A FRAMEWORK FOR MANAGING ACCESS OF LARGE-SCALE DISTRIBUTED RESOURCES IN A COLLABORATIVE PLATFORM

Su Chen[1]*, Tiejian Luo[2], Wei Liu[3], Jinliang Song[4], and Feng Gao[5]

College of Information Science and Engineering, Graduate University of Chinese Academy of Sciences, No. 19A Yuquan Road, Beijing, China
Email: chensu@mails.gucas.ac.cn[*1], *tjluo@gucas.ac.cn[2], songjl@mails.gucas.ac.cn[3], liuwei06b@mails.gucas.ac.cn[4], gaofeng05@mails.gucas.ac.cn[5]*

ABSTRACT

In an e-Science environment, large-scale distributed resources in autonomous domains are aggregated by unified collaborative platforms to support scientific research across organizational boundaries. In order to enhance the scalability of access management, an integrated approach for decentralizing the task from resource owners to administrators on the platform is needed. We propose an extensible access management framework to meet this requirement by supporting an administrative delegation policy. This feature allows administrators on the platform to make new policies based on the original policies made by resources owners. An access protocol that merges SAML and XACML is also included in the framework. It defines how distributed parties operate with each other to make decentralized authorization decisions.

Keywords: Federation, e-Science, Identity and Privilege Management, Access Control, Collaboration

1 INTRODUCTION

E-science relies on distributed collaborative platforms that aggregate the large-scale distributed resources across organizational boundaries and link people together for research. On such a platform, Virtual Organizations are dynamically created for research tasks. For this purpose, we need a framework that enables flexible, secure, coordinated resource sharing among dynamic collections of individuals, laboratories, institutions, and government agencies. This framework fulfills the following two goals:

- Identity management. In most cases, resource owners need to verify users' real identity (e.g. who are you? where are you from?) for making authorization decisions. It is very hard for a central registry system to verify the real identities of all users in a large-scale collaborative platform. Moreover, users should be able to authenticate just once to access all their authorized distributed resources. For this purpose, a unified approach to transport users' identification data across autonomous domains is necessary.

- Privilege management. On a distributed collaborative platform, resource owners have rights to determine who can use the resources and how they can use them. However, because Virtual Organizations are dynamically created for a specific research task, VO administrators, who are not necessarily resource owners, often need to administer security policies of integrated resources on the VO-level. For example, they may assign privileges of accessing a particular resource to the target VO members who are unknown to the resource owner. In this situation, privilege management tasks need to be decentralized from the resource owners to VO administrators whom they can trust. At the same time, the VO-level security polices must be consistent with the original security policies (i.e. domain-level security policies) made by resource owners. This dynamic nature of a Virtual Organization calls for an efficient and flexible approach to privilege management.

Centralized approaches are not suited for managing large-scale distributed resources because they are not scalable. Instead, federated approaches should be adopted. Federated identity and privilege management (FIPM) are two co-related aspects of an effective decentralized access management solution. Federated identity management enhances a user's privacy protection and convenience as well as decentralizes user management tasks through the federation of identities among identity and service providers (Gomi, Hatakeyama, Hosono, & Fujita, 2005). Federated privilege management relates to security policy management and authorization decisions across organizational boundaries by decentralized means. Although a number of FIPM schemes have emerged, most of them focus on decentralized identity management and authorization across autonomous

domains. Specifically, they do not provide an integrated approach for managing privilege on VO-level.

In this paper, we propose an access management framework for this purpose. The novel feature of this framework is that it supports administrative delegation policy, which allows managing privileges on the VO-level based on the domain-level policies specified by resource owners. The main contribution of the paper includes two aspects. First, we show how the access management of large-scale distributed resources in collaborative platforms benefits from an administrative delegation policy. Second, we propose an access protocol supporting two-phase, decentralized authorizations between collaborative platform and autonomous domains. The protocol merges the two open technique specifications. Thus it is easy to be implemented and extended.

The rest of this paper is organized as follows. Related works are shown in Section 2. The framework is given in Section 3. After that, we present a project case study. We conclude with a discussion of future work and potential research areas.

2 RELATED WORK

PKI-based approaches are widely used for distributed identity and privilege management. They are also known as Trust Management (TM), which means a unified approach to specifying and interpreting security policies, credentials, and relationships. It allows direct authorization of security-critical actions. A trust-management system provides standard, general-purpose mechanisms for specifying application security policies and credentials (Blaze, Feigenbaum, Ioannidis, & Keromytis, 1999). Policy Maker (Blaze, Feigenbaum, & Lacy, 1996) and Keynote (Blaze et al., 1999) are two notable TM solutions. Those TM solutions use capability-based credentials and bind the access control credential with either a user name or key. The name-oriented credential is not suited for distributed authorization because it is difficult to ensure that a user name is globally unique. In contrast, although the key-centric TM credential removes the dependency on names, this strategy blurs the distinction between authentication and authorization, thereby tightly coupling them and limiting the expressiveness and effectiveness (Bhatti, Bertino, & Ghafoor, 2007). In order to remedy the drawback of the traditional TM mechanisms, a number of promising schemes have been proposed. Notable among them is the Role-based Trust Management framework (Li, Mitchell, & Winsborough, 2002), which merges TM and Role-based Access Control (RBAC). In that framework, property-based credentials are used to map users into roles, and privileges are assigned to roles instead of identities or keys.

