TOWARDS ENSURING SCALABILITY, INTEROPERABILITY AND EFFICIENT ACCESS CONTROL IN A MULTI-DOMAIN GRID-BASED ENVIRONMENT

Nureni A. Azeez* and Isabella M. Venter**

**Department of Computer Science, University of the Western Cape, Private Bag X17, Bellville, 7535, South Africa Email: 3008814@uwc.ac.za*
***Department of Computer Science, University of the Western Cape, Private Bag X17, Bellville, 7535, South Africa Email: iventer@uwc.ac.za*

Abstract: The application of grid computing has been hampered by three basic challenges: scalability, interoperability and efficient access control which need to be optimized before a full-scale adoption of grid computing can take place. To address these challenges, a novel architectural model was designed for a multi-domain grid based environment (built on three domains). It was modelled using the dynamic role-based access control. The architecture's framework assumes that each domain has an independent local security monitoring unit and a central security monitoring unit that monitors security for the entire grid. The architecture was evaluated using the Grid Security Services Simulator, a meta-query language and Java Runtime Environment 1.7.0.5 for implementing the workflows that define the model's task. In terms of scalability, the results show that as the number of grid nodes increases, the average turnaround time reduces, and thereby increases the number of service requesters (grid users) on the grid. Grid middleware integration across various domains as well as the appropriate handling of authentication and authorisation through a local security monitoring unit and a central security monitoring unit proved that the architecture is interoperable. Finally, a case study scenario used for access control across the domains shows the efficiency of the role based access control approach used for achieving appropriate access to resources. Based on the results obtained, the proposed framework has proved to be interoperable, scalable and efficiently suitable for enforcing access control within the parameters evaluated.

Keywords: authorisation, grid, role-based access control, scalability, interoperability, access control, security, multi-domain environment

1. INTRODUCTION

Grid computing is an environment that provides unhindered access to computational infrastructure across various domains in academia and industry. It allows the porting, running, sharing and distribution of applications [1]. Since grid computing involves many users from different organizations and domains, sensitive and classified information may be vulnerable if no control policy for regulating and securing all the domains on the grid, is present [2], [3].

The concept of a grid system is analogous to a "water grid system". The facilities of a water grid system make it possible for anyone in his home to open a tap to collect water without knowing exactly where such water is being processed [4]. Similarly grid computing is able to provide endless and ubiquitous access [5] to high quality computing resource without having to know exactly where the data is being processed [1].

Buyya [4], defined a grid as follows: The "grid is a type of parallel and distributed system that enables the sharing, selection, and aggregation of resources distributed across multiple administrative domains based on their (resources) availability, capability, performance, cost, and users' quality-of-service."

The South African Grid (SAGrid) is a typical example of a functional grid. It is a group of South African tertiary institutions (Universities, laboratories and also the Meraka Institute) that are collaborating in the sharing of resources [6].

1.1 Why secure a grid?

To prevent sensitive and important information from being copied, altered, divulged to unauthorized users or manipulated has brought about the need for security on a grid system [7]. Without security a grid cannot be considered to be dependable. However, security models on the grid are difficult to implement and to sustain, due to the complexity of the grid environment [8]. Traditional access-based control models are based on recognized inadequacies and there is thus a need to replace them with more flexible [9] models which are relevant to distributed environments [10].

1.2 Security challenges

Scalability: Scalability caters purposely for future expansion [11]. For a grid environment to be scalable, a centralized administration as well as regular update of the security policies is necessary [12]. In other words, scalability simply means the capability of a grid system

Based on "Towards achieving scalability and interoperability in a Triple-Domain Grid-Based Environment (3DGBE)", by Nureni A. Azeez and Isabella M. Venter which appeared in the Proceedings of Information Security South African (ISSA) 2012, Johannesburg, 15 to 17 August 2012. © 2012 IEEE

such that it can efficiently handle both a small or large number of nodes and users [13].

Interoperability: This can be simply defined as the ability of various systems on the grid to exchange, share and utilize information across platforms. It is a security challenge due to disparate and unequal security policies. The characteristics of an interoperable grid-based environment include:

- the presence of a central authority for security and trust;
- heterogeneous resources, service discovery and management as well as;
- the interdependence of security infrastructures [14], [15].

Efficient Access control (EAC): is intended to enforce control over whom (agent) can interact with resources on the grid network. The EAC can be achieved through different means such as authentication and authorisation with the aid of an appropriate access control model. EAC remains a challenge in grid computing mainly because a large number of users are involved. The users are often considered to be dynamic in their requests. This could be attributed to the fact that each domain on the grid has its own policies and the domains are autonomous [33].

To secure a grid based environment without compromising accessibility, interoperability and scalability the following questions can be asked:

- How should a common security policy for various domains on the grid be determined? and
- How should the security of the grid be managed to ensure accessibility of resources in an interoperable and scalable grid based environment?

To achieve the aim of EAC, it was concluded that regulation is required. To regulate and find a solution to the factors which impact EAC within the grid platform, a role based access control (RBAC) model was designed, a prototype built and the prototype was tested with the G3S simulator. The RBAC model is based on three primary rules: role assignment; role authorization and transaction authorization. It was found that the proposed framework is interoperable (in terms of resources; grid middleware, operating system and authorisation), scalable and suitable for enforcing access control within the parameters evaluated.

The remaining part of the paper is organised as follows. In Section II, a summary of related work is presented. A brief analysis of the various security requirements on the grid is explained in Section III. Section IV gives a stratum of the proposed architecture with Subsections A and B presenting the stages of the architectural model. Section V provides a comprehensive overview of the components of the architecture. Section VI gives an operational overview of the model while Section VII gives an approach for evaluating security in a triple-domain grid-based environment (3DGBE). Section VIII deals with the implementation and evaluation. Finally, the paper is concluded in Section XI.

2. SECURITY REQUIREMENTS IN A GRID ENVIRONMENT

The security requirements defined by the International Organization for Standardization (ISO) and the International Telecommunication Union (ITU) are ITU-T Provision X.805 and X.800 [22].

2.1 Authorization

For any organization to allow its resources to be jointly shared between all parties involved there is a need for authorization: who should have access to any particular resource and who should not [23][18]. Globus Toolkit Gridmap files [24], Community Authorization Service (CAS) and Virtual Organization Membership Service (VOMS) are authorization measures usually adopted in grid computing [25].

2.2 Authentication and Access Control

Impersonation has been identified as a threat [11] in grid environments. Authentication is thus important to prevent illegal access [26]. The main purpose of authentication is solely to confirm that the user is who he claims to represent and not any other person. In both the shared and personal computer system, authentication is usually carried out with the use of a password and username. It has been established that when a password is used to log onto the system [4], the authenticity of a user is usually fully guaranteed. However a password can be stolen hence the information on the system can be vulnerable. Digital certificates, verified by a Certificate Authority [26], are taken as the best way to ensure authentication on the Internet.

2.3 Data Confidentiality

The purpose of data confidentiality is to protect data from being divulged to the wrong or an unintended party [27]. Two processes can be used to achieve data confidentiality: data encryption and data decryption. Also, two main types of cryptography can be used to provide data confidentiality [28], i.e. symmetric and asymmetric.

3. RELATED RESEARCH

Research done in terms of securing the grid can be divided into three main categories: security-policy aggregation, access control and reliability in grid security.

3.1 Security-policy aggregation

In a bid to ensure aggregated security policies across different domains Tari and Fry proposed Global Access Control. A distributed object kernel security service was provided for enforcing and aggregating local and general security policies on the grid. In order to allow control of data aggregation, they provided a security framework Federated Logic Language (FELL) and a logic-based language [16]. The security constraint was enforced by mapping state-transition graphs which model different nodes on the grid. This approach is good and enforces various security measures but it is not scalable since it does not allow more nodes to be added to the grid [6]. Security-policy aggregation in terms of scalability and interoperability still needs to be addressed.

3.2 Access control

In the work of Yanxiang et al. a model was developed based on a public key and double-identity authentication on a grid. The model was developed to ensure both authenticity and confidentiality. For the implementation of this model, they applied an RSA (Ronald Rivest, Adi Shamir, and Leonard Adleman) cryptosystem. Furthermore, a double identity authentication approach was employed, to include a time parameter on the server side. Finally, both the server and client produce passwords which change over time. However, this model is not scalable and dynamic as provision was not made for adding users [17].

