Research on Access Control Strategies for Medical Data Interaction Platform Based on Cloud Services

Bing Xu\textsuperscript{1,3}, Xingyuan Chen\textsuperscript{1} and Jian Ding\textsuperscript{2}

\textsuperscript{1} Information Technology Academy, KunMing University, Yangpu Road 2nd, KunMing, China. Email:xubingqd@163.com
\textsuperscript{2} Children's Hospital of KunMing, QianXing Road, KunMing, China
\textsuperscript{3} Yunnan Applied Basic Research Project: 2017FD089

Abstract. Cloud computing technology is widely used in the field of medical information, providing strong technical support for the information construction of medical and health institutions, but at the same time it also brings privacy disclosure of medical data and hidden dangers of medical information insecurity. Control is one of the key technologies in the privacy and security protection of medical data in cloud services. This paper aims at the limitations of the RBAC model applied to the SaaS model, improves it, and proposes a Service-Oriented Multi-tenant Access Control (SMTAC) strategy to meet the needs of the software rental service to provide security for multiple healthcare organizations in the region at the SaaS model.

1. Introduction
Cloud Computing provides Internet-based processing, storage, infrastructure, and software services for large-scale resource pools to users, enabling low-cost, automated, rapid provisioning, and flexible IT services. In the medical field, medical institutions can lease the software and hardware services provided by cloud computing through the Internet, that is, cloud services, thereby reducing operating costs. A variety of medical service information systems can be deployed on the regional medical information sharing and coordination platform, and the SaaS model based on single-instance multi-tenancy technology is used to provide software services for some medical institutions that have not yet established (such as most primary medical institutions) or whose business information systems are not yet sound, to assist and promote the sharing of medical information and medical collaboration of activities.

The single-instance multi-tenant SaaS model means that an application instance is provided by a service provider can serve multiple tenants, and at the same time it can achieve the effect that the instance only serves itself, using this model can save resources, reduce development costs, and provide tenants with lower-cost software rental services, so that small and medium-sized organizations can enjoy the "low threshold" of information construction.

Considering the complex relationship between the role and authority of platform users, it is particularly important to study the access control strategy to ensure that tenants can enjoy software lease services that match their own authority and roles. This article mainly studies the access control strategies based on the management control layer in the regional medical information sharing collaborative platform to ensure that the medical and health institutions under the SaaS model can lease software services flexibly and independently.
2. Comparison of Common Access Control Strategies
Access control is a defensive measure taken against unauthorized use of system resources by explicitly permitting or restricting subject access to objects and capabilities to prevent unauthorized users from intruding or to prevent damage caused by inadvertent operations by legitimate users to ensure that system resources are controlled and used legally.

The access control model is a method of describing a security system based on access control rules and establishing a security model. This article mainly analyzes the three commonly used access control models DAC, MAC and RBAC to explore the improved method.

2.1. DAC
DAC (adaptive access control) is a kind of more flexible access control technology, the most important characteristic is "autonomy", and that is, any authorized subject can authorize the already owned permission or undo the operation without the permission of the system administrator. Obviously, this method can reduce the burden on system administrators and break the limitations of authorization management. When new requirements arise during the work, the authorized subject can set the authority within the scope of authority accordingly. However, this kind of flexible feature has security risks. The main manifestation is that system administrators cannot fully control the authorization status and easily trigger unauthorized operations.

2.2. MAC
MAC (mandatory access control) is an access control technology mandated by administrators. The administrator defines the information security level and the user privilege level in advance, that is, binds the user's access privilege to the security level of the accessed resource. When the user requests an access to a certain resource, the system uses the predefined access mechanism to match the security attributes of users and resources. The advantage of this method is that it has strong security and confidentiality, and a high degree of controllability. The disadvantage is that it requires a large amount of work, the authority is difficult to change dynamically and it is not flexible enough.

2.3. RBAC
Do not add any footnotes until all the author names are linked to the addresses. For example, to format RBAC (Role-Based Access Control) is a role-based access control technology. By introducing the concept of roles, users and permissions are not directly related, but through specific roles and permissions associated with users and roles is a one-to-many relationship, roles and permissions are also one-to-many relationship. The system administrator can simplify the multi-level authorization directly to the user by setting the permissions of the role. This technology not only increases the flexibility and controllability of authorization management, but also reduces management costs greatly.

3. Proposed and Design of SMTAC Model
By contrast, RBAC is proposed to solve the problems of the MAC model and the DAC model. It has obvious advantages and is well applied in the SOA architecture. However, in the single-instance multi-tenant environment, RBAC cannot meet the isolation access requirements between platform tenants, nor be able to implement unified and effective management of access requests for each tenant. If the allocation of authority is limited to the platform manager, the workload of the manager is too large and the authority control is inflexible.

This paper proposes a service-oriented multiple tenant access control (SMTAC) strategy model for regional medical information sharing collaborative platform. This is a service-oriented multi-tenant access control model. It addresses the limitations of the RBAC model applied in the SaaS model. By adding elements such as tenants and software services, it improves access control rules and adapts the RBAC model.

