Infrastructure Smart Service System Based on Microservice Architecture

Xiaojun Li[1,2,3], Yue Xi[3](✉), Hehua Zhu[1,2,3], Jiaxin Ling[3], and Qi Zhang[4]

[1] State Key Laboratory for Disaster Reduction in Civil Engineering, Tongji University, Shanghai 200092, China
[2] Key Laboratory of Geotechnical and Underground Engineering of Ministry of Education, Tongji University, Shanghai 200092, China
[3] College of Civil Engineering, Tongji University, Shanghai 200092, China
xiyue2088@126.com
[4] School of Civil Engineering, Southeast University, Nanjing 210096, Jiangsu, China

Abstract. Traditional infrastructure information systems are often designed as closely coupled applications because of professional analysis functions needed. This often leads to complexity and poor scalability of the system. A distributed microservice based architecture is proposed in this paper. This architecture converts the complex professional analysis functions into microservices, which can be provided by computers in different numbers and locations so that this architecture is scalable. Taking a rock tunneling information system as an example, the microservices include data service and analysis service. The key of the microservices are identified as: (1) standardizing the request and response data of the services, and developing a registration center for the consistent access to the services; (2) analyzing the data exchange methods of the services; (3) introducing discovery mechanism for the service management. Finally, an implementation of microservice architecture and microservice examples are introduced, and the microservices are integrated into the service layer of infrastructure smart service system.

Keywords: Microservice · Infrastructure · Digitization technique · Smart technique

1 Introduction

To facilitate the infrastructure information management, commonly used technologies include geographic information system (GIS) and building information modeling system (BIM). GIS is mainly used for input, storage, query, display and analysis of geographic data [1, 2], but it lacks effective management for non-geographic data. BIM is successfully used in building engineering, but the transition of BIM into the domain of infrastructure projects has not been so easy [3–5]. Several studies have analyzed the adoption and use of BIM in infrastructure projects, such as road and tunnels, and discussed the evolution of BIM uses [6]. The extent to which level of BIM implementation has achieved in infrastructure industry has also been explored [7]. In

A. G. Correia et al. (Eds.): ICITG 2019, SSGG, pp. 131–143, 2020.
https://doi.org/10.1007/978-3-030-32029-4_12

summary, GIS and BIM technologies for infrastructure information management and analysis are still under-developed.

Considering the above reasons, it becomes a development trend to develop comprehensive smart service platforms. For example, digital underground space and engineering were proposed by Zhu and Li in 2007 [8]. Thereafter, shield tunnel digital system [9], integrated digital system of infrastructure construction and maintenance [10], and infrastructure smart service system (iS3) [11] were proposed. Among them, iS3 is designed as a service platform based on information flow, as shown in Fig. 1.

