World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:5, No:2, 2011

A Context-Aware based Authorization System for Pervasive Grid Computing

Marilyn Lim Chien Hui, Nabil Elmarzouqi, and Chan Huah Yong

***Abstract*—**This paper describes the authorization system architecture for Pervasive Grid environment. It discusses the characteristics of classical authorization system and requirements of the authorization system in pervasive grid environment as well. Based on our analysis of current systems and taking into account the main requirements of such pervasive environment, we propose new authorization system architecture as an extension of the existing grid authorization mechanisms. This architecture not only supports user attributes but also context attributes which act as a key concept for context-awareness thought. The architecture allows authorization of users dynamically when there are changes in the pervasive grid environment. For this, we opt for hybrid authorization method that integrates push and pull mechanisms to combine the existing grid authorization attributes with dynamic context assertions. We will investigate the proposed architecture using a real testing environment that includes heterogeneous pervasive grid infrastructures mapped over multiple virtual organizations. Various scenarios are described in the last section of the article to strengthen the proposed mechanism with different facilities for the authorization procedure.

***Keywords*—**Pervasive Grid, Authorization System, Context-awareness, Ubiquity.

I. Introduction

GRID computing was introduced for the purpose of integrating the distributed resources together to solve scientific and industrial complex computational problems [1]. Multiple groups of the resources collaborate together under a Virtual Organization (VO). Classical grid is able to provide high performance of computational processes. However, the ability to support exclusively static scenarios becomes the limitation of grid computing [2]. In recent years, grid technologies had been evolving to Pervasive Grid. This new research area had overcome the limitation of classical grid computing by considering the user environment factor and specifically the dynamic context in their authorization decision making process.

In a grid infrastructure, many resources are integrated together to provide and facilitate high performance resource access and accommodate services for different users in a transparent environment. Some of the data are confidential. We need distinguished and specific mechanism to ensure the authorized user to access the resource or system. On the other hand, it will protect the stakeholder. By integrating authentication system, authorization system and access control, the chances of resources misused by unauthorized user or even authorized user can be reduced. However, current classical authorization systems are not suitable for pervasive grid environment. The authorization system must be able to have the mechanism to detect and respond to the changes of context information, in order to adapt into Pervasive Grid environment

In this paper, we present some existing authorization systems and our ongoing work on a suitable architecture for the conception and development of a suitable authorization system in pervasive grid environment. In Section II, we give an overview of Authorization System, Pervasive Grid, and existing Grid Authorization system. In section III, we list out the requirement needed for the architecture of authorization system in pervasive grid. In Section IV, we outline our proposed architecture that collaborates with PERMIS as our credential provider; we also describe a scenario on how the architecture works. Finally in Section V we provide a summary of our work, and some future works.

II. Authorization Mechanism in Pervasive Grid

Pervasive Grid [3] is an advance Grid technologies by integrating sensing instruments and devices with classical grid. The basic discipline of pervasive grid enables ubiquitous concept, in which the ability to detect and dynamically respond to changes of conditions surrounding the object and user [2] in the execution environment becomes crucial. Currently, the research challenge in pervasive grid affects multiple advances in grid computing and pervasive environments as an integral part of same framework. There are some concerns regarding to (1) enhancement of the abstraction, programming model and the system to support dynamic context attributes and context-awareness notion, (2) improvement of the data quality management to characterise the information for better decision making, and (3) extension and improvement of runtime execution and middleware services to support context-aware and dynamic context.

The authentication and authorization notions are often mixed up. The same goes to authorization and access control [4]. Authentication is a mechanism to identify users of a given system, while Authorization is the mechanism to determine

Marilyn Lim Chien Hui is with the Universiti Sains Malaysia, Penang, Malaysia, (corresponding author to provide phone: 6-016-4221707; e-mail: marilynlch@ hotmail.com).

Nabil Elmarzouqi, is with the Universiti Sains Malaysia, Penang, Malaysia. (e-mail: nabil@cs.usm.my).

Chan Huah Yong is with the Universiti Sains Malaysia, Penang, Malaysia, (e-mail: hychan@cs.usm.my).

the level access of a user [5]. This means the authentication is responsible for verifying user, and provide credentials to user, meanwhile authorization is responsible for storing information about user access levels, permissions or roles [5], and checking the authority of a user on a specific sets of resource [4]. On the other hand, authorization is also defined as granting access rights, while access control is circumscribed by access verification and rights [6]. In this section we are going to look further detail into authorization system for pervasive grid environment.

A. Overview of Authorization System Mechanisms

In a virtual organization (VO), authorization system is responsible to filter the user from gaining access to resources [7] by checking their groups, roles, or permission. It's an important stage because it reduces the risk of the virtual organization misused by outsiders or even the user themself. There are two types of mechanisms in authorizing a user: (1) push-based mechanisms and (2) pull based mechanisms. For the ease of understanding, we will explain the relationship between the user, resource and the authorization authority [8] for both models using some example process.

In push-based model [4], before the user sends the service request to the resource, they need to have valid certificates. Firstly, they need to send an authorization request to the certificates generator along with their identity credential. The generator will check the credential and generate a certificate to the user which contains roles and group memberships. These certificates normally will have a time of validity before it expired [8]. When user receives the certificate, they will push the certificate to the resource. The resource sites will grant or deny user's access based on the certificate validity [4]. Push model are more scalable, as the certificate generator is loosely couple with the resource's access controller, which means the process of assigning certificates and access controlling can be done in separate time [4] to avoid traffic conjunction at resource site.

