The Design and Implementation of Single Sign-on Based on Hybrid Architecture

Zhigang Liang  
College of Computer Science & Engineering  
South China University of Technology  
Guangzhou, China  
Email: bumengxin@126.com

Yuhai Chen  
Asset Management IT Department  
HSBC Software Development (Guangdong) Limited  
Guangzhou, China  
Email: yuhai_chan@163.com

Abstract—For the purpose of solving the problems of user repeated logon from various kinds of Application which based on hybrid architecture and in different domains, single sign-on architecture is proposed. On the basis of analyzing the advantages and disadvantages of existing single sign-on models, combined with the key technology like Web Service, Applet and reverse proxy, two core problems such as single sign-on architecture mix B/S and C/S structure applications and cross-domain single sign-on are resolved. Meanwhile, the security and performance of this architecture are well protected since the reverse proxy and related encryption technology are adopted. The results show that this architecture is high performance and it is widely applicable, and it will be applied to practical application soon.

Index Terms—Single sign-on, web service, cross domain, reverse proxy, B/S, C/S

I. INTRODUCTION

With the information society, people enjoy the progress in the huge interests, but at the same time also faced the test of information security. With all system users need to log in the system increased, users need to set a lot of user names and passwords, which are confused easily, so it will increase the possibility of error. But most users use the same user name and password, this makes the authentication information is illegally intercepted and destroyed the possibility of increased, and security will be reduced accordingly. For managers, the more systems need more corresponding user databases and database privileges, these will increase management complexity. Single sign-on system is proposed a solution to solve the problem. Using single sign-on, we can establish a unified identity authentication system and a unified rights management system. It not only improve system efficiency and safety, but also can use user-friendly and to reduce the burden on administrators.

<table>
<thead>
<tr>
<th>SSO Achieve-Model</th>
<th>Action ability</th>
<th>Manageability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broker Model</td>
<td>The large transformation of the old system</td>
<td>Enable centralized management</td>
</tr>
<tr>
<td>Agent Model</td>
<td>Need to add a new agent for each of the old system, transplantation is relatively simple</td>
<td>Management more difficult to control</td>
</tr>
<tr>
<td>Agent and Broker Model</td>
<td>Simple, transformation of the old system with limited capacity</td>
<td>Enable centralized management</td>
</tr>
<tr>
<td>Gateway Model</td>
<td>Need to use a dedicated gateway to access various applications</td>
<td>Easy to manage, but databases between the different gateways need to be synchronized</td>
</tr>
<tr>
<td>Token Model</td>
<td>Implementation of relatively simple</td>
<td>Need to add new components and increase the management burden</td>
</tr>
</tbody>
</table>

Single sign-on refers to when the user needs to access a distributed environment which has different applications to provide the service, only sign on once in the environment,
no need for the user to re-sign on the various application systems[1]. Now there are many products and solutions to implement SSO, such as Passport of Microsoft, IBM Web Sphere Portal Server although these SSO products could do well in the function of single sign-on, but most of them are complex and inflexible. Currently, the typical models to achieve SSO include broker model, agent model, agent and broker model, gateway model and token model [2]. In table 1, it analyses these models can be implemented and manageability. Based on the above comparison, agent and broker model has the advantages both centralized management and revised less original application service procedure. So I decide to adopt agent and broker model as the basis for this model. In order to integrate information and applications well and with the B/S mode in-depth application software, there has been the concept of enterprise portal, offer a best way to solve this problem. Enterprise portal provides business users access information and applications, and complete or assist in a variety of interactive behavior of a single integrated access point. The appropriate system software portal provides a development, deployment and management of portal applications services. Enterprise information portal concerns portal, content management, data integration, single sign-on, and much other content.

