In: Cyberinfrastructure Technologies and Applications ISBN
Editor: Junwei Cao

Chapter 11

FEDERAL MANAGEMENT OF VIRTUAL ORGANIZATIONS WITH TRUST EVALUATION

Zhen Wang[1]
Junwei Cao[2]
Research Institute of Information Technology
Tsinghua National Laboratory for Information Science and Technology
Tsinghua University, Beijing 100084, P. R. China

Abstract

Dynamical and flexible resource aggregation tools are required in 21st century research. Scientists need to aggregate various digital equipments and cooperate with each other in different organizations through Virtual Organizations (VO) on the Internet in a flexible and dynamical way. In this cooperation and resource sharing process, trust evaluation is of great importance for flexible VO management. Traditional tools such as VOMS for grids are short in dynamism and trust evaluation. In this chapter, we propose a new scheme providing federal VO membership management based on trust evaluation, with which researchers can achieve appropriate trust relationships with each other and establish a particular VO dynamically to aggregate resources for their own purposes.

1. Introduction

1.1. Background

Modern science research has great requirement for experimental instruments, computational and storage capability, and cooperation across organizations and discinplines

[1] E-mail address: zhen-wang07@mails.tsinghua.edu.cn
[2] E-mail address: jcao@tsinghua.edu.cn

[1]. However, grid technology, which is considered as a traditional solution enabling resource integration and sharing within a virtual organization (VO), can not meet current requirements for multiple VO management. Scientists and researchers require a more general and flexible digital platform for data analysis, information integration, instruments sharing and so on.

In 2003, CI [1], short for Cyberinfrastructure, was proposed as a new infrastructure in Cyberspace in future by National Science Foundation in USA. In the year of 2006, Office of Cyberinfrastructure [2] was founded for CI implementation and a detailed plan [3] was established. Compared with grid or other similar technologies, CI is a more general and flexible platform, with which everyone can contribute their resources or obtain sufficient distributed resources to meet their own requiurments. In a CI environment, resource providers contribute their resources (Cyberresources) and users benefit from these resources via a Cyberenvironment. A member of CI may be a *RP* and a *User* at same time, depending on his requirement. CI knocks down the barriers between different grids VOs and makes it possible to share resources and cooperate across them.

The implementation of CI will be a large distributed system and brings a lot of challeages. The problem we are trying to address in this chapter is how to provide a mechanism to support trustable cooperation and resource sharing dynamically and flexibly, as we called Trustable Federal VO Management (TFVOM). TFVOM help Users or RPs to realize effective and trustable resource sharing and access control. There are already some traditional mechanisms which achieve similar functions: Mandatory Access Control (MAC) [4][5], Role-Based Access Control (RBAC) [6][7], Discretionary Access Control (DAC) [8], Virtual Organization Membership Service (VOMS) [9][10][11], Grid User Management System (GUMS) [12], PRIvilege Management and Authorization (PRIMA) [13], Privilege and Role Management Infrastructure Standards Validation (PERMIS) [14] and so on. DAC is implemented by maintaining an account list. In CI environment, the number of CI members is huge and they also change dynamically. On another hand, Users or RPs require flexible resource privilege management, which cannot be implemented using DAC. RBAC and MAC cannot be adopted either because of the different jurisdiction distribution. These two mechanisms are more suitable for centralized organizations with fixed architecture, where there is an account with the highest privilege to all the resources. However, in the CI environment, all the resources are owned by RPs who have the highest privilege over their own resources. This means CI environment is an incompact system and the privilege locates on terminals. Other mechanisms, including VOMS, GUMS, PRIMA and PERMIS, are all used in grids without any trust management mechanism since Users have already built trust relationships to some extent before a grid is enalbed. How to make cross-domain users and resource providers achieve appropriate trust relationships is the main purpose of TFVOM.

In this chapter, we will introduce TFVOM mechanism in details with corresponding implementation. How to deploy the TFVOM mechanism in the CI environment is also described.

1.2. Challenges

Compared with other resource aggregation environments, such as grids, a CI environment is more dynamic, complicated, and open. This is why CI is regarded as the future advanced infrastructure for 21st scientific discovery, but this also imposes significant challenges.

The openness of CI leads to the complexity and wideness of origins of Users and RPs in a CI environment, which is quite different from that of the grid. Before a grid is established, Users or RPs have already achieved an agreement about the purpose of the grid and how to contribute and share the resources. This agreement or protocol is established by non-technological manner. A grid usually has more fixed organization architecture and members of corresponding VO are within a specific domain. A grid is used to enable resource sharing between RPs and Users that already form a VO beforehand. Members of a grid believe that all other members are trustable and resources are shared following the existing agreement. Members of a CI environment do not have any agreement with each other on sharing their resources before they join in the CI. The CI provides an environment to enable members to build such agreement and thus fosters VOs and grids. In this process, RPs want to make sure that they have full control on their resources, and meanwhile Users also have requirements to ensure quality of services (QoS) when using these resources. TFVOM is designed to provide RPs and Users with a negotiation mechanism to achieve agreement on resource sharing with trust evaluation supports. This process should be implemented using advanced computing technologies that can adapte to various situations.

Compared with centralized organizations, the privilege of resources in a CI environment is distributed to each RPs, as mentioned before. No matter what has happened, RPs, resource owners, always have full control of their resources. Each RP has specific and various policies on how to share his resource, which makes traditional aggregation and access control mechanisms not feasible in the CI environment, since most of them are suitable for centralized organizations. To implement a CI environment, a new mechanism should be proposed that can ensure that RPs has the highest privilege on their resources.