In recent years, SAML (Security Assertion Markup Language) (Cantor, Kemp, Philpott, & Maler, 2005a), XACML (Extensible Access Control Markup Language) (Moses, 2005), and WS-Federation (Lockhart, Andersen, Bohren, Sverdlov, Hondo, Maruyama, et al., 2006) are emerging technical specification of federated identity and privilege management.

SAML is an OASIS XML-based framework for exchanging security information. It defines two implementation profiles (i.e. Post/Browser and Post/Artifact), which can meet most federated identity demands in web-based environments. Both identity and service providers publish information about themselves with special XML files named metadata files (Cantor, Kemp, Philpott, & Maler, 2005b). Metadata files can be used to locate identity and service providers, as well as publish basic attributes policy. However, it is not a fine-grain security policy model. For example, a metadata file does not allow a service provider to publish different security policies for users from the same identity provider. Finally, it focuses on supporting distributed authentication (i.e. identity management), not specifying and enforcing distributed authorization (i.e. privilege management). Shibboleth (Scavo & Cantor, 2005) is well-developed SAML middleware supporting web-based federated identity management. Another similar framework supporting federated identity management is Liberty ID-FF (Cantor & Kemp, 2003).

XACML is an OASIS standard for access control. It provides a unified method and format for expressing access control elements' policy. Following the data flow model and specification of XACML, it is possible to create a standard, modular, and pluggable access control mechanism for all services. In a complex distributed collaborative environment, it is useful to combine SAML with XACML to provide a comprehensive FIPM solution (Anderson & Lockhart, 2005). Besides XACML, the XML-based Generalized Temporal Role Based Access Control (X-GTRBAC) (Bhatti, Joshi, Bertino, & Ghafoor, 2005) is an expressive specification that uses RBAC to define dynamic fine-grain access control in an enterprise environment. X-GTRBAC supports fine-grained attribute-based access control together with a modular authentication and authorization mechanism such as SAML (Bhatti et al., 2007).

In a Web Service (WS) environment, federated identity and privilege management can be built upon standard WS-* family protocols. WS-Security (Nadalin, Kaler, Monzillo, & Hallam-Baker, 2006) is the foundation of the protocol family. It specifies how to ensure the confidentiality, integrity, and non-repudiation of Web services messaging. Built upon WS-Security, WS-SecurityPolicy (Nadalin, Goodner, Gudgin, Barbir, & Granqvist, 2007a) provides a flexible and extensible grammar for expressing the capabilities, requirements, and general characteristics of entities in an XML web services-based system; WS-Trust (Nadalin, Goodner, Gudgin, Barbir, & Granqvist, 2007b) provides methods for issuing, renewing, and validating security tokens, as well as ways to establish and assess the presence of broker trust relationships; WS-Federation leverages an existing WS-* family of specifications providing a rich extensible mechanism for both identity and privilege management, which means authorized access to resources managed in one realm can be provided to security principals whose identities are managed in other realms.

However, none of these proposed solutions directly supports privilege management on the VO-level. The VO-level privilege management requires that a VO administrator can only reassign all or some of privileges limited by the original security policies. This strategy is also known as delegation, which is considered as a useful and effective method to enhance the scalability of a distributed system and decentralized access control tasks (Gomi et al., 2005). Although some solutions, including Grid Delegation Framework (Ahsant, Basney, & Mulmo, 2004), WS-Federation, X-GTRBAC and Gomi et al. (2005), support delegation across autonomous domains, none of them can fully support VO-level privilege management. A well-designed VO-level privilege management scheme should have two important features. First, it should provide an expressive policy language to resource owners that declares whom to trust and how the privileges can be reassigned by trusted VO administrators (i.e. constrained delegation) (Bandmann, Dam, & Firozabadi, 2002). Second, by aggregating domain-level security policies, it should let VO administrators know which privileges are allowed to be assigned.