Some Attribute-Based Access-Control systems such as Akenti and PERMIS have been in use for several grid applications [18]. These authorization systems apply their own rules. As a result, a dynamic attribute based access control is required for the grid computing environment [19]. In this model, there is no room for interoperability across various domains on the grid.

John McLean [20] came up with a framework in which Mandatory Access Control (MAC) models, allow for changes in security to be formalized. He employed algebra to construct his model that paves the way for the discretionary access control for n persons. This model is good but does not handle the problem that emanates from the separation of duties and cyclic redundancy as a result of roles and hierarchy among participants on the grid.

3.3 Reliability in grid security

Laccetti and Schmid [21] came up with a framework for reliable grid security infrastructures using Grid Security Infrastructures (GSI) and Community Security Policy (CSP). Their analysis captured the policies and rules upon which GSI and CSP were based. Trust relationship based on a cryptographic key was used as a guiding principle. It was finally revealed that authentication implemented at grid levels develop a trust relationship that is transitive which is not the case when authentication is used at operating system tier. Formal model algebra was adopted in developing the security of the grid [21]. This model is not flexible as it has limited application.

4. STRATUM OF THE PROPOSED ARCHITECTURE

The proposed architecture constitutes two stages, each of which involves two phases (see Figure 1).

1. The first phase involves various domains. Each of the domains is characterised by a user and a local security-monitoring unit (LSMU).
2. In the second phase, the central security-monitoring unit (CSMU) interacts directly with all the domains of phase.
3. The third phase is a processing phase. All activities that result in the granting of resources are carried out in this phase.
4. The fourth phase is a grid environment phase where many resources are available. A user is allowed to access this phase based on a decision made in the third phase.

4.1 Stage 1 of the architecture

This stage involves the interaction between various users and the domains' LSMU with the CSMU. The architecture in Figure 2 and Algorithm 1 give comprehensive information with respect to this interaction and message passing between grid entities. In Figure 2, a theoretical framework of the interaction between the user and the LSMUs of three domains, as well as its interaction of the three domains and the CSMU is depicted.

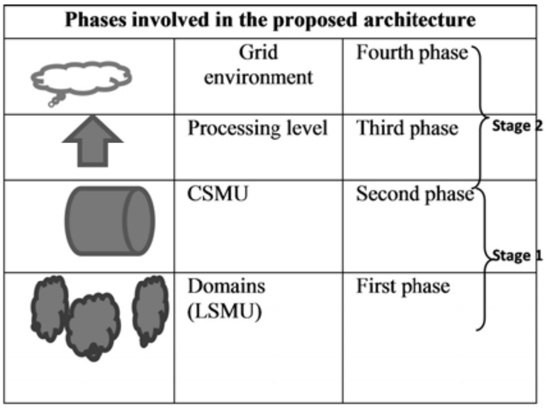
To explain the process of the architecture presented in Figure 2, let us assume the following scenarios:

1. Adam, a grid user (GU) in Domain A, forwards his request to his domain's LSMU, where his authorisation is verified and confirmed. Adam's status (eligibility as a user) is thus determined. This phase makes Adam's access right to the intended domain known.
2. The LSMU then sends Adam's request to access a resource in any intended domain to the CSMU to reconfirm his authorisation right in his own domain and his rights to access resources of any other domain. The CSMU verifies whether Adam qualifies to access the required resource. There are two outcomes: YES (acceptable) or NO (not acceptable).
3. If NO, the process (request) terminates and the feedback message is communicated to the user.
4. If YES, a "clearance" certificate will be given to the user (Adam) by the LSMU of the intended domain and the user can proceed to stage 2.

5. If there is a successful processing in stage 2, the user will proceed to access resources in the grid environment.

4.2 Stage 2 of the architecture

This stage deals with the interaction between the processing phase and grid environment. This stage comes into play if and only if there is a positive feedback during Stage 1 (See Figure 3 and Algorithm 2).



Phases involved in the proposed architecture		
	Grid environment	Fourth phase
	Processing level	Third phase
	CSMU	Second phase
	Domains (LSMU)	First phase

Figure 1: Phases involved in the proposed architecture

Algorithm 1: Algorithm describing the working relation of components in Figure 2

```
Required by Domain A, Domain B, Domain C, LSMU, CSMU
Begin:
    feedback [authorisation] = "Yes or No";
        GU{Domain A, B, C}: $\xleftarrow[\text{request}]{\text{authorisation}}$
LSMU:
            if authorisation = "No"
                then : terminate (process)
else:
        if authorisation = "YES"
          Then: LSMU $\xrightarrow[\text{request}]{\text{forward}}$ CSMU

    CSMU ⟶ { (GU (role))}:
      If CSMU [permission (decision)]: = "yes"

            then: CSMU $\xrightarrow[\text{to}]{\text{moves}}$ stage 2;
Stop
```

The operation of the architecture is presented in Figure 3:

1. Through the grid entry link, the GU requests access (with the role authorisation-certificate) from the grid user authentication service (GUAS). The request is either granted or not.
2. If the feedback is negative, the entire process will be terminated immediately and the request will cease to continue.
3. However, if the feedback is positive (YES), then the request will be forwarded to the policy information point (PIP) (a protocol of XACML (eXtensible Access Control Markup Language) for access control). This is to source detailed information about the user.
4. The request will further be directed to the policy decision point (PDP), which is another XACML protocol for access control. The PDP is responsible for making a decision on whether the user may access the requested domain. The feedback of the PDP will either be positive (YES) or negative (NO). If the feedback is negative, the entire process stops.
5. If the feedback is YES the request is conveyed to the PEP.
6. The PEP will demand an updated version of the user permission certificate from the PDP (grid virtual organisation (VO)-PDP).
7. A certificate validation/update will be transferred to the centralized resource database server (CRDS) from the PDP (grid VO PDP).
8. Finally, a message will be sent to the user to proceed and access resources on the grid.

The procedure is applicable from either of the domains available on the grid i.e. either Domain A to Domain B or from Domain A to Domain C, and vice versa.

In order to ensure a smooth and efficient access control mechanism on the grid and also to improve the performance of the architecture, the LSMU works with the CSMU. That is, there is smooth correspondence between the local security units of all the domains with the central security unit for the entire grid. They both communicate and work hand-in-hand to achieve a flexible, interoperable and scalable grid environment.

Algorithm 2: Describing the working scenario of the architecture presented in Figure 3

```
Require: role, user, PIP, PEP, PDP, GUAS, CRDS

Feedback: [yes/no]:
  Begin: from stage 1:
     request: ⟶ GU role certificate [GEL]
          then: GUAS $\xleftarrow{\text{Verifies}}$ Role (GU);
              Else: if feedback (GUAS) = "No"
                   Then: terminate (process);
              If feedback (GUAS) = "YES"

 Then: request ⟶ PIP;

           PIP: $\xrightarrow{\text{Requests}}$ PDP;   //request for appropriate
decision
             If feedback = "YES"
        proceed: ⟶ PEP;
           Else if feedback = "No"
                         then: stop (process)

Getupdate: ⟶ PDP-VO;
update (certificate): $\xrightarrow[\text{by}]{\text{obtained}}$ CRDS;
         finalDecison: $\xrightarrow{\text{Pass to}}$ (VO (Grid))
         Begin [GU] :access [resource]

Stop
```

5. OVERVIEW OF THE BASIC COMPONENTS OF THE ARCHITECTURE

In the proposed model, each of the domains available in the virtual organisation (VO) has an LSMU saddled with the responsibility of the domain's local security access control and management. The CSMU is an advanced access control and management system that handles access control and authorisation for the various grid entities across the three domains of the model. For any access request by a grid user, the LSMU would verify the user's access privilege. The model is based on the adoption of the XACML's (eXtensible Access Control Markup Language) request-response protocol which makes use of four basic components. The components are: PEP, PDP, PIP and PAP. However, in this model, only PEP, PDP and PIP are used because of their relevance, usefulness and application in the proposed architecture.