Another challenge sources from the variety of resources. Any resource that can be connected through Cyberspace can join in the CI resource pool. Cyberresources include hardware facilities, e.g. a computer, a sensor or an astronomical observatory, driven by different middlewares running on different operating systems. In general, a VO is founded usually for some specific research purpose, used for a certain discipline and enabled using grid technologies, e.g. Network for Earthquake Engineering Simulation (NEES) [16], National Ecological Observatory Network (NEON) [18], The Geosciences Network (GEON)[19], National Center for Atmospheric Research (NCAR) [20], US National Virtual Observatory (NVO) [21] and TeraGrid [15]. CI provides an environment to operate on multiple VOs or grids, e.g. Open Science Grid (OSG) [17], where cross-VO resource sharing becomes available.

In the CI environment, researchers and scientists can easily form a VO, aggregate sufficient and appropriate resources, and collaborate together to work for a specific project. After the project is finished, these resources are released again and can be used for other VOs. Existing VOs should also be able to share resources with each other. From this point of view, a CI could be a platform with many grids. The access control mechanism must be robust enough to handle various situations. Moreover, this mechanism should not be centralized. The primary challenges faced by the new access control mechanism can be summarized as follows:

- Trust evaluation mechanism: With the help of certificates, members can know identity of each other. However, members still can not trust each other based on pure authentication. We must provide a trust evaluate mechanism to help them establish appropriate trust relationships.

- Dynamic and flexible VO membership management: requirements of Users and RPs are various and dynamically changing over time, so we must provide appropriate tools help them change VO membership easily and dynamically when their purpose and requirements change.
- Communication with other middlewares seamlessly. The new access control implementation must be deployed in different environment and communicate with other CI components seamlessly.
- Reliability: as this mechanism is one of essential tools in a CI environment, we must ensure high reliability since it will influence all Users and RPs of a CI.

To address to these challenges, we propose TFVOM as the solution for the access control mechanism in CI. The details will be introduced in the following. In Section 2, we will introduce some related and similar technologies used in grids or other applications currently. In Section 3, TFVOM mechanism is introduced in details. We will demonstrate the implementation architecture and several typcial applications in Section 4. Then we will evaluate this mechanism, make conclusion and indicate future work in Section 5.

2. Related Work

In this section, we will introduce several mechanisms widely being used in the current systems. These implementations are developed for some specific purposes. For example, VOMS is developed by European DataGrid (EDG) [22] and Data TransAtlantic Grid (DataTAG) [23] to knock down the barrier between the two grids originally, and be accepted and used by other grids. Privilege and Role Management Infrastructure Standards Validation (PERMIS), supporting authentication of the personal idnode and determination of the role, status, entitlements, or other socio-economic attributes, is developed by ISIS, Institute for Science and International Security. These different schemes all support access control but focus on different aspects. In this section, we will introduce these schemes to show current technology landscape.

2.1. VOMS

VOMS, short for the Virtual Organization Membership Service, is one of the most famous and widely used implementation about authorization and authentication of access privilege over resources to the members in grids. The project is supported and developed by European DataGrid (EDG) and Data TransAtlantic Grid (DataTAG) and used by many other grids [24] such as Laser Interferometer Gravitational-wave Observatory (LIGO) [25], Structural Biology Grid (SBGrid) [26], Open Science Grid (OSG), Georgetown University Grid (GUGrid) [27] and so on.

VOMS is developed for the purpose of authorization and authentication on the organization level. VOMS maintains a database to manage and store the information of user roles and capabilities and provides user a set of tools for accessing and manipulating the database. Then VOMS can generate Grid credentials for users through the database contents when needed. The VO is established by the administrator, who is in charge of managing the VO, e.g. adding new member, creating new group, changing roles. Every member in VO is

assigned with specific role, with which the member has the privilege corresponding to the role assigned in VO. The role assignment is described in the certificate in local sites and stored and managed by the administrator. Access control over the resource is achieved by the role definition and assignment. In the VO, there are two important facts: administrator and user. Administrator is the one who has the responsibility to manage and maintain the VO and is in charge of role assignment and maintaining the membership information, while the User is a part of the VO and can request information onVO memberships when needed.

VOMS consists of four modules, User Server, User Client, Administration Server, Administration Client, which have different functions, respectively:

- User Client: contact to the Server with certificate and obtain the information of membership in the VO after confirmation, e.g. member lists, role assignments, sub-groups, and capability of a User.
- User Server: receive the request from the User Client, and return the results for user requests.
- Administrator Client: this client is used by the administrator who is in charge of management tasks, e.g. adding new user, creating sub group, role assignment.
- Administrator Server: this server is mainly a data server, which is used to maintain the database and response to the request for the membership information from the client.

VOMS adopts the GSI security control mechanism provided by Globus Toolkit package. User can use the command “voms-proxy-init” to get the certificate generated in the VOMS server. This certificate adopts RFC 3281[29] format and signed by the VOMS server. In order to make sure user can be a member of several VO and may have communication with other non-VOMS GateKeepers, this certificate is extendable and can be an aggregation of several certificates. The VOMS combines two different mechanisms: RBAC and VO. The policy on how the User uses the resources is defined by two aspects: which VO the User belongs to and which role he plays.

2.2. GUMS

The Grid User Management System (GUMS) is a system running in the local site in the grid. The major function of GUMS is to manage the mapping process from User’s grid certificate or credential to the local site-specific certificate or credential. In the grid, User shares the distributed resources through mapping User grid account to the RP local account, which is similar with remote access to the resources using the account RP assigned to the User. One RP may belong to many VOs or grids, so it is a big challenge how to map the grid certificate to the local account according to the access policy. The accounts provided for the remote access may be different according to different Users jobs. For example, a RP who owns a computer joins in two different VOs. He may provide two different kinds of accounts with different privileges locally for the two VOs. The accounts may even be different because of the job recieved.

