

DYNAMIC POLICY MANAGEMENT IN MOBILE GRID ENVIRONMENTS

Tariq Alwada'n¹, Hamza Aldabbas¹, Helge Janicke¹, Thair Khmour², Omer Aldabbas³,

¹Faculty of Technology, De Montfort University, UK
{tariq, heljanic, hamza}@dmu.ac.uk

²Department of Information Technology, AlBalqa Applied University, Jordan
khdour@bau.edu.jo

³Department of Engineering, AlBalqa Applied University, Jordan
omer_aldabbas@yahoo.com

ABSTRACT

Mobile Grid Services has given the ability to move jobs, data and application software from nodes to nodes during jobs execution in the grid environment. These migrations depend on the grid's users and the grid's nodes policies. The heterogeneity, diversity of policies and attributes leads to a need for policy management tools that can handle these problems. Also, before the users can submit their jobs or run their applications on a certain resource or system they may need to guarantee that this resource or system has not been compromised, which could result in their own application or data being stolen or asking for certain users to be allowed to access the service. To date, not enough attention has been paid to policies that deal with such concerns. Most existing grid systems have support only limited types of policies (e.g. CPU resources). A few designs consider enforcing data policies in their architecture. Therefore, we propose a dynamic policy framework that addresses these issues (user-submitted policy, data policy and multiple Virtual Organizations (VOs)).

KEYWORDS

Grid, Policy Management, Mobile Grid Services, Security, Grid Resource Broker.

1. INTRODUCTION

A grid is a system that has the ability to manage and organize resources and services that are distributed across several control domains, utilize protocols and interfaces and supply high quality of service [12]. Grimshaw et.al. [14] define grid computing as "coordinated resource sharing and problem solving in dynamic, multi-institution virtual organizations".

Security is an essential element in grid computing. For any grid environment, there should be methods to offer security, including authentication, authorisation and data encryption. One of the important issues that research into grid environment tries to solve is how to keep distributed resources from unauthorized users and at the same time how to allow the sharing of resources and the accountability for resource handling. Every resource applies a local security policy that may result in the refusal of requests for utilizing of its resources. This results in problems for both resource requesters and providers who want to share resources in the lack of global security policies in grid systems. Because of the fact that there are a lot of elements, like users and resources contributing to the grid, security has become a critical aspect in checking the element trying to use a service (authentication), and in verifying whether this element is allowed or not to use the service (authorization). Securing the grid, therefore, is vital to give confidence to both resource providers and users to join the grid. Any secure grid environment should

provide mechanisms to secure authentication, resource protection, authorization, communication, data transfer and encryption [18]. One of the most important security challenges that face the grid environment is coordinating users' identities among local and wide networks and dealing with the variety of local security techniques for either resource or user, trust relationships between resources and users, end-user key, credential organization. Supporting security to resources in opposition to unsafe actions from grid users can be considered as another security challenge [13].

Policies are groups of regulations, standards and practices written by one or more owners of jobs or administrators of resources about how their resources or jobs can be handled and used. Policies decide how a job should be done, how security is applied in a domain and how an organization organizes, secures and distributes their resources. Depending on the Globus Toolkit [4], before the job submission, there should be many steps for authenticating the users who ask to use resources [10, 24]. However, after the authentication, there are no further resource access restrictions on how to use the resources. This is known as "all or nothing". At present, there is no existing well-defined model to set and check authentication policies of remote resources for grid job submissions.

Currently, there has been much research that focuses on policy management in the Grid environment [7, 17, 20, 21, 23]. Nonetheless, most of these approaches presume a single VO. The aim of Policy Management is "to apply an integrated management system so that system management, network management and application management can cooperate in Grid computing" [25]. Before the users can submit their jobs or run their applications on a certain source or system they may need to guarantee that this source or system has not been compromised which could result in their own application or data being stolen or which could result in asking for certain users to be allowed to access the service. The previous security considerations have to be taken into account when creating a grid system[15]. Another aspect that should be taken into account is the user policy. Before the users can submit their jobs or run their applications on a certain source or system they may need to guarantee that this source or system has not been compromised. This could result in their own application or data being stolen or could result in asking for certain users to be allowed to access the service. Most methods do not consider this in their policy management designs. To date, not enough attention has been paid to policies that deal with such concerns. Most existing grid systems have support only limited types of policies (e.g. CPU resources). We propose a policy-managed grid environment that addresses these issues (user-submitted policy, data policy and multiple VOs). The contribution of this paper lies in the fact that it introduces a new dynamic framework that supports policy management within and between VOs along with supporting the data policy and without forgetting the external users' policies rights when making final decisions. Traditional authorization policy management frameworks act well in authorization policy for a single VO where the contributing hosts grant the permission to follow a global authorization system. However most of policy management tools do not provide a clear support for sharing mobile resources between multiple heterogeneous VOs. Therefore; the question is: How does the grid interact with policies for different domains and organizations in the case of Mobile sharing and data movements.

This question raises a number of other questions. These are:

- How to introduce a policy framework that supports a multi-organization environment with different domains?
- How to introduce policy management tools that provide a clear support for sharing mobile resources between multiple heterogeneous VOs?

- How to present a policy framework that can support the user policy in its final decision?
- How to present a policy framework that considers enforcing data policies in its designs?

The rest of the paper will be organized as follows: The next section presents related works to our architecture. Section three describes our new framework for both single and multiple virtual organizations, while the fourth section describes the component of our grid architecture and describes each component in a separated section. In section five, we give an explanation of the grid portal as part of the grid architecture and its advantages. Section six presents our resource broker and its architecture including the suggested framework for the mobile grid policy services and a scenario that explains the advantage of mobility mechanism and the role of policy server in it. In the last section we discuss future possibilities and conclude the paper.

2. RELATED WORKS

The standardized architecture of the grid makes procedures and exchange parts easier between different organizations. Because grid tools and equipments are from multivendor, interoperability becomes important and high standards must be identified. For standardizing grid requirements, protocols and interfaces, the Globus alliance and Open Grid Forum (OGF) were launched, as described below.

2.1 Open Grid Forum

Open Grid Forum (OGF) [11] is a public society forum for discussing grid technology matters. The aims of OGF involve designing of open procedures for the improvement of grid agreements and specifications and create a grid architecture documents and most suitable applying guidelines. Many research groups within OGF have established various standards such as Open Grid Service Architecture (OGSA) to offer a service oriented view of the shared physical resources or services provided for these resources, Open Grid Services Infrastructure (OGSI) to describe methods for establishing and organizing grid services, GridFTP and JSDL [1, 2]; a lot of other subjects are at present being worked on.