5.1 Assumptions

1. A user from domain A (Adam) may intend to access a resource in domain B and a user in domain B (Ben) may also be interested in accessing resources from domain A;
2. A user in domain A (Adam) may wish to access resources in domain C while a user that is in domain C (Charles) may equally be interested in resources of domain A.

These are two possible scenarios when a three domain based architecture is being considered. Scenario 1 is illustrated in Figures 2 and 3 and it is equally applicable to other scenarios. Adam, Ben and Charles are users in the domains A, B and C respectively. Each of them is bound with the security and access framework in their respective domains. There are six ways in which access could be requested: request can come from Domain A to Domain B, from Domain A to Domain C, from Domain B to Domain C, etc.

6. OPERATIONAL OVERVIEW OF THE MODEL

The security of each individual domain is quite dependable and efficient; because each domain has its own access control and monitoring policy which is monitored by the LSMU. If a user, however, wishes to access resources in another domain, the user from the designated domain will first need to be verified by his domain.

This is achieved by translating the certificate of his domain to the domain in which he wishes to access resources. The translation (or conversion) targets the access privileges and the identities in other domains on the grid. CSMU is mainly in charge of monitoring and overseeing access and security relationship from one domain to another domain depending on where an entity requires access. Also, CSMU is equally responsible for maintaining the information for mapping interactions between domains (see Figure 2 as well as 3).

7. DETERMINATION OF SECURITY IN A 3-DOMAIN GRID VIRTUAL ORGANISATION

7.1 Definition of simulation parameters

In order evaluate the effectivity of the security of the domains; the following parameters defined below were taken into consideration.

Definition 1: Let DSR(A,B) and DSR(A,a), denote the *direct security rate* which is determined and evaluated when the CSMU finds and grants permission and access privilege to a user from domain B to domain A or from an entity a ∈ domain A to domain A depending on from where the access is requested. DSR(A,B,C) denotes the DSR between the three designated domains.

Definition 2: Similarly let SR(A,B) or SR(A,a) denote the security rate for accesses from domain B to domain A or for an access from entity a ∈ Domain A to domain A. SR(A,B,C) denotes the security rate between the three designated domains.

Definition 3: Let Assess$(a_i \ldots a_j)^m$ denote assessment for entities $a_i \ldots a_j$ when $a_i \ldots a_j$ terminate at time step m, and $-1 \leq$ Assess$(a_i \ldots a_j)^m \leq 1$ shows either rejection or satisfaction during the assessment of the entities involved. While '-1' indicates the rejection, which will reduce the value of SR, '+1', however, indicates satisfaction, which will increase the value of SR.

Definition 4: Let DSR$(a_i \ldots a_j)$ stands for "Direct Security Rate" in a grid for entities $a_i \ldots a_j$.

Definition 5: Let Rep(A, a) denote *reputation* and *status* of entity a in Domain A on a grid.

Definition 6: Let Approv$(a_i \ldots a_j)^m$ stand for the *approval* in the service request for $a_i \ldots a_j$ after m time steps.

8. SECURITY EVALUATION IN A 3DGBE

Determining or evaluating the security rate in a multi-domain grid-based environment is completely different from what is obtainable in a single-domain environment. The main reason for this is the interaction and relationship between the grid entities involved. Unlike in a single-domain environment, a multi-domain grid environment has more entities from one domain to another to interact with. Hence, to handle the complexities that arise from the user's accessibility to different domains resources, the security rate (SRs) for the entities of each domain is useful for quick and accurate evaluation of the security within different domains. The approach adopted for determining the inter-domain security rate value is simple and provides the benefit of feedback that is flexible and dynamic in nature.

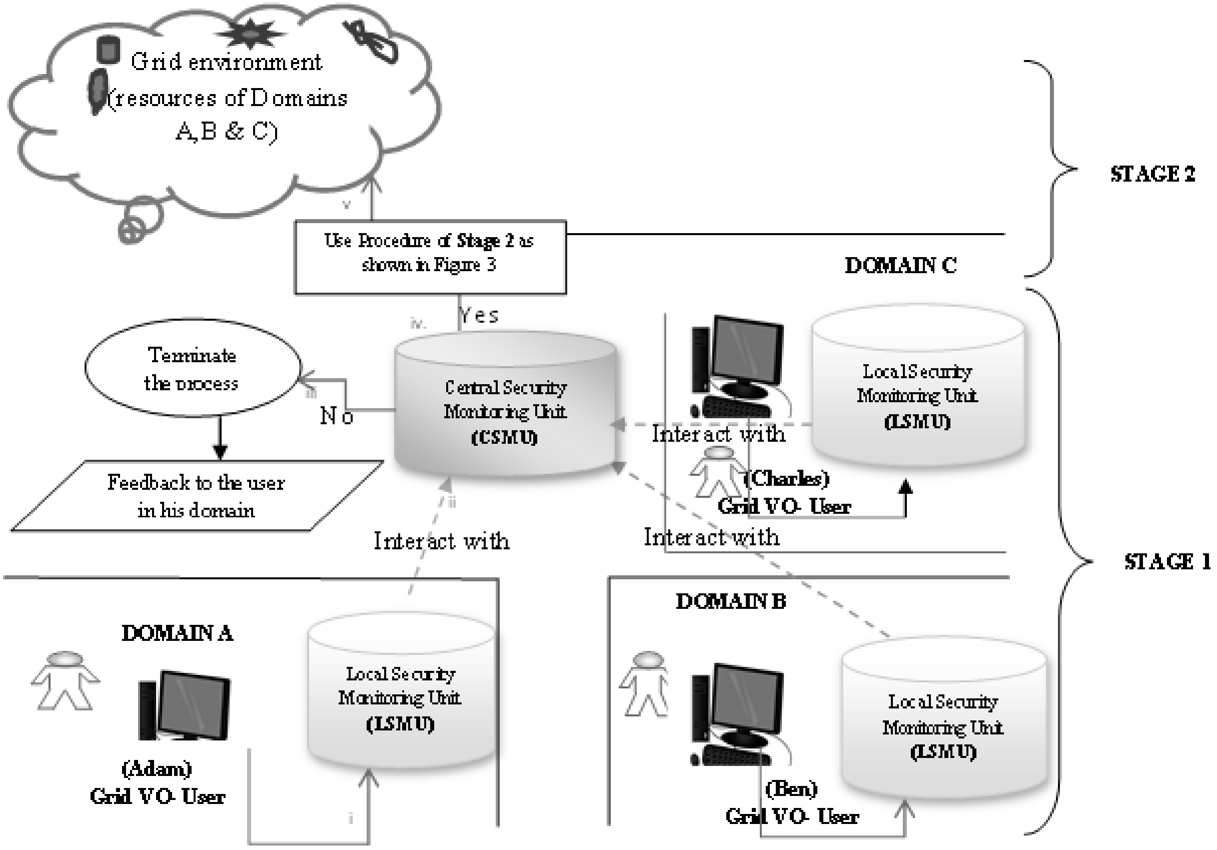

Grid environment (resources of Domains A,B & C)
STAGE 2
v
Use Procedure of Stage 2 as shown in Figure 3
DOMAIN C
iv.
Yes
Terminate the process
iii
No
Central Security Monitoring Unit (CSMU)
Interact with
Local Security Monitoring Unit (LSMU)
(Charles) Grid VO- User
Feedback to the user in his domain
ii
Interact with
Interact with
STAGE 1
DOMAIN A
DOMAIN B
Local Security Monitoring Unit (LSMU)
Local Security Monitoring Unit (LSMU)
(Adam) Grid VO- User
(Ben) Grid VO- User
i

Figure 2: A 3-domain role based access control architecture showing interaction between users, CSMU and LSM