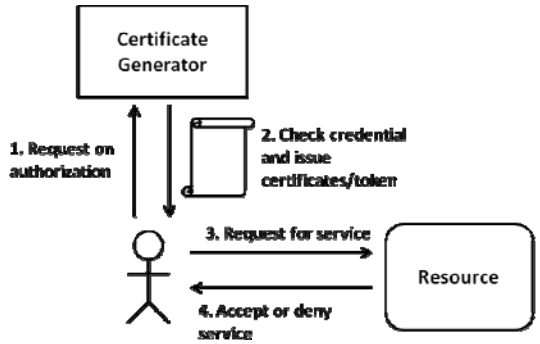

Fig. 1 The Push based model

In pull-based model, user can directly send their request for service to the resource, with minimum credential [8]. The resource's access controller will map user's name and password to a set of policies in a database. Once the user's name found in the database, the access controller of the resource will pull the corresponding permissions into the controller [4]. Base on the permissions, the user will allow performing operations that permitted on the resource. Pull based model are more user friendly compare to push model, as it doesn't request the user to predefine certificates in an authorization process [4].

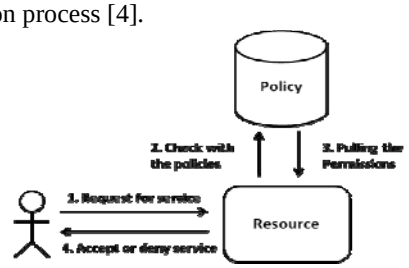

Fig. 2 The Pull based model

B. Characteristics of Existent Authorization Systems

In this section, we will compare among nine authorization systems by looking into five main characteristics of authorization system: scalability, security, revocation, interoperability and push/pull. For this, we selected nine well known authorization systems as: (1) *Community Authorization Service (CAS)* (2) *Virtual Organization Membership Service (VOMS)* (3) *Enterprise Authorization and Licensing Service (EALS)* (4) *Context-Constrained Authorisation (CoCoA)* (5) *Security Expert Initiative (SEINIT)* (6) *Akenti* (7) *Privilege and Role Management Infrastructure Standards Validation (PERMIS) Project* (8) *Gridmap*,(9) *Context-Aware Access Control for Pervasive Access to Process-Based Healthcare System (HDGPortal)*, where the CAS,VOMS, EALS, CoCoA and SEINIT are categories as centralized models, while Akenti, PERMIS, Gridmap and HDGPortal are under non-centralized models.

Generally, the mechanisms of authorizing a user to a resource play an important role in determining the scalability. The push mechanisms are considered to be more scalable than the pull based mechanisms, due to the decoupling between the certificate generator and the access controller [4]. In terms of performance scalability, CAS and VOMS are adopting push based model, and they distributed those credential to the user themselves [7]. This reduces the number of necessary trust relationships between consumers and resource providers [9], hence they are able to support more users in the VO even the number is increasing significantly. As a result they scored highly in scalability in size [10]. On the other hand, EALS, Akenti and Gridmap that using pull based model are less scalable but they take advantages of user friendliness as the access controller take care of granting access to the user without requesting them to obtain certificates from the certificates generator. PERMIS can be configured to use either a pull or a push model. In terms of administrative scalability, CAS and VOMS, are adopting a centralized policy database, thus they allow the admin to manage the database easily with only a single place update when there is some modifications for the user community. However, there might be the risk in single point failure [4]. Hence, if the community policies are not changing frequently, single master server is more efficient

with the add-on of routinely replication of policies to read-only slave servers, else multiple peer server are more preferable [9]. Furthermore, centralized solutions may cause infeasibility of synchronization [10]. In PERMIS, only users belonging to the centralized main domain are authorized to access the resource by receiving credential. Akenti is designed specially for high-performance distributed network environment, for example: web resource and Grid environment [11], hence it's using distributed policies database. Whereas, Gridmap has the highest administrative overhead, as the administrator need to update each system individually when there is modification in the user community.

Another important characteristic for Authorization System is security. There are two types of security attack in grid authorization system: masquerade attack and Denial-of-Service (DoS) attack. Masquerade attack happens where an adversary break into the system as an administrator, and modify the authority of certain user, whereas DoS attack happens when excessive authorization request has flooded the server and caused it to be down. Since CAS, VOMS and Gridmap are centralized database for storing policies, hence they are using GSI credential based authentication mechanism [4] for communication encryption and authentication [11]. In the latest version of VOMS, it uses a short lived credential which includes Attribute Certificate (AC) to make it more compatible with other authorization system [4]. Besides, VOMS also support multiple stakeholder just as EALS and Akenti do, which means if one of the database was hacked, other resources wouldn't be affected. Both Akenti, and PERMIS are using certificates which follow the X.509 specification to prevent security attack. Akenti does not use true AC [7]. Instead it stores its policies in three types of certificates, which is policy certificates, use condition certificates, and attribute certificates [4]. Policy certificates are responsible for the sources of authority for the resource use condition certificates and keeps the rules that control the access to the resource, and attribute certificates specify the attributes to the users that are needed to satisfy the use conditions. Whereas PERMIS store its policy attributes certificate in LDAP server [12]. Since EALS are more on enterprise purpose, for the portability of the wide range of usage, it was designed to support different types of security protection feature like: passwords, certificates and other credentials. However in case when there is a DoS attack, Akenti and EALS will distribute the requests to multiple servers, to avoid the server down. Among the grid system discussed above, Gridmap is the most unaffected to DoS attack; the impact can be neglectable as the database is distributed into different resources. It will only have impact if and only if the adversary attacks a significant number of resources. Besides those methods mentioned above, some of the techniques like resource level checks and ingress filtering are also recommended to reduce the DoS attacks.