To our knowledge, DyVOSE (Sinnott, Chadwick, Koetsier, Otenko, Watt, & Nguyen, 2006) may be the only scheme that has similar functionalities with our novel framework. It is a delegation authorization model based on X.509 Attribute Certificates. However, in DyVOSE, delegated privilege management is achieved through assigning particular authorization attributes to users based on the attribute assignment policies. This scheme demands resource owners and VO administrators agree upon a set of specific attributes. Moreover, an attribute assignment policy is not directly related to some access privileges. In our framework, privilege management is delegated through defining new policies based on parent administrative delegation policies. An administrative delegation policy is always grouped with some assignable access privilege. This unified policy-based delegation model is simpler and clearer than attribute-based ones. For example, by parsing the aggregated administrative delegation policies made by resource owners, a VO administrator can easily know what privileges can be assigned to VO members.

In short, large-scale distributed resources and the dynamic nature of VO in collaborative platforms are challenging access management models and implementations, some of which were not well developed in the past. For this reason, we propose a new framework, consisting of a policy administrative delegation model and an access protocol, in the following section.

3 FRAMEWORK

In this section, we first describe the administrative delegation policy and why it benefits VO-level privilege management. Then, we propose an access protocol to interpret how it supports decentralized authorization. Finally, some practical issues are discussed.

3.1 Access Policy and Administrative Delegation Policy

In a large-scaled distributed collaborative platform, it is often not realistic to build and maintain a central authority database that can define all access policies. Instead, we need to allow VO administrators to make security policies on behalf of resource owners. In order to make sure that the policies made on the VO-level are consistent with the policies desired by resource owners, we also need a unified model to manage access policy itself. A feasible solution is to define the policy of policies, that is, a parent policy determining who can define the child policies and how they can be defined. Such a parent policy is not used to make access authorization directly. It is used to allow a particular group of subjects (i.e. VO administrators) to delegate some privileges to others (i.e. VO members) by making new policies. This is called administrative delegation policy. On the other side, access policy is for making actual access authorization decisions. This actual access authorization process is accomplished with an access protocol shown later.

Figure 1 shows an exemplar policy chain consisting of two administrative delegation policies and an access policy. Each administrative delegation policy can pass all or some of its privileges with optional delegation conditions to its child policy. An access policy is a policy without delegation conditions. It is the leaf node of a policy chain. The non-leaf node in a policy chain is the administrative delegation policy. When making authorization decisions, after a non-root policy is successfully applied to an authorization decision request, the authorization component should automatically reconstruct another authorization decision request, which can be evaluated by one's former parent policy. The process is recursively performed along the chain from the access policy to the root policy. If the root policy is applicable, the final authorization decision response is made by the authorization component. Rissanen and Firozabadi (2004) have discussed this policy chain idea in more detail in their work.

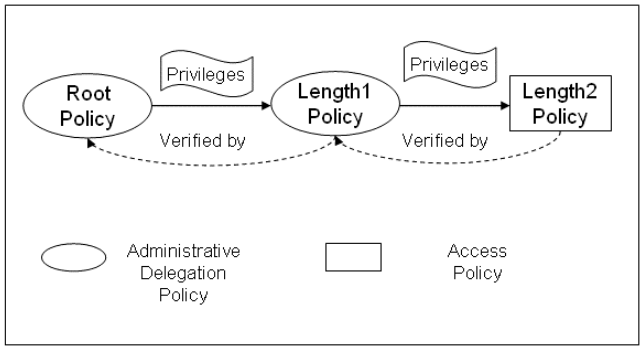

Figure 1. Diagram of a policy chain

Because XACML provides a common model for expressing and verifying policy among distributed partners in a unified format and method, we adopted it as the policy model in our framework. In XACML (Moses, 2005), access policy consists of a target, a set of rules, an identifier for the rule-combining algorithm, and (optionally) a set of obligations. The key component in an XACML specification is called the Policy Decision Point (PDP). It is responsible for making access authorization decisions according to security policies and received access requests. Although XACML does not directly support administrative delegation policy in its policy model, it can be extended to meet the requirement by adding new elements (e.g. Issuer) in its schema, as long as the relying parties can understand and process the new elements.

The details of the relationships and differences between XACML administrative delegation policy and access policy are shown in Figure 2. In general, the delegation condition element in the parent policy determines who can make the child policy. The new policy inherits elements including Target and Rule from the parent policy and contains additional Policy Verification Obligation elements. The term *inherits* means that a child policy should contain the subset of the access privileges from its parent set. Otherwise, the child policy conflicts with its parent policy and thus cannot pass the policy chain evaluation. The element Policy Verification Obligation occurs in any non-root policies. Each policy containing the element auto-generates another authorization request after being applied by the PDP. If the child policy is consistent with the parent policy, the parent policy is always applicable to the auto-generated request.