II. SYSTEM CONSTRUCTION WHICH REGISTERS BASED ON THE WEB SERVICE MIX CONSTRUCTION SINGLE SIGN-ON

The system consists of multiple trust domains. Each trust domain has much B/S architecture of the application servers; in addition to B/S architecture of the application servers also included C/S architecture application servers. All the applications are bound together through a unified portal to achieve functionality of single sign-on. You can see that this architecture is based on the agent and the broker model. A unified agent portal is playing a broker role, and various applications are playing an agent role. The B/S architecture applications are installed on the Client side of SSO Agent, and the unified portal is installed on the Server side of SSO Agent. Between them is through these two Agents to interact. In addition, in Fig 1, the external provision of authentication server is LDAP authentication interface. Token authentication Web Service server provides the interfaces of single sign-on token of the additions, deletions, editions and queries. But the permission Web Service server provides the appropriate authority information system, to achieve unified management authority for accessing unified portal application system.

The system supports cross-domain access, that is, the domain D1 users can access the application domain D2, and the domain D2 users can access the application domain D1. At the same time, the system also supports the application of different structures between the single sign-on, that is, user after accessing the application A of the B/S structure access the application E of C/S structure without having to repeatedly enter user name and password, or user access the application A after the application E without re-enter login information.

The whole structure of Single Sign-on is as Fig 1 shown.

A. The login process

The whole single sign-on process is as Fig 2 shown:

Below is the process specific steps description:

1) User login in the client browser to access A application, SSO Client of A system intercept and redirect the URL to the landing page of Unified Portal System

2) Enter the user name and password, Unified Portal System submits to the authentication server for authentication. If the information is correct, Unified Portal System automatically generates, saves notes and the role of the user ID to a local, and calls the increase-note interface of Web Service to insert the information.

3) Unified Portal System returns a list of application resources pages to the user. The user clicks any one application system (e.g. A system). The SSO Client-side of A application system read the notes information and call the query-notes interface of Web Service. If it is consistent and within the time limit, it will get the role information of the user in A application system and log in A application system. At the same time, it will call the update-note interface of Note Certification Web Service to update the log-in time of this current note. Then call the interface of user rights Web Service to get this user’s permission information with corresponding application system.

© 2012 ACADEMY PUBLISHER
4) If user end to access A application system, exit and click on the link of B application system, system implementations will be are as the same as steps (3).
5) If user complete all the required access-applications and need to do the log-off operation, it will mainly call the deletion-note interface to destroy the corresponding note information.

The following is this system process of achieving cross-domain access:

1) User log in Unified Portal system successfully.
2) User accesses A application system within the trusted domain D1, complete the access and then exit this application.
3) User clicks the URL of B application system within trusted domain D2 of the resources list of Unified Portal.
4) SSO Client of B application intercepts the request, gets the note behind URL, and calls the query-note interface of Web Service.
5) Query interface of Web Service gets back the legal information of this note to the SSO Client.
6) SSO Client redirect to B application system, the user access B application.

**C. The Solution of Single Sign-on between B/S and C/S Structures**

As we know, the implementation principles of applications are quite different between B/S and C/S structures. In this system, the applications of B/S structure can be accessed through by clicking URL of the application-resources-list page of Unified Portal. Since the browser security restrictions, the page does not allow users to directly call the local exe files, so need to adopt an indirect way to call C / S architecture applications. This article uses the way of Applet to call local exe files, the implementations as below:

For all C/S structure, create a common Agent. This Agent's role is an interceptor, which means it need browsers to access after the C/S structure joined up Unified Portal system. (Please note that: Since the original B/S architecture and C/S structure is not using the same authentication method. For the C/S application access to the unified portal framework to achieve single sign-on system, the need for a unified authentication management, and in order to change the amount of compression to a minimum. Implementation of this system is to create a needless user name and password authentication code for all applications which are accessed a unified portal, and land on the unified portal system certified landing page. When a user uses browser to log into the unified portal system successfully and then can access any application, including the B/S architecture and C/S structure of the application. To be ensure the security of C/S application framework, when the user clicks directly to the desktop shortcut to open applications still using the original authentication.)
Applications of C/S architecture are all using the same Applet of URL. The received parameters of this common Applet include bills, application name, unified login-name and password. When a user does not do the login operation before, the first visit a C/S application will be intercepted to the login-page of Unified Portal system for sign-on. If a user logged in before, when visiting a C/S application, this Agent will call the interface of Web Service note-validation. When a user does not do the login operation and the handle of login button through by JNI.

