CHAPTER - 5

UNIFICATION OF SCALABLE AND DYNAMIC ACCESS CONTROL MECHANISM IN GRID COMPUTING

This chapter describes the experimental schema of enhancing the unification of scalable and dynamic access control mechanism in grid computing environment.

5.1. INTRODUCTION

The growth of Grid services and service providers are directly proportional due to the immense requirement of shared resource accessing feature among the Grid service consumers. Grid systems are a mixture of autonomous domains, which are frequently open and dynamic. Grid systems treats large number of users with its services, who are often changeable, and different domains has their individual policies.

The existing identity based models are not scalable, closed, infeasible and inflexible. Scalability and the dynamicity for Grid control services have unluckily not extended well to grid access-control policies and mechanisms.

This experimental schema deals with the issues raised by scaling up of measures such as number of users, protocols, applications, network elements in a grid topological constraints, and functionality expectations along with the enhancement in dynamic access control.

The proposed enhancement method is a unification approach with the unique individual implementation towards the fine tuning of scaling and dynamic access

---

improvements. The results and discussions of our proposed method lead to the implementation of enhancement of scalable and dynamic access control mechanism in grid computing.

A grid 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 [15].

A grid 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 grid 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 [9]. 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 [6].

A grid resource in a Grid environment can be any sharable hardware or software asset in a domain and upon which an operation can be performed [5].

Grid Access control’s role is to control and limit the actions or operations in the Grid system that is 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 [2].
A Grid access control mechanism policy [3] 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 [8], which is capable of preserving support for the reasoning of the system policies through a conceptual framework [1]. The Grid service architecture is represented in Figure 1.1.

![Grid Service Architecture](image)

Figure 5.1: Grid Service Architecture [10]

### 5.2. METHODOLOGY

**Scalability Access Control Issue**

Scalability represents improving the quantitative value such as user count, Rules and constraints for communication, user interface, network components, topological design, and desired procedures.
The enhanced scalable system must be implemented, so that the grid management, grid maintenance and grid operational costs must be inversely proportional to the addition of the quantitative values for scalability in a positive way [7].

**Dynamic Access Control Issue**

Dynamic access control represents the access control mechanism that dynamically adjusts *role assignments* to users based on their context information.

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. However, they are applied during the run-time.

**Enhanced Unification of Scalability and Dynamic Access Control Issue**

The individual Scalability enhancement towards dynamic nature of access control due to multiple roles for a single user affects the Session arrival count, Session arrival frequency, Session duration extension and Session topology.

5.3. **PROPOSED EXPERIMENTAL SCHEMA FOR UNIFIED SCALABLE AND DYNAMIC ACCESS CONTROL**

The proposed methodology not only focuses on scalability at the architectural level where management, maintenance and operational costs do not increase as the number of system components (users, applications, policies and enforcement points) increases along with the dynamic access control through multiple roles for the user access simultaneously towards the Grid service environment.
The proposed methodology for the dual enhancement in Grid scalability and dynamic access schema diagram is as follows,

![Diagram of Proposed Schema for Grid Scalable & Dynamic Access Enhancement](image)

**Figure 5.2: Proposed schema for Grid Scalable & Dynamic Access Enhancement**

5.3.1. Phase-1: Scalability Enhancement

5.3.1.1 Proposed Hybrid Heuristic Architecture

The development of Hybrid architecture is based on the Service type categories. The services are divided into three types such that exclusive services are superior to the prime services which are superior to the General basic services. The following Figure 3.2 represents the hybrid architecture with different user service preferences.
5.3.1.2 Proposed Persuasive Administration Structure:

The enhanced Grid scale affects the ability of human administrators to maintain its complexity. The proposed methodology

The assignment of different administrators towards different services is the main focus. The assignment is based on the capability of handling the request and with minimized cost structure. The following Table 3.1 represents the multiple administrators' assignment with its efficiency based on the grid service access.

<table>
<thead>
<tr>
<th>Service Request</th>
<th>User/service: Administrators</th>
<th>Efficiency improvement ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>No of users &lt; Resource</td>
<td>1:1</td>
<td>0%</td>
</tr>
<tr>
<td>No of Protocols &gt; Resource</td>
<td>1:2; (Max 2 protocol only)</td>
<td>100%</td>
</tr>
<tr>
<td>Topology=low/medium</td>
<td>2:3; (finite topology-bus)</td>
<td>150%</td>
</tr>
<tr>
<td>Topology=Hybrid</td>
<td>1:((m/2); (Two layers only)</td>
<td>M*50 %</td>
</tr>
<tr>
<td>Complex user interface</td>
<td>1:n; (n number of applications)</td>
<td>N*100 %</td>
</tr>
</tbody>
</table>
5.3.1.3 Proposed Time Division Credential System

Short-term credentials are provided with Time stamps in order to preserve the security along with the user refreshing frequently with the Grid server.

