ISSN 2616-5775 Vol. 6, Issue 7: 8-14, DOI: 10.25236/AJCIS.2023.060702

# Research and Implementation of High Available Digital Campus Unified Identity Authentication System Based on LDAP

**Zhiyuan Wu[1,a], Yinqian Cheng[2,b,*]**

[1]*School of Information Engineering, China University of Geosciences (Beijing), Beijing, China*
[2]*Information Network Center, China University of Geosciences (Beijing), Beijing, China*
[a]*wuzy@cugb.edu.cn,* [b]*chengyq@cugb.edu.cn*
[*]*Corresponding author*

***Abstract:*** *In this paper, some problems in the previous use of the unified identity authentication system in campus are first analysed. Combined with the actual application of the campus, a high-availability unified identity authentication architecture scheme based on LDAP is proposed. This solution applies ETL technology to data synchronization between relational databases and LDAP, and realizes a unified identity authentication system for digital campus through customized development of the single sign-on CAS system. Due to the deployment of LDAP and CAS single sign-on systems in cluster mode, the high availability of unified identity authentication service has been effectively improved.*

***Keywords:*** *Digital Campus, Unified Identity Authentication, SSO, ETL, High Availability*

## 1. Introduction

In traditional campus, application system is relatively independent. The function of identity authentication is repeatedly designed and implemented, and is scattered in each application system, which easily leads to the phenomenon of multiple accounts and multiple passwords to users. This brings great inconvenience to users and managers, and also poses security risks [1]. As an important part of the digital campus platform, the unified identity authentication system centrally stores the unified identity account information of all teachers and students in campus. The single sign-on module which is a part of unified identity authentication system is the authentication entrance of the entire digital campus. Once the user is authenticated by single sign-on, he can be authorized to access the services provided by all application systems in the digital campus, which can effectively solve the problems of decentralized login and decentralized management of accounts in different application systems. Therefore, building an efficient, stable, safe and reliable unified identity authentication system is of great significance to the entire digital campus platform.

## 2. Situation and Issues

Many colleges and universities adopted the solution of relational database plus single sign-on CAS (Central Authentication Service) [2] to build unified identity authentication system. In that unified identity authentication system, the backend uses a relational database to uniformly store the user identity information of all teachers and students in the campus, and the frontend single sign-on system CAS unifies and integrates a large number of application systems. Through the unified identity authentication system, teacher or student only needs one password to log in and can access most application systems of the campus, which can greatly facilitate the use of the digital campus platform. However, as the campus continues to put new application system into use, some problems have gradually emerged with the unified identity authentication system of this solution.

### *2.1. System integration issue*

Although the CAS single sign-on system can realize integration with most application systems, it is not suitable for all systems. For some product-level systems, such as VPN system and mail system, it is impossible to customize and modify the system source code, and it is difficult to achieve integration with CAS.

ISSN 2616-5775 Vol. 6, Issue 7: 8-14, DOI: 10.25236/AJCIS.2023.060702

### *2.2. Data synchronization issue*

Many application systems need to obtain the campus organization and user identity account information synchronously from the unified identity authentication system. These application systems mostly support configurable LDAP interfaces, but some do not support obtaining the data from relational database, or some need source code modified to support the function. In this case, there are problems with data synchronization between the application system and the identity authentication system.

### *2.3. Single point of failure issue*

The CAS single sign-on system is used as the login entrance of the entire digital campus platform. Once there is a downtime or operation failure, all digital campus application systems integrated with CAS will not be able to be logged in normally, which will cause a relatively large negative impact on the daily business of the entire digital campus platform.

In order to solve the above problems, this paper redesigns the above architecture of the digital campus unified identity authentication system. By changing the relational database to the current mainstream LDAP, and applying ETL technology to the data synchronization between the relational database and LDAP. After applying high availability mode to deploy the customized CAS single sign-on system, the highly available digital campus unified identity authentication system based on LDAP can be realized.