Revocation [4] depends on the push/pull based model as well. Since CAS and VOMS do not have explicit revocation mechanisms, in the case that there is an adversary hacked into the system, it can access entire resource according to the gained credential. EALS, Akenti and PERMIS have inherent revocation mechanisms; hence they are able to terminate the adversary immediately by updating the policies manually. Gridmap also have inherent revocation, but it's different in the sense that the administrator needs to change the policy in each and every resource in the system during revocation.

Since there are different categories of grid system, it is important to have a communication protocols between multiple authorization systems. For the ease of communication, standardization method is introduced. CAS, VOMS and PERMIS are using SAML Standard (Security Assertion Markup Language) as the mark-up language [4]. However, in terms of interoperability, CAS is more compatible comparing to VOMS. This is because SAML allows CAS to work with Web service and OGSA tools, where as VOMS cannot be easily integrated into Web services and OGSA tools due to it is not based on OGSA framework [12]. CAS certificates are structured as: CAS server Distinguish Name as the subject; and the authorization information is included in an extension. This will reduce the effectiveness as the system needs to check the extension as well in order to decide who the owner [7]. Whereas VOMS authorization information is separated into two categories: relationship and privileges between users and VO; and the possible operations allowed to the resources [11]. VOMS adds the AC's in a non-critical extension of a standard proxy certificates [7]. In another words, Grid system service might need to be modified to use the CAS certificates, whereas VOMS certificates does not require any changes to the services. Akenti are using XML (Extensible Markup Language) standards in their policies management. EALS are designed for enterprises purposes, hence it needs to be able to cater wide range of system. Thus, it has an adapter for most industry products. It can exposed the access control policies through (eXtensible Access Control Markup Language) XACML standards and exchanging authorization credentials using SAML. Although Akenti are designed for web resources purpose [9], and it might potentially be used for enterprise purpose, but due to the issues in standards and licenses, it cannot be directly used in enterprise scenario.

CAS, VOMS, EALS, Akenti, PERMIS and Gridmap are not suitable to be use in Pervasive Grid environment, as they are not considering the context attributes during decision making. The CoCoA, SEINIT and HDGPortal authorization system are able to support pervasive grid by including the context attributes into their decision making process.

CoCoA [2] and SEINIT [13] are adopting push based model, hence they have high user scalability as other push based authorization system does. While HDGPortal [14] is using Pull based model where it stores policies at resource site. CoCoA and SEINIT is using GSI in their authorization decision [13]. The authorization decision mechanism in HDGPortal is certificate-based, and it's relies on CAS certificates [14].

According to the analysis given in this section, we summarize the characteristic of the existent authorisation systems into Table I.

TABLE I
COMPARISON OF EXISTING GRID AUTHORIZATION SYSTEM

	Centralized models					Non-centralized models			
	CAS	**VOMS**	**EALS**	**CoCoA**	**SEINIT**	**Akenti**	**PERMIS**	**Gridmap**	**HDPortal**
Support Pervasive Grid	No	No	No	Yes (Context Aware)	Yes (Context Aware)	No	No	No	Yes (Context Aware)
Push/ Pull	Pull	Push	Pull	Push	Push	Pull	Push or Pull	Pull	Push
User Scalability	High	High	Medium	High	High	Medium	High	Medium	High
Authentication Method	Using GSI	Using GSI	Password/ Certificates	Using GSI	Using GSI	Certificates	Certificates	Using GSI	Certificates
Revocation	No	No	Fast	No	No	Fast	Can be fast	Have to be updated	No
Decision making	Requires separately	Requires separately	Integrated in scheduler and License Manager	Based on policies	Based on policies	Single step, through capability certs	Two steps, "Yes/No answer"	Based on policies	Based on policies

C. The Limitation of Existent Authorization Systems

By studying the existing authorization system and the research effort being done by previous researcher, a number of studies have been done on the improvement of the scalability, security, revocation and interoperability for classical authorization system (CAS, VOMS, EALS, Akenti and GridMap). However, these research efforts are more focusing on the entity to verify the identity of user. The main limitation of classical authorization system is they only work with static attributes such as roles and group memberships, but not context attributes such as location of user, history, times and other environment information. The access request is granted as long as the subject has a valid Grid credential [2]. This caused classical authorization system unable to works well in pervasive environment, as the context attributes are changing from time to time.

Although there are some existing system which introduce context-awareness notion into the authorization system, to make them adapted into pervasive environment such as CoCoA, SEINIT, and HDGPortal, but most of them are using either push based mechanism or pull based mechanism. Both of the mechanism has their advantages and disadvantages. A hybrid mechanism which integrated push and pull based mechanism will utilise the advantages of both mechanism and increase the efficiency of decision-making.