GUMS can be configured to map the grid certificate to a local account in two manners: 1) generate statistic map-files according to the Users 2) or map the grid certificate to the local account dynamically according to the job submitted. For example, a user wants to submit some jobs to a certain resource. When the job arrives at the resource with the grid certificate

and be passed to the job manager, the gatekeeper must obtain a local account for this job. The gatekeeper will either consults with the map-file generated by the GUMS or pass a request to the local GUMS for a local site, depending on the GUMS configuration. If the GUMS is configured to map the grid certificate dynamically according to the job, the gatekeeper will act as later case, or just consult with the map-file.

The function of the GUMS can be summarized as follows [12]:

- Retrieve membership information from a VO server such as LDAP or VOMS.
- Maintain a manual group of people, and stored in the GUMS database (this is useful to handle special cases).
- Map groups of users to the same account (a group account).
- Map groups of users to an account pool, in which one account will be forever assigned to each user.
- Map groups of users according to the information present in NIS or LDAP.
- Map groups of users according to a manual mapping, stored in the GUMS database.

However, GUMS do not perform authentication but provide information to the gatekeeper. In this view of point, GUMS is just a Policy Decision Point (PDP) not a Policy Enforcement Point (PEP). It must cooperate with other middlewares.

2.3. PRIMA

PRIMA, short for PRIvilege Management and Authorization, is a system combining the resource access request with the appropriate privilege. In PRIMA model, a privilege is independent and self-contained, taking files access privilege for example. The privilege of access files are configured by the administrator and stored in the database. In order to ensure the seamless communication, PRIMA describe this privilege in XML-based language.

2.4. DAC, MAC and RBAC

DAC (Discretional Access Control), proposed in the 1970s, is based on the access control matrix. In this mechanism, any member in the system can empower other member's the privilege that is the subset of the privilege he has. And this information is stored and managed by the access control matrix. In the access control matrix, in which the line factors represent the User, the row factors represent the RP, and the elements represent the access privilege. If a User wants to use a certain resource, the DAC monitor will check the element on the intersection of the User and RP who own the resource. If this kind of access is allowed according to the element record, the access to the resource can be established, or be forbidden. However, the advantage of discretion of DAC also brings a big problem the DAC can not deal with: security. In DAC, the information and privilege always flows and be empowered from on member to another member. A User U_1 forbidden to the resource R_1 may get the privilege over it because another User who has the privilege over the R_1 empowers U_1 the privilege accessing to the R_1. Another problem is that this scheme is too complicated to maintain for the members, Users and RPs, and system managers.

MAC, short for the Mandatory Access Control, determines the privilege of access by the security level between Users and RPs. All the members, Users and RPs, are assigned with

security tags recording the security level by the system security administrator and these tags can not be changed by other members. User can only access to the resources whose security level is not higher than him. Multiple privilege management can be achieved based on the security tags. This mechanism is suitable for the centralized organizations and cannot be adopted a CI environment.

Role-based Access Control (RBAC), proposed by National Institute of Standards and Technology (NIST) in the 90's, is another access control mechanism that is widely used. RBAC binds the privilege with roles defined by the administrator. User can get the privilege only through the roles assigned to him. Administrator in charge of assigning the roles to the right Users according to the functions they have in the organizations. A role may be assigned to several Users, and meanwhile a User may have different roles in different departments. RBAC cut the direct relationships between privileges with specific Users, which makes the access control system more secure and easy to maintain. This scheme is only suitable for centralized organization for it has an administrator with the highest privilege over all the resources to assign the roles.

2.5. PERMIS

PERMIS [29], founded by ISIS, is designed to address the issues of an authorization of the personal idnode and determination of socio-economic attributes. Based on the RBAC mechanism, this scheme provides an authorization system that complements an existing authentication system. PERMIS is a privilege management with two major functions: provide policy editor for the owners to construct policies and assign appropriate privilege to the remote users. There are two kinds of policies: authorization policies define how to empower a remote user an appropriate privilege on local sites; delegation policies determine how to delegate to a trustable member the power to assign roles to other users in the same group. All these policies are in XML format. PERMIS also provides the Attribute Certificate Manager (ACM) and the Bulk Loader for managers to allocate privilege to users. The generated privilege information is stored in X.509 Attribute Certificate format [30]. With this policies and privilege information, PERMIS can provide following services:

- When users request access to your resources, PERMIS makes the access control decisions for you based on your access control policies and the roles of the users.
- Edit policies according to the requirement by the owner.
- It allows you to delegate to trusted individuals the ability to assign roles to users on your behalf.

PERMIS is kept in the local site which RP controls. So if being deployed in a dynamical environment, it is hard for PERMIS to maintain consistency among various repositories. Besides, PERMIS is more likely a policy engine without negotiation mechanism.

3. Federal VO Management

In a CI environment, everyone, including scientists, researchers, institutions and common PC users, are potential RPs. They can contribute their digital equipments, computers, sensors, instruments and other resources, to others and benefit from sharing resources through the CI

platform. All Cyberresources are various from access policies defined by the RPs. On the other hand, Users have different requests for resource sharing. TFVOM is designed to achieve trustable cooperation and resource sharing based on the agreement accepted by both RPs and Users. TFVOM helps Users to achieve agreements and trust relationships with RPs and then aggregates sufficient resources by establishing a suitable VO. In a VO, there can be three different kinds of members: Users, RPs and sub-VOs and two types of policies: resource policy and VO policy. Sub-VO is both a User and a RP in a VO. Resource policy, formulated by the RP, is the description on how much privilege the User can have to use the resource. VO policy, formulated by the VO administrator, describes what resources can join in and what privilege the User should have over the resources in the VO. VO policy is used to balance User requirements with RP interests. In this section, we will introduce the details of the TFVOM.