The following Figure 3.3 represents the Timestamp based survival for different user services...

Figure 5.4: Proposed Time Stamp division credential System

5.3.1.4 Proposed Safety Credential system:

Implement different access keys for different services for the same user also enhances the security. A customer can separate the permissions and revoke the access keys for individual applications if an access key is exposed. The following Table 3.2 represents the different safety access key codes with its access level representations for the web or mobile access.
Table 5.2: Safety Credential Access keys

<table>
<thead>
<tr>
<th>Access Keys</th>
<th>Access Type for Web Grid Service</th>
<th>Access Type for Mobile Grid Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0,0,0,0,0)</td>
<td>No Read, No Write, No Update</td>
<td>No Read, No Write, No Update</td>
</tr>
<tr>
<td>(1,1,1,0,0)</td>
<td>Read, Write, Update</td>
<td>No Read, No Write, No Update</td>
</tr>
<tr>
<td>(0,0,0,1,1)</td>
<td>No Read, No Write, No Update</td>
<td>Read, Write, Update</td>
</tr>
<tr>
<td>(1,1,1,1,1)</td>
<td>Read, Write, Update</td>
<td>Read, Write, Update</td>
</tr>
<tr>
<td>(1,0,0,1,0)</td>
<td>Read only</td>
<td>Read only</td>
</tr>
<tr>
<td>(0,1,0,0,1)</td>
<td>Write Only</td>
<td>Write Only</td>
</tr>
<tr>
<td>(1,1,0,0,0)</td>
<td>Read and Write only</td>
<td>No Read, No Write, No Update</td>
</tr>
<tr>
<td>(0,0,0,1,0)</td>
<td>No Read, No Write, No Update</td>
<td>Read and Write only</td>
</tr>
<tr>
<td>(1,1,0,1,0)</td>
<td>Read and Write only</td>
<td>Read and Write only</td>
</tr>
</tbody>
</table>

5.3.1.5 Test for Enhanced Scalability

The hybrid architecture reduces the architectural complexity by 66% due to the merger of 3 architectures into one, deploying role based multiple administrators reduced administration complexity with a maximum of m*100% ,time division and safety credential system improves stable security with increased scalability.

5.3.2 Phase-2 Dynamic Access Control Enhancement

5.3.2.1 Hybrid User Association

A user to user association will be permitted based on the end user if location is safe, Network Bandwidth is optimal, Decentralized or semi centralized policy architecture with the secured environment, otherwise deny the service responses to the users. The following code illustrates the Hybrid user association implementation using Grid MapleSoft Tool in C# Script.

Using System;
Using System.Data;
Using System.Collections.Generic;
Using System.Linq;
Using MSUniversityResults.Data;

namespace MSUniversityResults.Rules
{
    {
        protected override void EnumerateDynamicAccessControlRules(string controllerName)
        {
            RegisterAccessControlRule("UserID", AccessPermission.Allow, "JOHNSON", "DURAIAR");
            RegisterAccessControlRule("UserID",
                
                
                "[Location] = @Location and [Policy] = @Policy",
                "Select UserID from ResearchScholarsPrePhdViva +
                "Where Location = @Location and Policy = @Policy",
                AccessPermission.Allow,
                new SqlParam("@Location", "Tirunelveli"),
                new SqlParam("@Policy", "Decentralized"));
            RegisterAccessControlRule("UserID",
                "[Preference] = @Preference and [Trust] = @Trust",
                "select UserID from ResearchScholarsPrePhdViva " +
"where Preference=@Preference and Trust=@Trust"

AccessPermission.Allow,
new SqlParameter("@Preference", "Best"),
new SqlParameter("@Trust", "Worth");

RegisterAccessControlRule(
    "UserID",
    "[Autonomy]=@Autonomy and [Application] = @Application",
    "select UserID from ResearchScholarsPrePhdViva " +
    "where Autonomy=@Autonomy and Application=@Application",
    AccessPermission.Allow,
    new SqlParameter("@Autonomy", "Self Protection"),
    new SqlParameter("@Application", "Mobile Data");
}

5.3.2.2 Heuristic Component Interaction

High preferences user has higher priority to use the resource by changing the low accessed. Similarly no trust relationship with the user component restricts other components to access the critical service. Also in order to maintain congestion control only limited resource accessing will be permitted. The following figure 3.4 represents the sample scenario implementation for priority based service access.
5.3.2.3 Associative User Resource Collaboration

The network topology will be dynamically updated, the new resource will be dynamically included, and the user who accesses the resource can be anyone. In such a dynamic and heterogeneous environment, if the user moves to another site which is not secure, the delegate application will possible be denied to continue access some resource because the potential risk of leaking information to malicious user. The following table 3.3 represents permission access level for different resource-user access collaboration.
Table 5.3: User resource Collaboration Permission