We plan to design an architecture for authorization system to overcome the limitation of current authorization system. It will adapt the context-aware feature and using the hybrid push-pull based mechanism. The objective of this research work is (1) to integrate context-aware with existing grid authorization system (2) to improve the decision making by adopting integrated push and pull based model into context-aware based authorization system.

III. REQUIREMENT FOR AUTHORIZATION SYSTEM IN PERVASIVE GRID ENVIRONMENT

In pervasive grid, there are two type of information needed for the authorization module to make a decision [2]. The first type is the entity that can verify the identity of the user, for example: identity number, roles, group membership, permission and so for. The second type of information is the context attributes. Context attributes is the environment information of the user or resources. For example: time, location, network usage, history and so for. Hence, there should be a mechanism to handle those two types of information either parallel or sequentially.

Since the context attributes will always change from time to time, we need another function to monitor the changes of the environment factor. The function should be able to update the decision making module to re-authorise the authorised request as the environment changed. The pervasive authorization system must be able to adapt into different situation or scenario and various devices.

In order to have a more efficient decision making process, we suggest to combine both push and pull mechanism, become a hybrid model of authorization which will increase the efficiency of decision making process with lower cost. The push mechanism will mainly be responsible to push the certificates information to the decision service module, while the pull mechanism will handle in pulling the context aware attributes in an ad hoc situation. For example: changes of context attributes in the middle of accessing to VO.

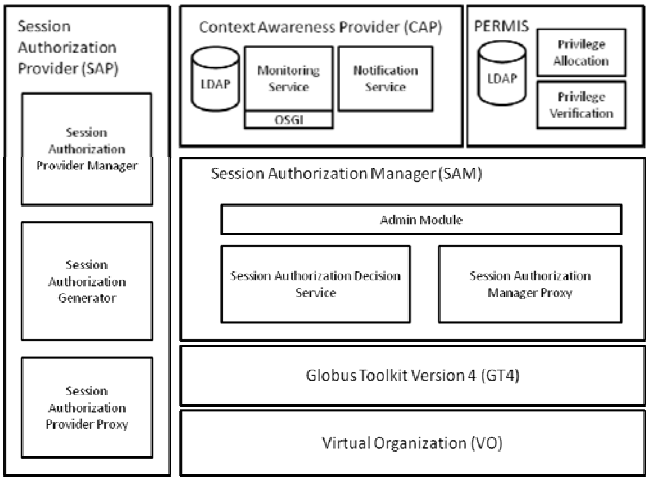

Fig. 3 The Context-Aware based Authorization System Architecture

IV. CONTEXT-AWARE BASED AUTHORIZATION SYSTEM ARCHITECTURE

Base on the requirement above, our proposal for an authorization system architecture aim to provide an context-aware based authorization process that is able to combine the existing grid authorization system with context assertions for

an authorization system that suit the pervasive environment using the hybrid model consist of both push and pull mechanism. The architecture supports both push and pull based mechanism. There are four main components in this architecture (Fig. 3): PERMIS, Context Awareness Provider (CAP), Session Authorization Manager (SAM), and Session Authorization Provider (SAP).

A. Privilege and Role Management Infrastructure Standards Validation (PERMIS)

The PERMIS PMI architecture [12] consists of two subsystems: (1) privilege allocation and (2) privilege verification. Privilege allocator is responsible to issue X.509 attribute certificates to users. The role assignment attributes certificates are stored in the LDAP directory. Those attributes include permisRole and ISOCertified [12]. Privilege verification is in charge of authenticating and authorizing the users.

In order to show that our system can be integrated with existing authorization system, we adopt one of the existing classical grid authorization system as part of our module. The reason we choose PERMIS is because of its lightweight, role based, and most importantly the incorporation of X.509 certificates that will contribute as the basic ITU-T (International Telecommunication Union-Telecommunication standardization sector) certificates standard for a public key infrastructure (PKI) to integrate with context attributes provided by our Context-Awareness Provider. Furthermore, the PERMIS PMI supports two configuration mechanisms through push and pull models. Therefore, it will be easier to collaborate with our hybrid model later.

B. Context Awareness Provider (CAP)

The Context Awareness Provider (CAP) is a tool to monitor and provide context information to Session Authorization Provider (SAP) and Session Authorization Manager (SAM). The Monitoring Service checks the state of the context attribute each time the user makes an authorization request. For example: location, time of access and others. Attributes are kept in the LDAP directory for subsequent use by Session Authorization Provider Manager. Context attributes might change according to the environment and its constituents. After those attributes had been stored in LDAP directory, the Monitoring Service will continue to monitor the attribute. In Figure 4, if changes are detected, Notification Service will acknowledge the Session Authorization Manager to make the authorization decision based on new attributes. The Session Authorization Decision Service can either continue to grant the access rights to the user, or suspend the current access rights.

C. Session Authorization Provider (SAP)

Session Authorization Provider is responsible for integrating attribute, issuing certificates and creates Session Authorization Manager (SAM) for each user.

In Figure 5, when the newly registered user sends a request for service, a simple message including the user identity (name, roles) will be sent to the SAP Manager. The SAP Manager will receive the request, and started to collect the certificate from PERMIS LDAP directory and Context attributes from CAP LDAP directory. It filters context attributes and passing only the selected attributes to the Session Authorization Generator.