2.2 GridFTP

GridFTP [22, 3] is an extension of the standard FTP protocol for grid computing. It designed to supply effective and protected access and transport large amounts of data between multi-distributed resources in the grid. The FTP protocol was selected because it is one of the most widespread data transfer protocols and because it contains a lot of characteristics such as; it is widely implemented, has a clear architecture, transparency and its support for third party transfers. GridFTP offers a lot of advantages such as parallel and partial data and file transfer and enhance Grid Security Infrastructure (GSI).

2.3 Globus Alliance

Globus Alliance [4] is an international association of establishments and privates conducting research for the enhancement of elemental grid technologies. Globus Alliance presents open source software named Globus Toolkit for creating grid environments and applications. The Globus toolkit supports a group of essential services required for grid computing. For example: security, data management, execution managers and information services. These are illustrated in Table 1 and shown in Figure 1.

3. DYNAMIC POLICY MANAGEMENT: NEW FRAMEWORK

The grid infrastructure allows contribution and sharing at the level of a Virtual Resource (VR). The VR can be one device, a group of devices or a virtual partition on the correspondent device. Each grid institute has many VRs that are accepted to participate with other contributors in the VO [21]. Our framework uses well-established concepts from [27] that deal with multiple VOs. In this paper, we propose an extension that provides the features of supporting the external User Policy (UP) along with enforcing policies for data movements within the grid. The following two sections illustrate our framework within a single VO institute and multiple VO institutes [5].

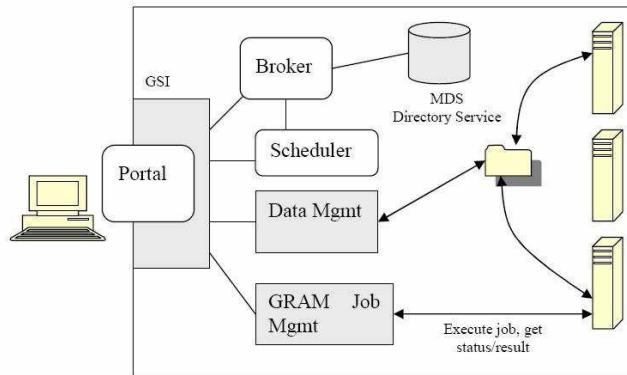


Figure 1. Globus Toolkit Architecture [4]

Table 1. Globus Services

Name	Service	Description
GSI	Security	Remoter Authentication Services
GRAM	Resource Management	High Grid Resource Management
Data Management	Transfer Data	Manage Data using GridFTP
MDS	Information	Grid Meta Directory Service
GEM	Executable Management	Managing Location if Executables

3.1 Single Virtual Organization

Our framework consists of three agents: Policy Agent (PA), Policy Management Agent (PMA) and Grid Information Agent (GIA) Figure 2 shows the framework for a single VO. Each Virtual Organization (VO) should have at least one policy agent that has the ability to access the policy repository. For PAs of the same virtual cluster, there should be a PA leader that coordinates other PAs in the cluster and at the same time performs a homogenous and a heterogeneous policy management across different policy frameworks. This agent is called (PMA). The proprietor of the organization allocates and stores policies at the Policy Server, in other words at the (PMA). The policy server can be considered as a combination of the policy management tool and policy repository. The (GIA) which is owned by the grid administrator is responsible for providing PMAs with the necessary information that is needed to perform the heterogeneous

policy management, if it is necessary, across different policy frameworks. Grid services are a field of web-services and for this reason the policy server would be a web service that publishes the set of services that can provide for an institute into the grid registry. For remote access, the policy server supports a SOAP/HTTP protocol binding in order to swap documents easily over SOAP. The policy documents conforming to the specification of the common information model are encoded in XML [21]. From Figure 2, it can be seen that three main features have been added to the policy management framework mentioned in [27]; First, it enforces the data policies management by using NETGridFTP protocol [9]. Although the GridFTP protocol offers security for grid data movements, there is no clear policy support to apply the resource employment policies that are saved in the policy server. The NETGridFTP, which is an implementation of GridFTP on the Microsoft NET framework, can execute grid data transfers bidirectional between windows machines and can enforce the resource employment policies that are saved in the PMA. The second feature is using the Active Network Middleware to connect the PDPs (PAs) with the PEP. The advantages for both Policy Management and Active network technology are equal. Yang et.al [25] has stated the approach to add programmability to grid management to expand the broadly used grid supporting tool (Globus) in the means of middleware. Together, active network middleware and policy grid management middleware may be used by grid supporting environment to ease the operation so that they can obtain better handling and management of multi-grid resources such as, computing resources, massive storage resources and special scientific instruments. The core of the active network is the active node, which is founded on ABLE. ABLE is an architecture that mainly deals with the network organization challenges. The main element in ABLE is the active engine. This engine is added to any IP router or huge storage or resources to develop an active node [19].

Policy Enforcement Points (PEP) represents the end point where the policy is finally applied. To achieve this application, a transport protocol should be presented for the purpose of communication between Policy Decision Point (PDP) and PEP so that the user can send policy regulations or configuration data to the end point(s), or can read configuration and get information from the device. Active network technology has become the most popular way to achieve policy enforcement [25]. It takes the external User Policy (UP) into the account when making the final conflict decisions.

3.2 Multiple Virtual Organizations

Figure 3 shows our framework for Multiple VO. Allowing PEPs (etc. external users) to obtain policy instructions from subjects outside their physical institute exposes them to security defencelessness. To avoid this issue, each PEP should be remaining only under the administrative control of the policy server (PMA) in its physical institute. Our framework deploys PAs to divide VOs into virtual clusters according to their security policy framework. Nevertheless, the PAs do not accomplish the ideal trust relationship. For example, some VOs do not trust other VOs, or none of the VOs are trusted by all other VOs in the virtual cluster. For that reason, dynamic management framework requires a PMA election procedure in the non-ideal trust relationship. The election method chooses a PA with the highest number of followers to be the PMA. The PMA initializes connections to PAs corresponding to the trust relationships of the PAs. As a result, some PAs do not trust the elected PMA where the security policy information for these PAs will not be available to access by the PMA. In this case, PMA must apply conflict analysis with partial information [26]. To reduce bottleneck in the PMA, a PMA can request the leader PA to execute conflict analysis. Leader PAs are selected by the relative high levels between PAs and are virtually connected to the PMA directly [28].

