Implementation of Universal Microservice User Privileges Management and Control

Huiyong Luo
Taizhou Central Branch of the People's Bank of China, Taizhou, 225300, China

Abstract: Microservices have the advantages of independent deployment, on-demand expansion, and heterogeneous technology, and are suitable for application systems for sensitive management. Microservices splits the application of traditional single-frame architecture into multiple services, and its user rights control and single application are quite different. This paper analyzes the control requirements of resource access under the micro-service architecture, designs the business process, and gives solutions to the overall architecture, refines the important modules, and builds a user rights management and control system suitable for most microservice applications. The structural transformation of SMEs needs to be.

Keywords: Microservice architecture; Authority control

1. Introduction

Monolithic Architecture applications generally treat all application functions and data as a whole. Applications are the smallest deployment unit [1], and all business logic processing requests can run independent processes [2]. With the rapid development of the Internet industry, the traditional single application architecture cannot meet the demand changes, and the problem of fast function iterations is becoming more and more prominent. In this case, the microservice architecture has the flexibility, scalability, flexibility, and stability of the architecture. More and more attention is being paid.

User rights management and control is a process of auditing user access behavior according to the access rights set by the system. This function is mainly used for user login and permission verification. User login confirms whether the user is a legitimate user by verifying the user account and password. Permission verification generally verifies whether the user-submitted access has the relevant permissions after the user logs in, no exception is returned, and the business operation is performed.

From single architecture, to service, to finer-grained microservices, application development strives to balance the complexity and efficiency [3]. Since user rights management and control is the core content of the information system, which directly affects the security and efficiency of the system, the micro-service splits the original single application into multiples, so the authority management and control in the micro-service architecture will become more complicated. The problem has a more significant meaning [4].

2. Security Analysis

Traditional single-frame architecture applications generally implement user login authentication by maintaining stateful user login information on the server side, such as Session [5], and then authenticate user rights by comparing user rights information and request operations. The access control of the single system usually adopts a centralized management mode. First, the login verification function responds to the login request to identify the user's identity information. Secondly, the access behavior is audited according to the set access rules. Finally, the user access request is reviewed, and then the user access request is passed. Business module execution.

The microservices form a number of microservice instances by splitting, and the user rights management and control functions can be separately implemented as one service, or can be independently implemented by each service service, and the implementation manners are various. Compared with the user rights management and control of the single system, the microservice security has the following difficulties:

Each microservice instance is relatively independent, and the user can be called across services once authorized. The server generally does not save user information, that is, the server is stateless [6].

User authority management and control is relatively concentrated, generally not performed by the microservice instance, but by the microservice gateway to undertake the authentication function [7], the gateway load pressure is relatively large.

The interface of the microservice instance generally conforms to the RESTful style [8]. In addition to the consideration of submitting microservices and addresses, the authentication needs to increase the consideration of the submission method.

In addition to the user rights control generated by user access, the mutual call of important microservice instances also needs to consider authentication.

3. Business Analysis

3.1. General business needs

By combing the characteristics of privilege management and control in the micro-service environment, and arranging common business requirements, the versatility of the user privilege management and control system can be better realized. This paper summarizes the aspects of rights allocation, page presentation, gateway performance, interface versatility, etc. The main contents are as follows:

Meet the requirements of universal authority configuration. The first is to ensure that the scenario of complex authority assignment is met, and the user's authority should be determined according to the user's role, rank, organization, department, etc., to achieve flexible authority configuration. Resources protected by access control should include pages that users can access, background addresses for data interaction, and so on. Second, it is able to quickly implement micro-service users, user-related configuration of permissions, and reduce the complexity of configuring micro-service user permissions.

Meet the requirements of universal page display. The first is to design a unified user login interface, user management and authorization interface. When the business needs related functions, it can be integrated into the business page by means of jump, embedding, etc., to achieve the uniformity of the style and avoid the repeated development of the business microservice. To enable developers to focus on business function development. The second is to provide data related to the login user rights, so that the front-end page can customize the user interface according to the user role, all resources, etc., and intercept illegal operations.

Meet the performance requirements of the gateway. The audit of the user access request can implement the “white and black list” function, and can quickly release the suffix files commonly used in the webpage, such as js, css, jpg, etc. according to the micro service setting, and only perform authentication processing on the resources to be controlled [9] .