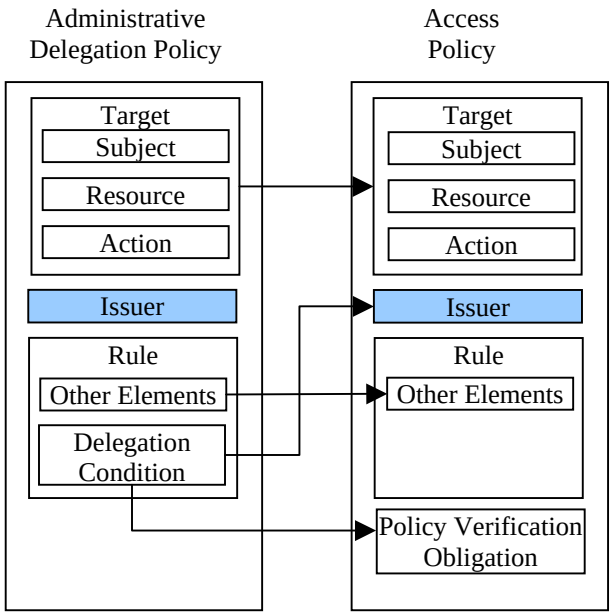

Figure 2. Comparing Administrative Delegation Policy and Access Policy

By supporting the administrative delegation policy, the policies made on the VO-level are under control of the resource owner as long as they publish administrative delegation policies to the Portal of the collaborative platform. Now we use a simple scenario to interpret how this model supports privilege management in domain-level and VO-level respectively. Assume that a resource owner has deployed a monitoring service in an institute named Institute-1, and the owner wishes to publish the resource to the collaborative platform. She or He first defines a root administrative delegation policy, which assigns delegation privileges to all the staff from Institute-1. The root policy is published to the platform's Portal. The monitoring service is also integrated into the Portal by developing a portalet web application. An example follows.

Step1: Bob who is a staff member from Institute-1 has created a VO on the Portal. At his workbench, he finds that he has rights to assign monitoring service access privileges. For collaborative research purposes, he defines a new access policy on the VO-level that allows John, who is from Institute-2 and has joined Bob's VO, to access the service.

Step2: John will see the monitoring service at his workbench. He makes an access request when he needs to use the service. Utilizing federated identity management, John is authenticated by the monitoring service in Institute-1. His subject name and some other attributes are sent to the monitoring service. Now, the monitoring service needs to make an authorization decision according to the authorization request. However, it does not find any local access policy applicable for current authorization request. So the monitoring service asks the platform Portal whether it can provide additional policies.

Step3: The Portal responded with applicable access policies and related administrative delegation policies (if existing) generated on the platform. After that, the PDP of the monitoring service first verifies the original authorization request with the access policy from the Portal. Additionally, another authorization request is auto-generated to make the policy chain verification.

Step4: The PDP of the monitoring service verifies the second authorization request using the local administrative delegation policy. Because it is a root policy, the recursive process terminates. John is allowed to access the monitoring service in Institute-1.

3.2 Access Protocol

The administrative delegation policy model illustrates how VO-level policy is defined and verified based on domain-level policy. When using the policy model to support decentralized authorization, an access protocol is also necessary to specify how the information is exchanged between the platform and autonomous domains. In this sub section, we use a data flow model to show the protocol.

3.2.1 Terminology

The access protocol merges SAML and XACML technique specifications, so most of terms are derived from Scavo and Cantor (2005) and Anderson and Lockhart (2005).

Identity Provider (IdP) - The identity provider maintains subject credentials and attributes. Upon request the IdP asserts authentication statements or attribute statements to relying parties, specifically service providers

Service Provider (SP) - The service provider protects secured resources from unauthenticated access. The resources are protected by a **SSO Service (SPSSO)**. The **Service Provider Assertion Requester and Consumer Service (SPARCS)** is responsible for requesting and consuming authentication or attribute assertions with IdPs. Moreover, the SPARCS make authenticated users' attributes available for authorization authority in the same security domain.

Authorization Authority (AuthzA) - The authorization authority receives access requests and makes authorization decisions after a subject has been authenticated by the SP in the same security domain. An AuthzA consists of a PEP, a PDP, a PIP, and a PAP.

Policy Enforcement Point (PEP) - The system entity that performs access control by making decision requests and enforcing authorization decisions.

Policy Decision Point (PDP) - The system entity that evaluates applicable policy and renders an authorization decision.

Policy Administration Point (PAP) - The system entity that creates and maintains policies. In our framework, the PAP can provide policies for other querying entities in the federation for decentralized authorization.

Policy Information Point (PIP) - The system entity that aggregates subject attributes, environment attributes, and resource attributes and provides the aggregated result to PEP to make authorization decision requests.

Environment - The set of attributes that are relevant to an authorization decision and are independent of a particular subject, resource, or action

Resource – Any data, system components, or services that a subject can access if authorized by PDP.