To validate the note which was transferred. If the validation is successful, Applet object will be downloaded to the user's local to implement. In order to transform the original applications as little as possible, the method of this article is to open the login window of the corresponding application through by Applet. Below are the codes:

```java
public void OpenExe(String appName){
    Runtime rn=Runtime.getRuntime();
    Process p=null;
    p=rn.exec("c:\" + appName + ".exe");
}
```

After opening the log-in window of the application, the operation steps of this Applet as follows:

1) Applet needs to call the bottom API of windows to get the user-name of login window, password-input box and the handle of login button through by JNI.

2) Locate the user-name-input box to send unified login name. Locate password-input box to send the password. (Password information is arbitrary and in order to distinguish it from the user clicks on a shortcut directly landing system, also need to send a code that uses a unified portal access without a password authentication system.) Locate the login button to send the click event.

3) At last, Applet will minimize the IE window, the related windows of applications will be placed to the forefront.

These are the implementation process of C/S architecture application single sign-on. The application codes which have not been changed at all before will join up the Unified Portal system using a loosely coupled way. Need to explain that, due to the Applet JVM security restrictions, cause Applet can not directly call the user's System32 directory of local native windows dll. Now the method is first to start to use C or C ++ to write the class which got the corresponding input box and button of the login window, and generate a JNIWindowUtil.dll file (JNIWindowUtil is a user-defined dll's name). And it is to place the dll in the same directory with the Applet. When the Applet is downloaded to the client side, dll is also downloaded to the user's System32 directory of local at the same time. Applet process also needs to execute statement: System.loadLibrary("JNIWindowUtil"). After completing these above steps, it can really use JNI in Applet internal to achieve the corresponding functions.

D. Authentication server

The old system user authentication information is usually stored in a database, but this architecture used LDAP to store user information. LDAP, short for Lightweight Directory Access Protocol, is the standard directory access protocol based on a simplified form. It also defines the way data organization; it is based on TCP/IP protocol of the de facto standard directory service, and has distributed information access and data manipulation functions. LDAP uses distributed directory information tree structure. It can organize and manage various users' information effectively and provide safe and efficient directory access. Compared with the database, LDAP is the application for reading operation more than writing operation, and database is known to support a large number of writing operations. LDAP supports a relatively simple transaction, but the database is designed to handle a large number of various transactions. When the query in Cross-domain data is mainly read data, modify the frequency is very low. When Cross-domain access to the transaction, it does not require a large load, so in comparison with the database, LDAP is the ideal choice. It is more effective and simple. This framework is applied to a large bank, the bank's systems can belong to different regions, and use of personnel may come from different geographies. In order to achieve distributed management, the use of three-level management, respectively named the Bank headquarter, Provincial and City branches of the three levels of branches, as shown in Fig 4:

- **Bank headquarter**
  - LDAP Server of Bank headquarter
  - Provincial Branch 1
    - LDAP Server of Provincial branch 1
      - Copy
      - Quote
  - Provincial Branch N
    - LDAP Server of Provincial branch N
      - Copy
      - Quote
  - City Branch 1
    - LDAP Server of Master of City 1
      - Copy
      - Quote
    - LDAP Server of Slave of City 1
      - Copy
      - Quote
    - LDAP Server of Master of City 1
      - Copy
      - Quote
  - City Branch N
    - LDAP Server of Master of City N
      - Copy
      - Quote
    - LDAP Server of Slave of City N
      - Copy
      - Quote