3.1. MTAC Model Terms
The following terms are involved in this model:
(1) Tenants: An organization that leases platform resources and represents a collection of all tenants
on the platform. The tenant ID is TID;

(2) Users: Users can independently access the resources of the system; there are two main categories, namely platform service administrators and end users. The administrator of the platform service is referred to as PUS; the end user is designated as US, and the identifier is UID;

(3) Roles: Refers to departments or departments in an organization and represents a collection of various roles on the platform. Role ID is RID;

(4) Permissions: refers to the various operations that can be performed on resources, representing a collection of all access rights. The permission identifier is PID;

(5) Objects: A collection of all operable objects provided by SAAS that represents a collection of all operable objects. The object identifier is OID;

(6) Soft-Service: Represents an instance of a SaaS service set, that is, a software service supported by the model, identified as SID;

(7) Operation: An executable image of a program. When the user performs certain functions, the corresponding operation is called, denoted as OP.

3.2. The Structure of the SMTAC Model

SMTAC The structure of the model is shown in Figure 1. In this model, the platform administrator PUS and the tenant set TS are one to many relationships, the platform administrator and the software service are one to many relationships, and the tenant and the end user are one to many relationships, such as a platform administrator can grant access control permissions to multiple tenants; software services and tenants, end users and corners. There are many to many relationships between colors, roles and objects. The platform administrator is responsible for managing the tenants and software services; the tenant is responsible for managing the role mutex rules and role rights set in the software service ordered by the agency, and realizing the management of the user's multiple roles by the user assignment operation, and the management of a single role through the authorization operation. By assigning permission to roles instead of users, it can increase the flexibility of permission allocation and refine the specified granularity of privileges.

![Figure 1. SMTAC model structure](image-url)
3.3. The Working Mechanism of the SMTAC Model

In the SaaS mode, the regional medical information sharing cooperative platform needs the security authentication and access control technology to meet the following three aspects: on the one hand, it can reduce the load of the server and the work capacity of the system as much as possible under the premise of providing the normal software service for the users; on the one hand, the tenants are allowed to try to use the same level. The software service of the platform, and then pay for the purchase, for the purchased service can be in a certain range of independent configuration; and one is to grant the tenant corresponding authority, allowing tenants to flexible within the authority within the authority to adjust the access rights of the end users within the organization. Therefore, the working mechanism of SMTAC model involves three processes: registration, authentication, authorization and change of permissions.

3.3.1. Registration process. The process involves two types of users (referring to medical and health institutions) and end users (reference room personnel), because the end users can only belong to one tenant organization, so the registration order of these two types of users is fixed, that is, after the tenant user is registered successfully first, the end user will register again.

The medical and health institutions submit the registration information of the tenant name (EID), the password, the name of the organization to the platform. After the audited by the platform administrator, it allows the user to try the platform service, and tries to try the user to pay for the rental software service after the trial. After the payment is successful, it can configure the operation within the scope of the software. When the end user is registered, it needs to submit the registration information such as the affiliated institution, the affiliated department, the user name, the password and so on. The platform administrator first judges whether its affiliated institution has become a legitimate tenant. If it is not terminated, the registration information is submitted to the tenant user for audit and the audit passes the minimum. The authority rules give the initial roles and privileges of the departments they belong to.

3.3.2. Authentication process. The authentication server extracts the user name and password recorded by the user on the login page and compares it with the local user identity authentication information. If the authentication is successful, it allows it to operate within the scope of the authority; otherwise the error information will be prompted.

3.3.3. Authorization process. In order to improve the flexibility of authorization, users and permissions are not directly related, but to grant users permission through roles. Verify that the server passes the authentication of the login user, issue a certificate for the user with the private key, and attach the public key at the same time. The contents of the certificate include user name, role set, permission set, verifying the server signature, and verifying the public key of the server. The user uses the certificate to access the server that provides the software service. The corresponding server decrypts the certificate with the public key. If the signature of the authentication server is found, the certificate is considered to be legal and responds to the user's request according to the role set and the permission set in the certificate.

3.3.4. Change permissions process. The minimum permissions rule is used to give users the most basic rights through the role intermediary, but in practice, users will have the need to change their rights. When a user applies for a change of authority, the administrator must decide whether to change the role or part of the permissions within the role.

If the role is changed, it is necessary to use the duty separation rule to judge whether the user's existing role conflicts with the application role, if the conflict refuses to change the request, and if it does not conflict, it will be allowed to change.

