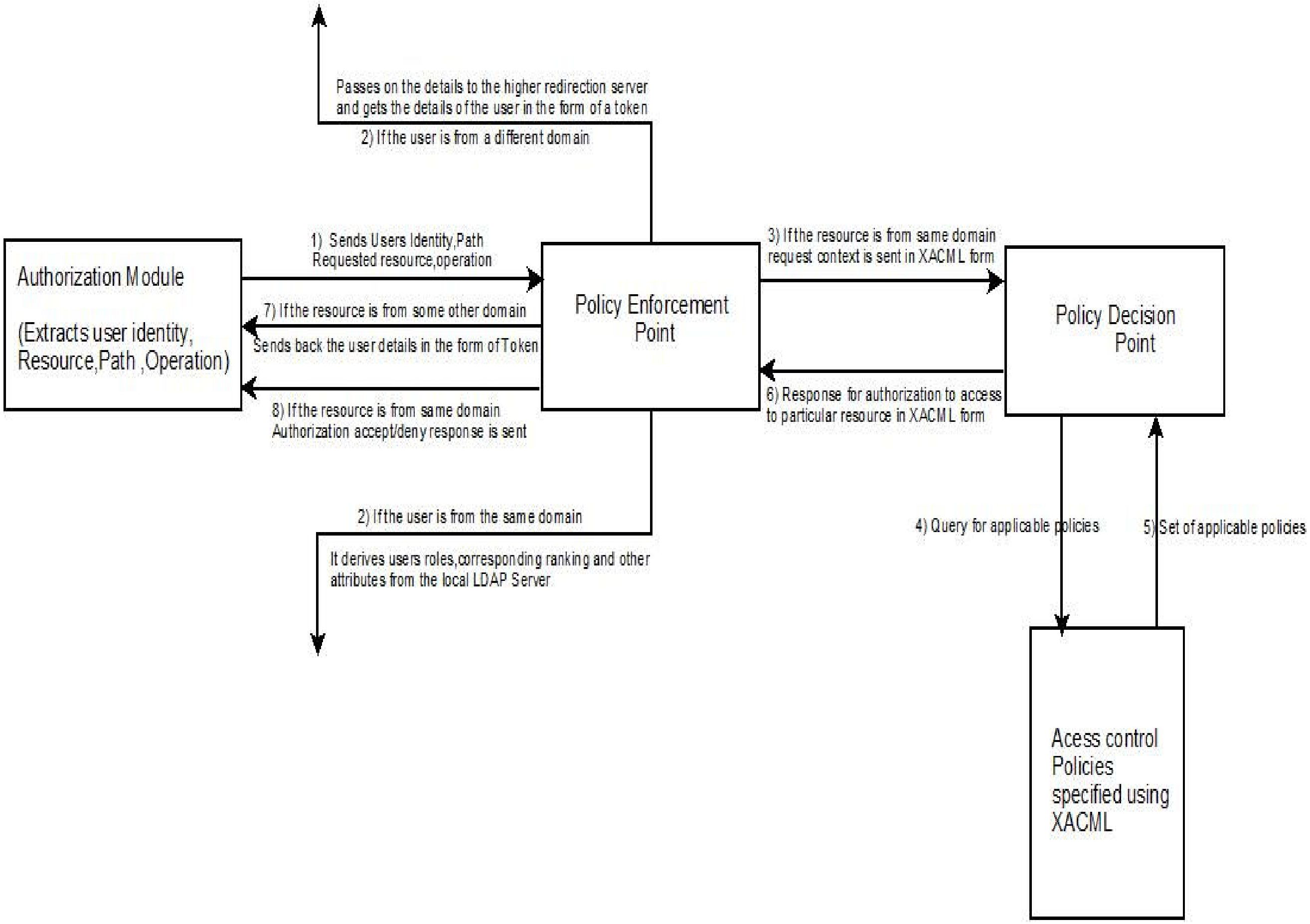
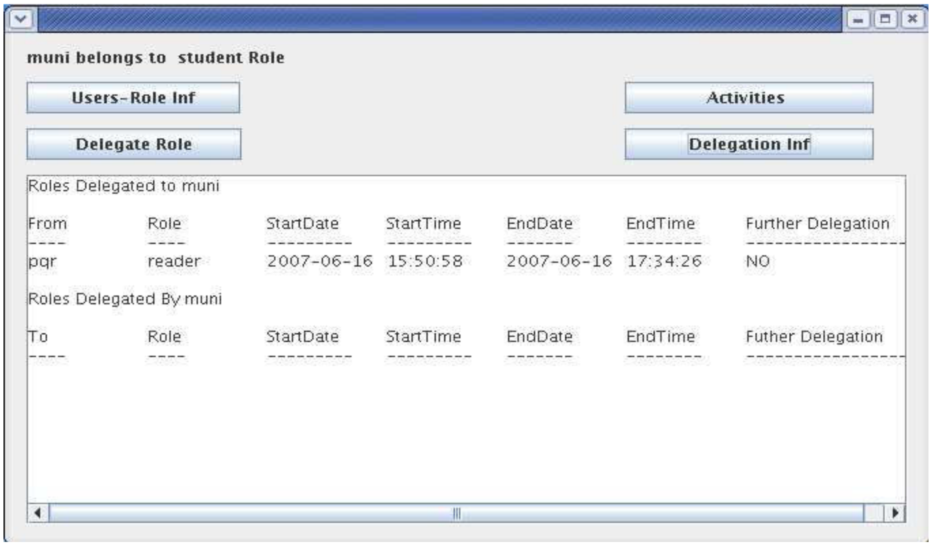