3.2.2 Data Flow Model

We only show two organizations in this figure (See Figure 3), Institute-1 and the platform Portal, in this data flow model. The model can be extended, however, to any number of participating institutes and Portals.

Platform users access and manage integrated resources at their workbench in the Portal. There are two types of authorization in the data flow model: VO-level authorization and domain-level authorization. The VO-level authorization determines which resources are presented at a user's workbench and how a user can administer VO-level policies. The domain-level authorization determines whether a user can access resources governed in autonomous domains. These two types of authorization are co-related. The PAP of a participating institute should first publish its domain-level security policies to the PAP of the Portal. Any feasible mechanisms can be used for this purpose. Then the Portal parses received policies and determines the initial administrative and access privileges of subjects (i.e. users). In domain-level authorization, the VO-level applicable policy is queried by the PDP of the service in participating institute.

Now we use the same scenario as discussed in section 3.1 to show how the subject named John@idp.institute-2.ac.cn gets authorization to the monitoring service at Institute-1. We assume that the service's administrative delegation policies have been published to the Portal, and the access policy that allows John to access the monitoring service has also been made by Bob in his VO.

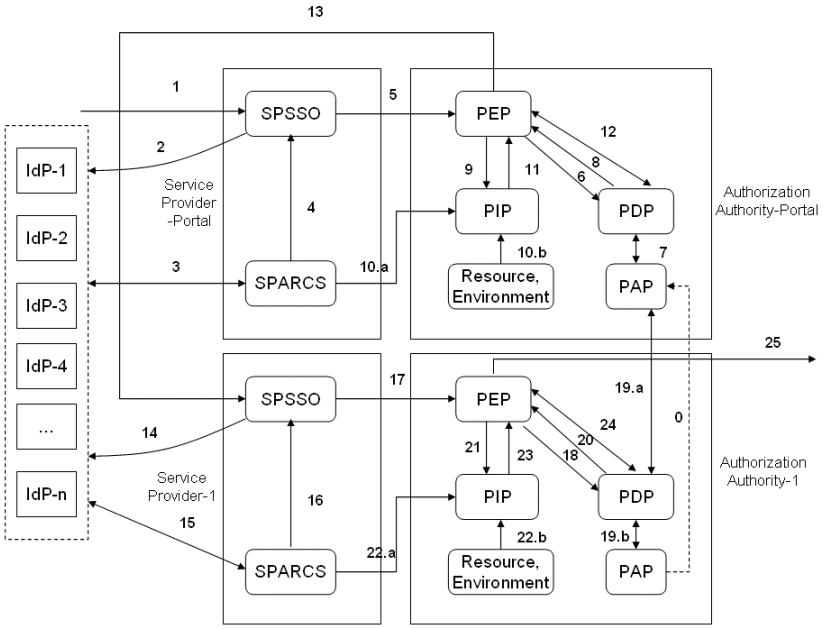

Figure 3. Data Flow Model of Access Protocol Supporting Decentralized Authorization

Phase 1: VO-level Authorization.

(1) John accesses the Portal through a user agent (e.g. web browser). The access request from the user agent is captured by the SPSSO of the Portal.

(2) The SPSSO redirects the user agent to perform a standard SAML authentication with a legal IdP selected by the user.

(3) John is from Institute-2, so he authenticates to IdP-2. After that, IdP-2 sends an SAML authentication assertion to the SPARCS of the Portal. The SPARCS may request additional attribute assertions from IdP-2.

(4) The SPARCS informs the SPSSO that the current subject has been authenticated.

(5) The SPSSO redirects the initial access request to the PEP. Note that the user agent has not sent an access request for the monitoring service until now. Actually, John never sees the monitoring service at his workbench unless the PDP of the Portal makes an authorization decision, so John's initial access request is to visit his own workbench. In the next step, the Portal must determine how to present the workbench to John. For this reason, the PEP needs to construct a set of authorization decision requests for the task. In what follows, we only show the decision request for the portalet wrapping the monitoring service.

(6) The PEP constructs the native decision request in the format <subject, resource, action> and sends it to the PDP.

(7) The PDP receives applicable policies from the PAP and checks whether any additional attributes are needed.

(8) (Optionally) The PDP requests any additional subject, resource, and environment attributes from the PEP.

(9) The PEP sends an attributes request to the PIP.

(10) The PIP retrieves attributes from the SPARCS, the Resource, and the Environment Components.

(11) The PIP responds with attributes to the PEP.

(12) The PEP sends additional attributes to the PDP, and the PDP responds with the authorization decision.

(13) Now John has been permitted to visit the portalet wrapping the monitoring service. Assuming that the only function of the portalet is redirecting the user agent to the monitoring service's access point, the VO-level authorization is finished.

Phase 2: Domain-level Authorization

(14) – (18) The processes are the same as (2) – (6). For SSO purposes, the IdP should provide a mechanism (e.g. encrypted cookie) to prevent a subject from re-inputting authentication information.