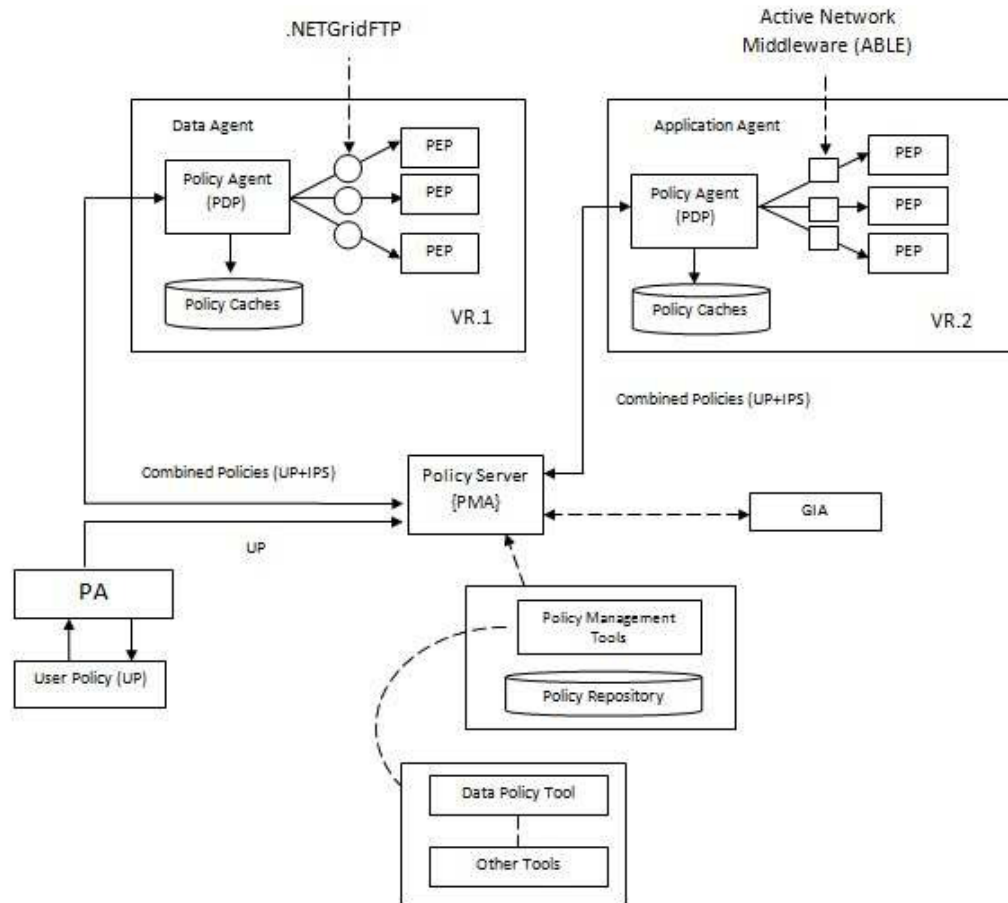


Figure 2. Single Virtual Organization Policy Management Framework

After the authorized grid users submit their jobs to the core of the grid system (resource broker), it asks the Grid Information Services (GIS) and Replica Catalogue about the free resources in the grid. Later, it sends this information along with the related policies (Users policies) to the Grid Policy Server or to the VO policy server to make the policy decisions. The VO policy server forwards the users' policies along with the VO policy to the PMA which is responsible for the target resource to make the final decisions. PMA checks if the requests are situated in a single virtual cluster or multiple virtual cluster. If the requests are situated in the same virtual cluster, a homogenous conflict analysis takes place without any need to retrieve any information from the GIA. Where PMA receives a request from one of the PAs asking for a service that is situated in multiple virtual clusters, a heterogeneous policy management mechanism takes place.

