Design and Implementation of Network Information Service Platform Based on Microservice Architecture

Jin Lian1, Ming Gao2*

1School of Artificial Intelligence, Jianghan University, Wuhan, Hubei, 430056, China
2Network Information and Teaching Equipment Management Center, Jianghan University, Wuhan, Hubei, 430056, China
*Corresponding author’s e-mail: gaoming@jhun.edu.cn, 171188648@qq.com

Abstract. In this paper, based on the development of network information microservice platform, and according to the business characteristics of the platform, the overall architecture model of the microservice platform was designed. Besides, the system is hierarchically managed by basic equipment area, standard area, hub area, and application service area. Hence, the problems of independent deployment, data transmission sharing, service monitoring, and authority management of the microservice system are solved. Moreover, the system development process, the development efficiency, and the system reliability are also improved. The project which is supplemented by a sound management mechanism implements the framework of service platform based on Spring Cloud technology, establishes the business container of microservice, analyzes the value of the system, and then forms a sustainable optimization system.

1. Introduction
Microservice Architecture, the landing architecture of Service Oriented Architecture (SOA), is further loosely coupled under the basic framework of SOA and inspires new technical routes and integration ideas for multi-system business integration, such as online business hall and integrated business platform. With the implementation of the national cyber development strategy, it is necessary to establish a large platform of "Internet + education" in colleges and universities so as to realize the standardization of the list of items for college management, the standardization of the business guideline, the standardization of the review work rules, and the coordination of business handling [1]. Microservice platform can not only break down the departmental management boundary formed by the traditional monomer system, but also can effectively realize business claim, overall planning, as well as data sharing and circulation. Hence, building a microservice platform has become a technical trend that cannot be ignored in the information construction. The microservice platform divides the driver into several small service granularities according to the business responsibility, clarifying the business ownership by fully decomposing the business process. In this way, every business has its own process and the process can be open, fair, transparent, traceable, and checkable [2].

2. Microservice Architecture
Microservices first appeared in 2011 by a software architecture team in Venice, and the name—microservices was determined in 2012 to represent a software architecture style. It was jointly proposed by Martin Fowler and James Lewis in 2014: “Microservices decompose a complex application into multiple tiny services. These services operate independently and use a lightweight communication
mechanism to complete the process of building a single service independently or in coordination with each other. [3]“ Microservice aims to reduce the coupling of the system by decomposing the functional modules into separate discrete services so as to provide more flexible service support [4].

The microservice architecture is independent and distributed. After the user logs in, the session server invokes a session management, which is responsible for the maintenance of the user session state throughout the whole process. The registry center is used to maintain the service function lists and interacts with the session server to form service discovery and logout. Service providers offer service functional capability and maintain heartbeat monitoring with the registry center. The basic framework of the microservice architecture can be deployed in multiple places to maintain heartbeat coordination and realize a PASS-based platform service cluster[5].

![Microservice Architecture diagram](image)

### 3. Demand Analysis of Network Information Service Platform

It is the basic work for the design and development of the network information microservice platform to accurately understand the specific requirements of the system (such as its function, performance, reliability, and extensibility) and to combine these with the basic characteristics of the microservice. The main requirements of the platform are summarized in the following four categories:

#### 3.1. Network equipment maintenance and failure repair business

The network information center is faced with a lot of real-time maintenance and repair business, so it needs to deal with the reporting business timely and effectively. Hence, the microservice platform is adopted to realize the real-time management of repair reporting business, improve the convenience of reporting, and add additional information such as geographic location information and pictures, which increases the information amount of fault preprocessing and improves the accuracy of failure prediction.

#### 3.2. Resource application and approval business

Requirements declaration business mainly consists of the approval business of each business department, department head, and other relevant business department head. The microservice platform can well solve the problem of remote connection and realize online approval. In this way, the cause of every business can be traced and every business can automatically flow from one link to the next.

#### 3.3. Data center and data sharing

The network information center is the information hub of the university, bearing the production, use, storage and re-extraction, sorting, analysis, and mining of the data and other important responsibilities. To share and apply data according to the different responsibilities of each department and to realize data flow is not only the construction target of the network information center but also a sign to show the degree of school information construction.
3.4. Requirements for scalability
College management is generally characterized by numerous departments (administrative departments), multifarious business types, and overlapping and diverse departmental responsibilities. The comprehensive service hall for college management should not only meet the business needs of the college, personnel, student work, finance, educational administration, and other departments, but also meet the cross-business processing of the school office, OA system, and other offices. Therefore, the system is supposed to be of high scalability.

