International Journal of Database Theory and Application
Vol.7, No.4 (2014), pp. 139-154
http://dx.doi.org/10.14257/ijdta.2014.7.4.11

An Unified Identity Providing Model of Heterogeneous Resource

Jing Liu[1], Xuanyu Liu[2], Lianzhong Liu[3] and Chunfang Li[4]

[1]Engineering Teach and Practice Training Center, Tianjin Polytechnic University, Tianjin, China

[2,3,4]School of Computer Science and Engineering, Beihang University, Beijing, 100191 China

[1]tfliujing@tjpu.edu.cn, lz_liu@buaa.edu.cn

Abstract

Existing identity providing models are generally user-oriented or service provider-oriented, which can hardly meet the requirements of integration for heterogeneous resources of enterprise. This paper presents a unified identity providing model for accessing to heterogeneous resources: providing resource identity, identity mapping from heterogeneous resources identity to unified identity, and also providing identity providing interface for application software developing. Unified identity shields the differences of identity information among resources in heterogeneous environment, which makes users get the ability to access resources with one unified identity.

Keywords: *Identity providing; heterogeneous resources; access control*

1. Introduction

With the development of information technology, large amounts of heterogeneous software and applications are constructed or extended when enterprises or governments expand their infrastructure to meet the ever-growing business requirement. The identity management mechanism each heterogeneous resource is different from each other. Users have to access to multiple resources and input user name and password frequently. In addition, there are those organizations cooperating with each other, and they require cross-organization resources access. Because of above, a unified identity providing mechanism for heterogeneous resources of cross-organization in a trust federation is extremely necessary.

Identity providing, also known as identity supply, refers to the automation of operations such as creation, maintenance and authentication of user identity within user identity's life cycle [1-3]. It is also interrelated with security services, such as creating user's account, resetting password, password synchronization, and synchronizing all kinds of certifications among application systems. Identity providing simplified the process of software installation and policy configuration, and it has been widespread concerned in the information industry.

Though identity providing technology and products are available, existing studies are mostly concerned with implementation of the identity providing inside a single organization or same kind of applications, such as single logon in web applications. With the increasing of interaction between organizations, it is necessary to provide mechanisms to implement cross-organizations identity providing, or identity providing in the federation scope [4]. Existing identity providing models did not take resources into account, so they are lack of the ability of integrating heterogeneous resources efficiently. This paper presents a new identity providing model named Heterogeneous resources-orient Unified Identity Providing model (HUIP).

ISSN: 2005-4270 IJDTA

2. Related Works

2.1. Exsiting Approaches

At present, identity providing models can be classified as user-oriented [5], service provider-oriented [6] and network-oriented [7].

In user-oriented identity providing models, user is the center of the system that every kind of information is under user's control [5]. Nowadays, this kind of model is implemented in several ways, such as SAML(Security Assertion Markup Language), UAC(User Account Control) module in Windows Vista OS, and SUDO(SuperUser Do) module in Linux OS. User-oriented identity providing models can make users obtain and update trust values more efficiently and have the ability to protect user's privacy in certain degree. Shortcoming of these models is that security policy configuration is complicated and it is difficult to manage identity information.

Service provider-oriented(SP-oriented) identity providing models mainly concern service providers. This kind of models maintains mechanisms to select security service dynamically, including authentication, authorization and access control. Kerberos is an authentication protocol implements this kind of model. SP-oriented model is designed from the viewpoint of service provider and it's usually inexpensive and easy to deploy. Because service provider completely controls identity management, it manages identity information safely and efficiently. But this kind of model is not convenient for users to utilize.

Network-oriented identity providing model mainly concerns configuration and management of network, and related security and access control issues. Advantage of this kind of models is to reduce the cost and fully reuse hereditary resources. It also has the ability of controlling interoperation between systems and ensures security transformation on the transport layer. But this kind of model does take user's experience and service provider into account either.

2.2. Unified Identity Providing

There are two types of identity providing, centralized model [8-9] and distributed model [10-11]. Centralized model uses a centralized server to maintain the mapping between identity of the user and applications. Identity providing server stores user account configuration information in a unified data storage system. Distributed model does not have the identity providing server; user account information was stored in local application system. When foreground system receives requests to create a user account in local system, it will send providing request to all applications. Nowadays, there are a lot of identity providing products support of centralized model, such as Xellerate of Thor and ManagerID of Blockade. Some other products support for both centralized model and dispersed model, one of them is Sun Java System Identity Manager of Sun. Distributed model is suitable for highly decentralized business environment, but the synchronization of identity information database is very complex and possibly lead to inconsistent user interfaces and identity providing procedure. Centralized model overcomes these shortcomings of distributed model; it has the ability to provide unified service interfaces and support for automated identity providing. Having summarized up the feature of existing techniques, HUIP utilize the centralized model.

After we introduced the concept of unified identity and resource, we can renew the relationship between identity and entity: entity is still a set of identity, but with only one unified identity. So one entity has only one unified identity that can be mapped to many resource identities. The comparison of ununified (a) and unified (b) relationship between entities, identities and resources identities is shown by Figure 1.