## 3. Unified identity authentication related technologies

The so-called identity authentication is the process of judging whether a user is a legitimate user. The commonly used authentication method is that the system judges whether the user's account and password are consistent with those stored in the system, thus judges whether the user's identity is correct. Complex authentication uses some encryption algorithms and protocols to require the user to present more information, such as a private key, to prove their identity. Unified identity authentication requires unity on the basis of identity authentication. Generally, an authoritative identity authentication center in a distributed environment needs to be created, and the authentication center will take over the functions of the original identity authentication scattered on all systems. The user information database of the authentication center is responsible for uniformly verifying the authenticity of the user identities. The security issues in the process of storing and exchanging information involving user identities are all guaranteed by the security protocols and policies provided by the unified identity authentication center [3].

### *3.1. Single sign-on*

Single Sign-On, referred to as SSO [4], is the core function of unified identity authentication. The definition of SSO is that in multiple application systems, users only need to log in once to access all mutually trusted application systems. At present, the open source enterprise-level single sign-on solution named CAS has been widely used. CAS is an open source project initiated by Yale University, and is designed to provide a reliable single sign-on method for web application systems. CAS officially became a JA-SIG project in December 2004. It has the following characteristics: an open source enterprise-level single sign-on solution; CAS server is a web application that needs to be deployed independently; CAS client supports a lot of clients (referred to here as Each Web application in the single sign-on system), including Perl, Java, .Net, u Portal, Apache, Ruby, PHP, etc.[5] The two parts of CAS, CAS server and CAS client, need to be deployed independently. CAS server is mainly responsible for user authentication, and CAS client is responsible for processing access requests to protected resources on the client side. When login is required, the request will be redirected to CAS Server.

### *3.2. LDAP protocol*

The Lightweight Directory Access Protocol (LDAP), developed by the University of Michigan, is a widely accepted directory access method [6] and an open industry standard. LDAP is a special database, in which data is organized in a directory, and the directory is composed of objects, which have attribute information. The attributes of the directory are essentially a key-value pair, which is a way to store data in the directory. LDAP writes data slowly, and the modification operation is only implemented using a simple locking mechanism. It does not support complex transactions, and there is no transaction rollback mechanism [7]. Therefore, the main task of LDAP is not data storage and operation, and is not suitable for

ISSN 2616-5775 Vol. 6, Issue 7: 8-14, DOI: 10.25236/AJCIS.2023.060702

storing a large amount of data that requires operations other than query (such as adding, deleting, modifying, etc.). But the reading performance of LDAP is stronger than its writing performance, and the query speed is much faster than that of general relational databases [8]. LDAP is based on X. 500 standard security protocol, providing access control by the Simple Security Attestation Layer (SASL) protocol, and using the SSL/TLS authentication mechanism to protect data integrity and privacy [9]. Therefore, it is very efficient and safe to use LDAP for unified identity authentication of read-intensive operations in a network environment.

LDAP has a directory information tree (Directory Information Tree, DIT), which stores objects (entries) in a tree structure [10]. The quality of its design is directly related to the overall query performance of the authentication system. Therefore, it is necessary to try to reduce the structural level of the directory information tree. This is because the less the level, the shorter the identification name of the object (entry), and the less affection by other factors. When there is an organizational adjustment in a certain department, the fewer the structural levels, the smaller the impact on other departments or branches [11].

## 4. System design

### *4.1. System architecture design*

In the digital campus platform, the basic user identity data comes from the public database, which is a relational database. In this paper, the user identity information database of the unified identity authentication system uses the current popular open source LDAP directory database – OpenLDAP. Since the user identity data in LDAP must be obtained synchronously from the public database, it is necessary to synchronize the data regularly from the relational database to LDAP in order to achieve the accuracy and immediacy of the user identity information in LDAP. The single sign-on module of the unified identity authentication system still adopts the relatively mature scheme with CAS. Integrated with CAS, other application systems of the campus can realize the function of single sign-on. Considering data security and single point of failure, the LDAP directory server is deployed in dual-node mode, and the dual-node cluster mode is adopted in the deployment of the CAS single sign-on system, which can be helped to achieve high availability for the unified identity authentication system. The system architecture design is shown in figure 1 below.