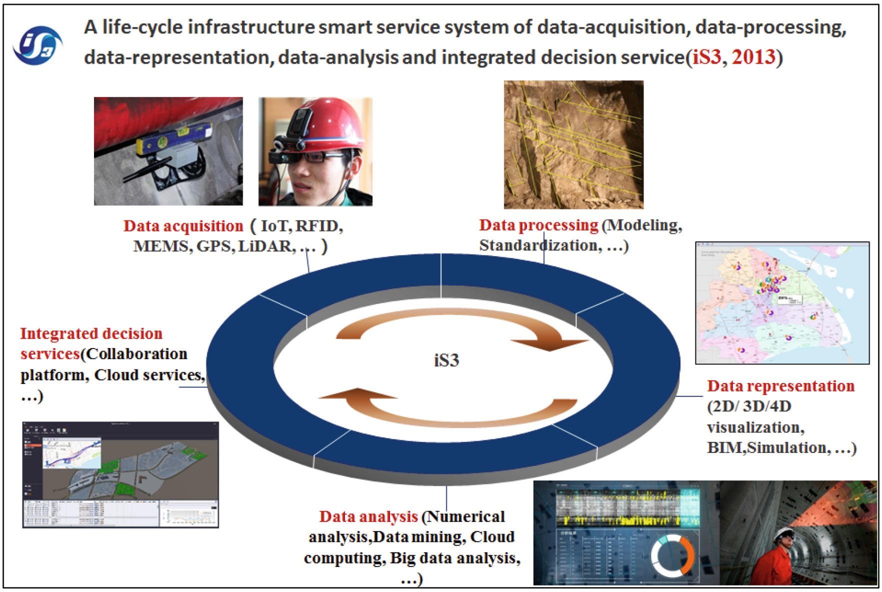

A life-cycle infrastructure smart service system of data-acquisition, data-processing, data-representation, data-analysis and integrated decision service(iS3, 2013)
Data acquisition (IoT, RFID, MEMS, GPS, LiDAR, ...)
Data processing (Modeling, Standardization, ...)
Integrated decision services(Collaboration platform, Cloud services, ...)
iS3
Data representation (2D/ 3D/4D visualization, BIM,Simulation, ...)
Data analysis (Numerical analysis,Data mining, Cloud computing, Big data analysis, ...)

Fig. 1. Concept of infrastructure smart service system (iS3).

The architecture of an information system designates how to divide the application system into different parts, and makes these parts working together. Three-tier architecture is often used, which includes presentation layer, business logic layer, and data access layer. Although the three-tier architecture divides the application into three layers logically, it does not specify how to implement. If each layer is compiled and run in the same process on the same machine, the development would be easier but eventually the system would becoming complex and poor in terms of scalability. In fact, currently platforms are mostly developed as standalone applications [12, 13], such as EXE format of C++ and C# and JAR format of Java. Although applications under such architecture has advantages such as easy to develop and deploy, the applications would finally become more complex and difficult to understand and maintain. In addition, problems occurred in one module may lead to the collapse of the entire system. With the increase of application functions, there will be more combinations of possible failures. Therefore, such architecture is unable to adapt to future rapid changes.

In this paper, a low-coupling, highly reliable, easily scalable distributed microservice architecture is adopted for the iS3 system. Request and response standards for services are developed, and data exchange modes of service functions are analyzed. Service discovery mechanism is introduced for the management of all analytical services to provide consistent calls to outsiders by registry.

2 Introduction of Infrastructure Smart Service System (iS3)

iS3 is aimed at managing infrastructure projects from the perspective of information flow, and it is an integrated decision service system of life-cycle data acquisition, processing, expressing, analysis of infrastructure. The overall framework is shown in Fig. 2, and it is divided into five layers: basic layer, data layer, service layer, application layer and user layer.