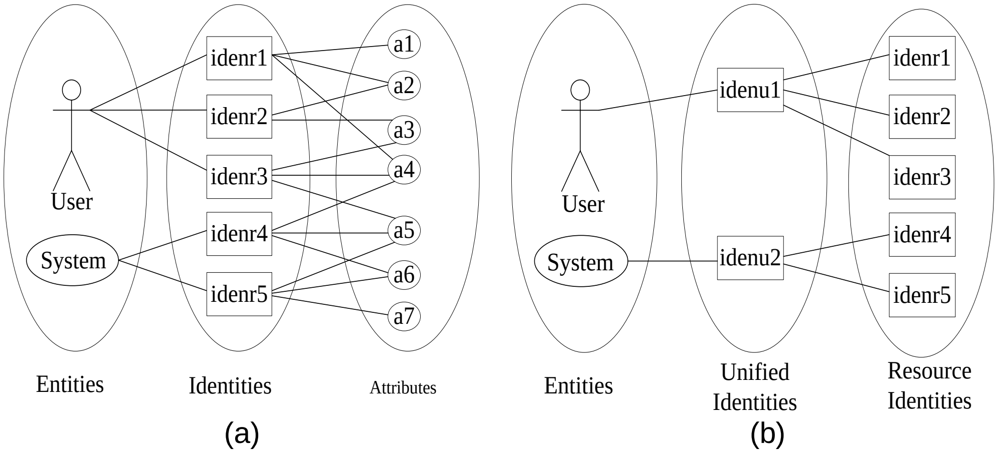


Figure 1. Comparison of Ununified Identity and Unified Identity Relationships between Entities, Unified Identities and Resources Identities

To solve the problem of identity providing for the heterogeneous resources, the usual solution is to provide identity to resources directly. It is complicated to provide identity and integrate the providing process because of identity information format, order of attribute and contents of resources are different from each other. Here proposed the concept of unified identity. Unified identity information is mapped to resource identities by identity mapping, so that users can use only one unified identity to complete providing operation. By this way, we can solve the complicated question of heterogeneous identity.

3. HUIP Model

3.1. Structure of HUIP

Heterogeneous resources-orient unified identity providing model(HUIP) is illustrated in Figure 2, which have 9 main fuction modules, as following:

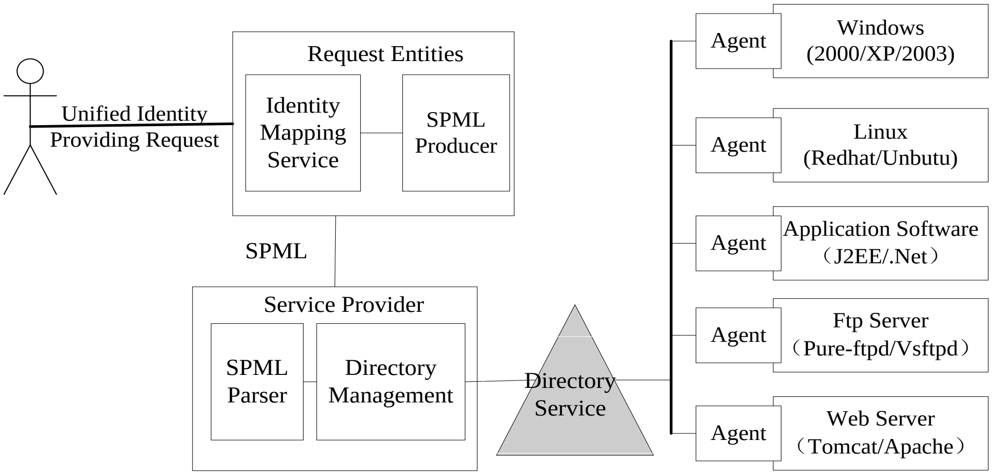


Figure 2. Structure of HUIP

1. Unified Identity Providing Request: sent by users and to request entities.
2. Request Entities(RE): receive identity request and transform into SPML (Service Provisioning Mark-up Language) message.
3. Identity Mapping Service: map identity information into specific resource identity.
4. SPML Producer: package the identity and request into SPML message.
5. Service Provider: receive SPML request and transform into directory service command.
6. SPML Parser: receive SPML request message, verity their legality and parse request type and identity information.
7. Directory Management: commit identity providing operation, and read and write identity information in LDAP(Lightweight Directory Access Protocol).
8. Directory Service: lib of identity information with tree data structure, and provide JNDI(Java Naming and Directory Interface) interface.
9. Agent: capture user request and redirect to directory service to obtain identity information and identify the identity.

In general, getting access to various heterogeneous resources need multiple user identities. In federation context cross-organization access requires allied user identity. HUIP abstract these identities and establish a unified main body identity and store it in the directory service. HUIP has storage, sharing, synchronization and distribution for identifying information. HUIP shields the differences of various heterogeneous resources to the lower layer, provides various services and interfaces for identity management and support for single sign-on to the upper layers. HUIP is an easy to extend and deploy unified identity providing solution. Key step of the unified identity providing is to create unified main body identity for various users access to heterogeneous resources.

We have put forward formal definition of HUIP. Providing operation must be described in semantics which can be recognized by machines in system. Here we use message model based on SPML and XML. Identity providing request only contains unified identity information and providing requiring instruction. Identity mapping will convert the information from unified identity to resource identity which contains all attributes that of certain resource. Request parser transforms unified identity providing to standard XML documents according to request type and resource identity data. This paper constructs standard XML schema for request of providing based on SPML. Standard XML documents are encapsulated by SOAP and send to SPML parser.

First, SPML parser tests the legality of the received request according to XML schema of SPML. If the result is valid, parser will divide and analyze this request message on the basis of SPML semantic. After parsing, system will store the information into persistent directory service. All kinds of heterogeneous resources can obtain identity information by connecting to directory service.

Identity providing is designed for all kinds of heterogeneous resources, including applications, operating systems and other hardware and software services. In each resource, an authentication agent is equipped. The agent is responsible to interact with the identity providing server to obtain user identity information. There is an identity acquire module in the authentication agent. This module is responsible for obtaining the identity information and sending it to the identity providing server, and then returns the results to this authentication agent to complete the authentication process.