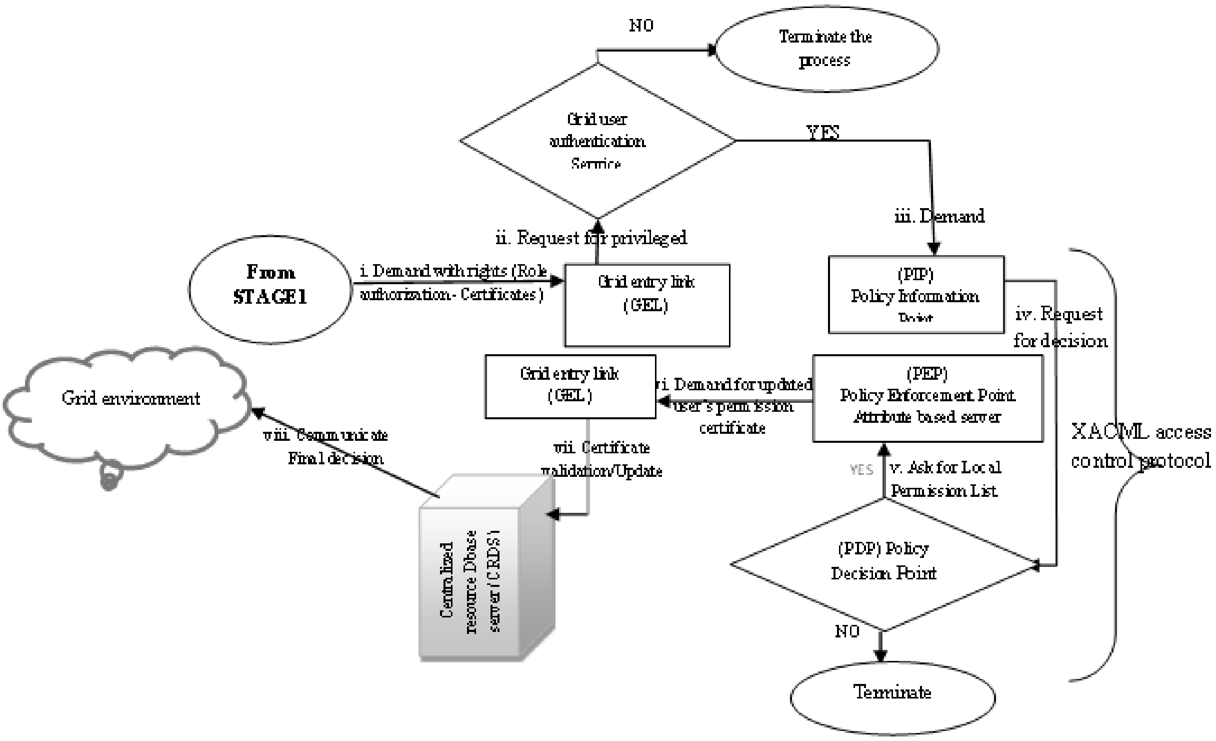

NO
Terminate the process
Grid user authentication Service
YES
iii. Demand
ii. Request for privileged
From STAGE1
i. Demand with rights (Role authorization- Certificates)
Grid entry link (GEL)
(PIP) Policy Information Point
iv. Request for decision
Grid environment
Grid entry link (GEL)
vi. Demand for updated user's permission certificate
(PEP) Policy Enforcement Point Attribute based server
XACML access control protocol
viii. Communicate Final decision
vii. Certificate validation/Update
YES
v. Ask for Local Permission List
Centralized resource Dbase server (CRDS)
(PDP) Policy Decision Point
NO
Terminate

Figure 3: A 3DGBE with RBAC architectural framework of the proposed model

Rep(C, a_i) yields status/repute of entity a_i to domain C in a virtual organisation considered that a_i is not an entity in domain C. It is worth mentioning that A, B, and C represent three different domains being considered while a_i, b_i and c_i are entities in the three domains.
Hence,

$$\mathrm{SR}(A, B, C)=\lambda_1 DSR(A, B, C) + \lambda_2 Rep(A, B, C) \qquad (1)$$

Equation 1 is used to evaluate the SR in the three domains A, B and C where the weight λ_1 and λ_2 are positive and $\lambda_1 + \lambda_2 \leq 1$.

$$\mathrm{DSR}\left(A_i, A_j\right)=\frac{\sum_{a\in A_j} \mathrm{DSR}(A_i, a)}{|A_j|} \qquad (2)$$

Where a is an entity from the domain A. Given two different domains A_i and A_j with i, j ∈ 2 [1…n], where i≠ j, and n is the number of domains.

Therefore,

$$\mathrm{DSR}\,(A, C) = \frac{\sum_{c\in C} \mathrm{DSR}(A, c)}{|\,C\,|} \qquad (3)$$

When considering any domain, A, B or C, Equation 2 is generic and can therefore be used to compute direct security rate (DSR) between them. The same is applicable to Equation 3 where domains A and C were specifically considered.

9. REPUTE AND STATUS ACROSS DOMAINS

For domains A_i to A_j with i ≠ j, the status of entities is determined as follows:

$$\mathrm{Rep}\,(A_i, A_j) = \sum_{a\in A_j} \theta_a \; Approv\,(A_i\,, a) Rep\,(A_i\,, a) \quad (4)$$

Where $\theta_a > 0$ is the weight given to Approv(A,a) for a ∈ A and $\sum_{\mathbf{a}\in\mathbf{A}} \boldsymbol{\theta}_\mathbf{a} = 1$. Equation 4 implies that the Rep can be determined from any desired domain and can be extended to any number of domains.

10. IMPLEMENTATION AND EVALUATION

Various simulation experiments were carried out using different simulators however in this example, Grid Security Services Simulator (G3S) was used [29]. To carry out an empirical evaluation of the access control architecture, the simulation was developed in Java making use of Jbuilder. In the three domains in this experimental grid based environment: domain A was made up of a cluster of seven nodes (or computers while the other two domains were LANs (Local Area Network) comprising of 13 computers each. The simulated grid environment was developed using the Globus toolkit 5.0.5. All the hardware of the test bed was embedded in Linux Ubuntu 12.04. A computer hosted a database with the information of all users and acted as the LSMU for each domain while a computer server with a static IP address was chosen as the CSMU for the experimental grid. For an efficient and reliable evaluation, the resources and entities considered were accessible when a grid user requested their services.

Table 1: Simulation parameters with their corresponding values

Parameters	Corresponding values
λ_1	0.25
λ_2	0.36
DSR (a_i…..a_j)	0.34

10.1 Evaluation of 3DGBE and MAC

In the experiment, 3DGBE access control was compared with MAC, which is a popular access control method. Table 1 provides the detail of the parameters used in the simulation experiment. Users were provided and assigned with both MAC-based and 3DGBE access control simultaneously.

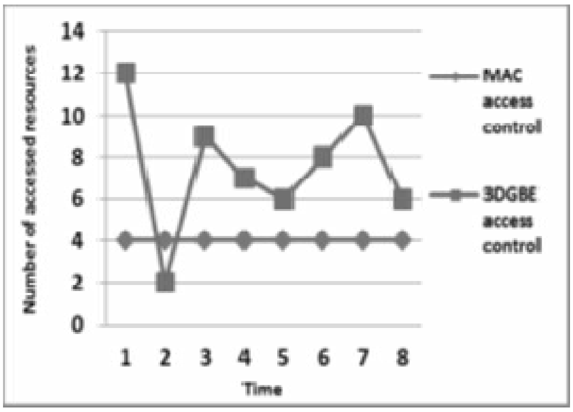


Figure 4: Number of available resources in the two access control policies 3DGBE and MAC

The number of resources varied over different time periods. It was noted that the number of available resources varied over time in the 3DGBE access control architecture whereas it remained unchanged in the MAC-based access control system (see Figure 4).

It can thus be deduced that access to resources would be flexible when deploying a 3DGBE architecture.

Equation 2 was used to evaluate the security rate without considering any weights. Entities in either of the domains A, B or C could request resources from any desired domain and the destination domain then evaluated such requests. The result of the SR was thereafter obtained (see Figure 5).

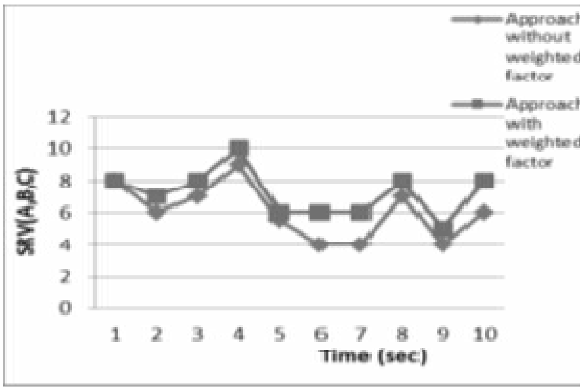


Figure 5: Secure rate compassion using two approaches

Equation 1 was used for calculating the SR between the domains. The security rate value will vary if there are no weighted values for θ_j. Table 2 gives a summary of the required parameters. The simulation result revealed that the available number of grid nodes has a direct influence on the turnaround time as shown in Figure 6.