<table>
<thead>
<tr>
<th>User Access</th>
<th>Type</th>
<th>Access Permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Same Location</td>
<td>Granted</td>
</tr>
<tr>
<td></td>
<td>Different Location</td>
<td>Granted with Minimal Time stamp</td>
</tr>
<tr>
<td></td>
<td>Mobile Data</td>
<td>Granted</td>
</tr>
<tr>
<td></td>
<td>Self-WiFī</td>
<td>Granted</td>
</tr>
<tr>
<td></td>
<td>Public-WiFī</td>
<td>Denied</td>
</tr>
<tr>
<td>Simulation</td>
<td>Distrust</td>
<td>Denied</td>
</tr>
<tr>
<td></td>
<td>Suspect</td>
<td>Denied</td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>Granted with Minimal Time stamp</td>
</tr>
<tr>
<td></td>
<td>Good</td>
<td>Granted</td>
</tr>
</tbody>
</table>

5.3.2.4 Refreshing Secure Credential System

Implement different access keys for different services for the same user also enhances the security. A customer can separate the permissions and revoke the access keys for individual applications if an access key is exposed [4]. The following table 3.4 represents refreshing credentials based on user status for security in terms of revoking the access permission.

Table 5.4: Refreshing Credentials

<table>
<thead>
<tr>
<th>Access Type using Access Key</th>
<th>Revoke Possibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home/Office System with Wired connection</td>
<td>1 %</td>
</tr>
<tr>
<td>Home/Office System with Home/Office WiFī</td>
<td>5 %</td>
</tr>
<tr>
<td>Home/Office System with Public WiFī</td>
<td>99 %</td>
</tr>
<tr>
<td>Mobile Application with Mobile Data</td>
<td>1 %</td>
</tr>
<tr>
<td>Mobile Application with Home/Office WiFī</td>
<td>5 %</td>
</tr>
<tr>
<td>Mobile Application with Public WiFī</td>
<td>99 %</td>
</tr>
</tbody>
</table>
5.3.2.5 Test for Dynamic Access Control

Consider the case studies of a single user want to access dynamically the grid computing resources which is YET TO FULL for our proposed methodology as follows,

Case-1: Privacy User having 1 MBps speed resides on semi centralized policy architecture within a secured environment. The user is a prime user with trust worthy and self configurable, able to access through application or simulation.

Case-2: Public isolated location having more than 10 MBps speed, moving towards unknown environment of decentralized policy architecture with average privileges, normal trust and self configurable, able to access through application or simulation in mobile data.

Case-3: Non secured location having <64KBps speed, resides on a centralized policy architecture within a distrust region, able to access through application or simulation using public Wifi.

The Final result is the User with Case-1 and Case-2 scenario using Application based dynamic accessing will be permitted. But Case-2 with Simulation based Dynamic Grid accessing will not be permitted.

Moreover User with Case-3 Scenario will not be permitted for dynamic access of resources in Grid computing environment.
5.3.3 Phase-3 Ensuring Scalable Integrity

5.3.3.1 Session Arrival Count

When scalable increased by m users and the dynamicity of role splits into m roles. Such that

Users = \{u_1, u_2, u_3, \ldots, u_m\} = \text{Total of m users}

Roles = \{r_1, r_2, r_3, \ldots, r_m\} = \text{Total of m corresponding roles}

Then the Session Arrival Count \( SAC = \sum_{i=1}^{m} u_i \cdot r_i \) where \( i = 1 \) to \( m \)

If additional j users appended with corresponding j roles such that \( m+j = n \), i.e. (n-m) new users scaled up. Then

The new Session Arrival Count \( SAC_{\text{new}} = \sum_{i=1}^{n} u_i \cdot r_i \) where \( i = 1 \) to \( n \). The final expansion slowly integrates with the function of Distribution, Which is equivalent to the Poisson distribution \( f(\mu, \delta) = e^{-\mu} \frac{\mu^\delta}{\delta!} \)

5.3.3.2 Session Role Updates Frequency

The minimum role update frequency = static role for m users = m

The average role of r for m users = mC_r = m! / r! (m-r)!

The Maximum role update frequency = m combinations = m!

So the frequency lies between m and m!

5.3.3.3 Session Duration Optimization

The Pareto distribution is used for the computation of extended optimal duration by using the formula

\[
f(x; \mu; \delta) = (\mu / \delta) (x / \delta) ^ {\mu-1} e ^ {-(x / \delta)}
\]
Where $\mu, \delta$ are scale and skew parameters which optimizes the duration with available skew of congestion representations.