Identity provision servers usually run on application servers or web servers. It supplies functions such as request processing, creating, modifying or deleting user accounts. System stores user configuration information and other relevant data in the

LDAP directory service. LDAP directory service stores all types of resources access information in the unified user identity information.

3.2. Definition

To discuss convinently, here a group of concepts are defined for identity providing and relationship between entities.

Definition 1: *Resource*, is the physical entity or information entity that executes providing requirement operation and stores identity information. Set of resource is denoted as *Res*. *Resource* is described by a unit doublet:

$$res = (rid, rtype), res \in Res$$

In the description, *rid* is a unique mark of resource in the system, and *rtype* is the type of resource.

Definition 2: *EqualR*, is a binary relation on *Res*.

$$EqualR = \{((rid_1, rtype_1), (rid_2, rtype_2)) \mid rtype_1 = rtype_2\}$$

EqualR is self reciprocal, symmetrical and transitive, so *EqualR* is an equivalence relation on *Res*. So *Res* can be described as a division by equivalence class of *EqualR*:

$$Res = \bigcup_{r \in Res} [r]_{EqualR}$$

Definition 3: *Attribute*, is a kind of property of a person or an organization. Set of attributes is denoted as *Attr*. *Attribute* can be described as a unit doublet:

$$attr = (k, v), attr \in Attr$$

In the description, *k* is the key of *attribute*, and *v* is the value. For example, if a person's name is *Alice*, it can be described as (*name*, *Alice*).

Definition 4: *EqualK*, is a binary relation on *Attr*.

$$EqualK = \{((k_1, v_1), (k_2, v_2)) \mid k_1 = k_2\}$$

EqualK is self reciprocal, symmetrical and transitive, so *EqualK* is an equivalence relation on *Attr*. So *Attr* can be described as a division by equivalence class of *EqualK*:

$$Attr = \bigcup_{a \in Attr} [a]_{EqualK}$$

Definition 5: *Identity*, is the description of user's information in system. Set of identities is *Iden*. *Identity* is a set of finite attributes. Two different attributes contained in one identity is not a part of *EqualK*.

Definition 6: *EqualI*, is a binary relation on *Iden*. Let identity $iden_a$ = (i_{a1}, i_{a2},…, i_{an}), and identity $iden_b$ = (i_{b1}, i_{b2},…, i_{bn}). | $iden_a$ |=| $iden_b$ |, if any $i_{ak} \in iden_a$, there exists and only exits one $i_{bk} \in iden_b$ that (i_{ak}, i_{bk})$\in$*EqualK*, that is what we call it ($iden_a$, $iden_b$)$\in$ *EqualI*.

It is easy to prove that *EqualI* is an equivalence relation on *Iden*. So *Iden* can be described as a division by equivalence class of *EqualI*:

$$Iden = \bigcup_{i \in Iden} [i]_{EqualI}$$

Definition 7: *IBR*, is a binary relation from *Iden* to *Res*. If $i \in iden$ is stored in *Res*, we call *it* (i, r)$\in$*IBR*.

Definition 8: *Resource Identity Set*. Let $(i,r)\in IBR$, so that $[i]EqualI$ is the Resource identity set of $[r]EqualR$. Identities in $[i]EqualI$ are called resource identity of r. Resource identity set is a equivalence class that represents a certain kind of resource uses the correlated identities. For short, we denote the resource identity set $[r]EqualR$ as $Iden(r)$.

Definition 9: *Unified Identity Set*. Unified identity set is a equivalence class established according to the requirements of system. Unified identity set is denoted as $[U]EqualI$. Attributes contained in U are assigned by the request of system. To any $r\in Res$, $[U]EqualI\cap Iden(r)=\varnothing$. Identity in $[U]EqualI$ are called unified identity. Unified identity is a kind of logical identity. It is a summary of resource identity. For short, let us denote $[U]EqualI$ as $Iden(U)$.

3.3. HUIP Identity Mapping

Unified identity is the core concept of HUIP model. This paper presents the concept of identity mapping base on unified identity and resource identity. As we mentioned before, one entity may has numbers of identities.

After we introduced the concept of unified identity and resource, we can renew the relationship between identity and entity: entity is still a set of identity, but with only one unified identity. So one entity has only one unified identity that can be mapped to many resource identities. The relationship between entities, unified identities and resources identities is shown by Figure 1.

Definition 10: *Corr*, is a binary relation from unified identity set to resource identity set. If the attribute of unified identity $attr_{ui}$ and the attribute of resource identity $attr_{rj}$ has the same semantic, we define that $(attr_{ui},attr_{rj})\in Corr$. For instance, unified identity has an attribute called *name*, and resource identity has two attributes, *username* and *password*. Name in unified identity and username in resource identity has the same semantic, so $(name,username)\in Corr$; name in unified identity and password in resource identity does not share the same meaning, so $(name,password)\notin Corr$.