Table 2: Simulation parameters for λ_1 DSR, REP of Domains A, B, C alongside the number of entities

Parameters	Corresponding values
λ_1	0.6
First (initial) value of DSR (A,B,C)	0.58
First (initial) value of Rep(A,B,C)	0.44
Entities in domain A	20
Entities in domain B	15
Entities in domain C	23

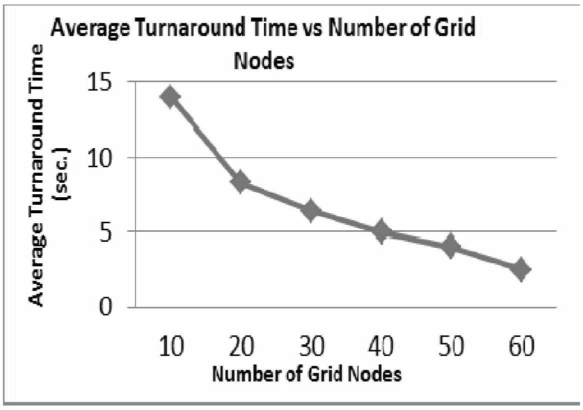


Figure 6: Average turnaround time versus number of grid nodes

This implies that as the number of grid nodes increases the average turnaround time reduces and thereby increases the number of service requesters (grid users) on the grid.

To further prove and sustain the argument that the model developed and implemented is scalable, Figure 7 shows that as the number of service requesters increases, there is little and slight effect on the turnaround time which does not impact on the users' services and request time.

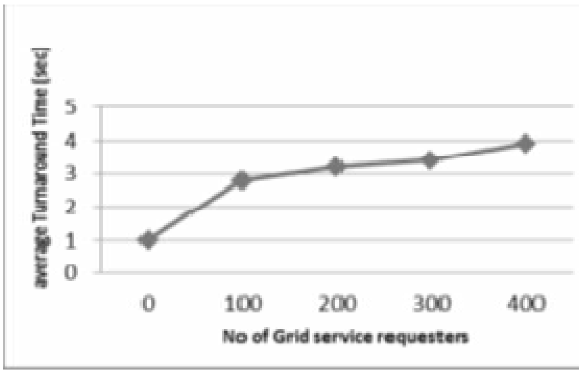


Figure 7: Average turnaround time versus number of service requesters

In order to further sustain the argument that the 3DGBE architecture is scalable therefore, the effect of increase in the number of nodes against the volume of data that is transferred within a given period of time (throughput), were observed and measured.

The result of the comparison of 3DGBE (which uses X.509 certificates) with MAC, CAS, AKENTI and PERMIS (that use own their certificate formats) is presented in Figure 8. The result shows that 3DGBE has the highest degree of interoperability when compared to the others.

In the initial setup indicates domains A, B and C contain

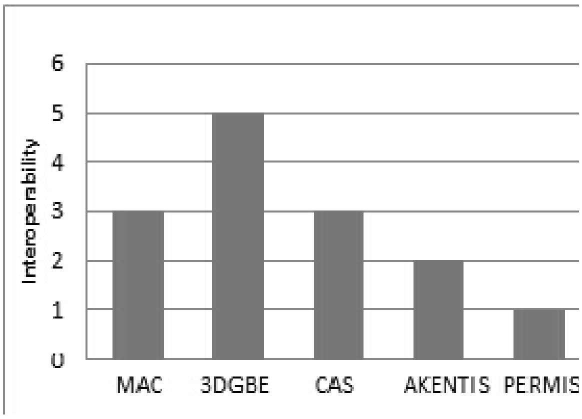


Figure 8: Comparative evaluation of interoperability of 3DGBE with the existing system

7, 13 and 13 nodes respectively. To ascertain the effect of an increase in nodes on the performance of the throughput, the number of nodes in each domain was increased as follows: domain A had 12 nodes; domain B had 20 nodes while domain C had 25 nodes.

The result obtained (see Figure 9), shows an increase in throughput as follows: when the number of grid nodes in domain A comprises 12, the throughput is 100MB/s, when the number of grid nodes in domain B is increased to 20, the throughput is 2200MB/s, while 3100MB/s is attained when the number of grid nodes in domain C is increased to 25.

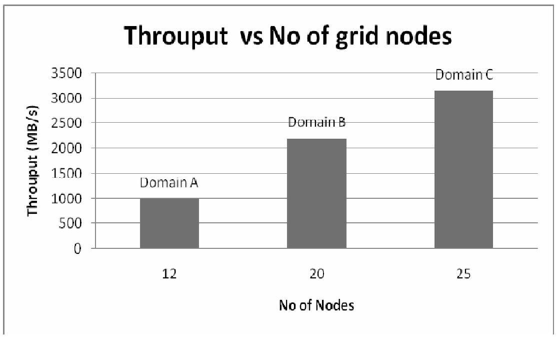


Figure 9: Throughout (MB/s) vs No of nodes

From Figure 9, it can be deduced that as the number of grid node increases, the throughput also increases thereby increasing the number of resources being accessed within a given time. This proves that the scalability of the 3DGBE architecture.

The use of grid middleware has been identified as one of the ways for solving the challenge of interoperability among multiple administrative domains. This model adopted the XACML access control protocol, which showed the highest level of interoperability when compared to others.

Appropriate handling of authentication and authorisation through LSMU and CSMU: The central security-monitoring unit (CSMU) maintains a high degree of interoperability between the users on the grid. For a resource request to be allowed, approval needs to have been given by the local security monitoring unit (LSMU). The CSMU serves as the central point which makes the final decision for grid resources to be accessed. There is smooth correspondence between the LSMU of each of the three domains and CSMU. The purpose of this is to ensure adequate and an efficient data sharing mechanism among the domains with the view of achieving interoperation of authorisation. CSMU forwards requests and authorisation from all the domains across the domain to access any required resources.

Operating system interoperability: Aside from the fact that the architecture permits various applications to run (application interoperability), the architecture has proved to be interoperable in both the LINUX (Ubuntu 10.04) and Windows operating system with different middlewares (Globus, Glite and UNICORE). The evaluations that were carried out were done on both operating systems, LINUX performs better than Windows. The feature of operating system interoperability is noticeably weak in some of the existing models such as MAC, AKENTIS and PERMIS.

Interoperability with Grid middleware: Middleware can be regarded as:

> *"A mediator layer that provides a consistent and homogeneous access to resources managed locally with different syntax and access methods" (Priol, 2005: 32)*

Aneka, Alchemi , Cosm P2P Toolkit , Globus ,Gridbus, Grid Datafarm , GridSim (Toolkit for Grid Resource Modeling and Scheduling Simulation), Jxta Peer to Peer Network, Legion, NorduGrid middleware, PUNCH, Simgrid, Storage Resource Broker (SRB), ProActive, Unicore and Vishwa are prominent grid middleware [4]. With some of the listed grid middlewares, the virtual organisation interoperability issue remains a problem. This is because of the absence of upper-level semantic concepts in their grid middleware layers [30].

To address this challenge, a tri-middleware integration approach was used. The 3DGBE was enabled with three different middlewares across the three available domains on the grid, namely Globus 5.0, gLite and Alchemi for domains A, B and C respectively (see Figure 14).

With this approach of grid interoperability that is based only on the middleware integration, various middlewares were deployed on different domains and allow the same set of users to share and access resources with well-established and defined virtual origination's policies, irrespective of the grid middleware they intend to use.

The problem of middleware differences was solved effectively by using common standards. The middlewares were implemented as a subset of specifications of the different grid middlewares. With this interoperation based approach to middleware integration, different middlewares need not necessarily communicate with each other to be able to merge and share resources.

The various computing resources (installed in the three domains) were all made accessible to all the grid users regardless and independent of their middlewares they intend to adopt. The middleware integration of three or more grid resources is easier and grid users across the three domains can access resources without any hindrance.