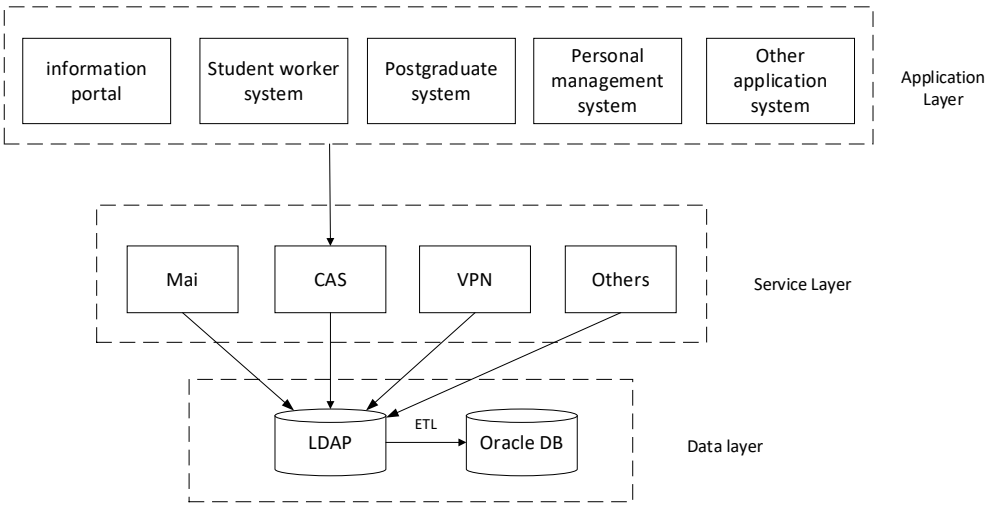


*Figure 1: The architecture of digital campus unified identity authentication system.*

As shown in the figure above, the entire system architecture is divided into three layers, which are data layer, service layer and application layer from bottom to top.

▪ The data layer mainly includes a LDAP directory database, which provides data sources for user identity information authentication and school organizations to the upper layer. In this layer, the user identity data and organizational data in the public database, such as Oracle DB, should be synchronized to LDAP through ETL tools.

▪ The service layer calls the LDAP authentication interface from downwards, and provides the single sign-on integration with the campus application system to upwards. It mainly includes CAS single sign-on system, mail system, and VPN system, etc. Due to the existence of highly productization systems such

ISSN 2616-5775 Vol. 6, Issue 7: 8-14, DOI: 10.25236/AJCIS.2023.060702

as VPN system and mail system of the digital campus, it is difficult to modify the source code of the systems, and thus cannot realize the integration with CAS through customized development. For this kind of application system, generally the LDAP configuration is provided. Through the LDAP configuration, the LDAP interface from the data layer is used to call to authenticate the user account and password. At the same time, the campus organization data can also be synchronized through the way. Thus, the role of unified identity authentication by logging in with the same set of account and password is achieved.

▪ The application layer includes various application systems of the digital campus platform. These application systems realize the single sign-on integration with the CAS of the service layer, including most of the campus's application systems such as information portal, student work system, postgraduate school system, etc.

### *4.2. LDAP directory design*

The directory in LDAP stores data in a tree-like hierarchical structure [12]. The basic information unit of the directory is an entry. An entry can be composed of multiple attributes, and each attribute identifies a feature of an object. An attribute has a type and one or more values. The attribute type describes the information contained in this attribute, while the attribute value contains the actual data. Each entry determines its position in the tree structure through a unique identifier DN (Distinguished Name). Each DN is composed of several elements, called relative DN (Relative Distinguished Name) [13]. According to the identity information structure of the campus user, by classifying user-related data, the LDAP tree organization structure is obtained as shown in the figure 2 below.