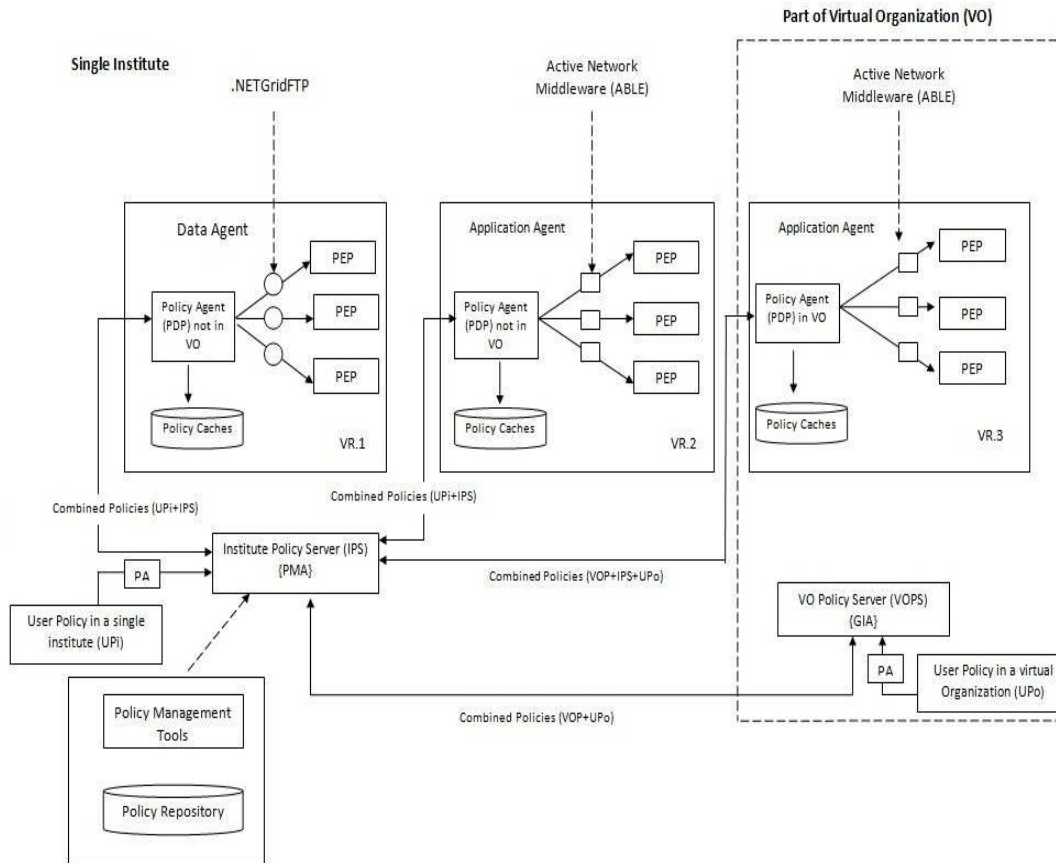


Figure 3. Multiple Virtual Organization Policy Management Framework

The heterogeneous conflict mechanism can take place in any of the following cases (or both). Chiu-Man Yu and Kam-Wing in [26] state the first case once the PMA receives a request from one of the PAs that has a framework different from that in the PMA. The second case when the User Policy framework is different from that in the PMA. This conflict analysis can be either with full or partial information. In [26] the authors have specified the conflict analysis mechanism which can be applied for both previous cases. This mechanism depends on converting the policies of the target service into the policy model of the PMA. The PMA asks the GIA for the account maps and the policy scheme maps of the target services. At the same time, the PMA asks the service requester for its authorization policies through the PA of the service. When the PMA receives this information, it starts the policy conversion to the PMA policy model. First it applies the Account Mapping to make it possible for users, whether trusted or not, to access services in a remote VO. Here, a map mechanism should be applied to map those users to local accounts and later to perform the policy mapping to generate an inter-scheme map which maps the scheme of the policy model of the service's VO to that of the requester's VO. In other words, a map mechanism is used to convert policies between heterogeneous policy models to a one that can be understood by the PMA. Finally, the PMA applies the conflict analysis mechanism on the policies of all target services to find suitable permissions for the requester of service.

4. ARCHITECTURE STRUCTURE AND COMPONENTS

Grids depend on enhanced software that guarantees seamless communication between components nodes. It uses an effective mechanism which determines the suitable policy(s) that should be applied to achieve the best way to utilize resources in a way that guarantee privacy and security for both grid users and grid resources. Figure 4 shows our proposed architecture [6]. It applies Client/Server architecture since this architecture is the most favorable type in heterogeneous environments [8]. Client/Server network includes clients and servers who operate on the proper hardware and software for their jobs. There are two forms of client/server architecture; two and three-tier (multi-tier). Our architecture employs the last model which comprises of the client (grid portal) as the first tier, the resource broker as second tier and grid nodes as third tier. The following describes the functions for each one of them.

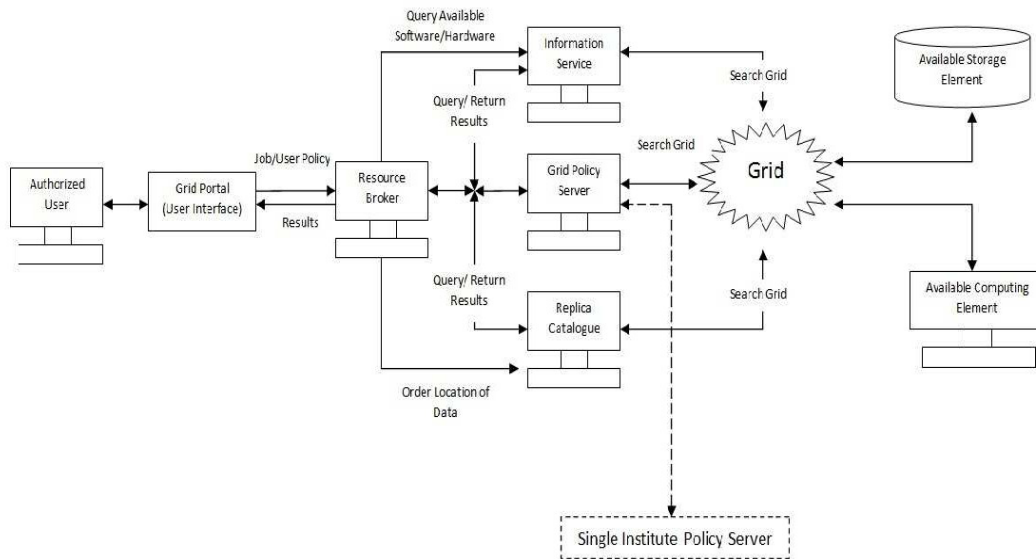


Figure 4. Grid Architecture

5. GRID PORTAL

A grid portal or grid interface is a virtual computing resource performing an interface on behalf of grid users to approach the grid. A portal has many features such as hiding the complexity of the grid from users via a simple interface which facilitates the classification of grid job necessities. Grid portals support identical services to users. For example, web browsers offer a single interface which can be used to reach internet resources, while grid portal is used to illustrate and send job/applications to be accomplished by grid resources.

6. RESOURCE BROKER

The Resource Broker is one of the major grid elements, it performs significant functions in building a valuable grid environment by arranging user jobs onto grid resources to reach particular accomplishment targets, like cutting communication delays, raising the resource exploitation, reliability and distributing jobs across resources without depending on a particular resource. The main job for the broker is to discover and choose suitable resources for jobs by sending jobs input files to the resources, monitoring jobs and sending outputs to users. The resource broker presented in this paper is based on the mobility framework and isolates the user from the grid's middleware. All of this helps in automating the operation from the point of

receiving user's resource during the job execution on the suitable resources till the submission of results. In other words, the resource broker is connected with all grid elements.

6.1 Resource Broker Architecture

The resource broker accepts job requirements from the portal and looks for appropriate resources that can fit these requirements. First it asks for all information about the available resources from the information service and the data information stored in the replica catalogue. Then it chooses the resources that can fit the job requirements and asks the grid policy server about policies for those resources. According to that, the resource broker's architecture comprises of three components indices: information service, the replica catalogue and the grid policy server.

6.2 Information service

Information service is a crucial element in grid computing. It is a directory service holding data about all the grid resources and the entire grid activated jobs operating on those resources. This information can be either dynamic or static information. The last one is for the hardware conditions and the operating system, while dynamic information related to the resources available time, the job presently running, type of application software, disk space and policies. In order to advertise their information the resource broker communicates to both resources and the information service to ask for this information.

6.3 Replica Catalogue

This is also an important component for the grid, because it presents information which helps in accessing the stored data in the grid. It determines the places of data in the grid, updates data resources and maps logical file names to the actual physical places on grid resources. In order to use the data on the grid the resource broker communicates with a replica catalogue to ask for information about data place and the access control needed to utilize this data.