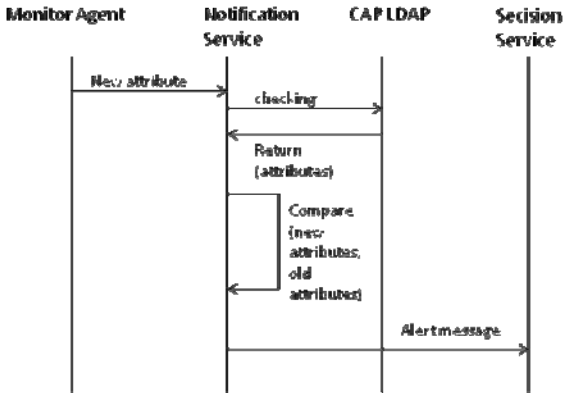

Fig. 4 Sequence Diagram of Notification

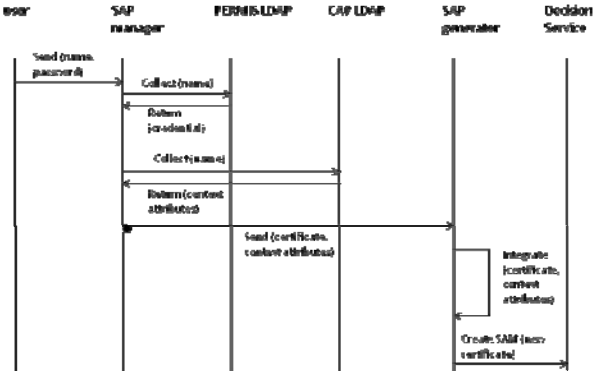

Fig. 5 Sequence Diagram of Creating a Session Authorization Manager

The Session Authorization Generator will integrate the context attributes into the certificate. The certificate will be containing the user's role, group membership, subject domain, location of user, time of usage and others, after combined with the context attributes. The newly generated certificate is suitable to be used in Pervasive Grid environment. Session Authorization Generator will create a Session Authorization Manager (SAM) for the user with the new certificate. The certificates will be used by the SAM in future to make the authorization decision.

If the same user wishes to start a service in the future, the flow of the process will be slightly different. When the Session Authorization Manager receives the request from the user, it will check with the history list. If the user name found in the history list, it will redirect the request to the Session Authorization Provider Proxy. Since each user have their personal SAM, the SAP Proxy will then match the user with their belonging SAM, and escalate the user's request to the SAM for authorization decision. This will save the time by avoiding the unnecessary process.

D.Session Authorization Manager (SAM)

Session Authorization Manager (SAM) is the core part of the authorization architecture. Here are the places where authorization decisions are made.

In Figure 6, when Session Authorization Decision Service receive the user certificate send by SAP Generator, it will use the attributes and the policies in the certificates to make the decision either to grant access to the user request or deny the request. Base on the user policy, different role will have different level of access to the resource. User will be given the access rights on certain resource sets, base on their role and membership.

If the decision service authorise the user request, it will forward the user's service request to the SAM Proxy. The Session Authorization Manager Proxy will distribute the jobs summit to the resources in the virtual organization through the Globus Toolkit Interface (GT4).

If user were to submit another request for service in the future, instead of receiving the message from SAP Generator, the Session Authorization Decision Service will receive the user request forwarded by SAP Proxy. In this case, Session Authorization Decision Service will acknowledge the CAP to check the context attributes. If changes of context attributes are detected, Session Authorization Decision Service will pull the new context attributes list, and swap the new attributes into the existing certificates. Base on the modified certificates, the Decision Service will re-authorise the user's request of service.

After the request have been authorised, user can access to the resources to run required jobs. However in the middle of using the VO, if changes of context attributes were detected, Decision Service receives the alerts from notification service. According to the new context attributes received from notification service, Decision Service will reauthorise the user's request, and will choose to continue authorise the user to the resource access or suspend their access.

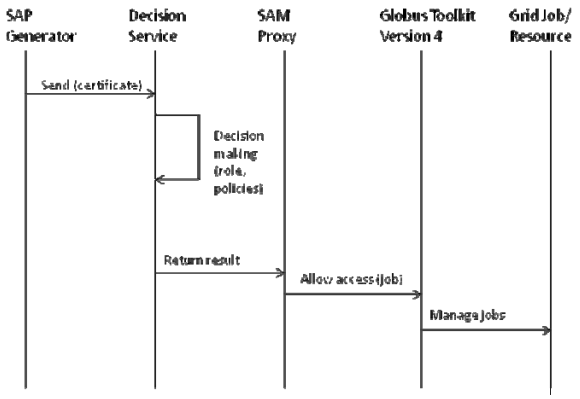

Fig. 6 Sequence Diagram of Making Authorization Decision

V.Experimental scenarios

In Figure 7 and Figure 8, we present a sample of our testing pervasive grid environment, which consists of heterogeneous infrastructures including computers, PC tablets, cameras, robots projectors, printers and servers distributed in three halls, and a grid cluster hall. These pervasive grid infrastructures are then structured into several Virtual Organizations (VO): here we present an example of three VOs. All infrastructures will be tagged with the RFID tag, the RFID will be playing a very important role in analysing the location of user and infrastructure. In this environment, we present the following services:

- RemoteControlRobotsService – This service allow the user to remote control the robots base on the tablet. Some API will be installed on the tablet to communicate with the robots. The API functions as the controller for the robots. When the user entered data on the controller, it will send to the robot and change the movement of the robot.
- MonitoringService – This service enable to user to access control the IP camera to monitor the physical environment of the halls. The user can control the rotation of the camera angle or zoom in to observe the situation.
- ProjectorService – This service allow the user to project their presentation slides or video onto the projection slide.
- PrintingService – This service allow user to print their documents with the available printer.
- SendingJobService – This service allows user to submit computational jobs to many commonly used distributed hosts through CoGKit toolkit (Globus Java Commodity Grid), which communicates with different Globus gatekeepers, includes the GSI (Grid Security Infrastructure) specification and uses the IAIK java SSL libraries to delegate credentials.