Aside from Globus 5.0.2, two other middlewares were installed for the interoperability integration testing: gLite and Alchemi. These three middlewares were chosen because they were available.

Local clusters and their security were considered as the basic elements of infrastructure that could be reused to aid interoperability across the three domains. Globus 5.0.2, gLite and Alchemi aid Torque/MAUI as local scheduler, which supports the ability to share local clusters and all available resources. These middlewares use X.509 certificates hence the same grid resources can be accessed, shared and distributed by these three different grid middlewares. Achieving interoperability with three different middlewares is simple with Globus 5.0.2, gLite and Alchemi respectively. Both gLite and Alchemi adopted the Grid System Infrastructure (GSI) model developed by Globus for user's authorisation.

The model (GSI) makes use of a digital certificates and proxies for the authentication and authorisation of hosts and users. Established on X.509 digital certificates and proxies, GSI was extended in both the gLite and Alchemi with the agreement of the Virtual Organisation Membership Service (VOMS), which released fully X.509 compatible signed extensions to proxies. Additional information about the users, which is required for the mapping on various levels of authorisation, is achieved through these extensions. Since VOMS proxy is compatible with the X.509 proxy, therefore the former's proxy can be taken as authentication and authorisation credential when deploying it on the three grid middlewares.

The distribution of resources across a tri-middleware based architecture is the second focus in achieving interoperability. The cluster manager in charge of the local resources was configured in such a way that jobs could be submitted despite differences in middlewares. The local scheduler in the architecture is Torque/MAUI. This scheduler is supported by Globus 5.0.2, gLite and Alchemi hence it is very easy to express new queues new and added middlewares in order to utilise the same resources.

Declarative language (queries) for interoperability: A big challenge for developing and implementing an interoperable 3DGBE lies in the ability to efficiently, sufficiently express "cross-queries" (inter-domain queries) that relate information from different domains. To overcome this challenge, a Meta-Query Language (MQL) [31], which is similar to the Structures Query Language (SQL) was adopted.

MQL was used for querying and restructuring tables containing information across different domains. As illustrated in the scenario presented, MQL was used to query and restructure information across Domains A, B and C within the federation. Hence, interoperability was achieved across these three domains through MQL dynamic query mappings.

Test scenario:

Three different databases were created for domains A, B and C: a University, Hospital and Banking database respectively (see Tables III, IV and V).

10.2 Inter-domain queries

CASE 1: Databases were created for domains A and B and they were aggregate and joined to enhance further operations (see Figure 10).

CASE 2: To combine the information of the databases of domains A and C, the query as depicted in Figure 11 was issued and the report generated shows a UNION of both databases for both domains.

```
SELECT DB1.*, DB2.*
FROM DomainA.Database1.dbo.myTable AS
DB1
INNER JOIN
DomainB.Database2.dbo.myTable AS DB2
   ON DB1.id = DB2.id
```

Figure 10: Query for aggregating data from Domains A and B

```
-- FROM Domain_A
SELECT * FROM
[MyDatabaseOnDomain_A].[dbo].[MyTable]
Table1
   INNER JOIN
Domain_C].[MyDatabaseOnDomain_C].[dbo].
[MyOtherTable] Table2
        ON Table1.ID = Table2.ID
```

Figure 11: Query for aggregating data from Domains A and C

```
SELECT a.ID_NO,
a.Surname,a.Nationality,
b.File_No,b.Patient_Condition,
c.Service_Code,a.Tax_ID,b.Age
```

Figure 12: Cross-domain query for joining data from Domains A, B and C

Cross-domain queries were applied to each of the newly obtained tables. For example to obtain ID No, Surname, and Nationality from Table 3; File No, Patient_Condition and Age from Table 4; as well as Service_Code and Tax_ID from Table 5, from respectively domains A, B and C, the cross-domain query depicted in Figure 12, was used.

Table 3: University Database for Domain A

ID No	Student No	Surname	First name	Initial	DOB	Passport No
1000	3008814	Azeez	Nureni	N.A	1978/07/10	QOO12345
1001	3066278	Adewale	Abiola	M.D	1980/04/08	RE0047476
1200	2340078	Abidoye	Philip	P.O	1976/06/09	WE345678
1220	2357858	Scholtz	Josue	S.J	1981/04/05	VB7878784
1260	3400993	Magnuth	Henry	H.M	1975/06/09	FD8787878
1320	3476002	Andy	Liu	X.L	1984/02/09	RE7878784
1400	3455266	Achmed	Imran	I.A	1986/02/09	UD5785785
1523	2004556	Jonathan	Magnus	I.M	1974/07/10	TY8989896

Table 4: Hospital Database for Domain B

Patient ID	File No	Surname	First name	Patient Condition	Date Of Admin
1000	0031	Azeez	Nureni	Medical checkup	2012/09/04
1200	0045	Abidoye	Philip	Eye problem	2012/06/04
1320	0067	Andy	Liu	Hearing problem	2012/03/06
1400	0012	Achmed	Imran	Back pain and X-ray	2011/09/04
1523	0056	Jonathan	Magnus	Pregnancy	2012/01/02
1001	0023	Adewale	Abiola	Blood test	2012/09/09
1220	0013	Scholtz	Josue	Car accident	2011/09/01
1260	0011	Magnuth	Henry	Head injury	2012/03/06

Table 5: Banking Database for Domain C

ID No	Customer Name	Service Code	Account ID	Current Balance
1000	Azeez, N.A	FD9989	45454599XX	R 56.5XX
1200	Abidoye, P.O	YU7878	57757577XX	R 100XX
1320	Andy, X.L	HJ7880	47747744XX	R 562XX
1400	Achmed, I.A	WE4545	67677676XX	R 00.7XX
1523	Jonathan, I.M	QW567	56565655XX	R 06.7XX
1001	Adewale, M.D	NH7676	86868612XX	R 129XX
1220	Scholtz, S.J	YE85852	67676768XX	R 451XX
1260	Magnuth, H.M	BG2323	13243535XX	R 000XX

The cross-domain queries were introduced purposely to handle heterogeneity of information represented in different structures, to provide distinct aggregation capability in addition to the principal objective of multi-domain database interoperability.

10.3 Efficient access control

Access control remains a bottleneck when accessing resources in a multi-domain environment such as a grid. Each user participating in grid resource sharing tends to gain access to resources within its jurisdiction. Some grid users might want to access resources for which they are not authorized.

To achieve efficient access control, hierarchical role based access control was adopted for specifying role, services as well as permission for each user from any domain. To explain this, consider a specific scenario (Health); where each of the domains has roles, services and permission defined among the users (see Algorithm 3).

Terms and Definitions as used in this context:

- Let H1, H2 and H3 denote the hierarchies and let the role hierarchy (RH) denoted as H1, H2 and H3 be assigned to domains A, B and C respectively where H1 > H2 > H3

It could be recalled that a *"... hierarchy is mathematically a partial order defining a seniority relation between roles, whereby the seniors' roles acquire the permission of their juniors, and junior roles acquire the user membership of their seniors"* [35]

Algorithm 3: Algorithm for efficient access control in a 3DGBE

```
Required: Domains A, B and C, LSMU, CSMU
Grid User (GU) identification;
Get the Domain's hierarchy as {H1, H2, H3};
Assign hierarchy to the chosen domain;
Obtain GU role;
Retrieve GU services - permission;
Proceed to the grid
```

Thus,

- Let Role_Domain A denotes all roles defined in domain A;
- Let Role_Domain B denotes all roles defined in domain B; and
- Let Role_Domain C denotes all roles defined in domain C.