Meet the universal interface requirements. The interface for universal user rights control should be RESTful and easy to call.

3.2. Business service process

According to the general business requirements, the general user rights management and control should include three core functions: user login authentication, resource list acquisition, and user authority verification. User login authentication means that the user submits personal information, implements user identity authentication, and obtains the verified certificate. Through the acquisition of the user resource list, it is convenient to display the page refinement and the need for the foreground permission interception. User authority verification, that is, authentication function, the system can perform permission review on the user-submitted access. If there is no relevant permission, the access failure or abnormality is returned. Otherwise, the relevant micro-service is submitted to implement the specific service.

The key to the business process of universal privilege management is to link the three core functions reasonably and ultimately realize the business requirements. The user submits the username and password, and the user logs in to the authentication module to complete the verification, and sends the returned result in JWT (JSON Web Token) [10] format. User authentication needs to consider preventing retransmission attacks, and a "challenge-response" mechanism can be introduced [11]. After the user logs in, the user resource list is obtained. After the microservice gateway intercepts the user to initiate the access, the destination address (microservice), the submission mode, and the JWT forwarding user authority verification interface are actually verified. The actual business service flow is shown in the figure 1 below.

4. Overall Architecture Design

Based on the reference to oAuth [12], Shiro [13], SpringSecurity [14] and other rights control components, this paper combines common business requirements, optimizes the access process, adjusts the data structure, and forms a general micro-service user rights management and control design. Its uniqueness Mainly embodied in the micro-service idea, the user rights management and control core functions are independently designed as micro-services, and at the same time provide login authentication to the user and provide an authentication interface to the gateway.

In the micro-service, the gateway takes over the interception and forwarding of user access, and the user right authentication actually completes the authentication. The user rights verification function has a performance bottleneck, splits the user login function and the user authority verification function, and disassembles it into a registration center from the architecture. Service center two micro-services. When the service center is overloaded, the dynamic scalability of the microservices can be utilized to achieve dynamic load balancing [15]. The iteration and exception of the registration center microservices do not affect the authority verification, ensuring the stability of the microservice system. The overall architecture design is shown in Figure 2 below. The main functions of the two are as follows:

The registration center microservice can implement user management and authentication interfaces, including: firstly, user login authentication, user basic data synchronization, and authority-related data management; second, the use of XML permission configuration file to realize microservice permission. The third is to provide the user permission data related to the foreground request such as the user's permission list and role list to the outside.

The service center micro-service can realize the synchronization and verification of user rights, including: firstly, the user's basic data is obtained from the traditional relational database and the "key-value" paired data is sorted by calculation, and synchronized to the Redis memory database.[16], to improve the efficiency of data query; second, according to the access address (microservice) submitted by the front end, the submission method, the user JWT, etc., to confirm whether the user has the relevant authority.

5. Important Module Design

5.1. User permission calculation and synchronization

The user authority calculation and synchronization module is an important module for data synchronization between two micro-services. Since the service center uses an in-memory database to facilitate frequent reading of micro-services, the module can obtain data in the registry micro-services periodically or actively. The final result is stored in the in-memory database by calculation. Since the calculation of content such as "microservices: users" and permissions is completed in advance, the frequent calculation of user rights is avoided, and the overall efficiency of the rights management is effectively improved.

5.2. User rights acquisition module and user rights authentication module

The user permission acquisition module and the user authority verification module have similar functions. The user permission data is obtained from the Redis database, but the business logic is different: the user permission interface mainly provides the user-related permission list to the foreground, and facilitates the front-end customization interface, etc., generally After the user logs in, the user rights authentication interface is mainly used as the authentication service of the gateway. After receiving the call request, the micro-service gateway performs user rights verification and authenticates according to the information transmitted by the gateway.

The user rights authentication module introduces a graylisting mechanism, and the graylist is introduced to facilitate quick access of resource class files. The gray list does not need to be set independently, and is calculated for the user permission calculation module. For example, if a micro service requires permission to control only the address of the jsp and action suffix, the gray list calculation result is a set composed of jsp and action, and the user authority verification first determines the gateway. Whether the forwarded microservice request address is a jsp or action suffix, if not, the direct return verification is passed, and if yes, the further verification is performed according to the user authority information.