(19) The PDP of the monitoring service requests the VO-level applicable polices by sending the SAML Assertion containing the <XACMLPolicyQuery> element (Anderson & Lockhart, 2005) to the PAP of the Portal. The PDP also gets domain-level applicable policies from the PAP of the monitoring service. Then the PDP checks whether any additional attributes are needed.

(20) – (24) The processes are the same as (8) – (12).

(25) Now John has been permitted to access the monitoring service. The domain-level authorization is finished.

The above access protocol seems complicated only because the SAML and XACML have been merged to work together. The data flow model therefore follows specifications of the two techniques in order to reuse existing components (e.g. Shibboleth and Open XACML). As discussed below, the data flow model may be simplified in some scenarios.

3.2.3 Practical Issues

Besides the administrative delegation policy model and the access protocol proposed in the former section, some practical issues influence the implementation of the framework.

The first issue concerns the logical delegation policy language. The effectiveness of the administrative delegation policy model depends on an expressive logical delegation policy language, which allows the policy maker to put any delegation constraints on an administrative delegation policy. In our model analysis, we did not focus on logical delegation policy language because putting delegation constraints on a subject name is acceptable in our implementation cases. However, powerful logical delegation policy languages, such as the proposed solutions by Bandmann et al. (2002) and Yin, Wang, Shi, and Teng (2007), should be considered by system designers if it is necessary to put more flexible constraints on delegation. Because XACML is a XML-based language, it is easy to add a powerful logical delegation policy language.

The second issue concerns the authorization decision process. In our framework, we adopt two types of authorization: VO-level authorization and domain-level authorization. This scheme is designed to support decentralized privilege management. However, it also brings an extra workload if we do not limit the recursive policy evaluation processes because both the PDP of the Portal and the PDP of integrated services evaluate the full applicable policy chain for a decision request. Fortunately, in most cases, none of them needs to evaluate the full applicable policy chain. In VO-level authorization, because the PAP of the Portal ensures that each applicable conforms to its parent policy, the PDP of the Portal just needs to evaluate the applicable access policies for a decision request. In domain-level authorization, if the PDP of an integrated service trusts the PAP of the Portal, it also just needs to evaluate the access policy from the Portal. Moreover, it is also possible to let the PEP of an integrated service query an authorization statement directly from the PDP of the Portal. This scheme simplifies the data flow model, but it also has some drawbacks. First, this scheme demands a higher trust relationship among the Portal and the involved participating institutes. Second, in order to make the authorization decision in some cases, the Portal has to query additional attributes (e.g. Environment attributes) that are only available at integrated services. System designers should carefully evaluate the trade-offs in the particular environment.

4 CASE STUDY

The initial motivation for developing the access management framework is to support the Collaboration for Bio-safety Laboratory project (http://www.cbl-science.cn). The goal of the project is to develop a portal-based, grid-enabled, and extensible collaborative platform, which integrates large-scale distributed resources to facilitate education and research for improving the means of detecting, preventing, and treating infectious diseases. It is a vital part of the strategic plan of “National Facilities and Information Infrastructure for Science and Technology,” which was initiated by the Ministry of Science and Technology (MOST) of China in 2003. We aim at building an infrastructure that enables flexible, secure, coordinated resource sharing among dynamic collections of individuals, laboratories, institutions, and government authorities. Within the collaborative environment, administrators, researchers, educators, and public users will be able to easily access a massive data sharing, large-scale instrument sharing, and computing power sharing repository for research and educational purposes. To achieve this goal, we need a functional, scalable, and manageable framework to fulfill the

requirements.

We have developed an architecture supporting VO management that can easily to be tailored and extended (Luo, Song, Chen, Liu, Xu, Du, & Liu, 2007a, 2007b). By developing a service-bus communication protocol, we achieve a lightweight architecture with customizability and expandability. It can integrate self-made and third party collaborative tools, then select and customize such tools to meet with specific collaborative platform needs from various domains. Because domain experts rely on grid-enabled services to support remote control of large-scale instruments, high performance computing, and experiment monitoring, we have made efforts to develop computational and instrumental tools with Grid technology under the unified VO management architecture (Luo, Liu, Chen, Song, Jin, & Du, 2008). Presently the platform has more than 17 service components supporting more than 10 collaboration scenarios. We also have 10 participating institutes with 5000+ researchers in seven provinces around China. This amount exceeds 60% of the total BSL-3 laboratories authorized by the Chinese government. The remaining institutes are also the integrating target in our work agenda.