**Figure 4: LDAP Authentication Structure**

Directory replication and directory reference is the most important technology in LDAP protocol. It can be seen from the figure, Provincial and City branches of the LDAP server branch data are copied from the floor, but not a simple copy of all information, just copy the relevant data with their own information. Because for a particular application system, its users are mostly belong to the same
region, so that implementation can greatly simplify the management of directory services and to improve the efficiency of information retrieval. When a user outside the region to use this system, because of its user information in the region can not retrieve LDAP server, you need to refer to the other regions of the LDAP server to query, and therefore requires a way to use up the reference queries, first Provincial branches of the server search, without further reference to the Bank headquarters of the server up until the search to the appropriate user information.

The management of the regional City branch, using the LDAP directory replication model of Single Master/Multi Slave. When a directory user queries the directory information, Master LDAP Server and Slave LDAP Server (Slave server can have more than one) can provide services to the directory, depending on the directory user makes a request to which the directory server. When the user requests the directory update directory information, in order to ensure the Master LDAP Server and Slave LDAP Server in the same directory information content, the need for replication of directory information, this is achieved through the LDAP Replica server data synchronization. Using directory replication, when the directory number of users increases or the need to improve system performance, only simply add Slave LDAP server to the system and then can immediately effective in improving system performance, and the whole directory service system can have a good load balancing.

E. Permissions Web Server

Access Control technology began in the computer age of providing shared data. Previously, the way people use computers is mainly to submit the run-code written by user or run the user profile data. Users do not have much data sharing, and do not exist to control access to data. When computer comes into user's shared data, the subject of access control is nature to put on the desktop. Currently, the widely used access control models is using or reference to the early nineties of last century the rise of role-based access control model (Role-Based Access Control - RBAC). RBAC model's success is that it is inserted the "role" concept between the subject and object, decouples effectively between subject and the corresponding object (permission), and well adapts to the subject and object associated with the instability. RBAC model includes four basic elements, namely the user (User - U), roles (Roles - R), session (Session - S) and permission (Permission - P), also in the derived model also includes constraints (Constraints - C). The basic idea is to assign access rights to roles, and then the roles are assigned to users. In one session, users can gain the access rights through roles. The relationship between the elements: a user can have multiple roles, a role can be granted to multiple users; a role can have multiple permissions, a permission can be granted multiple roles; user can have multiple conversations, but a conversation is only to bind a user; a conversation can have multiple roles, a role can share to multiple conversations at the same time; Constraints are that act on specific constraints on these relationships. As shown in Fig 5:

Figure 5: RBAC Basic Model

This system is to use this very sophisticated permission access control model.

Rights management, not only protects the safety of system, but also facilitates management. Currently most using the manner of code reuse and database structure reuse, rights management module is integrated into business systems. Such a framework has the following shortcomings.

1) Once the permissions system has been modified, the maintenance costs will be very high. This is the general shortcoming of using code reuse and database structure reuse. Once revised, we will have to update the code in all business system and database structure, and also to ensure that existing data can smooth the transition. Some processes may require manual intervention, which is a "painful" thing for the developers and maintenance personnel.

2) Did not facilitate management of Permission data. Need to enter permission management module of various business systems to manage the corresponding rights. It is complex operation, and not intuitive.

3) For different architectures, different software operating environment, we must develop and maintain different permissions system. For example, B/S and C/S architecture system must each develop their own rights management system.