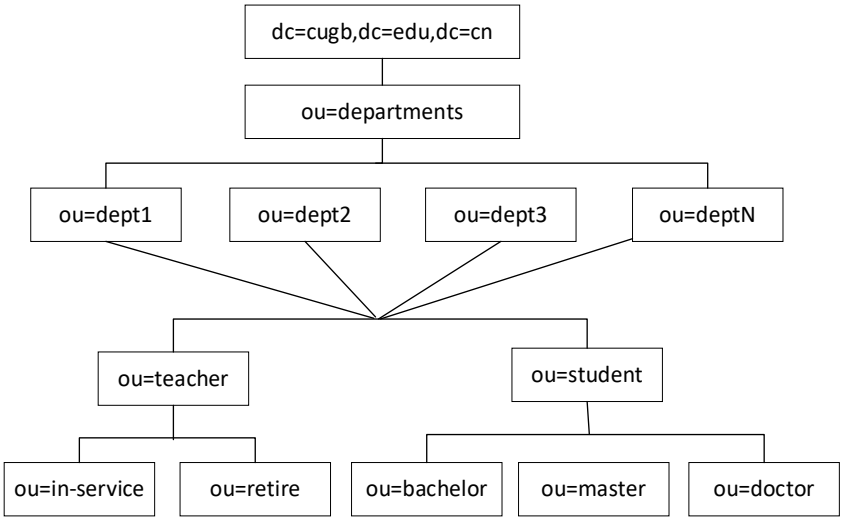


*Figure 2: Customized LDAP directory structure diagram.*

Since the entries of the LDAP directory tree are determined by object classes and attributes, the construction of unified LDAP directory service needs object classes and attributes to be designed based on the user basic data in the public database. In the LDAP information model, a schema is a collection of object classes grouped according to the principle of similarity. Attribute types and object classes are defined in the schema. The RFC2758 document defines an LDAP object class named InetOrgPerson and a set of attributes available for this object class, which usually used in directory services. According to the practical application of the unified identity authentication system, the information represented by the existing attributes in the InetOrgPerson object class cannot meet the usage requirement, so the system introduces a custom schema, which is extended on the basis of the person object class, to meet the storage requirement of digital campus users identity information.

## 5. System implementation and key technologies

### *5.1. Using ETL technology to realize the synchronization function from database to LDAP*

ETL [14] is the abbreviation of Extract, Transform, and Load, which represents the technical implementation process from data extraction to loading. It is an integration tool used to integrate data from heterogeneous and multiple data sources. In this paper, the open source ETL tool named Kettle is

ISSN 2616-5775 Vol. 6, Issue 7: 8-14, DOI: 10.25236/AJCIS.2023.060702

used to realize the synchronization of user identity data from the digital campus public database Oracle to OpenLDAP. By designing the conversion process in Kettle, data mapping between heterogeneous data sources Oracle and OpenLDAP can be established. The customized transform process designed in Kettle is shown in Figure 3 as below.

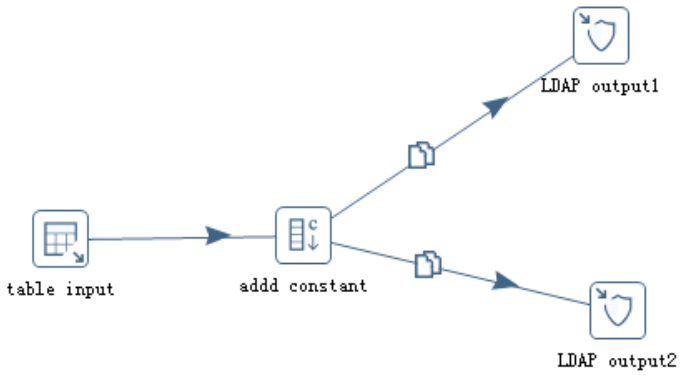


*Figure 3: Customized transform process designed in Kettle*

In the Figure 3 above, the "table input" is used to extract user identity data from the user basic information table in oracle, and the "add constant" is used to add constants required by LDAP custom schema. After the user identity data has been extracted and converted through the above steps, it will be distributed to two peer LDAP node servers which have identical schema. Since the directory data in the two target LDAP node servers are exactly the same, single point of failure can be avoided, and load balancing from calling the LDAP interface can also be realized.