3.1. VO Architecture

In a CI environment, VO is established for the purpose of cross-domain resource sharing and member collaboration. A VO consists of two different types of members: RPs and Users. A member can be both a RP and a User in one VO. A sub-VO can be considered as a RP and a User simultaneously.

All Users, RPs and VOs are regarded as Nodes in a CI environment. If node A directly belongs to node B, we call node B is a *Father Node* of A and node A is a *Child Node* of B. If node A belongs to node B indirectly, we call node B is an *Ancestor Node* of A. If nodes A and B belong to the same Node directly, we call they are *Brother Nodes*.

As shown in Figure 1, VO2 is a Child Node of VO1 and VO1 is a Father Node of VO2. VO1 is an Ancestor Node of RP3. One VO, RP or User can belong to multiple VOs. As shown in Figure 1, VO2 belongs to both VO1 and VO4. RP3 is both a member of VO3 and VO4. In a CI environment, Users, RPs and VOs are same logically, so we manage them using a uniform abstract: *Node*. VOs in CI consist of nodes and has a hierarchical architecture.

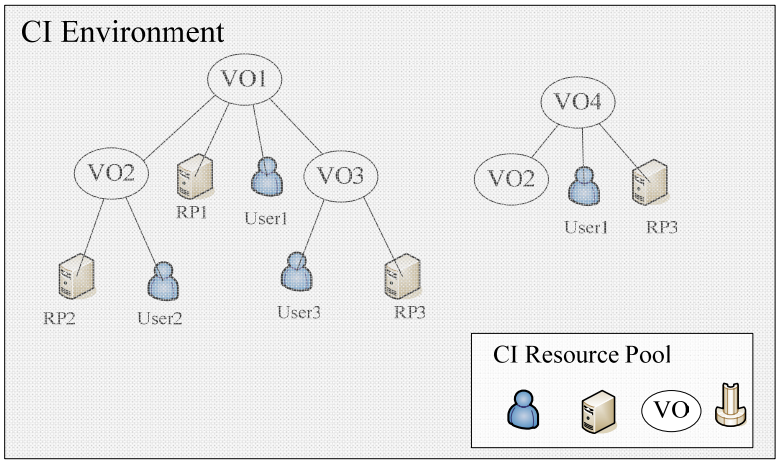

Figure 1. Hierarchical VO architecture.

3.2. Resource and VO Policies

In a CI environment, there are two types of nodes with policies: RP and VO. The policy defined by the RP is different from that defined by a VO administrator.

Resource Policy: This policy represents RP configuration about how to share their resources. A PC owner may only allow sharing his PC when the CPU utilization ratio is lower than 10 percents, providing no more than 30 percents EMS memory of the PC and reading on the hard disk is not allowed. All these are configured by the RP according to his preferences. Cyberresources consist of various types of instruments and services, policies of which are quite different from each other. For scientific computing, some jobs are computation intensive and others may be data intensive. More over, Cyberresources do not just include physical instruments, but also Web Services developed by the scientists. There may be more limitations and definitions on how to share their services.

VO Policy: This type of policy defines rules all members in the VO should obey and rules other nodes out of the VO should obey if they want to cooperate with nodes in this VO. Former rules are use policy, representing Users' reqirements about what kind of resources and collaborators they need. These include instrument status such as memory usage, CPU frequency, core number, instrument type, and bandwidth, and the way to use the resource such as available time, memory limitation and cost. A program for data analysis would collect computers with large memories and bandwidth, which can be defined via the VO policy. Another rule type, share policy, is proposed for the nodes out of the VO. As mentioned before, VO has two roles in function: User and RP. So a VO may join in another VO as RP. Share policy, having the same function with the RP policy, are the policy for other nodes on how to combine and share this VO. If a scientist in Bioinformatics established a VO to analyze data in biology, he may not be happy to include any other VOs which have nothing to do with bioinformatics. He can write this policy as VO policy to avoid such a situation.

In summary, there are only two types of policies in function: policy for outer nodes and that for inner nodes. First policy, stipulating outer nodes how to share local resources, is regarded as use policy. Second type of policy, stipulating members how to share resources in the VO, is regarded as share policy. RP only has use policy while VO has both of them.

3.3. Federal Cooperation and Sharing Mechanisms

Organizations are usually formed to achieve aggregation and cooperation using two different types of mechanisms: centralized mechanism and federal mechanism. In the centralized mechanism, power is distributed in a pyramid way, centralized to the top level organizations. Most of organizations with fixed architecture apply this mechanism. If two organizations, organization 1 and organization 2, all have privilege on an instrument and organization 2 is a sub-organization and belongs to organization 1, organization 1 has higher privilege than organization 2 over the instrument when there are some conflicts. But in the federal model, a big organization, consisting of small organizations and individuals, has smaller privilege over resources of sub organizations. Organizations at bottom of the hierarchy have highest privileges over resources.