Assume a unified identity described as follow:

$$iden\ (U\ ) = \{attr_{U1}, attr_{U2}, \cdots, attr_{Un}\},\ attr_{Ui} = (k_{Ui}, v_{Ui})$$

And a resource identity described as follow:

$$iden\ (r\ ) = \{attr_{r1}, attr_{r2}, \cdots, attr_{rm}\},\ attr_{ri} = (k_{ri}, v_{ri})$$

Define $<V_{r1},V_{r2},\ldots,V_{rm}>$ the vector of attributes of $iden(r)$. After creating operation, system will assign default value$<V_{d1},V_{d2},\ldots,V_{dm}>$ to the vector, or the default attribute vector. $attr_{ui}$ is the number i attribute of $iden(U)$, and $attr_{rj}$ is the number j attribute of $iden(r)$. Define C_j the relation value of $attr_{ui}$ and $attr_{rj}$:

$$c_j = \begin{cases} 1, if\ (attr_{Ui}, attr_{rj}) \in Corr \\ 0, if\ (attr_{Ui}, attr_{rj}) \notin Corr \end{cases}$$

Let $C(attr_{ui})=<c1,c2,\ldots,c_m>$ be the relativity vector between $attr_{ui}$ and $iden(r)$. So we can get the relativity matrix of $iden(U)$ and $iden(r)$:

$$C_{Ur} = \begin{pmatrix} C\ (attr_{U1}) \\ C\ (attr_{U2}) \\ \vdots \\ C\ (attr_{Un}) \end{pmatrix} = \begin{pmatrix} c_{11} & \cdots & c_{1m} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nm} \end{pmatrix}$$

When mapping unified identity *iden*(*U*) to resource identity *iden*(*r*), the formula of computing resource identity attribute vector is as (1):

$$(v_{r1}, v_{r2} \cdots, v_{rm}) = (v_{u1}, v_{u2}, \cdots v_{un}) C_{Ur} \oplus (v_{d1}, v_{d2}, \cdots v_{dm}), a \oplus b = \begin{cases} a, a \neq 0 \\ b, a = 0 \end{cases} \quad (1)$$

First, according to *Corr*, construct the relativity matrix of *iden*(*U*) and *iden*(*r*). Change attributes contained in *Corr* from unified identity attributes to resource identity attributes through the relativity matrix; assign default values which are given according to the requirement of system and practical situation to attributes do not contained in *Corr*.

3.4. Cross-organization Structure of HUIP

The cross-organization structure of HUIP is shown by Figure 3.

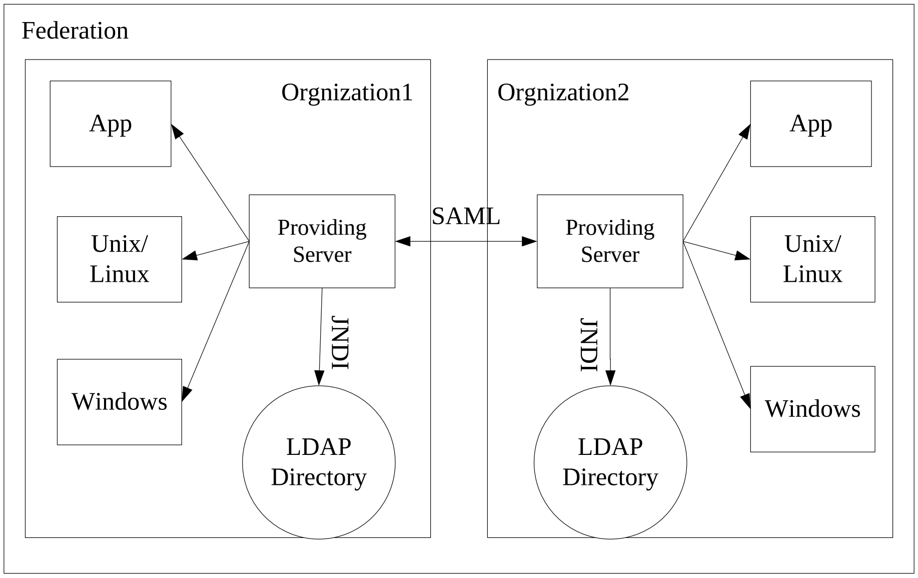


Figure 3. Cross-organization Structure of HUIP

Identity providing is available in cross-organization environment with trust relationships same as in unquie organization. Identity providing between the organization domains relies on the federation technology, using security transmission protocol SAML(Security Assertion Markup Language) to transfer identity information between servers in different organization domains.

To present it clearly, Figure 3 contains only two organization domains. In practice, a federation may contain a number of organization domains, and these domains make up a federation trust circle, and the identity providing model is completely suit for a federation trust circle.

4. Implementation

4.1. Directory Service Administration

Directory Services Administration allows administrators to remotely manage organizations and user information through the web browser. User administration subsystem updates and queries institutional staff information by accessing LDAP directory through JNDI. In the LDAP "C/S" mode, the LDAP server can let client gets access to useful information, JNDI beans mention in this paper implement six basic operations defined in LDAP protocol: (1)bind/certificate directory server; (2)query directory entries; (3)extract directory entry’s attribute; (4)add directory entry; (5)modify directory entry; (6)remove directory entry. Figure 4 is the JNDI directory service administration design.

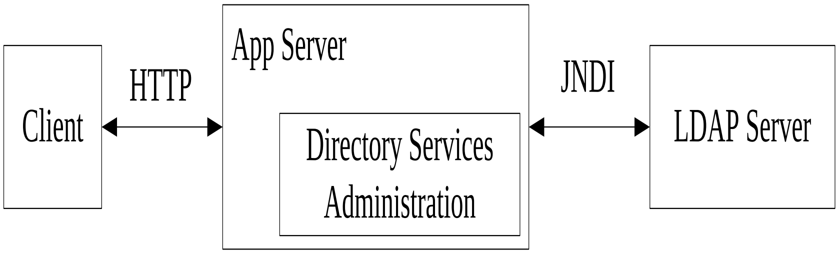


Figure 4. JNDI Directory Service Administration Design

4.2. Directory Service Agent Administration

Directory service agent administration implements synchronization of the existing directory services, and also provides an infrastructure synchronize with other directory services for future expansion.