### 5.3.3.4 Session Topology

The non scalable grid access control mechanism uses the virtual point to point topology, the static grid access control mechanism uses the virtual bus topology where the enhanced scalable grid access control mechanism uses the virtual mesh topology and the dynamic access control mechanism uses the virtual tree topology, finally the unification of enhanced scalable and dynamic access control mechanism uses the virtual hybrid topology. The following figure 3.5 represents the different topology structures for the Grid access control mechanism.

![Grid services session virtual topologies](image)

**Figure 5.6: Grid services session virtual topologies [11]**
5.4 RESULT AND DISCUSSION

5.4.1 Phase-4 Ensuring Dynamic Access Control Integrity

*Role Assignment

The role assignment operation is based on a set of role assignment rules and it respects a set of constraints imposed by the system administrator.

If user1

If role1="Director" or "Manager" then "Grant"

else if role2="Supervisor" or "Staff" then "Deny"

else "Delay"

End if

*Context Collection

It accomplishes the function of collecting a variety of data related to the user session environment such as the access request time, the user location, etc. Moreover, the Context Collecting agent remains listening to every interesting event that can occur during the user active session. It can respond instantly to the different contextual changes and notifies the others. The following table 3.5 represents the Context collection based permission access for the Office functioning scenario.

<table>
<thead>
<tr>
<th>Role</th>
<th>Location</th>
<th>Time</th>
<th>Access Salary Details</th>
<th>Otherwise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Director</td>
<td>Internal/External</td>
<td>Office Time/Free Time</td>
<td>Grant</td>
<td>Nil</td>
</tr>
<tr>
<td>Manager</td>
<td>Internal/External</td>
<td>Office Time</td>
<td>Grant</td>
<td>Report to Director</td>
</tr>
<tr>
<td>Accountant</td>
<td>Internal</td>
<td>Office Time</td>
<td>Grant</td>
<td>Report to Manager and Director</td>
</tr>
<tr>
<td>Staff</td>
<td>Internal/External</td>
<td>Office Time/Free Time</td>
<td>Deny</td>
<td>Report to Accountant and Manager</td>
</tr>
</tbody>
</table>
* Trust Evaluation:

It will be executed on evaluating the user trust level based on the collected user session information. A zero to one point scale of trust relationship excluding the extremes are listed in the Table 3.6 as follows,

<table>
<thead>
<tr>
<th>Trust Relationship</th>
<th>Permission Membership Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distrust</td>
<td>0.1</td>
</tr>
<tr>
<td>Suspect</td>
<td>0.2</td>
</tr>
<tr>
<td>Normal</td>
<td>0.5</td>
</tr>
<tr>
<td>Good</td>
<td>0.7</td>
</tr>
<tr>
<td>Trust worthy</td>
<td>0.9</td>
</tr>
</tbody>
</table>

*View Assignment

Inference system is used to decide the appropriate view to be assigned to the user role. The View Assigning agent input is its knowledge base which is made up of the facts base and the rules base. The facts base maintains the role already assigned to the authenticated requestor, the user trust level evaluated by the Trust Evaluating agent and the events that can be occurred during the life time session. The rules base presents the different constraints imposed by the system administrator on the view assignment operation and a set of assignment permissions rules represented as a sample office scenario in table 3.5.
<table>
<thead>
<tr>
<th>Role</th>
<th>View Access for Accounts</th>
<th>View Access for Future Proposed Designs</th>
<th>View Access for Merger or Demerger Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Director</td>
<td>Grant</td>
<td>Grant</td>
<td>Grant</td>
</tr>
<tr>
<td>Manager</td>
<td>Grant</td>
<td>Grant</td>
<td>Deny</td>
</tr>
<tr>
<td>Accountant</td>
<td>Grant</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>Staff</td>
<td>Deny</td>
<td>Deny</td>
<td>Deny</td>
</tr>
</tbody>
</table>

**SUMMARY**

The combined enhancement of scalable access control and dynamic access control is a complex structure if any strategy deals it with as a single overlapping, but the merger of these two strategies in grid computing with unified enhancement focuses on all the aspects in terms of scalable and dynamicity during and after the amalgamation. This paper deals with the enhancement of scalable access control mechanism in Grid computing systems comprises the Upgradable construct for the architectural design, administrative structure, and Time division credential system along with the safety constraints. The optimization of dynamic access control mechanism in grid computing comprises user association, component interaction, resource collaboration and safety service access and the unification of enhancement in scalable and dynamic access control mechanism includes the individual components of both the strategies along with the ensured enhancement in session arrival rate, session context update frequency, session duration and session topology for the combined scalable side and role assignment, context collection, Trust evaluation and view assignment for the combined dynamic side enhancements. This novel composition scaled up the different services towards safety with proper care on service guarantees.
for different role of users in dynamicity... In near future the usage of high end 128 or 256 bit THz processors utilization along with the Neural networks and Fuzzy logic based Grid service allocation methodology will make this entity an admirable one for the different category of customers with simultaneous access of different level of multiple services.