When the user enters the Virtual Organization (VO), our proposed authorization system will generate a certificates which containing user entity for the user. Those information included encryption algorithm, subject, validity and so for. Meanwhile, RFID will collect the environment condition of the user and resources. For example: Location, Network Condition and so for. The context attributes collected will be integrated into the certificate, for further use in authorization decision making. Figure 9 presents a sample of the new certificate after integration of x.509 certificate with context attributes. This certificate is generated using the PERMIS Attribute Certificate Manager. It contains (1) user attributes, (2) context attributes, (3) Policies. Based on the new certificate, our authorization system will push the identity and context information to the authorization decision-making service to filter the accessibility of the user. Different users will be having different resource access based on their roles and permissions, and policy set by the resources site. While the context information changing from time to time, we will have IP camera, sensors, and RFID to collect the information of the VO environment. The context-awareness service will pull the environment attribute of the user and available resources information and monitor their changes for the decision service to re-evaluate the decision on access control.

RFID will be playing an important role in the location tracking for the scenarios above. When the RFID tag carried by the user detects any movement, a signal will be sent to the surrounding access point to obtain the current location of the

moveable object. The access point in that area will pick up the signal and calculate the signal strength and pass it to the RFID dongle which is connected into the host PC in cluster room. Those signals will be converted to readable information, which contains the location and other attributes as well. On the other hand, a floor plan of the experiment environment will be drawn and the location of access point will be pre-marked. Hence, the access point that picks up the signal will be able to derive the location of the user.

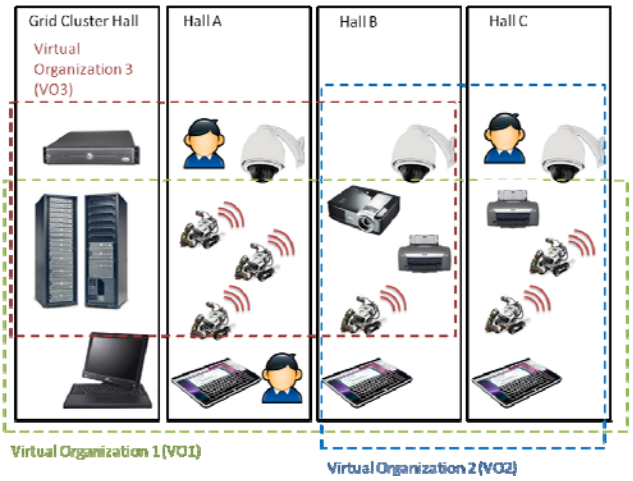

Fig. 7 The Virtual Organization and Infrastructures in our Prototype

Fig. 8 The Pervasive Grid Environment in our Prototype

For the development of our prototype, one of the servers from the School of Computer Science of Universiti Sains Malaysia, which is working in Linux environment have been allocated for this project. PERMIS infrastructure and openLDAP server has been installed and set up in this server. PERMIS infrastructure will focus more on policies creation and certificate generation. The PolicyEditor will create policy in XML form, whereas Attribute Certificate Manager will generate an X.509 AC like shown in Figure 9, openLDAP will function as light weight database to store the policies and certificates. The reason of choosing PERMIS had been explained in section IV.

As for the decision part, more constraints will be included in the access control process. Besides location, the priority of group and range are added in to the consideration when allocating the available resource to user request. Algorithm to process the range and priority will be implementing in the Session Authorization Decision Service module as mention in section IV and also manage the queuing request.

```
SEQUENCE { -- AttributeCertificate --
  acinfo = SEQUENCE { -- AttributeCertificateInfo --
    version = 1,
    holder = SEQUENCE { -- Holder --
      baseCertificateID = [OPTIONAL; omitted],
      entityname = directoryName: CN=A PERMIS Test User,O=PERMISv5,C=GB,
      objectDigestInfo = [OPTIONAL; omitted]
    },
    issuer = CHOICE { -- AttCertIssuer --
      v2Form = SEQUENCE { -- V2Form --
    issuerName = directoryName: CN=A PERMIS Test User,O=PERMISv5,C=GB,
    baseCertificateID = [OPTIONAL; omitted],
    objectDigestInfo = [OPTIONAL; omitted],
        }
    },
    signature = sha1WithRSAEncryption (1.2.840.113549.1.1.5) ,
    serialNumber = 381905652160288859671274699979569893396074291449,
    attCertValidityPeriod = SEQUENCE { -- AttCertValidityPeriod --
      notBeforeTime = Fri Aug 08 08:00:00 SGT 2008 -- GeneralizedTime --,
      notAfterTime = Thu Aug 08 08:00:00 SGT 2013 -- GeneralizedTime --
    },
    attributes = SEQUENCE { -- Sequence of Attributes --
      SEQUENCE { -- Attribute --
        type = 1.2.826.0.1.3344810.1.1.13 -- ObjectIdentifier --,
        values = SET { -- Set of AttributeValues --
          PrintableString = "<?xml version="1.0" encoding="UTF-8"?>
          <!DOCTYPE X.509_PMI_RBAC_Policy>
          <!--This policy tests integration of PERMIS with Shibboleth.
          SubjectDomain0: world
          SubjectDomain1: O=PERMISv5,C=GB
          guest: CN=guest,O=PERMISv5,C=GB
                 permisRole:Role0
          developer: CN=developer,O=PERMISv5,C=GB
                 permisRole:Role1
                 .
                 .
                 .
                 .
                 .
                 .

        </X.509_PMI_RBAC_Policy>

    issuerUniqueID = [OPTIONAL; omitted],
    extensions = SEQUENCE {
    Context:
        Location: 001
    }
  },
  signatureAlgorithm = sha1WithRSAEncryption (1.2.840.113549.1.1.5) ,
  signatureValue =
0001010110101111100110001100011100011101000111001001001001001010100110010101
```