CI is an open environment with high freedom and flexibility, in which cooperation and aggregation between nodes happens frequently. The reason we adopt a federal mechanism to

deal with cooperation and sharing in CI can be summarized in two aspects. 1) In CI, all the resources owned by various RPs who have the highest privilege over their resources. VO can not have higher privilege than the RP for security problems. Besides, VO can join to a higher-level VO for the cooperation. Joining a VO does not mean that the sub VO or resources will be controlled by the father VO. Node only collaborate based on the common goals. This feature is quite different from organizations with a centralized architecture. 2) In CI, VO is established, removed and combined in other VOs dynamically according to the various requirements. When a node needs to aggregate resources or collaborate with other User or VO, it will establish a VO or join in the established VO. Such actives happen frequently. In this case, the federal mechanism is more suitable for the CI environment than the centralized mechanism.

In CI environment, privilege is defined by the policies. VOs at different levels have different privileges over a specific resource, though this resource belongs to all these VOs. Figure 2 shows the difference of privilege scopes of VOs at different levels. We denote the rectangle as a node and the context covered by the rectangle as the privilege the node has over the appointed resource. RP has the highest privilege and can control the resource completely, and contributes part of his privilege for the members in VO2 based on its use policy. The same situation happens between VO1and VO2: VO2 has higher privilege over the RP than the VO1, for VO2 is one of the RPs of VO1 in this case.

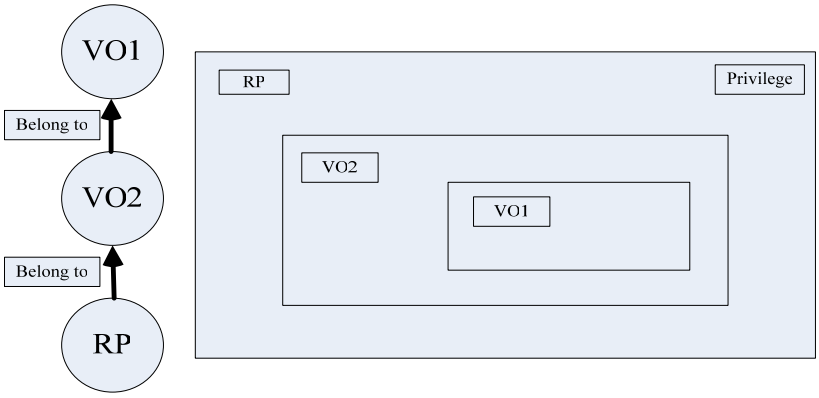

Figure 2. Privilege architecture in the federal mechanism.

4. Trust Management

Federal VO architecture can ensure the platform flexibility and dynamism to satisfy the various requirements, however, it is not so easy to be implemented, since privileges are quite different from one to another. A computer owner may just want to only contribute his computer to certain users or just provides computational capability without data storage. Because policies are defined by common users and RPs individually according to their specific requirements, it is hard to describe all these policies in a uniform way. A lot of privilege search and manage scheme which are all based on semantic analysis are proposed, such as SIMDAT (semantics-enabled service discovery framework in a pan-European pharmaceutical Grid) [30].

In fact, RPs define various resource policies just because they have different trust levels for different users. Resource policies will transform different trust levels to appropriate privileges on the local site. Since trust levels can be somehow charaterized in a uniform and quantfatitative way, we can just map different trust evaluation to different levels of privileges. This support can help RPs to assign appropriate privileges to the Users.

Schemes to manage and evaluate trust values are wildly used in E-commerce and P2P applications [31, 32, 33]. Considering characteristics of a CI environment, a trust management model is proposed that can help Users and RPs achieve appropriate, flexible and dynamic trust relationships automatically.

4.1. Current Trust Models

Trust is defined in different ways: In [34], trust was defined as "a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action and in a context in which it affects his own action." In [35] , the trust that Device A places in Device B is defined as: the level that A believes B will implement the desired operations and will not initiate or transfer attacks on Device A or a system that runs on Device A. In CI environment, trust value A places in B is the level of risk A would take to empower privileges to B. More privileges always lead to high risk. When a RP receives request from a User, he must first calculate the value of trust he places on the User, and then the RP will empower appropriate privileges to the User according to the trust values.

In the CI environment, trust value A places on B is not the same as that B places on A, for the trust value is not symmetrical. We use $T_{A\text{-}B}$ to denote trust value A places on B.

In an E-commerce environment and other virtual communities on the Internet, trust management schemes originate from real world scenarios. In real world, people get trust information in two ways: recommendation from others and their own judgment. Recommendation from others is usually called as reputation in real world. On the other hand, ones own judgment comes from his individual experience. So in a virtual environment on the Internet, researchers also adopt similar ways to evaluate and manage trust values via both *direct trust* and *indirect trust* [34, 36].

Direct trust: Direct trust value can be calculated from historic information Node A observes from Node B. When Node B wants to establish trust relationship with Node A, it will present its trust related information to Node A, such as certificate, membership, etc. Node A also has historic access records. Then A will calculate trust values according to these two aspects of information: trust related information Node B presents and historic records maintained by Node A:

D_{A-B}: Direct trust value Node A places on Node B;

Inf_B: Information Node B presents to Node A;

Rc_{A-B}: Historic records between Nodes A and B.

Direct trust values will be calculated as $D_{A-B} = f(Inf_B, Rc_{A-B})$. This trust value includes the individual judgment of Node A to Node B.