Directory service agent administration is an external agent, hide the internal details of a variety of agents through a proxy interface. The proxy manages the existing agents as a linked list. Proxy receives a unified identity administration instruction request from user interface, then redirect it to the specific agent (agent plug-in). Directory agent is responsible for other servers to connect with ActiveDirectory, encapsulates protocol details connected with the directory server communication.

In order to achieve Windows account’s integration with the system, we should replaces Windows Service Account Manager (SAM) a steering. The steering will require Windows NT directory service, not SAM., to query the system user information. Windows user management and authentication has been changed: the redirect method to provide account management services for the Windows will no longer be used, but synchronization between Windows Active Directory and the local system directory service. The system uses Name Service Switching (NSS) provided by Unix and Linux to get detailed information of user account from system directory. Using pluggable authentication module (PAM) API provided by Unix and Linux, it allows Unix and Linux to achieve user authentication through directory services. For J2EE applications, filters can be used to provide services by intercepting HTTP requests and forward the request back to applications after processing the identity provision services.

4.3. Federation Supporting Service

Federation is a kind of model that is used to share authentication results between identity providers (IDP) and Service Providers. It was present by liberty alliance in liberty phase1 and liberty phase2. Multiple security domains (organizational domain) with mutual trust and independence constitute a federation. Federation supporting service make it possible to cross the boundaries of different organization domains. When local domain needs to execute providing operation to resources in the target domain, the identity providing server will pack the types of operation and the identity providing information into an SAML request message, then send it to the identity providing server in target domain. After receiving the SAML request message, identity providing server in target domain will analyze the message and get the identity information and types of operations, and then execute providing operation to the resources inside this domain. After the operations, the server will create a SAML response message and send it back.

Identity providing cross-domain needs to transfer identity information between organization domains. There are two types of information: global identifier and fake name. Global identifier is a unique identifier in the federation such as users working number. Fake name is a random string, and a user uses different fake names in different domains. Fake name can hide the identity information of user and protect the privacy effectively, so HUIP uses fake name to transfer information. When a user wants to get access to resources in another organization domain, identity providing cross-organization will be triggered. Suppose user *X* wants to get access to the SP (in this context, SP is the identity providing server in another domain, while IDP is the identity providing server inside the local domain), but SP could not confirm the identity of *X*, the providing possess is as follow:

(1) *X* sends a request to SP;

(2) SP sends a request to IDP, require for identity information of *X*;

(3) IDP judges whether *X* is legal and has the right to access SP; if positive, go to the next step; else, terminate the process;

(4) IDP calculate the fake name of *X* as *X*’, with other metadata information and send them to SP;

(5) SP searches local fake name database, if it does not contains information of *X*’, then insert *X*’ and metadata into fake name database;

(6) SP allows *X* to access resources, then the process terminates.

4.4. Data Synchronization Service

Legacy applications have their own user management system, so user data is usually stored in relational database. Data synchronization service provides synchronization function between LDAP user account information and user relational data tables. It confirms the consistence of different types of user database.

4.5. Interface Supply Service

Interface supply services are a remote interface service for the applications to obtain identity information. Applications can query the user identity information stored in the LDAP directory service through the supplied interface. Interface supply service can be implemented through SOAP (Simple Object Access Protocol), which is a part of Web Service standards and defines rules to access remote object based on XML. SOAP does not provide the underlying communication protocol; it usually implemented through HTTP. The benefits of using SOAP is that it is independent free with the programming language. Nowadays, organizations such as governments and enterprises contains multiple heterogeneous distributed applications. They may run on different platform and different programming languages, and have access to centralized storage of user information, so providing platform and programming language independent interfaces is critical. SOAP is the preferred solution to this problem.

4.6. SPML Engine

SPML parser parses SPML requests, transforming them to API invoking. The nature of SPML parsing is actually XML parsing. This paper creates a SPML parser based on OXMapping. The parser is SAX-based with random access ability.

JavaBean is used to map with elements in SPML document. Annotation marks the mapping relation, attributes and invoking of converters, which used to define format in the mapping process. When transforming from SPML document to object, handler gets SPML document, maps it to JavaBean and out put the object. In contrast, system uses JavaBean to create object, then handler maps it to SPML document.

4.7. Identity Agent

In the system, several kinds of resources are concerned, such as Windows OS, Linux OS, J2EE/.Net application and Apache Server. Agent is responsible for integrating these heterogeneous resources. LDAP is adopted to store resource identity information so that all kinds of resources could connect to LDAP to obtain identity information.

At present, most kinds of Linux OS offer NSS module to implements remote identity obtain. Also PAM offer several useful services for users. The agent is implemented on the basis of NSS and PAM. In Windows OS, we utilize Gina DLL developed by Microsoft to replace default MsGina, so that information in LDAP could be obtained from Windows OS instead of local SAM. Data accessing interface is published as Web Service for applications to obtain identity information. For Apache server, we use mod_auth_ldap in NSS to implement the adapter.

Identity providing includes user from login to logout organization, and the process flow involves adding unified identity, identity mapping, authorizing identity, and assignment failure, as shown in Figure 5.