6.4 Grid Policy Agent

The grid policy agent contains all the policies information about all resources in the grid. Each institute should have at least one policy agent that has the ability to access the policy repository or policy information for that institute. All policy agents (PA) in all domains in the grid should be registered with the grid policy agent and should send their policy information (e.g. policy framework) or any changes or updated data about their policies to the grid policy agent. Grid services are an area of web-services; for this reason the policy agent would be a web-service that issues the group of services that it can support for an institute into the grid registry. For remote access, the policy agent supports a SOAP/HTTP protocol to exchange documents easily over SOAP, the policy documents meeting the requirements of the specification of the common information model are encoded in XML [21]. Grid administrator can specify the policies for units participated in the grid, but it does not have any policy agents that can directly use it. As an alternative, a grid policy agent operates as a proxy for the policy agents that run at each of the different institutes. Figure 5 shows the architecture of the single institute policy agent. Each grid institute has many Virtual Resources (VRs) that are accepted to participate with other contributors in the grid. The grid administrator defines policies using various management tools in the policy agent and stores those policies in the policy repository. As a result, the institute policy agent can be considered as a combination of the policy management tools and policy repository [21].

The main job now for the institute policy agent is to combine the policies from the institute administrator, the policies from the global grid and grid user's policies in order to obtain the

efficient set of policies for resources belonging to that institute. The efficient set of policies is the ones applied by the policy agents attached to each resource assigned to that institute in the grid.

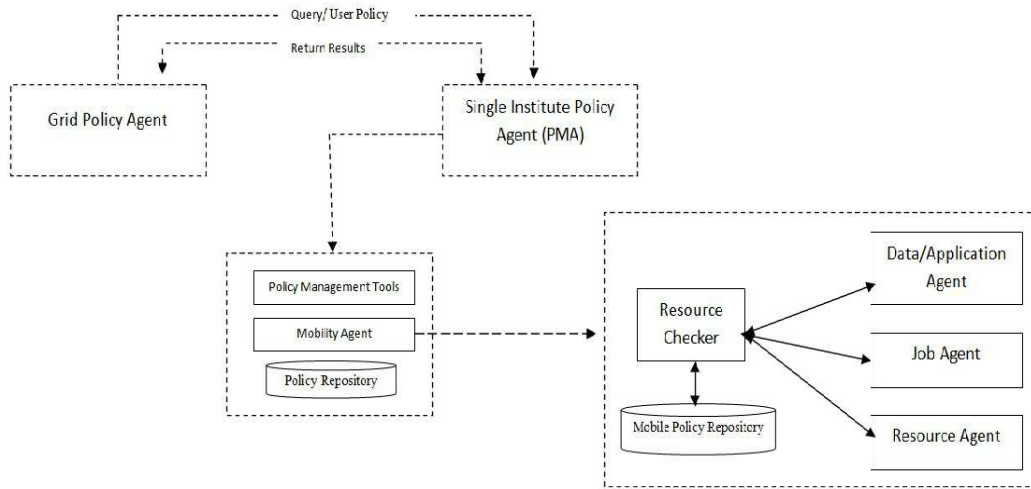


Figure 5. Mobile Agent Architecture

In many cases resource broker find itself in a situation that has to reject some jobs because the required resources may not be found. These are some of these situations, in short:

- The resource needed to fit the job requirements is busy at this time.
- Resource that fits the job hardware requirement that does not own the needed application software.
- The resource that fits the job hardware and application software requirement and does not have the required data. As a result the Mobility has created a new environment that can solve these cases. In our model mobile policy server plays a significant role to achieve these requirements. Figure 5 shows the architecture of our mobile policy server and its components. The following describes each one of them.

6.4.1 Data/Application Software Agent

This agent is responsible for the data and application software migration. Our grid architecture allows application software and/or data to migrate from one node to another in the grid environment so as to adapt the resources needed to fit the job requirements. If the resource that fits the job hardware conditions and the availability time does not have the application software or data needed for the job(s), the resource broker will search the grid for the nodes that have this data/application software by inquiring the information service and replica catalogue and put these nodes in a new list. Then it will check each one of these nodes, one by one, by asking the mobile policy agent to determine whether or not the data/application's software policy in these nodes allows their movements (or having a copy from this data or application software). The Data/Application Software Agent will check the policies for the nodes that contain the required data or application software and return the results to the resource broker. If one of the nodes does support the mobility feature for data/application software, the resource broker will migrate or copy appropriately and send it to the resource that meets the job hardware and time requirements along with its policy. If all the nodes' policies do not support the data/application software mobility, the broker will tell the user that the grid cannot execute the job.

6.4.2 Job Agent

This agent is responsible for checking the grid users' policies. Our grid architecture allows the job and its execution state (i.e. the context of execution) to migrate from one resource to another and restart on the new one, in order to fit the job conditions and requirements. If the resource that fits the job hardware requirements is busy at the time needed, our resource broker will vacate this resource by migrating the currently running job in that resource to other resources, (if they are presented and have the job requirements). This can be done by looking for jobs running on this required resource and obtain details of their requirements from the information service and the replica catalogue. If the job requirements can be satisfied using other resources, the resource broker will ask the mobile policy agent if the currently running job(s) is allowed to be migrated to another resources. The Job Agent and the Resource Agent in the mobile policy agent will check whether or not the grid user's policy and the new resource(s) policies allow migrating the running job to the new resources and returning the results to the resource broker. If the policy allows this kind of migration, then the resource broker will migrate these jobs to the new resource(s) and send the new job to the vacated resource which will fulfil its requirements.

6.4.3 Resource Agent

This agent checks whether or not the resources' policies allow the migration for jobs, data and application software between various resources. Our grid architecture allows jobs, data and application software to migrate from one node to another in the grid environment in order to adapt the resources required to meet the job requirements. In the case the resource that meets the job requirements is currently busy and there is a need to migrate the currently running job(s) to other resource that can meet the running job requirements, or there is a need for a data or application software migration. In both cases, the resource broker will ask the mobile policy agent to check the policy aspect in these situations. The Resource Agent in the mobile policy server will determine whether or not the current resource's policy allows the job migration from its node to the destination resource, or if the destination resource can accept jobs from the original resource. In both cases, it will inform the resource broker about the results. In the case of data/software migration the resource agent in the mobile policy agent will determine if the addition or migrating of data/application software policies are allowed in the current resource and the destination resources. If they do not, the broker will tell the user that the grid cannot execute the job. If they do, the broker will apply the migration between those resources.