Indirect trust/reputation: Indirect trust values are provided by a third party, which is regarded as a trustable node for Node A. Assume Node B wants to establish trust relationship with Node A. Node B should send a request to Node A. Then Node A needs to make a decision on if Node B is trustable. Besides direct trust values, it is also important to consider other trustable nodes' opinion about Node B. For example, Node C is trusted by Node A as a third party node. How Node C evaluates Node B influence the trust value Node A evaluate on Node B. The importance of the recommendation depends on the trust level Node A places on Node C. The recommendation from the third node is more likely regarded as reputation of Node B in real world:

I_{C-B}: Indirect trust values Node A places on Node B from Node C (recommendation from Node C);

Indirect trust values represent global judgment on a certain node. In E-commerce or distributed scenarios, final trust value is combined with two types of trust values with different weights and calculated as follows:

$$T_{A-B} = \omega D_{A-B} + (1-\omega) I_{C-B} \qquad 0 \le \omega \le 1;$$

where ω presents the level a node trusts its own judgment. ω=1 means Node A judges Node B totally according to its own experience and does not trust any recommedation from other nodes; ω=0 means Node A totally trusts the recommedation from others and do not adds any self experiences in it.

4.2. Trust Modeling in CI Environment

CI is a distributed environment in which all resources are aggregated and managed through VOs. VO is an aggregation of nodes in CI based on the agreed policies and trust levels. If Users or RPs evaluate trust values every time they want to share or contribute resources with others, the platform is not only complicated to use but also hard to support a large scope of cooperation and resource sharing. Modern science research always need large scope of cooperation across disciplines and huge number of various instruments. CI should support dynamism, usability and large scope of cooperation of VOs. In fact, nodes in a VO already establish a certain level of trust relationships between each other when they agreed on the VO policies and joined the VO. There are three different types of trust values in the trust model of CI:

- **Global trust value:** This trust value is managed and calculated by the CI Management Center (CIMC). It represents the reputation of a node in the whole environment, for it is an accumulated value which is calculated from all historical records of activities in the environment. CIMC is a trustable node which is in charge of managing all nodes' global trust values as recommendation values.
- **Local trust value:** This trust value is based on the specific relationship. When Node A receives request from Node B, Node A will examine historic records of Node B and idnode information presented by Node B. And then Node A will calculate local trust value according to all these information. Local trust value, which makes sense only for specific relationship, represents individual opinion. Nodes can establish flexible and individual trust relationship through local trust values.

- **VO trust value:** This trust value defines the basic trust level of a VO and assigned by the VO administrator. All members of a VO are trustable at the level of the VO trust value. In default situation, all members can establish trust relationship among each other without any authorization and negotiation processes. This trust value represents the agreement of nodes in a certain VO.

The first two types of trust values are calculated while VO trust value is assigned by the VO administrator according to the VO purpose.

4.2.1 Global Trust Value

When a task is submitted by a User to a specific RP, there are two main properties to describe the completion of the submitted task: duration time and quality of execution. Duration time of a task represents task complexity. Quality of execution of a task can be characterized in three grades: success in time, success but delayed and failure. Longer duration time and higher quality contribute positively to trust evaluation.

When a task is finished and the task result is returned to the User, User will send a task report to the CIMC, which records the duration time and quality of execution of the task. CIMC will calculate contribution value from the report. And then we need an appropriate scheme to assign this contribution value to the related nodes: User, RP and VO (if necessary).

The trust contribution of task execution is assigned to Users and RPs with different weights. Users and RPs are designed to take the responsibility of the task together to avoid spite activities from Users and RPs. This can also encourage Users submit suitable tasks and configure sufficient expected time while RPs provide QoS services.

Final global trust value is an accumulated value from contribution of many tasks. Last task status has highest influence and trustability as it can more accurately reflect current situation.

If the task is submitted and finished across VOs, VOs which the User and the RP directly belongs to take part in this trust relationship, because a high trustable VO is a guarantor aided to provide credit situation of nodes thatbelong to the VO. Further more, scientists or researchers usually collaborate with each other on the VO level to establish resource sharing among all members of the two VOs. This cooperation is always the result of negotiation between administrators of the two VOs. In this case, VO plays a key role in the process of establishing trust relationship between members from different VOs. As VO is a guarantor in this process, it also takes responsibility in the resource sharing.

Global trust values represent nodes' reputation and recommendation of CIMC. It is basic trust evaluation in CI which provides three functions:

- Determine whether or not a node can join in a VO, for the global trust value of the node must satisfy the VO requirement. Generally speaking, global trust value of a node must higher than the trust value of VO it belongs to.
- This value is also one part of idnode information which helps node to calculate local trust value.
- If a node wants to establish a VO for his applications, the VO trust value configured by him is limited by his global trust value. VO trust value must be lower than founder's global trust value.

4.2.2 Local Trust Value

Local trust value represents the individual judgment on specific relationships. This type of trust values is calculated from two types of information: idnode information from the applicant and historical records in the local site. Assuming Node B sends a request to Node A to establish a trust relationship. The local trust value is calculated using historic data.

Historic records are items that record former activities between Nodes A and B. Every item consists of several properties: task duration, completion time and subjective judgment. Completion time is the time when a task is finished. Recent items always are more important and influential. Task duration time for a task is also of importance, the longer duration is, the more influential this task for the local trust is. Subjective judgment comes from the local site: the RP will give a rough judgment on the User who submits his tasks while the User will also make a judgment if the RP provide high quality of services.

4.2.3 VO Trust Value

The VO trust value is initially configured by the VO founder, namely the VO administrator, under the limitation of administrator global trust value. After all, a node can not establish a VO with trust value higher than its global trust value. Generally speaking, administrator's global trust value should higher than his VO trust value in a certain number for he must be surplus to handle unexpected global trust value changing. When a VO trust value is configured, this parameter should keep stable in its entire life. There is only one limitation: administrator should have sufficient global trust value to ensure the VO trust value.