Fig. 9 Example of certificate created after combining the contextual attributes

Location is the core attribute to represent the context attributes in our prototype in current stage. Others context attributes will be included as well in our future work. By controlling the location attribute value, and other user attributes in the certificate, we will get different results from different set of test data. Result will be collected and compare with other similar authorization system base on scalability and request queuing time.

In order to demonstrate the functionality of our proposed authorization system, we have set a few scenarios to test our system. Some simple API has been developed in JAVA platform to mimic some of the modules in our architecture, where later we will transform them into agent in Java Agent Development Framework (JADE) platform. Several scenarios are supported by our proposed architecture as below:

• When the user request for different services, our authorization system will authorise the user to access the available resources which is nearest to the user. Thus the resource can be saved and the request can be completed in shorter time. For example, Figure 10 shows the steps performed for a success authorization based on location:

1. Nic is a user of our VO and his role is guest. He log in to our grid service.
2. The SAP will load the policy and AC of Nic from the openLDAP server.
3. Base on Nic's roles, he will only have access to certain

service like RemoteControlRobotsService and PrintingService.

4. Nic request for RemoteControlRobotsService.
5. The CAP will check and return a list of location and status of all robots to SAM.
6. The CAP will check and return the location of Nic to SAP which is "Hall B".
7. SAP will integrate the location of Nic into the AC.
8. Base on the new AC and available robots, SAM decision maker will authorize Nic to access those robots.
9. In this case, Nic is in Hall B, hence the SAM supposes to allocate him the robots in Hall B (Nearest to Nic).
10. If the robots in Hall B used by other user, then SAM will suggest robots in Hall A and Hall C.

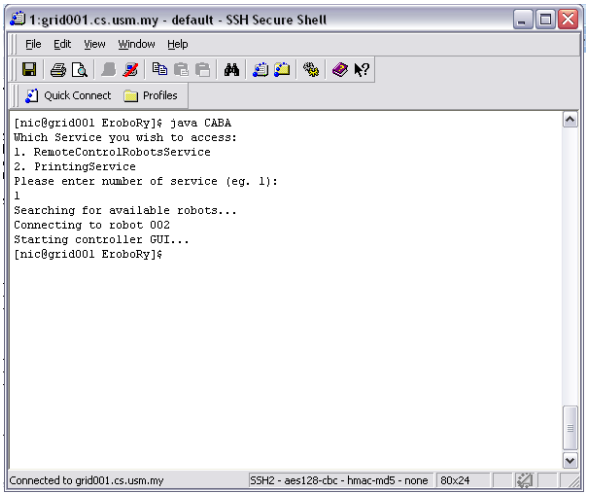

Fig. 10 Authorization Based On Location and Context-Awareness

• Normally, user will gain the access to the available resource using their roles. However when the network is in heavy load, the resources will deny the access of noncritical user and provide the access only to the critical user. After a while when the network is back to normal load, the authorization will grant the access to noncritical users as before. For example, Figure 11 and Figure. 12 shows the steps performed for a success authorization based on role priority:

1. Now there is another user name "Nabil" joined our VO as a developer role, which have higher role priority then guest.
2. Nic still maintain the role as guest, hence he only allows for certain service. The SAP will validate the resource that Nic able to access base on its role.
3. Nic request for PrintingService.
4. SAM will authorize Nic to access the printerB in Hall B.
5. Printer C in Hall C is accessed by Marilyn the admin.
6. When Nabil request for printing Service, the SAM will send an message to request the CAP to check for the status and location of all printers and return to SAM.
7. Since all printers are busy, SAM will check the role of Nabil.
8. If Nabil have higher role priority, the SAM will ask Nabil whether to access the printer which is nearest to him (in this case is printerB), or continue the request in queuing.
9. If Nabil take over the printerB, then his tablet's SAM will send message to Nic's SAM to reauthorize Nic access to printerB. If not, then Nabil will continue wait until there are available printer.