6.4.4 Resource Checker

As soon as the mobile policy server makes its decisions about any possible migration(s) either for jobs, data or application's software, it stores indexes for these decision using the resource checker and stores these indexes in the policy repository prior to submitting the decision's results to the resource broker. The aim of these indexes is to track any changes or updates in the target policy(s) and inform the resource broker about them. This helps in enhancing the mobile policy server performance and throughputs by returning to these indexes for any new requests from the resource broker instead of going for the whole checking operation again. After the authorized grid users submit their jobs to the core of the grid system (resource broker), it asks the Grid Information Services (GIS) and Replica Catalogue about the free resources in the grid. Later, it sends this information along with the related policies (Users policies) to the Grid Policy Server which forward it to the Single Institute Policy Server to make the final Policy decisions, then it sends the results back again to Grid Policy Server. The Grid Policy Server sends the results to the resource broker to enforce the policy results in its decisions [16].

6.5 Mobile Policy Server Example

The following scenario explains the advantage of mobility mechanism and the role of policy in it within grid systems. It is divided into three sections.

– First Section: Grid Resources Specifications

The grid contains five nodes; each node has different conditions and specifications. These specifications are: hardware, domain, application software, data and policies. Each node is responsible for defining its own policy. Also it contains the running jobs, if presented, as shown in Tables 2, 3 and 4.

– Second Section: Jobs Requirements

There are five jobs which need to be executed by the grid resources. The requirements needed to accomplish the jobs include hardware, software, input, output, domain and policies, as shown in Tables 5, 6 and 7. Grid’s users are responsible for defining their policies when submitting their jobs to the grid.

– Third Section: Fits the Jobs Requirement to Grid Resources

The resource broker is responsible for locating the optimal resource that can meet the job requirements and scheduling the jobs into grid resources with respect to the policies. All of these issues will be illustrated in the following. It is also shown in Figures 6 and 7.

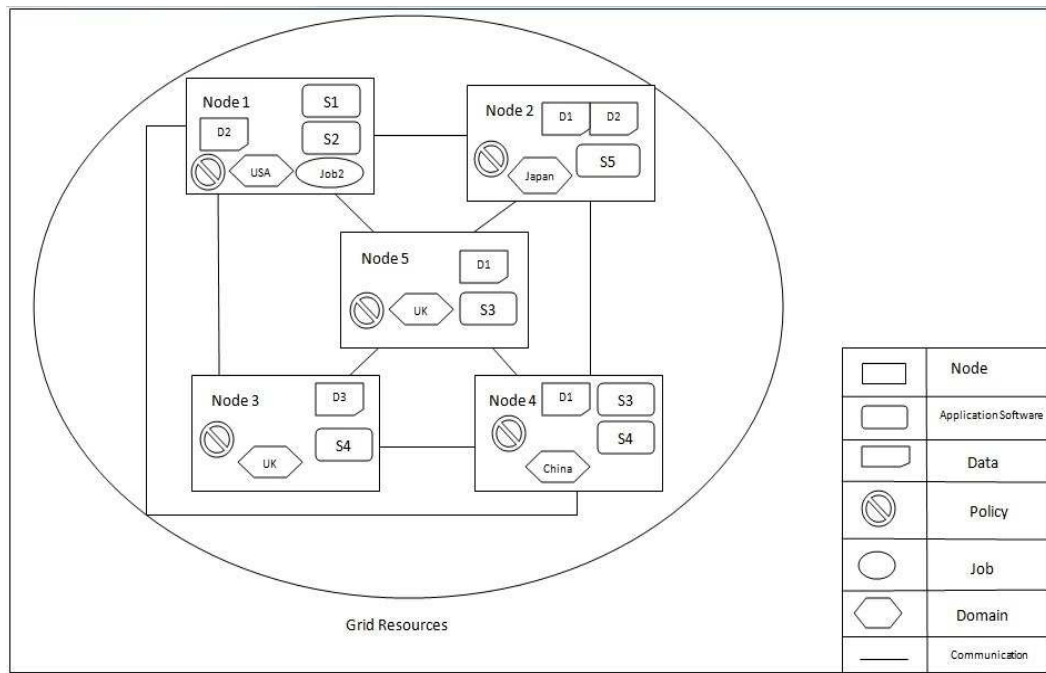


Figure 6. Grid Resources (Infrastructure)

Table 2. Grid Nodes Hardware Specifications.

Node Name	Hardware				Domain
	CPU		Memory		
	Speed	Count	RAM	Shared or Distributed	
Node 1	1	1	1024	-	USA
Node 2	1	1	2048	-	Japan
Node 3	2	2	2048	D	UK
Node 4	1	1	2048	-	China
Node 5	1	1	2048	S	UK

Table 3. Grid Nodes Application/Data Specifications.

Node Name	Application				Data	
	File	Version	Requirement		File Name	Size
			CPU Speed	Disk Space		
Node 1	S1	9.2	0.5	200	D2	1000
	S2	1.0	0.5	700		
Node 2	S5	2	1.0	300	D1/D2	900
Node 3	S4	1.0	1.0	500	D3	700
Node 4	S3	1.0	1.0	900	D1	700
	S4	5.0	1.0	250		
Node 5	S3	1.0	1.0	500	D1	700

Table 4. Grid Nodes Policy Specifications and Running Jobs

Node Name	Policy					Jobs Running
	Exclusive Execution	Move Data	Move Application	Move Job	Restricted (Domain/ User/ Job)	
N1	Yes	Yes	Yes	Yes	China/U4	Job2
N2	No	No	Yes	Yes	China	-
N3	Yes	Yes	No	Yes	None	-
N4	Yes	Yes	Yes	Yes	None	-
N5	Yes	Yes	Yes	Yes	None	-

Table 5. Node Specification Requirements.