4.3. Establishment of Trust Relationships

In a CI environment, all nodes are organized through VOs. The trust relationship between two nodes can be only established inner a VO directly or indirectly. If two nodes belong to the same VO directly, they can establish a trust relationship. If they do not belong to the same VO directly, they can establish trust relationship only if they have at least one common ancestor VO. Nodes many have many common ancestor VOs. In this case, we just choose the VO which has the highest VO trust value as the smallest common VO.

5. Implemtation

There are three components in a CI environemnt: CI Management Center (CIMC), VO Management Center (VOMC) and Clients. CIMC has three functions: authentication, trust management and node management. VOMC (VO Management Center) manages VO membership and monitors member status in real time. VOMC and CIMC are all implemented as Web Service to provide flexible and security information service. There is also a small optional client installed at client sides (Users and RPs). This client can help User manage local historic records and aid to make decision on the access control and privilege management. Common Users or RPs can also access to the VOMC or CIMC through a web browser.

Compared with VOMS, membership in a CI environment is recorded and authenticated by VOMC and CIMC rather than in the certificate. When two nodes want to establish a trust relationship, they must check the other node membership and global trust value from CIMC and VOMC. This scheme can ensure the dynamism and flexibility of VO memberships. Figure 3 shows how these three components work together.

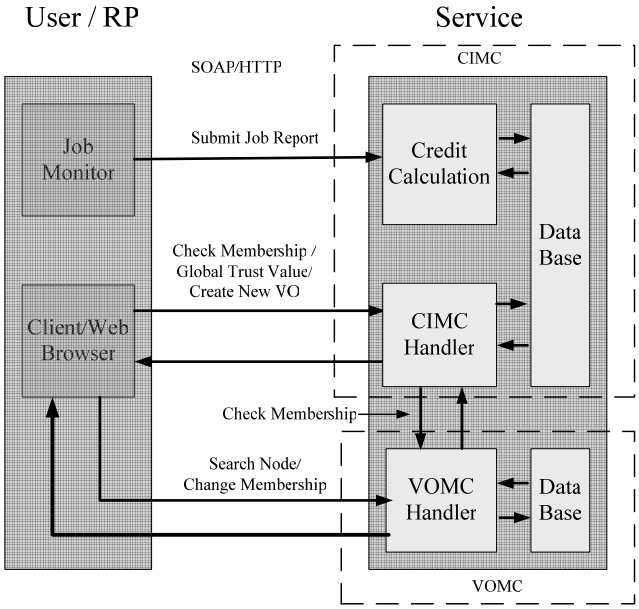

Figure 3. Architecture of three Components

5.1. CIMC

CIMC is composed by three function modules: Certificate Authority Module, Trust Management Module and Node Management Module and CI Database maintain all concerned information:

- **Certificate Authority Module**. This module is the authority center for all the nodes in the CI, signing and sending certificates. There are three sections in a certificate: a section for node basic information, an extendable section and digital signature.
- **Trust Management Module**. This component is mainly in charge of calculating global trust values. Input data is task reports from Users and RPs. When a new report is received, new global trust values are calculated and databased is updated.
- **Node Management Module**. Node Management Module is designed to manage and monitor all nodes in CI, including VOs, RPs and Users. This module is in charge of registering of VOs, Users and RPs and checking their qualifications. It is also used to provide identity information, e.g. membership of a node, global trust value of a node, etc.

5.2. VOMC

A VO is always established by a certain individual or another VO. Though this establishment may be the negotiation result among several individuals and organizations, there is only one administrator to found a VO. The administrator maintains a VO Management Center (VOMC) and undertakes the following responsibilities: managing VO membership, monitoring members status in real time, communicating with other VOs on VO level collaboration. VOMC has two components: VO Monitoring Module and VO Membership Module. There is also a VO database to store and manage all this information.

- **VO Membership Module**. This module records sufficient information about all the members in the VO and deals with dynamical changing of these information. Information on father nodes or brother nodes is also maintained. As mentioned before, nodes achieve resource sharing only when they belong to the same VO directly or indirectly. This module provides membership management and information service. Membership management handle requests related to the membership such as joining or leaving the local VO, and collaborating with other VOs and so on. Information service provides membership search and lookup services.
- **VO Monitoring Module**. This component is designed to monitor the member status in the VO in real time, especially the member global trust value, since every member should meet the requirement for the node global trust value from the VO and global trust value is dynamically changing.

5.3. Clients

The client is an optional component for Users or RPs. It is used to store individual historic records about past activities and evaluation. The client provides a tool to guide Users or RPs to record integrated information about a task and store them locally.

The client receives applications and provides processed information to help Users or RPs implement access control and privilege management. It calculates local trust value from past records, and final trust value as mentioned before. The client also retrieves identity information of applicants from CIMC and VOMC if necessary. All these information is provided to Users/RPs to help them make a decision. Users or RPs can also run a small component like GUMS to help them assign appropriate privilege to the applicant according to the information the client provides following the policies they are configured with.

6. Conclusion

TFVOM, as proposed in this chapter, provides federal VO management mechanism to achieve trusted collaboration. Compared with traditional technologies, TFVOM has the following features:

- **Trust Evaluation Supports**. This is one of core functions of TFVOM. Traditional resource aggregation or authority/authentication mechanisms can not provide this function. Members' credits cannot be evaluated only with certificates. The Grid,

which is based on the public key infrastructure, is widely used as a resource aggregation and cooperation platform. However, all the members of a grid have to achieve agreement on resource sharing beforehand. TFVOM provides an environment for members who know nothing about each other before to build credits, gain trust relationships with each other and form a VO together if required.