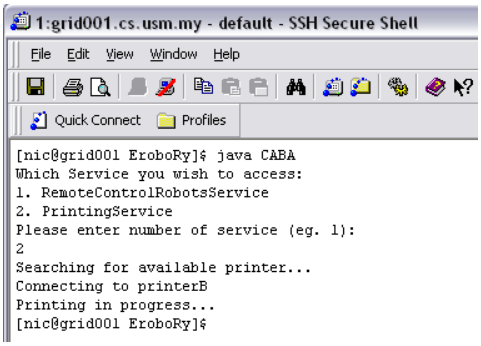

Fig. 11 Authorization Based On Role Priority by Role Guest

In the scenario given above, we have shown that our proposed context-aware based authorization system is able to adopt in pervasive grid computing environment and works with multiple Virtual Organization (VO). During the collaboration, new user, new resource and new VO will be added in. The authorization will update the decision making base on the policy. We plan to have more details about the implementation and performance of our prototype in the camera-ready version of this paper.

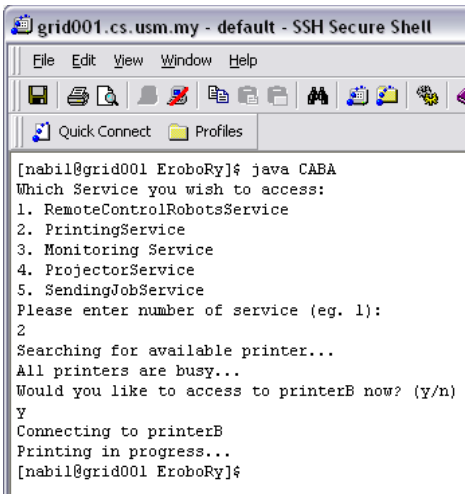

Fig. 12 Authorization Based On Role Priority by Role Developer

VI. Conclusion

In this paper, we have described the main characteristics of authorization system, the requirement for authorization system in pervasive environment and new authorization system architecture for pervasive grids. We aim to integrate the context-awareness with existing grid authorization system, as well as increasing the efficiency of authorization decision-making process by introducing a hybrid model of authorization mechanism. The proposed architecture provides the basic services for: (1) integrating user attributes and

contextual attribute by extending existing authorization system with context attributes and (2) reauthorizing the authorized service when there are changes in contextual attributes. Currently, we extend the architecture to support authorization for groups of users and other useful functions to locate RFID tagged objects.

REFERENCES

[1] Ian Foster , Carl Kesselman , Steven Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations, International Journal of High Performance Computing Applications, v.15 n.3, p.200-222, August 2001

[2] J. Chin, N. Zhang, A. Nenadic, and O. Bamasak, "*A context-constrained authorisation (cocoa) framework for pervasive grid computing*," *Wireless Networks*. [Online]. Available: http://dx.doi.org/10.1007/s11276-008-0135-0

[3] "Pervasive Grids: Challenges and Opportunities," M. Parashar and J-M Pierson, "Handbook of Research on Scalable Computing Technologies," Editors: K Li, C Hsu, Laurence T Yang, J. Dongarra and H Zima, Information Science Reference, IGI Global, ISBN: 978-1-60566-661-7, 2009.

[4] Chakrabarti, A., *"Grid Authorization System"*, In Grid Computing Security. Chapter 5. Springer-Verlag, 2007.

[5] Robert G. Carter. "Authentication vs. Authorization". Kerberos: What, Why, How? 29 May 2010. http://www.duke.edu/~rob/kerberos/authvauth.html

[6] Andrew S. Tanenbaum, Maarten Van Steen, Distributed Systems: Principles and Paradigms, Prentice Hall PTR, Upper Saddle River, NJ, 2001 ISBN: 978-0-13088-893-8, 2002.

[7] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell' Agnello, A. Frohner, et. al., *"VOMS, an Authorization System for Virtual Organizations"*. In *Proceedings of the 1st European Across Grids Conference – Santiago de Compostela, Spain, 13-14 February 2003*, 2003.

[8] Markus Lorch, Bob Cowles, Rich Baker, Leon Gommans, Paul Madsen, Andrew McNab, Lavanya Ramakrishnan, Krishna Sankar, Dane Skow, Mary R. Thompson, *"Conceptual Grid Authorization Framework and Classification"*. In *Authrization Frameworks and Mechanisms - WG,2004*.

[9] "Pervasive Grids: Challenges and Opportunities," M. Parashar and J-M Pierson, "Handbook of Research on Scalable Computing Technologies," Editors: K Li, C Hsu, Laurence T Yang, J. Dongarra and H Zima, Information Science Reference, IGI Global, ISBN: 978-1-60566-661-7, 2009.

[10] Hung-Min Sun, King-Hang Wang, Pa Saffiong Kebbeh, *"Distributed Authorization and Authentication Framework for a Grid Infrastructure"*, 2007.

[11] LIU Shengjian, *"A study on the Mechanisms of Policy-based Grid Authorization"*, In *2009 International Conference on multimedia Information Networking and Security*, 2009

[12] D.W. Chadwick ad A. Otenko. *"The PERMIS X.509 Role Based Privilege Management Infrastructure"*. In *Proc. Of 7th ACM Symposium On Access Control Models And Technologies*, 2002.

[13] Vassiliki KOUFI and George VASSILACOPOULOS, *"Context-Aware Access Control for Pervasive Access to Process-based Healthcare Systems"*. In *eealth the Horizon –Get IT There S.K. Andersen et al. (Eds.) IOS Press,2008*.

[14] Vassiliki KOUFI and George VASSILACOPOULOS, *"Context-Aware Access Control for Pervasive Access to Process-based Healthcare Systems"*. In *eealth the Horizon –Get IT There S.K. Andersen et al. (Eds.) IOS Press,2008*.