Role and services specification for DOMAIN A:

1. Role_Domain A = {Physician, Cardiologist, Neurologist, Obstetrician, Pathologist, Pulmonologist, Surgeon, Pediatrician, Oncologist, Dermatologist}
2. Services (permission)
    - {Physician (write patient record, read patient record, write prescription, read prescription, examine patient)}
    - {Cardiologist (treat heart disease, write patient record, read patient record, write prescription, read prescription)}
    - {Neurologist (treats brain, examine nervous system, write patient record, read patient record)}

Role and services specification for DOMAIN B:

1. Role_Domain B ={patient , nurse, pharmacist, dentist, Psychiatrist , Podiatrists }
2. Services (permission)
    - {patient (read prescription , read patient record)}
    - {nurse (write patient record, read prescription, read patient record)}
    - {pharmacist (read prescription, read patient record, select prescription)}

Role and services specification for DOMAIN C:

1. Role_Domain C ={ Ultrasound Technologist, X-Ray Technician, Clinical Technologist, Clinical Technologist, Dental Assistant, Dental Laboratory Technician}
2. Services (permission)
    - {Ultrasound Technologist (read patient record, take ultrasound, analyse images)}
    - {X-Ray Technician (read patient record, perform x-ray on patient, interpret and analyse x-ray result)
    - {Clinical Technologist (read patient record, perform medical test, interpret result)}

Whenever a grid user (GU) specifies his domain, the corresponding hierarchy of such a user will be instantly verified and produced. The hierarchy is divided into three layers; hierarchy 1 (H1) for domain A, hierarchy 2 (H2) for domain B and hierarchy 3 (H3) for domain C.

Any GU with H1 as hierarchy is from domain A and can access resources from any desired domain whose services are defined. The formulation is such that H1 > H2 >H3 thus H1 has the highest hierarchy and can access all the resources within its domain and the domain under it, that is, domains B and C.

Similarly, H2 permits grid users to access all available information in its domain and resources below it, that is, in H3. However, H3 permits grid users to access resources within its domain alone. This initial access control framework is efficient in a 3DGBE as users whose identities are not linked to a specific hierarchy will automatically be denied access to resources.

The prototype was implemented in a Java Runtime Environment 1.7.0.5 for the workflows that define the model's task. The implementation reveals that the access control adopted is efficient within the three domains considered.

Figure 13 is a comprehensive pictorial explanation of domain roles and their services as spelt out for each user. From the foregoing, it is clear that a cardiologist who has his roles defined in domain A of H1 has the corresponding listed services allocated to it. A dentist whose domain is B with domain hierarchy H2 can only access the allotted services. Any attempt to access other information or services, will result in a "rejection or denial of service" which signifies the efficiency of the access control put in place. Finally, the same condition is applicable to the ultrasound technologist who has his/her services defined in domain C.

11. CONCLUSION

Evidence from the literature reviewed, showed that scalability, interoperability as well as efficient access control are three basic security challenges that need to be addressed if the full scale-benefits of grid computing are to be realized.

Based on the results obtained, the architectural framework has proved to be scalable when the average

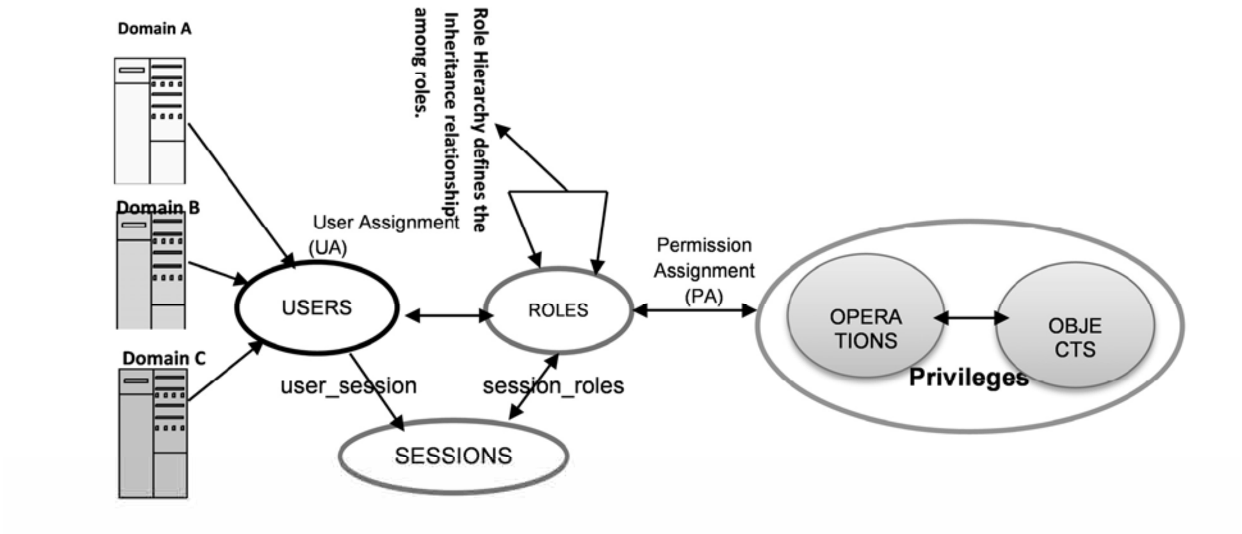


Figure 13: Implementation of hierarchical RBAC and 3DGBE

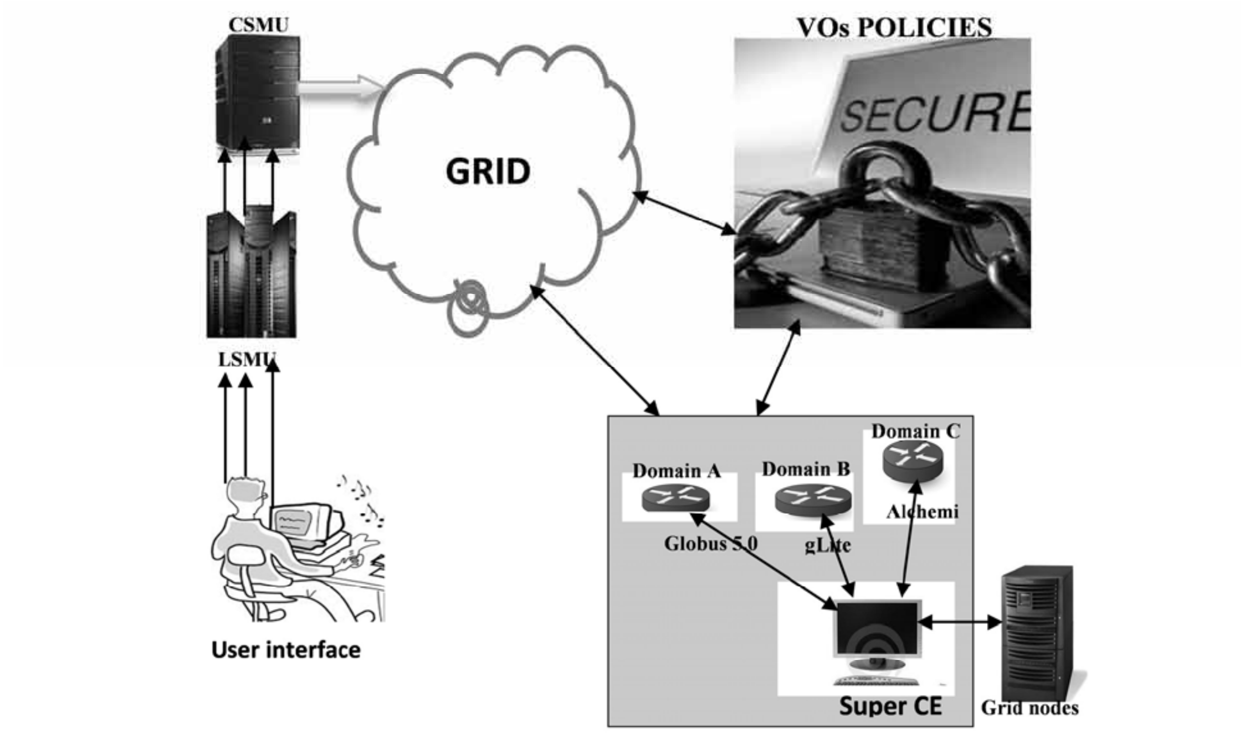


Figure 14: Tri-middleware based infrastructure for 3DGBE interoperability

turnaround time was measured against the number of grid nodes. More convincing results were achieved when the throughput and number of nodes as well as when the average turnaround was measured against the number of grid requesters.