4. Architecture design of network information service platform

4.1. Service business module
Service business module can handle the services of wired, wireless, server, and other network equipment. The system can offer many integrated services, such as online form filling, form submission, form approval, progress query, employee dispatch, repair feedback, and service evaluation.

4.2. Approval business module
The approval business module can realize the online application, approval, review, business opening, and resource usage monitoring of approval services, such as VPN, campus mailbox, temporary internet account, campus website, external IP address, and domain name service.

4.3. Message management engine
The message management engine is divided into two parts: one is the message management engine of the related business, and the other is the user message push and feedback management engine. The business management engine is used to manage the dependencies between business. The user message push and feedback engine handles the message pushes, feedbacks, and evaluations among various services. The two parts of the message management engine determine the intersection and balancing of microservice systems.

4.4. Authority management engine
The authority management engine manages the user’s permissions and determines the initiation of business processes. The activity-oriented BPEL4WS[6] dynamic access authorization model can realize the dynamic authorization recovery mechanism based on the business process context by removing the coupled relationship between the organization model and the business process model, solving the problem of authority management in the open and collaborative environment of the microservice architecture platform.

4.5. Standard data management
Standard data management is the foundation of the microservice platform and also the basic guarantee to realize the manageability, controllability, and optimization of microservices. According to DQAF (Data Quality Assessment Framework), to measure the integrity, timeliness, effectiveness, consistency, and accuracy of the microservice data center[7], can ensure the quality and sustainable improvement of the database.

4.6. Data sharing engine
Data sharing is the driving force to realize data exchange between microservice platform business. The establishment of the data sharing engine helps the data exchange from the data standard level to the business system level, so that the independent physical table of the microservice is mapped to the standard data center, so is the direction.
4.7. Overall platform architecture
The network information microservice platform has designed the system architecture into four main areas (see Figure 2): basic equipment area, standard area, and hub area and service area. The basic equipment area provides basic computing, storage, and network resources. The standard area standardizes, monitors, and maintains technical specifications such as various communication protocols, data structures, and interface definitions that are used in the microservice process.

![Figure 2 Architecture diagram of microservice platform](image)

The hub area which is the core area of system management is responsible for session maintenance, authority management, registration and maintenance of the interface, maintenance of microservice process monitoring information, and opening and closing management. The service area is the centralized deployment area of the microservice system application services. This area will be deployed in a distributed manner based on the microservices formed in the business process context. The business process context forms a business boundary based on the resource requirements, procedures, data production, and processing results of the business. Developers treat this boundary as a physical boundary to form fine granularity business microservices[8]. The services within the architecture have independent codebases, data storage, and even development teams. Also, the selected technology stack and language platform of the architecture can also be different[9].

This architecture well achieves the goal of independent development and deployment of small granularity business as well as a combination of production, operation, and maintenance processes. Besides, the underlying resources based on virtualization technology and Docker packaging technology are adopted to achieve cloud deployment of microservices. In addition, the management of the hub area is utilized to establish standards, monitor production, and synchronize data.

5. Implementation of the Network Information Platform based on Spring Cloud
Spring Cloud provides distributed common components, such as configuration management, service discovery and service registration, message bus, load balancing, circuit breaker, data monitoring, micro broker, distributed session, and cluster state [10], which provides strong support for the information-based microservice platform. In addition, Docker is used to package the developed microservices and their dependencies into a portable image, and then publish them to any machine equipped with Docker. This effectively solves the problem of development and deployment because of small service granularity and a large number of services under the microservice architecture[4]. This section mainly describes the implementation process of the platform architecture. The project uses JDK1.8 and Spring Cloud 2.0.1.RELEASE version.
5.1. Create registry server: eureka-server
1) Add a dependency in the pom.xml: spring-cloud-starter-eureka-server
   2) application.properties configures registry ports and Eureka-related parameters:
      server.port=3344
eureka.instance.hostname=localhost
eureka.client.register-with-eureka=false
eureka.client.fetchRegistry=false
eureka.client.service-url.defaultZone=http://${eureka.instance.hostname}:${server.port}/eureka
3) Server startup.