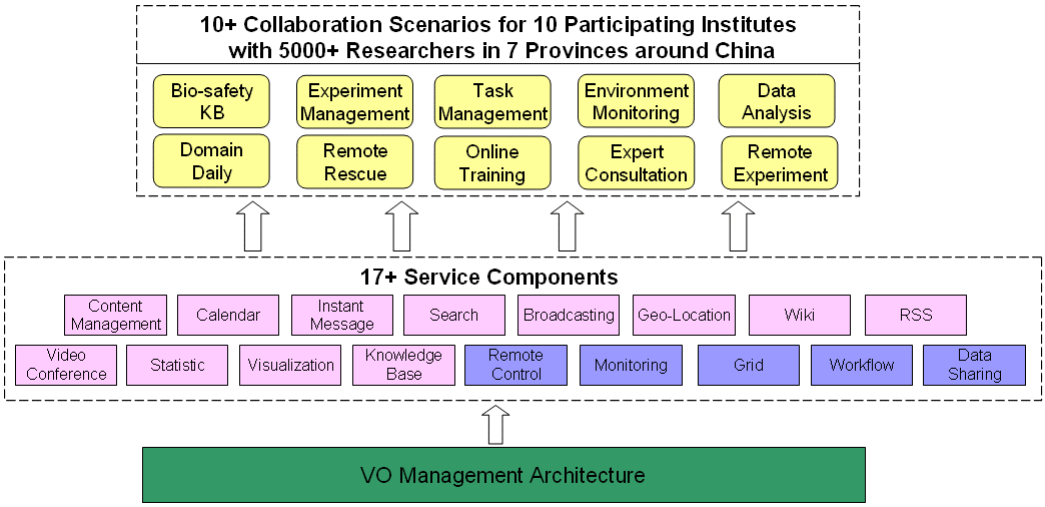

Figure 4. CBL Architecture Overview

The access management framework proposed in this paper is a part of the VO management architecture. As shown in Figure 4, five services including Remote Control, Monitoring, Grid, Workflow, and Data Sharing are deployed in participating institutes. Each participating institute determines its own administrative delegation policies for these services and publishes them to the platform Portal. The Portal aggregates the domain-level polices together with the VO-level policies to determine users' privileges and provide an integrated workbench for each user.

Additionally, with this access management framework, the CBL collaborative platform is interoperable with other partnership collaborative platforms. This feature is very important for the nation-wide integration of the information infrastructure that is being initiated by MOST of China.

5 CONCLUSION AND FUTURE WORK

In this paper, we propose a federated identity and privilege management framework for managing access to large-scale distributed resources in collaborative platforms. It decentralizes privilege management tasks from resource owners to VO administrators by supporting administrative delegation policies, which allows managing privileges on the VO-level to be based on the domain-level policies made by resource owners. An access protocol is also shown by a data flow model. Compared with similar proposed schemes, in our framework, delegated privilege management is achieved through defining new policies based on parent administrative delegation policies. An administrative delegation policy always goes along with some re-assignable access privilege. This unified policy-based delegation scheme is simpler and clearer than what has been done before. The framework has been successfully implemented in a large collaborative platform for bio-safety research in China.

In the future, we plan to improve the framework and publish it as an open-source middleware. Moreover, user activity in a distributed collaborative platform is also useful for authorization decisions. Further research also

includes incorporating the Reputation System (Resnick, Zeckhauser, Friedman, & Kuwabara, 2000) and Trust Metrics System (Paolo Massa, 2006) into our access management framework.

6 ACKNOWLEDGEMENTS

We would like to thank all the participants from the following organizations: Information Center at Ministry of Health, Peoples' Republic of China (PRC). Institute for Disease Control and Prevention, PRC; Center for Infectious Disease Control and Prevention, PRC; Institute of Disease Biology Research at Chinese Academy of Medical Sciences; Institute of Experimental Animal Research at Chinese Academy of Medical Sciences; Center for Infectious Disease Control in Jiangsu Province, PRC; Shanghai Medical College, Fudan University; Center for Infectious Disease Control in Fujian Province, PRC; Clinic Center for Public Health, Shanghai; Center for Infectious Disease Control in Yunnan Province; Center for Epidemic Disease Control in Heibei Province, PRC; and Medical College, Zhongshan University. This work is funded by Ministry of Science and Technology, PRC (Grant contract numbers 2005DKA64100 and 2005DKA10201) and EU-Asia Link Programme (Contract number CN/ASIALINK/020-103035).

7 REFERENCES

Ahsant, Basney, & Mulmo (2004) Grid Delegation Protocol. In *Proceedings of the Workshop on Grid Security Practice and Experience, 2004*.

Anderson & Lockhart (2005) SAML 2.0 profile of XACML v2.0, OASIS Standard. Retrieved Feb, 2007 from the World Wide Web: http://docs.oasis-open.org/xacml/2.0

Bandmann, Dam, & Firozabadi (2002) Constrained Delegations. In *Proceedings of 2002 IEEE Symposium on Security and Privacy*.