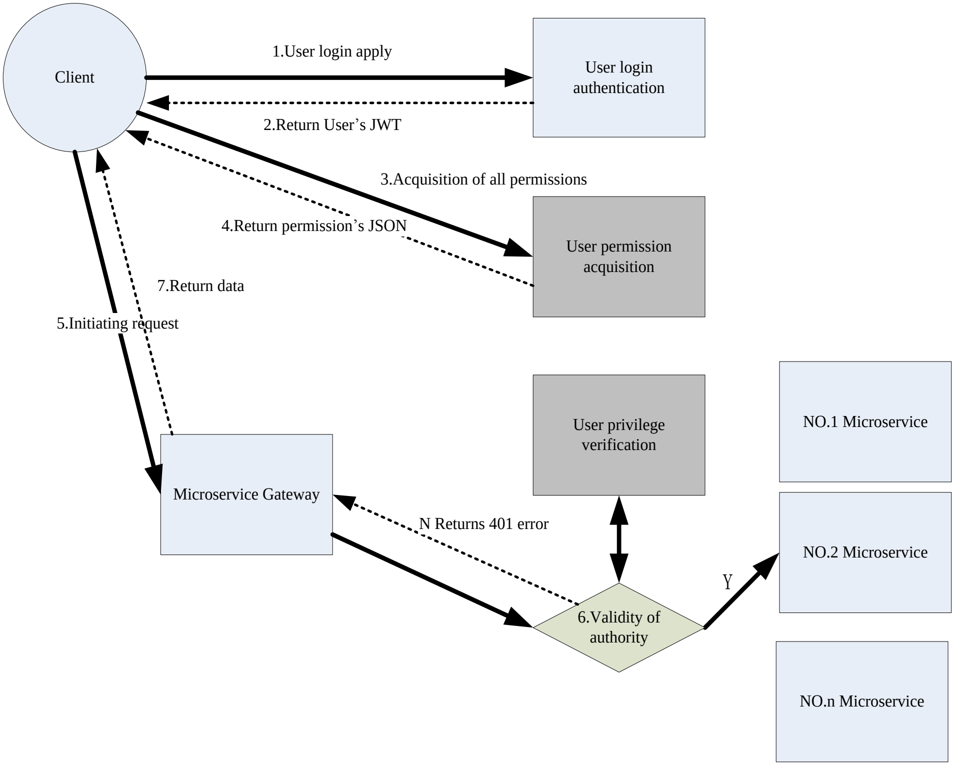

Figure 1. General microservice user rights control access business process

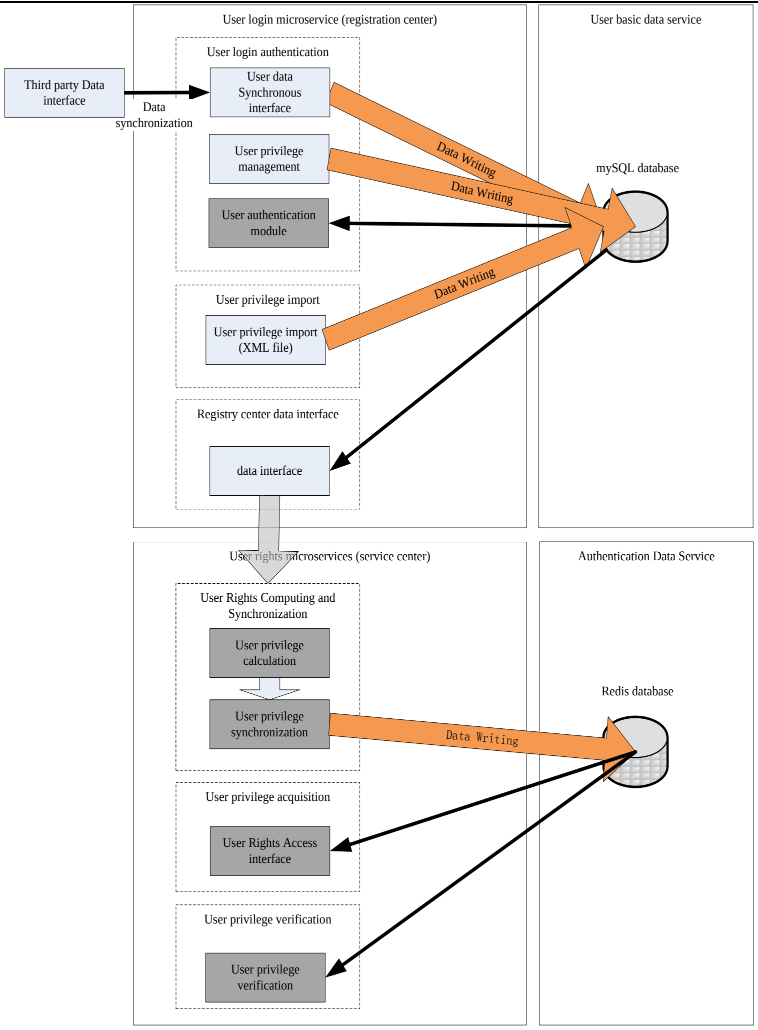
User login microservice (registration center)
User basic data service
User login authentication
Third party Data interface
Data synchronization
User data Synchronous interface
User privilege management
User authentication module
Data Writing
Data Writing
mySQL database
User privilege import
User privilege import (XML file)
Data Writing
Registry center data interface
data interface
User rights microservices (service center)
Authentication Data Service
User Rights Computing and Synchronization
User privilege calculation
User privilege synchronization
Data Writing
Redis database
User privilege acquisition
User Rights Access interface
User privilege verification
User privilege verification