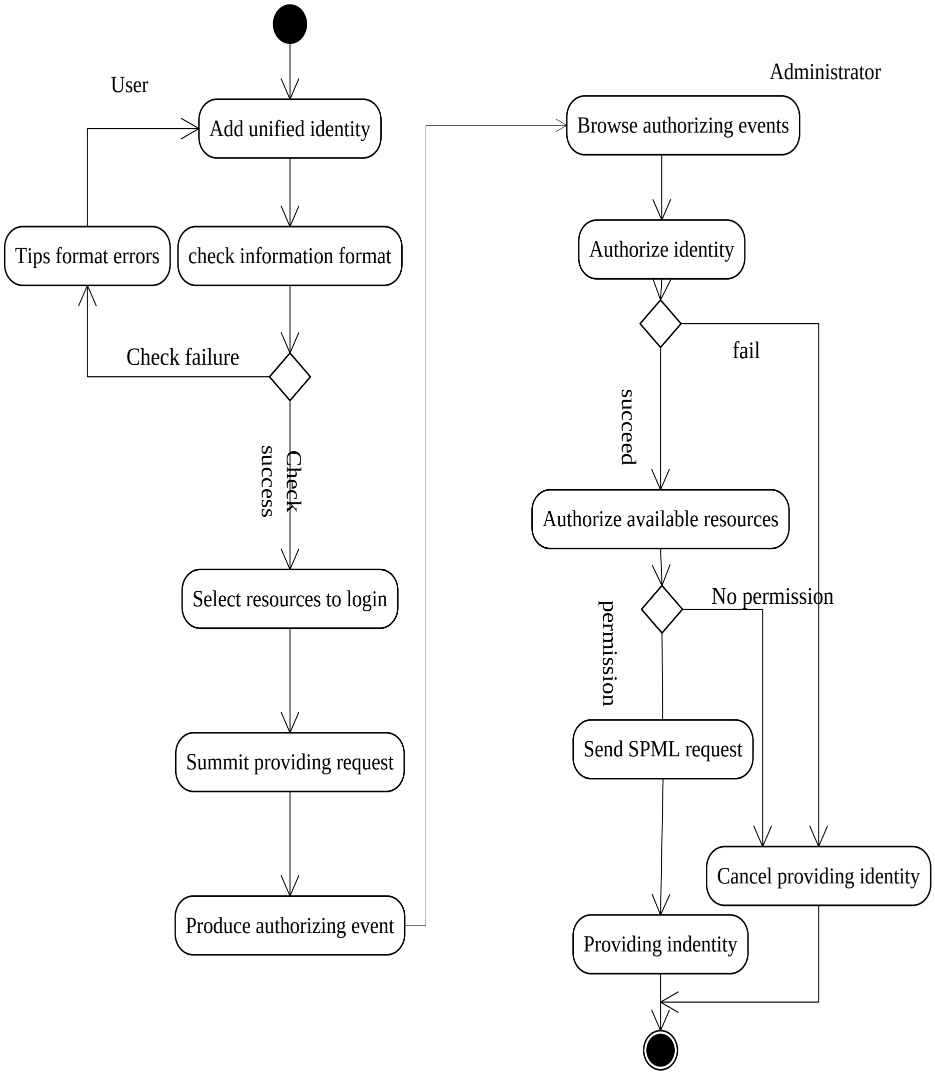


Figure 5. Active Chart of Identity Providing

There are two type users: user and administrator. User proposes identity providing request, and the request is sent to database in SPML format and wait to be authorized. Administrator has the right to authorize, browse, and cancel the identity providing from user request.

Password synchronizing. It's a control center for synchronizing password in multi resources. It sends the command, and SPML engine receive the request and check the validity, and deal with the request.

Resource agents. The unified identity information and resource indentity are saved in LDAP directory services, agents of different resources implement obtain and authorize the identity information. Agents changed old authorization process, and LDAP becomes the unified identity server. Though the authorizing techniques of different resources are different, they have same process procedure as illustrated in Figure 6.