### *5.2. CAS server customization*

#### *5.2.1. CAS server configuration and development*

Since the default configuration and function of the native CAS system cannot meet our needs, it is necessary to carry out customized development based on the source code of the public CAS system. The custom development of CAS Server mainly involves two aspects: environment construction and LDAP configuration. Among them, CAS server uses the officially recommended overlay which is a configuration template provided by the system to configure the server, and the default configuration file can be overwritten by customized template file which is modified from the original template (cas-overlay-template) according to requirements. An https security certificate needs to be configured when enabling the single sign-on function of the CAS server. Since the CAS server needs be deployed in the Tomcat application server, it is also necessary to add support for the HTTPS protocol in the configuration file which is named server.xml in Tomcat.

As the CAS server provides relatively simple user name and password authentication by default, its configuration needs to be modified to support LDAP-based user name and password verification. When verifying user identity information, CAS server needs to send relevant user information to the OpenLDAP. For this purpose, it is necessary to create a connection between CAS server and Open LDAP, which can be achieved by modifying deployerConfigContext.xml file in the WEB- INF directory.

#### *5.2.2. CAS login interface customization*

As the default login interface of the CAS system cannot meet the style of digital campus, it is necessary to customize the login interface. The main steps are as follows:

- Copy resource files. According to the overall style of the digital campus platform, design a set of login pages suitable for our campus. Then copy css, image, js to the source code directory of the CAS system. The target directories are "src/main/webapp/css", "src/main/webapp/images", and "src/main/we-bapp/js".

- Modify the casLoginView.jsp file. The casLoginView.jsp file is located in the directory that is "src/main/webapp/WEB-INF/view/jsp/default/ui". The modification method is that directly copy the original spring form component of the template file, and replace the form element according to our customized requirement.

ISSN 2616-5775 Vol. 6, Issue 7: 8-14, DOI: 10.25236/AJCIS.2023.060702

### *5.3. The implementation of high availability*

As the CAS single sign-on system is the entrance of the digital campus platform, in order to avoid single point of failure and meet a lot of concurrent requests from teachers and students in the whole campus, the performance of the CAS service needs to be considered. In this paper, cluster is used to deploy CAS system, and Redis server is used to store shared sessions and tickets which are necessary for the CAS system. Through these methods, high availability of the CAS system can be achieved.

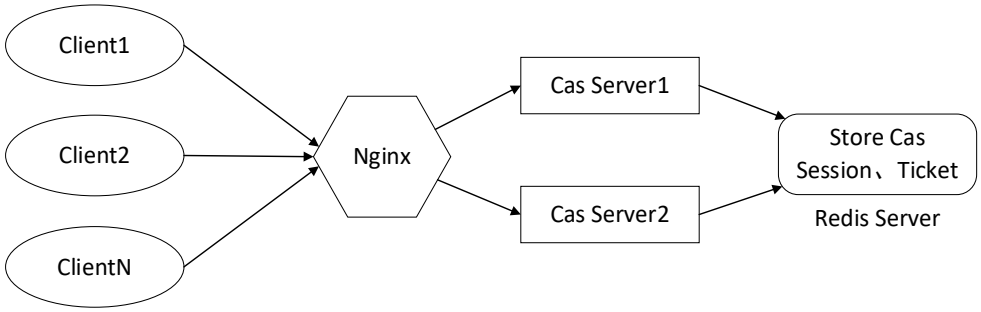


*Figure 4: CAS Server deployed using cluster mode*

The deployment of the CAS system using cluster mode is shown in the figure 4 above. The cluster consists of one Nginx node and two CAS server nodes. With the function of load balancing, Nginx is responsible for distributing user login requests to the two different CAS server nodes. Requests from client users are randomly assigned to one CAS server node. When login randomly in CAS server node, there will be a session sharing problem. At the same time, the tickets allocated by TGT on different CAS nodes are also different, so it is necessary to solve the problem of sharing sessions and tickets for different nodes. At present, there is a relatively popular and simple way to centrally store the shared session and ticket of the CAS server in the cluster using Redis. In this way, unified management of session and ticket can be realized. The specific deployment steps with Redis are as follows:

- Use Redis to configure session persistence, add “cas-server-webapp-session-redis” dependency in the maven pom file, and add related configuration in application.properties.
- Use Redis to configure ticket persistence, add “cas-server-support-redis-ticket-registry” dependency in the maven pom file, add Redis related configuration in application.properties.
- Configure load balancing in Nginx. The upstream parameter must be configured in Nginx. By default, the server in the group is polled to process the request according to the round-robin scheduling strategy, and its configuration needs to be modified to the ip_hash method. Since ip_hash is assigned to each request according to the hash result of the source ip of the request, each time the client accesses a fixed CAS server. With this configuration, the session sharing problem caused by the difference of the CAS node key can be solved.

## 6. Conclusions

With the continuous construction of the digital campus platform, highly available unified identity authentication system becomes more very critical. Based on some problems in the use of the digital campus unified identity system before, this paper designs and implements a high-availability unified identity authentication system based on LDAP. The solution uses the open source CAS as the single sign-on support system, selects the open source Open LDAP as the authentication directory server, and introduces ETL technology to realize the synchronization of the user directory data. By deploying LDAP and CAS systems in cluster mode, the high availability of the unified identity authentication service is effectively guaranteed. In this paper, the design and construction method of the high-availability unified identity authentication system based on LDAP can provide a reference technical solution for other campuses.

## References

*[1] Huang Xiufang, Wang Hai. Integrated implementation scheme of college digital campus unified identity authentication based on LDAP[J]. Journal of Jiangsu University of Science and Technology (Natural Science Edition), 2015, 6:580-584.*

ISSN 2616-5775 Vol. 6, Issue 7: 8-14, DOI: 10.25236/AJCIS.2023.060702

*[2] Apereo CAS - Identity & Single Sign On for all earthlings and beyond [EB/OL]. https: // github. com/ apereo/ cas, 2023.*
*[3] Wang Qun, Li Fujuan. Laboratory unified authentication scheme based on single sign-on [J]. Experimental Technology and Management, 2020, 37(5): 219-223.*
*[4] The Open Group. Single Sign-On [EB/OL]. http://www.opengroup.org/security/sso, 1995-2005.*
*[5] Yu Liangliang. The Application of Single Sign-on System Based on CAS Protocol in Digital Campus [J]. Computer and Information Technology, 2018, 26(6):21-23.*
*[6] Wang Weihua, Wang Changjie. Application of Unified Identity Authentication in Network Based on LDAP [J]. Journal of Qingyuan Polytechnic, 2014, 6:38-40.*
*[7] Wu Xiaobin, Zhang Yuelin. The design of an uniform identity authentication system based on LDAP [J]. Journal of Huazhong University of Science and Technology(Natural Science Edition), 2003, A1:332-334.*
*[8] Yuan Yi, Song Qiulin. User authentication system design and Implementation Based on LDAP[J]. Journal of Chongqing College of Electronic Engineeri, 2012, 3:161-164.*
*[9] Tan Shenglan. The Design and Implementation of the Campus Network Unified Identity System Based on LDAP [J]. Journal of Dongguan University of Technology, 2009, 16(3):82-86.*
*[10] Zhang Xiqi. Research on Unified Authentication and Log-on System Based on Directory Service [J]. Journal of Anhui Vocational College of Electronics & Information Technology, 2015, 14(2):1-4.*
*[11] Hua Jianxiang, Qu Xia. Uniform identity authentication system research based on LDAP protocol [J]. Intelligent Computer and Applications, 2019, 9(3):129-132.*
*[12] Xu Li, Wang Yao. Unifying authentication of moodle and e-mail system based on LDAP[J]. Computer Applications and Software, 2011, 12:232-235.*
*[13] Tang Mingjing, Chen Jianbing, The research and realization of true uniform identity authentication on digital campus [J]. Journal of Yunnan University(Natural Sciences Edition), 2013, 2:138-142.*
*[14] Miao Jiajia, Deng Su, Liu Qingbao. Overview on ETL Technology[J]. Computer Engineering, 2004, 40(3):4-6.*