Figure 2. General microservice user rights management overall architecture

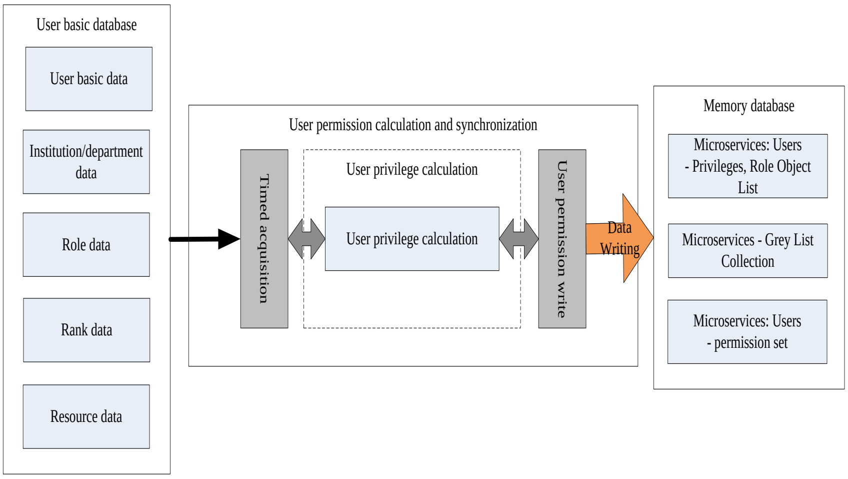
User basic database
User basic data
Institution/department data
Role data
Rank data
Resource data
User permission calculation and synchronization
Timed acquisition
User privilege calculation
User privilege calculation
User permission write
Data Writing
Memory database
Microservices: Users - Privileges, Role Object List
Microservices - Grey List Collection
Microservices: Users - permission set

Figure 3. User permission calculation and synchronization module design

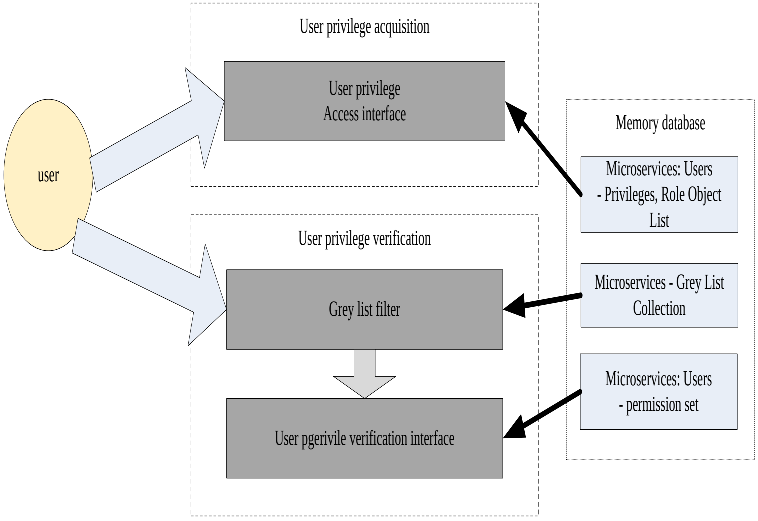
User privilege acquisition
User privilege
Access interface
user
Memory database
Microservices: Users - Privileges, Role Object List
User privilege verification
Grey list filter
Microservices - Grey List Collection
Microservices: Users - permission set
User pgerivile verification interface