This paper argues that most common function of the permission system can abstracted from business systems to form an independent system - "unified rights system". Business system only retains the rights inquiries, read common data system and the control rights function of this system specific fine degree (such as menus, buttons, links and so on). As shown Fig 1. How to achieve a unified rights management? This paper argues that there are two implementations, one way is to use Web services to provide rights data; the other is using Mobile Agent to provided permissions data. However, the second one run, maintenance costs are higher, and implement is more difficulty than Web services. So this architecture using Web services to provide authority data of the various systems in
a unified way. Business system using Web services client interface to query data and obtain system privileges to share data. The client is just a port, and specific implementation code is placed in "unified rights system". These client interfaces introduced to the business system by package. If we keep the client interfaces unchanged, modify and upgrade of the unified authority system will not affect the business system. Users and permissions through Web pages of "unified rights system" to unify management and to achieve the user's single sign-on. The biggest advantage of Web services is the integration of data between heterogeneous systems. This breaks the restrictions of B/S, C/S structure; there is no difference between Windows and Linux platform.

III. System Security Analysis

1) The interception of user name and password. The system for authentication of the user login and send the user name and password to Applet objects are used SSL protocol. And make sure that information during transmission confidentiality and integrity. Meanwhile, due to the key which is hard to get and time limited, so it can effectively prevent that intermediary attack to the transmission of information.

2) Replay attack. Many systems will use the ways of time stamp to avoid duplication attacks. However, this approach requires the computer clocks of communication parties to be synchronization. But it is difficult to achieve, while also appears the following situation: the two sides’ clocks which are connecting with each other, if they are out of synchronization occasionally, the correct information may be mistaken to discard for replay information, but the incorrect replay information may be as the latest one to receive. Base on the above, this system needs a simple method F of an appointment between query interfaces of Web Service provided and SSO Client of each application system or Agent. This system’s parameter value is a random string X. The whole process of bill validation as shown in Fig 6:

   a) When the user accesses to application system A, the SSO Client of system A intercept and call the query interface of Web Service provided, and the input parameters are a random string X and the corresponding note.

   b) Web Service server receives system A’s call, intercepts note to compare with the note’s information of Session queue. If the queue contains the note, it will return the value of F(X) for showing validation is successful. If not, it will return ‘failed’ for showing validation is failed.

   c) SSO Client of the application A receives the return information of Web Service server, and then compares the return value with F(X) of this system. If the two are the same, it will redirect to system A, otherwise it will not be allowed to visit.

The random string is different, which each interact with Web Service server. So you can limit replay attacks very well.

3) Use reverse proxy technology. Reverse proxy technology is a substitute, which is a reverse proxy server as to N identical application servers. When external access to this application, it just knows the reverse proxy server and can not see the back multiple application servers. This improves the security of this application system.

Through the above analysis, this system can provide users with a good safety Web environment.

IV. System Performance Analyzes

First, this system in addition to use SSL encryption in the transmission of user name and password, the interactions of between other servers and between user and servers are based on HTTP protocol to transmit. SSL encryption and decryption process requires a lot of system cost, severely reduces the performance of the machine, so we should not be use this protocol to transmit data too much. Since the data which need to encrypt is small, only a user ID value (note), so the performance of using MD5 to encrypt is quite satisfactory.

Second, when user accesses any application system of each domain, they will be redirected to Unified Portal system for identity authentication, or directed to Web Service server for note validation. User need to sign on the system only when he is certification first time. When the visitor volume is larger, the user switch to the new application system will easily handle an interruption, which is single sign-failure phenomenon [7]. This phenomenon has two reasons, one is the server load is too large, the other one is network bandwidth is not enough [8]. Among them, the method which is resolved the server load is too large is to use server cluster. Cluster is made up of multiple servers. As a unified resource, it provides a single system service to external. In this system, except for using reverse proxy technology to improve the security of accessing the applications, the more important is capability which can
help to implement cluster technology of load balancing. The whole structure of reverse proxy is shown in Fig 7:

![Reverse Proxy Structure](image)

**Figure 7: Reverse Proxy Structure**

Fig 7, reverse proxy server R provides the corresponding interface to implement the algorithm of load balancing except for providing cache for the behind A1, A2 and A3 application. That is, it can consider the arrival request to distribute to the server which has the best performance through by scanning the conditions of CPU, memory and I/O of A1, A2, A3 server. By LoadRunner8.1, the use of reverse proxy system before and after was related to stress testing. The test results are shown in Fig 8:

![Average Access Time of Concurrent Users Access](image)

**Figure 8: The system stress testing result**

It can be seen from Figure 8, at the beginning, when the number of concurrent users is not large, use the reverse proxy and out of use proxy is similar. But with the gradual increase of concurrent users, the performance difference between the two is more and more evident. To 100 concurrent users to access, the system response time of using the reverse proxy is almost twice as fast as the one out of use proxy.