Bhatti, Joshi, Bertino, & Ghafoor (2005) X-GTRBAC: An XML-based Policy Specification Framework and Architecture for Enterprise Wide Access Control. *ACM Transactions on Information and SystemSecurity 8(2)*, 187–227.

Bhatti, Bertino, & Ghafoor (2007) An Integrated Approach to Federated Identity and Privilege Management in Open Systems. *Communications of the ACM 50(2)*, 81-87.

Blaze, Feigenbaum, Ioannidis, & Keromytis (1999) The KeyNote Trust-Management System Version 2, RFC2704. Retrieved Sep, 2005 from the World Wide Web:http://tools.ietf.org/html/rfc2704

Blaze, Feigenbaum & Lacy (1996). Decentralized Trust Management. *In Proceedings of the 1996 IEEE Symp. on Security and Privacy*, Washington, USA.

Cantor, Kemp, Philpott & Maler (2005a) Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS. Retrieved Feb, 2007 from the World Wide Web: http://www.oasis-open.org/committees/security/

Cantor, Kemp, Philpott, & Maler (2005b) Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS SSTC. Retrieved Feb, 2007 from the World Wide Web: http://www.oasis-open.org/committees/security/

Cantor & Kemp (2003) Liberty ID-FF Protocols and Schema Specification. Version 1.2. Retrieved Jun, 2008 from the World Wide Web: http://www.projectliberty.org/resource_center/specifications/

Gomi, Hatakeyama, Hosono, & Fujita (2005) A delegation framework for federated identity management. In *Proceedings of the 2005 Workshop on Digital Identity Management.*

Li, Mitchell, & Winsborough (2002) Design of a role-based trust management framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, Piscataway, NJ.

Lockhart, Andersen, Bohren, Sverdlov, Hondo, Maruyama, et al.(2006) Web Services Federation Language

(WS-Federation).Version 1.2. Retrieved Feb, 2007 from the World Wide Web: http://www.ibm.com/developerworks/library/specification/ws-fed/

Luo, Song, Chen, Liu, Xu, Du, & Liu (2007a) A Services Oriented Framework for Integrated and Customizable Collaborative Environment. In *Proceedings of IEEE IRI-07*, LAS, USA

Luo, Liu, Chen, Song, Du, Liu, & Xu (2007b) A Collaborative Environment for the BSL3 Laboratories. *In Proceedings of ICOMP 2007*, LAS, USA

Luo, Liu, Chen, Song, Jin, & Du (2008)Utilizing Grid to Build Cyberinfrastructure for Biosafety Laboratories. In *Proceedings ofIEEE CWCW*, XiAn, China.

Moses (2005) eXtensible Access Control Markup Language (XACML) Version 2.0, OASIS Standard. Retrieved Feb, 2007 from the World Wide Web:http://docs.oasis-open.org/xacml/2.0/

Nadalin, Kaler, Monzillo, & Baker (2006) Web Services Security: SOAP Message Security 1.1. OASIS Standard Specification. Retrieved Feb, 2007 from the World Wide Web: http://docs.oasis-open.org/wss/v1.1/

Nadalin, Goodner, Gudgin, Barbir, & Granqvist (2007a) WS-SecurityPolicy 1.2, Committee Draft 02. Retrieved Feb, 2008 from the World Wide Web:http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/

Nadalin, Goodner, Gudgin, Barbir, & Granqvist (2007b) WS-Trust 1.3 OASIS Standard. Retrieved Feb, 2008 from the World Wide Web: http://docs.oasis-open.org/ws-sx/ws-trust/200512

Massa, P. (2006) A Survey of Trust Use and Modeling in Real Online Systems. In *Trust in E-services: Technologies, Practices and Challenges*, Idea Group, Inc.

Resnick, Zeckhauser, Friedman, & Kuwabara (2000) Reputation Systems. *Communication of the ACM, 43(12).*

Rissanen & Firozabadi (2004) Administrative Delegation in XACML. Position Paper. Retrieved Dec, 2006 from the World Wide Web: http://www.w3.org/2004/08/ws-cc/erbsf-20040902

Scavo & Cantor (2005) Shibboleth Architecture Technical Overview, Working Draft. Retrieved Dec, 2006 from the World Wide Web: http://shibboleth.internet2.edu/shibboleth-documents.html

Sinnott, Chadwick, Koetsier, Otenko, Watt, & Nguyen (2006) Supporting Decentralized, Security Focused Dynamic VirtualOrganizations across the Grid. In *Proceedings of Second International Conference on e-Science and Grid Computing, 2006.*

Yin, Wang, Shi, & Teng (2007) Rule Based Constrained Delegation Framework. *Chinese Journal of Computing 30(9)*, 1511-1519.