Figure 4. User permission acquisition and user permission verification module design

6. Conclusion

This paper analyzes the general micro-service user control requirements of small and medium-sized enterprises, and conducts relevant design according to the specificity, focusing on the calculation and synchronization of authority, user authority acquisition and user authority verification, and highlights the "general and flexible" system. Construction ideas. The user is associated with the organization, department, rank, etc., meets the requirements of flexible customization authority, and forms a user authority management and control system through multiple microservices, which facilitates the rapid iteration of each module and increases the stability of the microservice user control function. The xml file is used to implement the user rights configuration of the micro service, which simplifies the permission configuration problem after the micro service is online, and realizes the decoupling between development and operation and maintenance.

References

[1] Dragoni N., Giallorrnzo S., Lafuente A., et al.Microservices : Yesterday, today, and tomorrow.Present andUlterior Software Engineering. 2017, 4, 195-216

[2] Mi Woqi. Depth analysis of the nine characteristics of microservice architecture. Computer Knowledge and Technology: Experience and Skills. 2016, (10), 105-110.

[3] Chen Chunxia. Analysis of Container-based Microservice Architecture. Information Systems Engineering. 2016, (3), 95-96.

[4] Zhu Yongqiang, Fang Yi, Gong Xueqing. Design and implementation of access control model under microservice architecture. Computer Applications and Software. 2018, 35(12), 27-32+43.

[5] Zhuang Wei, Lu Xuegang. Discussion on authentication and authentication in microservice architecture. The era of financial technology. 2018, 278(10), 40-42.

[6] Xin Yuanyuan, Niu Jun, Xie Zhijun, et al. Overview of Microservice Architecture Implementation Framework. Computer Engineering and Applications. 2018, 54(19), 16-23.

[7] Zhang Jie, Si Weichao, Wang Lina, et al. Design and Application of a General Assessment System for Microservices. Computer and Digital Engineering. 2018, 46(12).

[8] He Jun, Chen Guimin, Huang Huihai, et al. Design of RESTful Architecture. Network Security Technology and Application. 2019, 217(01), 40.

[9] Liu Yaou, Huang Liusheng, Xu Hongli, et al. A Resource Authentication Method for WoT Architecture. Small Computer Systems. 2017(4).

[10] Jones M B. The Emerging JSON-Based Identity Protocol Suite [C]//W3C Workshop on Identity in the Browser. 2011, 1-3.

[11] Huang Chaoyang. Dynamic Password Authentication System Based on Random Numbers. Network Security Technology and Application. 2019, 217(01), 28.

[12] Xiao Meihua, Cheng Daolei, Li Wei, Li Ya'nan, Liu Xinqian, Mei Yingtian. Formal analysis and, verification of oauth 2.0 protocol improved by key cryptosystems. Chinese Journal of Electronics. 2017, (26), 484.

[13] Apache Shiro. Shiro reference documentation[EB/OL]. [2018-12-20]. Http: / / shiro. Apache. Org /.

[14] Liu Yao. Third party authorization framework based on spring and oauth2.0. Computer Technology and Development. 2017, (3).

[15] Xin Yuanyuan, Niu Jun, Xie Zhijun, et al. Overview of microservice architecture architecture framework. Computer Engineering and Applications. 2018, 54(19), 16-23.

[16] Li Faping, Wang Chengliang. Design and implementation of redis scalable and efficient replication scheme. Journal of Southwest China Normal University: Natural Science Edition. 2018.