- **Federal VO Management**. Most of related mechanisms, such as RAC and RDAC, are suitable for centralized organizations. RAC and RDAC requires a center with highest privilege managing all members in the organization, e.g. assigning roles, empowering privileges, determine security levels and so on. The federal mechanism makes TFVOM suitable for incompact organizations.
- **Portability and Extendibility**. TFVOM can handle various resources and dynamic membership changes. Any RPs or Users can join the environment. Besides, the extendibility also indicates that TFVOM can meet various requirements and requests since policies are extendable.
- **On-the-fly Collaboration**. VOMS is another toolkit to enable cooperation and resource sharing across VOs. VOMS is more suitable for stable cooperation and resource sharing. TFVOM provides a series of tools that facilitates the process of dynamic VO operations, e.g. creating, joining, leaving, removing or merging VOs.

References

[1] D. E. Atkins et al. Revolutionizing Science and Engineering through Cyberinfrastructure. National Science Foundation Blue – Ribbon Advisory Panel on Cyberinfrastructure, January 2003.
[2] OCI – Office of Cyberinfrastructure. http://www.nsf.gov/oci.
[3] NSF Cyberinfrastructure Council. NSF's Cyberinfrastructure Vision for 21st Century Discovery, Version 7.1. National Science Foundation, July 2006.
[4] Thomas, T.; A mandatory access control mechanism for the Unix file system, *Aerospace Computer Security Applications Conference*, 1988.
[5] Yixin Jiang; Chuang Lin; Hao Yin; Zhangxi Tan, Security analysis of mandatory access control model, 2004 IEEE International Conference on Systems, Man and Cybernetics, Vol. 6, 10-13 Oct. 2004.
[6] Andreas Schaad, Jonathan Moffett, Jeremy Jacob. The role-based access control system of a European bank : a case study and discussion, *Proceedings of the sixth ACM symposium on Access control models and technologies*, May 2001.
[7] Ninghui Li; JiWon Byun; Bertino, E, A Critique of the ANSI Standard on Role-Based Access Control, *IEEE Security & Privacy*, Vol. 5, No. 6, 2007.
[8] Sandhu, R.S. Samarati, P. Access control: Principles and Practice. *IEEE Communications Magazine*, Vol. 32 , No. 9 pp. 40 – 48, 1994.
[9] http://vdt.cs.wisc.edu/VOMS-documentation.html
[10] http://hep-project-grid-scg.web.cern.ch/hep-project-grid-scg/voms.html
[11] Dongguk Univ. Grid Information Retrieval Management System for Dynamically Reconfigurable Virtual Organization, *Fifth International Conference on Grid and Cooperative Computing (GCC 2006)*, Oct. 2006.
[12] https://www.racf.bnl.gov/Facility/GUMS/1.2/index.html
[13] http://computing.fnal.gov/docs/products/voprivilege/prima/prima.html
[14] Privilege and Role Management Infrastructure Standards Validation: http://www.permis.org
[15] TeraGrid. http://www.teragrid.org.
[16] NEES – Network for Earthquake Engineering Simulation. http://www.nees.org.
[17] OSG – Open Science Grid. http://www.opensciencegrid.org.
[18] NEON – National Ecological Observatory Network. http://www.neoninc.org
[19] GEON – The Geosciences Network. http://www.geongrid.org
[20] NCAR – National Center for Atmospheric Research. http://www.ncar.ucar.edu
[21] NVO – US National Virtual Observatory. http://www.us-vo.org

[22] EDG – European DataGrid http://eu-datagrid.web.cern.ch/
[23] DataTAG – Data TransAtlantic Grid http://datatag.web.cern.ch/datatag/
[24] VOMS Monitoring Documentation http://voms-monitor.grid.iu.edu/cgi-bin/index.cgi
[25] LIGO – Laser Interferometer Gravitational-wave Observatory. http://www.ligo.caltech.edu
[26] SBGrid – Structural Biology Grid http://www.sbgrid.org/
[27] GUGrid – Georgetown University Grid http://gugrid.arc.georgetown.edu/
[28] I. Foster, C. Kesselman and S. Tuecke, The Anatomy of the Grid, *International Journal of High performance Computing Applications*, 15, 3 (2001).
[29] Permis http://sec.cs.kent.ac.uk/permis/
[30] C. Upstill, and M. J. Boniface, "SIMDAT," *CTWatch Quarterly*, vol. 1, no. 4, pp. 16-24, Nov. 2005.
[31] Alexandria, Virginia, Trust Management for Trusted Computing Platforms in Web Services, Conference on Computer and Communications Security, *Proceedings of the 2007 ACM workshop on Scalable trusted computing*, Nov. 2007 - Nov. 2007.
[32] Guangwei Zhang, Jianchu Kang, Rui He, Towards a Trust Model with Uncertainty for e-Commerce Systems, *Proceedings of the 2005 IEEE International Conference on e-Business Engineering (ICEBE'05).*
[33] Alireza Pourshahid, Thomas Tran, Modeling Trust in E-Commerce: An Approach Based on User Requirements,
[34] D. Gambetta, "Can We Trust Trust?" in *Trust: Making and Breaking Cooperative Relations*, Basil Blackwell, New York, 1988, pp. 213-237.
[35] Tao Sun, Mieso K. Denko. A Distributed Trust Management Scheme in the Pervasive Computing Environment, *Canadian Conference on Electrical and Computer Engineering (CCECE 2007)*, pp. 1219-1222, 22-26 April 2007.
[36] Pavlou, P. A., Yao-Hua Tan, and Gefen, D. The transitional role of institutional trust in online interorganizational relationships, *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, 2003.