5.2. Build client service: eureka-client
1) Add a dependency in the pom.xml:
   spring-cloud-starter-eureka-server, mybatis-spring-boot-starter, mysql-connector-java, spring-boot-starter-web,
2) Application.class is officially launched.
3) The project is completed according to the following functions:
   (1) Register to eureka as a client;
   (2) Implement mybatis to connect to mysql.

5.3. Create configuration center: config-server
1) Add a dependency in the pom.xml: spring-cloud-starter-eureka, spring-cloud-config-server
   2) application.properties configures the center properties:
   3) Server startup, adds @EnableConfigServer, and then can be started as a configuration server.

5.4. Use Feign to invoke between services
1) Add a dependency in the pom.xml:
   spring-cloud-starter-eureka, spring-cloud-starter-feign, spring-boot-starter-web
2) Application.class adds the function of opening annotations.
   @EnableDiscoveryClient  // Enable publish and subscribe
   @EnableFeignClients     // Enable feign
   3) application.xml configures related properties:
       eureka:
       serviceUrl:
       defaultZone: http://localhost:3344/eureka/
       feign.hystrix.enabled=true   // Enable fuse function of feign
4) Create a service interface to invoke the service and fuse functions

5.5. Create gateway: Zuul
1) Add a dependency in the pom.xml:
   spring-cloud-starter-eureka, spring-cloud-starter-zuul, spring-boot-starter-web
2) Application.class adds the function of opening annotations.
   @EnableEurekaClient  // Enable publish and subscribe of eureka
   @EnableZuulProxy    // Enable the routing gateway function of zuul
   3) application.xml or application.yml configure related properties.
In addition to the above basic framework features, the project also implements Ribbon-based load balancing functions, uses Hystrix to achieve isolation and control services, uses Sleuth for tracking records, and uses Zipkin for log aggregation.

6. Conclusion

Based on the microservice application architecture, this paper discussed the ways of implementing the construction of the service platform of the network information center, meeting real-time requirements, process requirements, data analysis, storage and inquiry requirements, data sharing, and information security of the business. In response to some disadvantages of the microservice architecture, design supplements were made and some key issues were improved.

The microservice architecture has obvious advantages, but the overhead caused by a large number of microservice concurrency is also obvious. The microservice does not change the nature of WEB services, but provides a solution with fine management and distributed deployment to the business system. Therefore, it is necessary to do research on the effectiveness, efficiency, and benefits of the microservice. All of the research call for further study and continuous improvement on the basis of the actual application.

Acknowledgments

In this paper, the research was sponsored by Science Foundation of Wuhan Education Bureau of China (Project No.2011002) and Key topics in the 12th Five Year Plan of Education Science in Wuhan (Project No.2012A122).

References

[1] Ministry of Education. the Education Informatization 2.0 Action Plan [EB/OL]. [2018-10-26]http://www.moe.gov.cn/srcsite/A16/s3342/201804/t20180425_334188.html.
[2] Shi Yulong, XUE Chunrong. 2018. Design and research on the municipal online office hall [J]. Computer Era (5) : 105-109.
[3] Zheng Binbin. 2017. Reconstruction and Optimization of OJ System based on Micro-service [D]. Shanghai: Donghua University: 5-6.
[4] Wang Lei. 2016. Microservice Architecture and Practice [M]. 1st edition. Beijing: Publishing House of Electronics Industry, 13-38.
[5] Zhang Jing, WANG Yanjie, HUANG Xiaofeng. 2017. Implementation of Microservice Architecture [J]. Computer Systems & Applications (4) : 82-86.
[6] Shang Chaowang, LIU Qingtang, ZHAO Gang, et al. 2014. Study on Activity-oriented Dynamic Access Authorization Model for BPEL4WS [J]. Computer Science (7) : 102-104, 109.
[7] Laura Sebastian-Coleman. 2016. Measuring Data Quality for Ongoing Improvement [M]. Beijing: China Machine Press, 48-55.
[8] Sam Newman. 2016. Building Microservices [M]. Translated by Cui Liqiang, Zhang Jun. 1st edition. Beijing: Posts & Telecom Press
[9] Zhou Li. 2017. Spring Cloud and Dockers Microservice Architecture Practice [M]. 1st edition. Beijing: Publishing House of Electronics Industry: 60~80.
[10] Fendoumanong. Concept, Relationship and Differences of Spring, SpingMVC, SpingBoot, and SpingCloud [EB/OL]. [03-03-27, 2018].