The results obtained in terms of interoperability when the operating systems, grid middleware, LSMU and CSMU as well as database were implemented and experimented with, proved that the model's framework is interoperable. Finally, the efficient access control was evaluated with a role based access control and implemented with a health scenario, and it yielded the expected result.

Other issues that need to be investigated in grid computing are: grid maintenance, grid coordination, pricing, grid auditing and scheduling. These pose challenges that deserve attention for future work. The objectives set out in this research work were achieved. It is therefore, believed that a full-scale implementation of this model, on a real grid system, will ensure a secure, scalable and interoperable grid-based environment.

12. ACKNOWLEDGEMENT

The authors would like to thank the Research Committee of the University of the Western Cape for funding and the Center for High Performance Computer (CHPC) in Cape Town for assistance.

13. REFERENCES

[1] N. A. Azeez, T. Iyamu, and I. M. Venter, “Grid security loopholes with proposed countermeasures,” in ISCIS 2011. Springer Verlag, London, 2011, pp. 411–418.

[2] T. Herath and H. R. Rao, “Protection motivation and deterrence: a framework for security policy compliance in organisations,” European Journal of Information Systems, vol. 18, no. 2, pp. 106–112, 2009.

[3] M.-S. Hwang and W.-P. Yang, “A new dynamic access control scheme based on subject-object list,” Data and Knowledge Engineering, vol. 14, no. 1, pp. 45–56, 1994.

[4] R. Buyya, “Economic-based distributed resource management and scheduling for grid computing,” Ph.D. dissertation, Monash University, Melbourne, Australia, 2002.

[5] H. Baktash, M. B. Karimi, M. R. Meybodi, and A. Bouyer, “2L-RBACG: A new framework for resource access control in grid environments,” in 2010 Fifth Int. Conf. on Digital Information Management (ICDIM).Thunder Bay: IEEE Computer Society, 2010, pp. 359–366.

[6] GStat, “Grid gstat 2.0,” 2010, http://gstat.gridops.org/gstat/sa-gr.

[7] Z. Mao, N. Li, H. Chen, and X. Jiang, “Trojan horse resistant discretionary access control,” in in SACMAT 09: Proc. 14th ACM Symp. On Access Control Models and Technologies. Stresa, Italy: IEEE Computer Society, 2009, pp. 237–246.

[8] B. Bouwman, S. Mauw, M. Petkovic, and E. Philips Res.-Ordina, “Rights management for role-based access contro,” in Consumer Communications and Networking Conference, CCNC. Las Vegas, N: IEEE Computer Society, 2008, pp. 1085 – 1090.

[9] Z.-D. Shen, F. Yan, W.-Z. Qiang, X.-P. Wu, and H.-G. Zhang, “Grid system integrated with trusted computing platform,” Computer and Computational Sciences, International Multi-Symposiums on, vol. 1, pp. 619–625, 2006.

[10] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, “Role-based access control models,” IEEE Computer, vol. 29, no. 2, pp. 38–47, 1996.

[11] R. Lakshmish, L. Ling, and I. Arun, “Scalable delivery of dynamic content using a comprehensive edge cache grid,” IEEE Trans. on Knowl. Data Eng., pp. 614–63, May 2007.

[12] O. Rahmeh and P. Johnson, “Towards scalable and reliable grid networks,” in IEEE/ACS International Conference on Computer Systems and Applications (AICCSA-2008). Doha, Qatar: IEEE Computer Society, 2008, pp. 253–259.

[13] A. Detsch, L. Gaspary, M. Barcellos, and G. Cavalheiro, “Towards a flexible security framework for peer-to-peer based grid computing,” in 2nd Workshop on Middleware for Grid Computing, Toronto, Canada: ACM, 2004, pp. 52–56.

[14] S. Basit, B. James, B. Elisa, and G. Arif, “Secure interoperation in a multidomain environment employing rbac policies,” IEEE Trans. Knowl. Data Eng., pp. 1557–1577, 2005.

[15] G. Pankaj, “Application of a distributed security method to end-2-end services security in independent heterogenous cloud computing environment,” in IEEE World Congress on Services. Washinsgton, DC: IEEE Computer Society, 2011, pp. 379–384.

[16] Z. Tari and A. Fry, “Controlling aggregation in distributed object systems: A graph-based approach,” IEEE Trans. Parallel Distrib. Syst., pp. 23–32, December 2001.

[17] H. Yanxiang, L. Fei, and H. Wensheng, “The design and implementation of security communication model in grid networks,” in Int’l Conference on Computer Science and Information Technology, IEEE, ICCSI, 2008, pp. 421–424.

[18] A.-B. Ali, Z. Hussein, and S. Francois, “Access control mechanism for mobile ad hoc network of networks (MANoN),” Software Technology Research Laboratory, De Montfort University, Leicester, Tech. Rep., 2009.

[19] H. Mohteshim, “Passive and active attacks against wireless LAN,” in IASTED, 2004. Int. Assoc. of Sci. and Technology for Development, 2005. [Online]. Available: http://www.iasted.org/conferences/2004/ Innsbruck/pdcn.htm

[20] J. McLean, “The algebra of security,” in IEEE Symp. on Security and Privacy. Naval Research Laboratory, Washington, D.C.: IEEE Comput. Soc., 2008.

[21] G. Laccetti and G. Schmid, “A framework model for grid security,” Future Generation Computer Systems, vol. 23, no. 5, pp. 702–713, June 2007.

[22] NHSE, National HPCC Software Exchange, pp. 4–8, 2009. [Online]. Available: http://wotug.org/parallel/nhse/

[23] A. Imine, A. Cherif, and M. Rusinowitch, “An optimistic mandatory access control model for distributed collaborative editors,” INRIA, Tech. Rep., 2009.

[24] I. Foster and C. Kesselman, “Globus: A metacomputing infrastructure toolkit,” The International Journal of Supercomputer Applications and High Performance Computing, vol. 11, pp. 115–122, 1997.

[25] D. Chadwick, “Authorisation in grid computing,” Information Security Tech., vol. 10, pp. 33–40, 2005.

[26] C. Rongxing, Lu; Zhenfu, “A simpler user authentication scheme for grid computing,” International Journal of Network Security, vol. 7, pp.202–210, 2008.

[27] Z. Weide, W. David, D. V.and Glenn, and H. Marty, “Flexible and secure logging of grid data access,” in 7th IEEE/ACM Int. Conf. on Grid Computing, (Gridn 2006). Barcelona, Spain: IEEE/ACM Computer Society, 2006, pp. 1–8.

[28] MSDN, “Data confidentiality,” 2005. [Online]. Available: http://msdn. microsoft.com/en-us/library/ff650720.asp

[29] N. Syed and R. Michel, “Grid security services simulator (G3S)—a simulation tool for the design and analysis of grid security solution,” in Proc. First Int. Conf. on e-Science and Grid Computing (e-Science ’05). IEEE Computer Society, 2005, pp. 421–428.

[30] D.Welch, & S. Lathrop, 2003. Wireless Security Threat Taxonomy. Proceedings of the 2003 IEEE workshop on information assuarance United States Military Academy West Point. NY.

[31] M. W. W Vermeer, & P. M. G. Apers, 1996. On the applicability of schema integration techniques to database interoperation. In International Conference on Conceptual Modeling/the Entity Relationship Approach, pages 179-l 94.

[32] L. V. Lakshmanan, F. Sadri, & , I. N Subramanian, 1996. SchemaSQL - a language for interoperability in relational multi-database systems. In Proceedings of the 22nd VLDB Conference

[33] L. Bo, F. Ian, , S. Frank, A.Rachana, , & F. Tim, (1990,). Attribute Based Access Control for Grid Computing. pp. 1-13. International Journal of grid computing, vol.9 no.3, March 1990

[34] T. Priol, 2005. GRID Middleware. South Korea, Advanced Grid Research Workshops through European and Asian Co-operation published by the European Research Consortium for Informatics and Mathematics, pp. 1-11.

[35] BIBLIOGRAPHY Chuan-Lun, R., Xiao-Hui, Z., Zhong-Xian, L., Xin-Xin, N., & Yi-xian, Y. (2010). Towards Hierarchical-User RBAC model. International Conference on Machine Learning and Cybernetics (ICMLC), 2010 (pp. 2870- 2874). Qingdao, China: IEEE.