If some permissions in the role are changed, permission changes can be made according to the rules of permission derivation.
3.4. Access Control Rules of the SMTAC Model

3.4.1. Minimum permissions rule. The minimum permissions rule is a required rule that the user does not have the right to set up it. It specifies that a user (a process or a program) can only access the necessary information or resources that it runs in a specific abstraction layer. The rule is designed to avoid losses due to erroneous or malicious acts, thereby protecting data and functions. In the SMTAC model, the rule requires users to perform the operation permissions on the ordered software services, including the privileges that the users have for the software service. The algorithm flow of the minimum permissions rule is illustrated as follows:

![Flow chart of the execution of the minimum permission rule](image)

**Figure 2.** Flow chart of the execution of the minimum permission rule

3.4.2. Separation of responsibilities rules. The rule of separation of duties is controlled by controlling user assignment, role activation, assignments of authority and execution of operations. In this rule, when the role of the user application conflicts with the assigned and activated role, the application's role assignment request is not allowed, that is, the role of the newly granted user cannot be mutually exclusive with the role that the user has already had. If there is mutual exclusion, the new role can not be granted. Considering that medical and health information services involve personal secrecy information, institutional cost secrecy information, and privacy information and so on, the SaaS tenant institution has a strong demand for this rule when renting software services. Therefore, the software services ordered by the medical and health institution tenants need to be set up with the separation of responsibility rules. The main idea of the rule separation algorithm is as follows:

1. The user UIDi makes a request for a new role RIDj;
2. Get the current set of roles \( \{RS_i\} \) owned by the user UIDi;
3. to obtain the user's current role set \( \{RS_i\} \) exclusive set of roles \( \sum RS_i \{RID\} \);
4. It is judged whether RIDj is included in \( \{RS_i\} \{RID\} \), and if so, the application for adding the role is rejected, and the prompt for prohibiting the addition of the role is given; if not, the set of user roles needs to be updated immediately \( \sum RS_i=\{RS_i\{RID\}\} \) gives tips for increasing the success of the role.
Table 1. Role Table

| Field name     | Column name | Data type     | Major key/foreign key |
|----------------|-------------|---------------|-----------------------|
| Role ID        | RID         | Varchar(50)   | Major key             |
| Role name      | Rname       | Varchar(50)   |                       |
| Role description | Rdescription | Varchar(50) |                       |
| Permission     | PID         | Varchar(50)   | foreign key           |
| Exclusive roles | Exclusion   | Varchar(50)   |                       |

3.4.3. Permission derivation rules. In the SaaS model, multiple tenants can be served via the Internet. Because different tenant organizations have different requirements for the same SaaS roles and permissions, therefore, in order to be able to flexibly make multiple tenant organizations share the same software service, permission derivation rules are provided in the SMTAC model. The characteristics of this rule are to grant the minimum permission set and maximum permission set for each role in SaaS mode. For the same software service, only the minimum permission set of each role is fixed, allowing the tenants to independently select the permission for setting the maximum permission set. The execution process of permission derivation rules is shown in the figure.

![Figure 3. Partial permission derived flow chart](image)

4. Conclusion
In view of the security problems of access control in a single instance multi-tenant application in the cloud service, the SMTAC strategy model is proposed based on the regional medical information sharing cooperative platform. The SMTAC policy model not only inherits the advantages of the RBAC model, but also has the advantage of the privilege management and the flexible operation, and the hierarchical management of the authority. It reduces the workload of platform administrator's permission management in multi tenancy environment. This strategy is more helpful to solve the problem of the user's rights distribution and security access in the multitenant environment, and
provide a certain guarantee for the tenant to provide the tenant with the application software flexibly and safely.

5. Acknowledgements
We would like to acknowledge the sponsorship of the Yunnan Applied Basic Research Project (2017FD089) for funding this research. We also thank thank for the anonymous reviewers for their constructive suggestion.

6. References
[1] Feng Dengguo, Zhang Min, Zhang Hao, et al. Research on Cloud Jiusan Security [J]. Journal of Software, 2011, 22(11): 71 - 83.
[2] Han Daojun, Gao Jie, Ruan Haoliang et al. Research progress on access control model [J]. Computer Science, 2010, (11): 29 ~ 33.
[3] Wei Lifeng, Meng Kaikai, He Lianyue. Fine-grained autonomous access control mechanism for user roles [J]. Journal of Computer Applications, 2009, (10): 2809 ~ 2811.
[4] Sun Hu. Research on Access Control Model [J]. Computer Knowledge and Technology (Academic Exchange), 2007, (08): 390 ~ 392
[5] Jie Fengke, Zhu Aijun, Ma Guihang et al. Role-based management access control model in Oracle system [J]. Zhongyuan Engineering Journal of Chinese Academy of Sciences, 2003, (02): 33~35.
[6] Jiang Yufeng, Fu Yi, Wu Xiaoping. Design and Implementation of Permission System Based on RBAC [J]. Computer and Digital Engineering, 2009, (06): 98~101.
[7] Lin Li. Application of cloud computing in the construction of regional medical information system [J]. Electronic Technology and Software Engineering, 2017, (15): 174-174
[8] RAJRAMAN A, ULLMAN ID. Mining of massive datasets [M]. Cambridge United Kingdom: Cambridge Univery Press, 2012.
[9] MOHITMASKE, PRASAD D P. An introduction to real time processing and streaming of wireless network data [J]. Journal of Data Analysis and Information Processing, 2014(2): 6-11.