User Name	Job Name	Node Specification			
		CPU		Memory	
		Speed	Count	RAM	S/D
U1	Job 1	1	1	1024	-
U2	Job 2	1	1	2048	-
U3	Job 3	2	2	2048	-
U4	Job 4	1	1	2048	-
U5	Job 5	2	2	2048	-

– Job Migration

From Tables 5, 6 and 7 it can be noticed that Job1 requirements fit the Node1 specification in Tables 2, 3, 4, but Node1's policy, Table 4, is to allow only a single job to run at any time (exclusive execution), so there is a need to migrate the existing job (Job 2) on Node1 to another node that fits Job2 requirements. The resource broker looks for this substitute node and finds Node4 and Node5; but Node4 domain is in China which is against the policy of Job2 and Node1 policy. Therefore, the resource broker sends Job1 to Node1 and move Job2 together with its status (memory image) to Node5 for execution.

– Data Migration (case 1)

As shown in Tables 2, 3, 4, Job3's requirements fit Node3's specifications in Tables 5, 6 and 7, but Node3 does not contain data (D2); this data is available in Node1 and Node2, Node1 policy is to allow movement of this data as well as Node3's data requirements, while Node2 is not. The resource broker will therefore send a message to Node3 telling it to take data (D2) along with its policy from Node1 and execute Job3.

– Data Migration (case 2)

As shown in Tables 2, 3, 4, Job4's requirements fit Node4's specifications in Tables 5, 6 and 7, but Node4 does not contain data (D2); this data is available in Node2 and Node3 (after migration). Node2 policy is not to allow movement of data to China domain, but the policy in Node3 allows this kind of movements, but the data in Node3 was moved originally from Node1 which its policy does not allow to move data to China domain. Therefore, the resource broker will send a message to User4 which says that the grid is unable to execute Job4, because the needed data is unavailable.

– Application Software Migration

In the previous Tables, it can be seen that Job5's requirements fit Node3's specifications. But Node3 does not have application software (S5). Node2 does, however, and its policy is to allow this application software as well as node3's application software requirements. The resource broker will therefore send Job5 with a message to Node3 telling it to take application software (S5) from Node2 and execute Job5.

Table 6. Application Software Requirements.

User Name	Job Name	Application Software		Data
		Name	Version	
U1	Job 1	S2	1	-
U2	Job 2	S3	1	D1
U3	Job 3	S4	1	D2
U4	Job 4	S4	5	D2
U5	Job 5	S5	1.1	D3

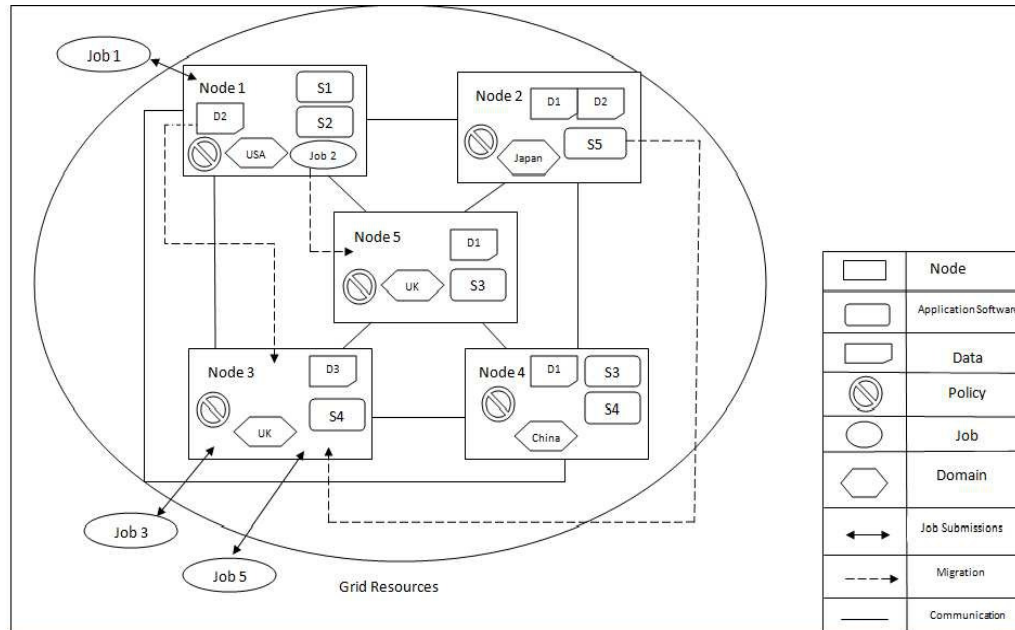


Figure 7. Grid Resources after Mobility

Table 7. Job Domain/Policy.

User Name	Job Name	Policy			Domain
		Exclusive Execution	Move Job	Restricted (Domain/ User)	
U1	Job 1	No	Yes	Non	China
U2	Job 2	No	Yes	China	UK
U3	Job 3	No	No	None	USA
U4	Job 4	No	No	None	USA
U5	Job 5	No	No	None	UK

7. CONCLUSIONS AND FUTURE WORK

We have presented in this paper a new dynamic policy management framework that has the capability to deal with policies of multiple virtual organizations and which at the same time has a feature of enforcing policies for data movement within the grid. We have utilized NETGridFTP protocol to enforce data policies management in the grid environment and have also operated an active network middleware in the grid environment to ease the operation of handling and managing multi-grid resources. The other framework that we presented in this paper is the dynamic policy management framework for mobile grid services that has the capability to deal with policies of multiple virtual organizations. The other advantage of this architecture is taking the policies of the external users of the grid into account when making policy decisions. Based on our contributions, we are confident that the mobile policy feature which supports enforcing policies for mobile applications, i.e. jobs and data, within the grid can be addressed efficiently in the future.