Passes on the details to the higher redirection server
and gets the details of the user in the form of a token
2) If the user is from a different domain
Authorization Module
(Extracts user identity,
Resource,Path ,Operation)
1) Sends Users Identity,Path
Requested resource,operation
7) If the resource is from some other domain
Sends back the user details in the form of Token
8) If the resource is from same domain
Authorization accept/deny response is sent
Policy Enforcement
Point
3) If the resource is from same domain
request context is sent in XACML form
6) Response for authorization to access
to particular resource in XACML form
Policy Decision
Point
2) If the user is from the same domain
It derives users roles,corresponding ranking and other
attributes from the local LDAP Server
4) Query for applicable policies
5) Set of applicable policies
Acess control
Policies
specified using
XACML

Fig 2 –Authorization Server Architecture

Fig 3 shows one of the screen shots of the implementation. We plan to extend this implementation for cross-domain authorization in the near future. We also plan to incorporate Role Based Delegation and Revocation models for multi-domain grid environments.



muni belongs to student Role

```
Roles Delegated to muni

From        Role        StartDate    StartTime    EndDate      EndTime     Further Delegation
----        ----        ---------    ---------    -------      --------    ------------------
pqr         reader      2007-06-16   15:50:58     2007-06-16   17:34:26    NO

Roles Delegated By muni

To          Role        StartDate    StartTime    EndDate      EndTime     Futher Delegation
----        ----        ---------    ---------    -------      --------    ------------------
```

Fig 3 – Implementation Details for Single Domain Grid Enterprise

5.0 Conclusion

The proposed role mapping architecture makes it possible for grid nodes across domains to interact and authorize users for resource access. This architecture supports reusability of role-ranking mechanism as the token once created between two nodes can be used for future interactions between them. Domains can also formulate additional access control policies like giving access to only semantically sensible organizations for that particular resource, apart from just comparing the global ranking of the user.

References

[1] Marty Humphrey, Mary R Thomson and Keith R Jackson, “Security for Grids”, *Proceedings of the IEEE,* Vol 93, No 3, pp 644-652, March 2005.

[2] Weizhong Qiang, Hai Jin, Xuanhua Shi, Deqing Zou, and Hao Zhang, “RB-GACA: A RBAC Based Grid Access Control Architecture”, LNCS 3032, pp. 487–494, 2004, .Springer-Verlag Berlin Heidelberg.

[3] Jiageng Li, David Cordes, “A scalable authorization approach for the Globus grid system”, *Future Generation Computer Systems* Vol 21, pp 291–301, 2005.

[4] R. Alfieri, R .Cecchini et al, “From Gridmap-File to VOMS: Managing Authorization in a Grid Environment, *Future Generation Computer Systems*, Vol 21, pp 549–558, 2005.

[5] Ravi S. Sandhu, David F. Ferraiolo, D. Richard Kuhn, “The NIST Model for Role-Based Access Control: Towards a Unified Standard”, *ACM Workshop on Role-Based Access Control*, 2000, pp 47-63

[6] eXtensible Access Control Markup Language, Version 2.0, OASIS Standard, February 2005, http://docs.oasisopen.org/xacml/2.0/access control-xacml-2.0-core-spec-os.pdf

[7] Basit Shafiq, James B.D. Joshi et.al, “Secure Interoperation in a Multidomain Environment Employing RBAC Policies”, IEEE Transactions on Knowledge and Data Engineering, Vol 17, No.11, November 2005.

[8] Ajith Kamath, Ramiro et.al, “ User-Credential Based Role Mapping in Multi-domain Environment”, *Proceedings of the Privacy, Security, Trust* (PST), 2006.

[9] Liang Chen and Jason Crampton, “Inter-domain Role Mapping and Least Privilege”, *Proceedings of the Symposium on Access Control Models and Technologies* (SACMAT), 2007