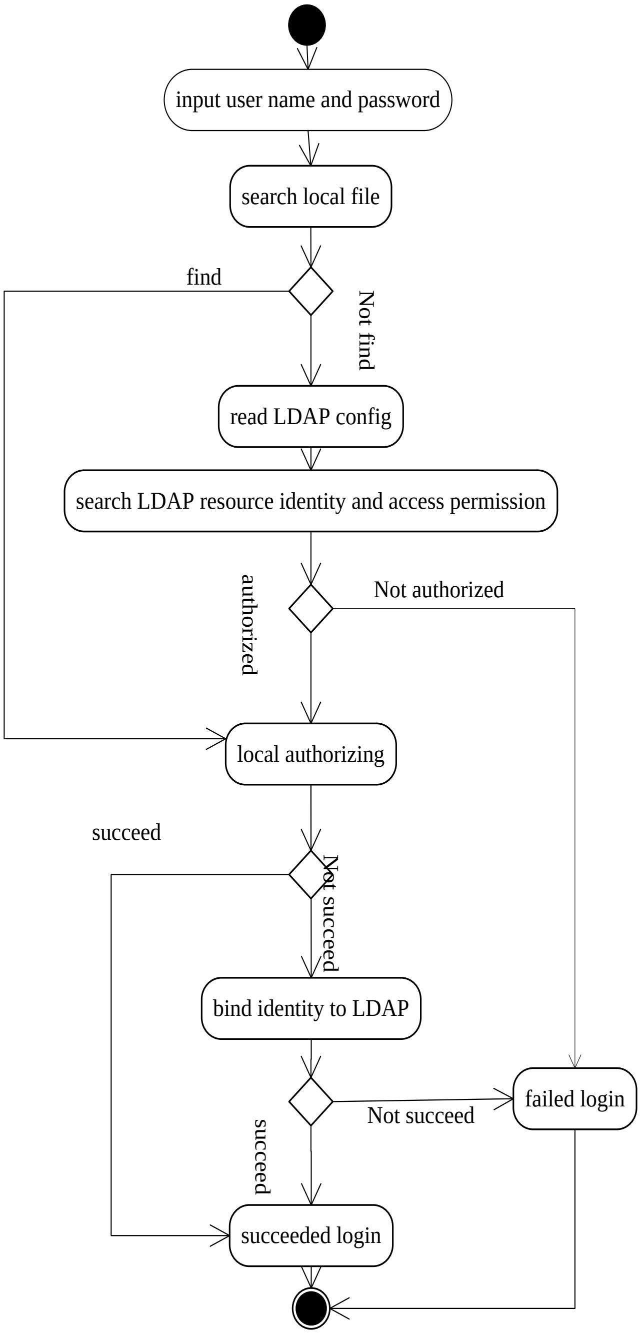


Figure 6. Active Chart of Identity Authorizing of Agent

After user inputs username and password, resource lookups local identity lib. If it fails to find the user and sends request to LDAP to query identity. When the resource authorizes the identity with local function firstly, and if failed, it sends the username and encrypted password to LDAP directory server to authorize identity. Here we implement the Linux, Windows, Application of J2EE, FTP server and Tomcat Server authorizing agents.

5. Experiment

5.1. Experiment Environment

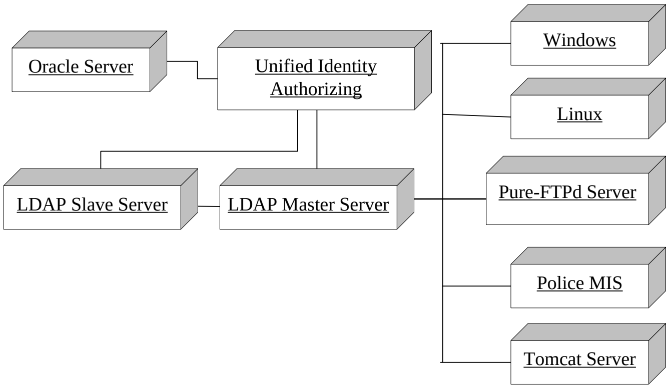


Figure 7. Experiment Environment of HUIP Model

Experiment environment of HUIP model is illustrated in Figure 7, Police MIS(management information system) is a J2EE application software developed by our lab. UIA(unified identity authorizing system) is an identity management system developed by our lab, and it can manage resources and users, and HUIP model is implemented and verified in UIA as a function module.

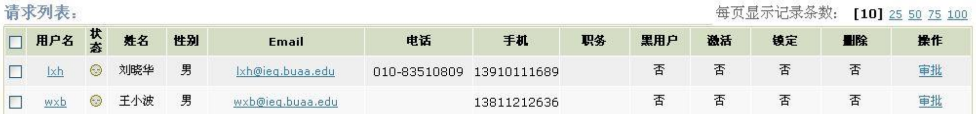
The user identity information is listed in Table 1. General user can create identity for new person of the organization, and set accessible resources and sent the request of providing identity. Administrator has permission of authorization of providing identity, password synchronization, and deleting identity besides of the right of general user.

Table 1. Users information in UIA

Users in UIA	Role	Authorized permission
user	general user	Providing identity
		Query identity
admin	administrator	Request authorization
		Password synchronization
		Providing identity
		Query identity
		Delete identity

5.2. Providing Identity

For example, user login UIA and will provide identity of a new person named *lxh*, *i.e.*, create a unified user information with exclusive ID, and will bind *lxh* to authorized resources. Here for verifying the HUIP model, *lxh* is permitted to all resources. HUIP module will map the identity based on unified identiy and accessible resources, and generate SPML request document to save in UIA server.

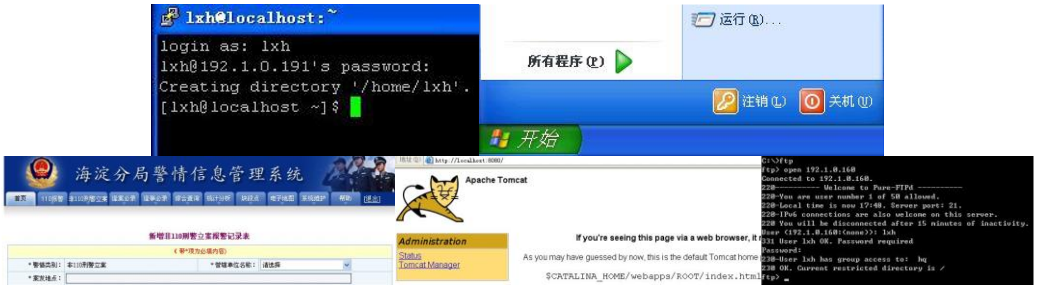


请求列表：　　每页显示记录条数： [10] 25 50 75 100

□	用户名	状态	姓名	性别	Email	电话	手机	职务	黑用户	激活	锁定	删除	操作
□	lxh	☺	刘晓华	男	lxh@ieq.buaa.edu	010-83510809	13910111689		否	否	否	否	审批
□	wxb	☺	王小波	男	wxb@ieq.buaa.edu		13811212636		否	否	否	否	审批

Figure 8. Identity Providing Authorization of Administrator

User *admin* authorizes the request of perviding identity after confirming the identity attribute and resources, and click the authorize button, HUIP module will search and commit the document of SPML. After success of providing identity, system will sent a email to inform *lxh*. User *lxh* will login all resources linux, windows, Police MIS, Tomcat, Pure-FTPd server with unified username and password.