REFERENCES

- [1] A. Anjomshoaa, F. B. and Michel Drescher, D.F.a.L.S.M.D.P., Savva, A.: Job submission description language (jsdl) specification, version 1.0 (November 2005), <http://www.gridforum.org/documents/GFD.56.pdf>
- [2] Aldabbas, O.: A framework for mobility and temporal dimensions of grid system. Ph.D.thesis, School of Computing, De Montfort University, UK (2008).
- [3] Allcock, W., Bresnahan, J., Kettimuthu, R., Link, M., Dumitrescu, C., Raicu, I., Foster, I.: The globus striped gridftp framework and server. In: Proceedings of the 2005 ACM/IEEE conference on Supercomputing. pp. 54–. SC '05, IEEE Computer Society, Washington, DC, USA (2005), <http://dx.doi.org/10.1109/SC.2005.72>
- [4] Alliance, G.: Globus toolkits, <http://www.globus.org>
- [5] Alwadan, T., Janicke, H., Aldabbas, O., Aldabbas, H.: New framework for dynamic policy management in grid environments. In: Recent Trends in Wireless and Mobile Networks, Third International Conferences, CoNeCo 2011. vol. 162, pp. 297–304. Springer (2011)
- [6] Alwada'n, T., Janicke, H., Aldabbas, O., Alfawair, M.: New framework for policy support for mobile grid services. In: Cuppens, F., Foley, S., Groza, B., Minea, M. (eds.) CRI-SIS. pp. 88–93. IEEE (2011), <http://dblp.uni-trier.de/db/conf/crisis/crisis2011.html#AlwadanJAA11>.
- [7] Carpenter, B.E., Janson, P.A.: Abstract interdomain security assertions: a basis for extra-grid virtual organizations. *IBM Syst. J.* 43, 689–701 (October 2004), <http://dx.doi.org/10.1147/sj.434.0689>.
- [8] Coulouris, Dollimore, J., Kindberg, T.: *Distributed Systems: Concepts and Design* (4th Edition) (International Computer Science). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2005).
- [9] Feng, J., Cui, L., Wasson, G., Humphrey, M.: Policy-directed data movement in grids. In: Proceedings of the 12th International Conference on Parallel and Distributed Systems – Volume 1. pp. 319–326. ICPADS '06, IEEE Computer Society, Washington, DC, USA (2006), <http://dx.doi.org/10.1109/ICPADS.2006.82>
- [10] Ferreira, L., Berstis, V., Armstrong, J., Kendzierski, M., Neukoetter, A., MasanobuTakagi, Bing, R., Amir, A., Murakawa, R., Hernandez, O., Magowan, J., Bieberstein, N.: Introduction to grid computing with globus. IBM Corp., Riverton, NJ, USA, first edn. (2003).
- [11] Forum, O.G.: open grid forum, <http://www.gridforum.org/>
- [12] Foster, I., Kesselman, K.: The grid: Blueprint for a future computing infrastructure. In: Morgan Kaufmann in Computer Architecture and Design (1999).
- [13] Foster, I., Kesselman, C., Tsudik, G., Tuecke, S.: A security architecture for computational grids. In: Proceedings of the 5th ACM conference on Computer and communications security. pp. 83–92. CCS '98, ACM, New York, NY, USA (1998), <http://doi.acm.org/10.1145/288090.288111>.
- [14] Grimshaw, A.S., Humphrey, M.A., Natrajan, A.: A philosophical and technical comparison of legion and globus. *IBM J. Res. Dev.* 48, 233–254 (March 2004), <http://dx.doi.org/10.1147/rd.482.0233>
- [15] Humphrey, M., Thompson, M.R.: Security implications of typical grid computing usage scenarios. *Cluster Computing* 5, 257–264 (July 2002), <http://portal.acm.org/citation.cfm?id=592899.593008>
- [16] Jameel, H., Kalim, U., Sajjad, A., Lee, S., Jeon, T.: Mobile-to-grid middleware: Bridging the gap between mobile and grid environments. In: EGC'05. pp. 932–941 (2005)
- [17] Joshi, J.: Access-control language for multidomain environments. *Internet Computing, IEEE* 8(6), 40-50 (2004).

- [18] Kaneda, K., Taura, K., Yonezawa, A.: Virtual private grid: a command shell for utilizing hundreds of machines efficiently. *Future Gener. Comput. Syst.* 19, 563–573 (May 2003), [http://dx.doi.org/10.1016/S0167-739X\(03\)00036-0](http://dx.doi.org/10.1016/S0167-739X(03)00036-0)
- [19] Kornblum, J.A., Raz, D., Shavitt, Y.: The active process interaction with its environment. *Comput. Netw.* 36, 21–34 (June 2001), <http://portal.acm.org/citation.cfm?id=376733.376735>
- [20] Lorch, M., Adams, D.B., Kafura, D., Koneni, M.S.R., Rathi, A., Shah, S.: The prima system for privilege management, authorization and enforcement in grid environments. In: *Proceedings of the 4th International Workshop on Grid Computing*. pp. 109–. GRID '03, IEEE Computer Society, Washington, DC, USA (2003), <http://portal.acm.org/citation.cfm?id=951948.952044>
- [21] Verma, D.C., Sahu, S., Calo, S.B., Beigi, M., Chang, I.: A policy service for grid computing. In: *Proceedings of the Third International Workshop on Grid Computing*. pp. 243–255. GRID '02, Springer-Verlag, London, UK (2002), <http://portal.acm.org/citation.cfm?id=645442.652675>
- [22] W. Allcock, J. Bester, J.B.A.C.L.L., Tuecke, S.: Gridftp: Protocol extensions to ftp for the grid (April 2003), <http://www.ggf.org/documents/GWD-R/GFD-R.020.pdf>.
- [23] Welch, V., Siebenlist, F., Foster, I., Bresnahan, J., Czajkowski, K., Gawor, J., Kesselman, C., Meder, S., Pearlman, L., Tuecke, S.: Security for grid services. In: *High Performance Distributed Computing, 2003. Proceedings. 12th IEEE International Symposium on*. pp. 48– 57 (2003)
- [24] Wu, Jin, Leangsuksun, Chokchai Box, R.V., Hong, O.: Policy-based access control framework for grid computing. In: *Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid*. pp. 391–394. CCGRID '06, IEEE Computer Society, Washington, DC, USA (2006), <http://dx.doi.org/10.1109/CCGRID.2006.80>
- [25] Yang, K., Galis, A., Todd, C.: Policy-based active grid management architecture. In: *Networks, 2002. ICON 2002. 10th IEEE International Conference on*. pp. 243 – 248 (2002).
- [26] Yu, C.M., Kam-Wing: Dynamic policy management framework for partial policy information. In: *Advances in Grid Computing - EGC 2005 European Grid Conference - Volume 1 / 1973 - Volume 6473 / 2011*. Springer Berlin / Heidelberg, Amsterdam, The Netherlands (2005).
- [27] Yu, C.M., Ng, K.W.: A heterogeneous authorization policy management mechanism for grid environments. In: *Multimedia and Ubiquitous Engineering, 2007. MUE '07. International Conference on*. pp. 381 –386 (2007).
- [28] Yu, C.M., Ng, K.W.: A dynamic management framework for security policies in open grid. In: *Proceeding of the Third International Conference of Grid and Cooperative Computing (GCC 2004)*. pp. 871–874. China (October,2004).