System Web Service server needs to store the information of note, so using Web Service server cluster to pay attention to this problem: the different Servers of cluster use different JVM, so an object of JVM can not be accessed by other JVM directly. For this problem, there are two methods to resolve:

1) Put the object in Session, and then configure cluster to the copy model of Session.

2) Use Memcache, put the object in Memcache, and then all Server get this object from Memcache. To be equivalent to open a public memory area, which everyone can access.

Any more, business system requires get rights information data through the Web services frequently. This performance of the system put forward higher requirements. The system has been taken two measures to improve performance:

1) It receives a request by using time-sharing patterns of authority data server. After that, if always be calculated in real-time data, it will not certainly respond in time as the server limited resources. This will cause the system to slow down. A "time-sharing patterns of authority data" can solve this problem. When the system data changes (such as a new operation is authorized to the role, etc), the system automatically calculates the affected user, and then re-calculate the relevant authority data, save to the specified field of database. When the business system requests data, only run "to read the database designated field corresponding to the specified data" such a simple action, you can greatly speed up the system response speed.

2) Designed the cache structure to rely solely on time-sharing model is not enough to improve response time. After all, access to the database is relatively resource intensive. This paper designs cache structure which is based on the memory to further improve the speed, to achieve the following functions: when the system starts, it will reside the public data which is often used to cache (such as organizational structure, role information, etc.), if the change in the operation of this part of the data, then also update the data in the cache. It is to adjust the priority of the data according to the frequency of access the cache object, and regularly to remove expired objects. It can improve the cache hit rate through the trade-off algorithm of cache object which is optimized.

V. CONCLUSION

This text develops a single sign-on. Compare with the traditional sign-on model, this text presents a new solution which is single sign-on of hybrid framework of being based on Web Service. This program can be applied to distributed environment of being based on multi-trust-domain, and also applied to the case which has both B/S structure and C/S structure applications. It optimizes the unified management of the administrator faces to user information, facilitate the user to access to each application resource, improve the efficiency of user use application resource, enhance the security of user access each site, and provide the basis for integrating more sub-application systems.

VI. ACKNOWLEDGMENT

This research project has been supported by Pansky Science and Technology Co., LTD
REFERENCES


Zhigang Liang, 1985-08, was born in Maoming City, Guangdong Province. Currently a graduate student at Computer Science and Engineering of South China University of Technology, major field of study is computer system structure and computer application.

He ever had an internship in HSBC Software Development (Guangdong) Limited and Pansky Science and Technology Co., LTD, job title is Java software engineer.

Yuhai Chen, was born in Maoming City, Guangdong Province on 30th October 1984. He graduated from Guangdong Polytechnic Normal University in 2008, major in Software engineering, has a Bachelor of Engineering. Major Research Interests include Software Development and Technology, Systems Analysis and Database System.

He ever worked at Complete Solution (Guangdong) Limited in Guangzhou City from Jun. 2008 to Nov. 2009, job title is JAVA SOFTWARE ENGINEER, and primary responsibility are System Development and Java programmer. From Dec. 2009, he joins HSBC Software Development (Guangdong) Limited, job title is SOFTWARE ENGINEER, and primary responsibility are System Development and AS400 programmer. He is familiar with AS400 (RPG, CL, SQL), C++, Java, Struts, Spring, Hibernate, HTML, JavaScript, CSS, XML, MySQL and Oracle Database.