lxh@localhost:~

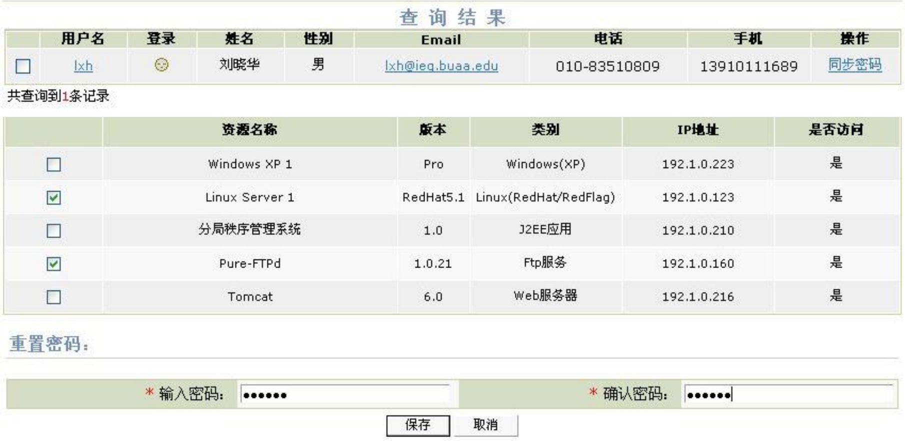
```
login as: lxh
lxh@192.1.0.191's password:
Creating directory '/home/lxh'.
[lxh@localhost ~]$
```

```
C:\>ftp
ftp> open 192.1.0.160
Connected to 192.1.0.160.
220---------- Welcome to Pure-FTPd ----------
220-You are user number 1 of 50 allowed.
220-Local time is now 17:40. Server port: 21.
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of inactivity.
User (192.1.0.160:(none)): lxh
331 User lxh OK. Password required
Password:
230-User lxh has group access to:  hq
230 OK. Current restricted directory is /
ftp> _
```

Figure 9. Unified Identity Providing Implementation

5.3. Password Synchronization

When *lxh* want to modify password of linux and pure-FTPd, he sends a request of password synchronization, *admin* will query *lxh* information, and select resources of password synchronization, input new password and save it, and HUIP module will commit password synchronization request which will be processed samely with login resources.



查询结果

□	用户名	登录	姓名	性别	Email	电话	手机	操作
□	lxh	☺	刘晓华	男	lxh@ieq.buaa.edu	010-83510809	13910111689	同步密码

共查询到1条记录

	资源名称	版本	类别	IP地址	是否访问
□	Windows XP 1	Pro	Windows(XP)	192.1.0.223	是
☑	Linux Server 1	RedHat5.1	Linux(RedHat/RedFlag)	192.1.0.123	是
□	分局秩序管理系统	1.0	J2EE应用	192.1.0.210	是
☑	Pure-FTPd	1.0.21	Ftp服务	192.1.0.160	是
□	Tomcat	6.0	Web服务器	192.1.0.216	是

重置密码：

* 输入密码： ••••••　　* 确认密码： ••••••

Figure 10. Password Synchronization Page

6. Conclusion

HUIP model achieves unified identity providing in organization and cross-organization for varies heterogeneous resources based on SPML, LDAP, and agent. Five kinds agents of resources including operating system, application, and standard network services are developed successfully. Identity mapping approach, system design and implement verify that the model is feasible and effective. As for future, other kind of identity agent, such as .NET, will be investigated and implemented to further verify the proposed model.

References

[1] J. Goode, "The importance of identity security", Computer Fraud and Security, vol. 2012, no.1, **(2012)**, pp. 5-7.

[2] J. Torres, M. Nogueira, G. Pujolle, "A survey on identity management for the future network", IEEE Communications Surveys and Tutorials, vol.15, no. 2,**(2013)**, pp.787-802.

[3] D. J. Lutz, B. Stiller, "A survey of payment approaches for identity federations in focus of the SAML technology", IEEE Communications Surveys and Tutorials, vol.15, no.4, **(2013)**, pp.1979-1999.

[4] C.G. Hocking, S.M. Furnell, N.L.Clarke, P.L. Reynolds, "Co-operative user identity verification using an Authentication Aura", Computers and Security, vol.39, no. PART B, **(2013)**, pp. 486-502.

[5] J. Vossaerta, J. Lapona, B. De Deckerb, and V. Naessensa, "User-centric identity management using trusted modules", Mathematical and Computer Modelling, vol. 57, no. 7-8, **(2013)**, pp. 1592-1605.

[6] M. Behan, O. Krejcar, "Open personal identity as a service", Advances in Intelligent Systems and Computing(Multimedia and Internet Systems: Theory and Practice), vol. 183,no. AISC,**(2013)**, pp. 199-207.

[7] K.A. Ajmath, P. V. Reddy, B. U. Rao, S.V.K. Varma, "Identity-based directed proxy ring signature scheme", Journal of Discrete Mathematical Sciences and Cryptography, vol.15, no. 2-3,**(2012)**, pp.181-192.

[8] A. Ng, P. Watters, S. Chen, "A consolidated process model for identity management", Information Resources Management Journal, vol. 25, no. 3, **(2012)**, pp.1-29.

[9] W.J. Luo, M. Xu, "Hierarchical identity-based key management in cloud computing", Journal of Convergence Information Technology, vol. 7, no. 20, **(2012)**, pp.343-350.

[10] A. Sarma, J. Girao, "Supporting trust and privacy with an identity-enabled architecture", Future Internet, vol. 4, no. 4, **(2012)**, pp. 1016-1025.

[11] C. Feher, Y. Elovici, R. Moskovitch, L. Rokach, A. Schclar, "User identity verification via mouse dynamics", Information Sciences, vol. 201, **(2012)**, pp.19-36.