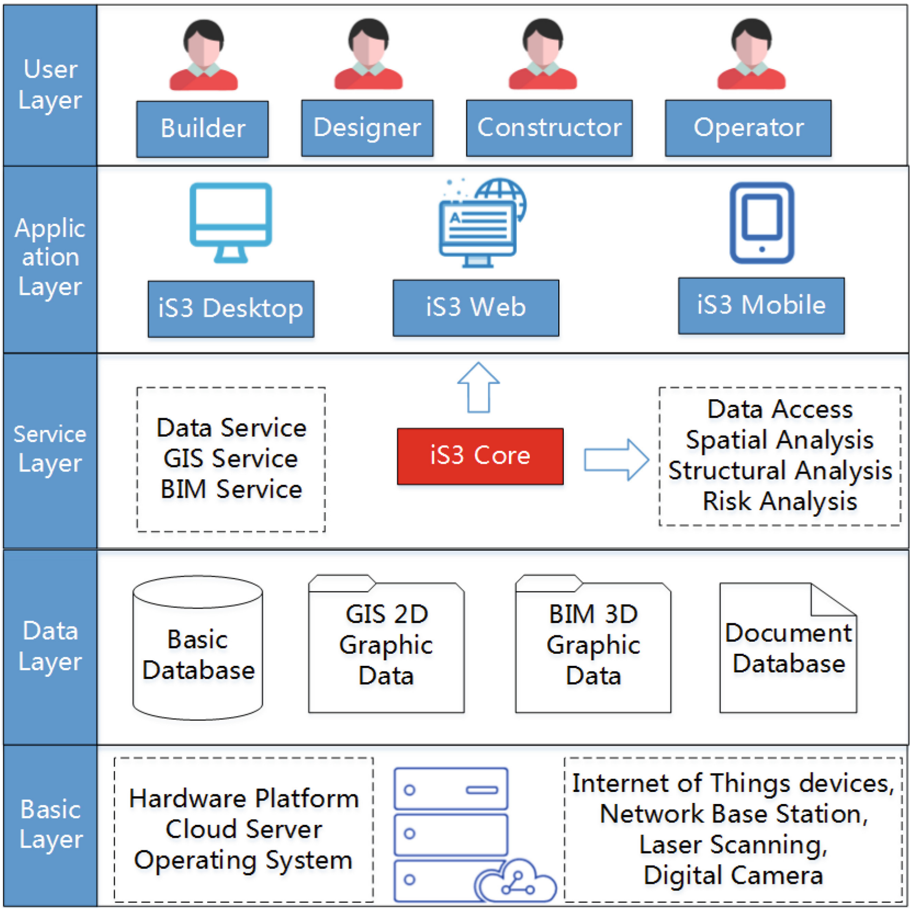


Fig. 2. Overall framework of infrastructure smart service system (iS3).

The basic layer is the collection of hardware devices of the whole system; the data layer is to provide the service layer with access, calculation and storage resources; the service layer is the logical layer that provide the application layer with data access interfaces and analysis service interfaces; the application layer contains user-oriented client programs; the user layer is the group that uses infrastructure smart service system (iS3). The main contribution of this paper is the design of the service layer of iS3 based on microservice architecture, which has advantages of scalability, distributed deployment and unified invocation.

3 Introduction of Distributed Microservice

Taking rock tunneling information management system as an example, this paper introduces the design of microservice architecture. Assume that the existing service layer has provided two kinds of services. The first kind is data service, including tunnel life-cycle data of design, construction, geology, environment and monitoring. The other kind is analysis service, such as the risk analysis of water inflow. Apart from the two above-mentioned service modules, the service layer also provides GIS and BIM service module, UI interface service module, and API interface service module.

All relatively independent services are implemented in the form of microservice, as shown in Fig. 3. Each microservice opens an API interface and is probably deployed on different machines. The purpose of microservice is to split system applications and realize agile development and deployment. In order to build iS3 service layer with microservice architecture, there are several issues needs to be resolved: (1) Establishing communication interface standards between independent microservices; (2) Designing communication modes between microservices; (3) Managing these increasing microservices effectively.

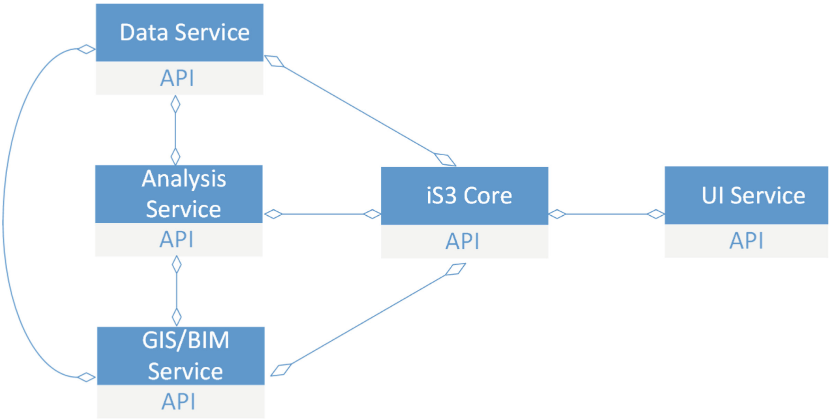


Fig. 3. Microservice architecture design of infrastructure smart service system (iS3).

4 Definition of Service Interface

4.1 Data Standard for Request and Response

Data Service Standard. Take the construction information management of a rock tunnel as an example. The request content of construction management data is shown in Table 1, including the project's name and the part of the tunnel. The corresponding interface can be called using the request content. The response content of construction progress data contains four properties: (1) the mileage that represents the construction progress of the initial lining, (2) the mileage that represents the construction progress of the secondary lining, (3) the bid section number, (4) the date of the construction progress. The response content of construction progress data as shown in Table 2. The response content of geologic forecast data contains seven properties: (1) the interval of mileage, (2) lithology, (3) weathering degree, (4) groundwater, (5) stability, (6) special geologic conditions, (7) surrounding rock grade. The response content of geologic forecast data as shown in Table 3.

Table 1. Request content of construction management data.

Field	Type	Property
Project	String	The project's name
Part_no	String	The left part or the right part

Table 2. Response content of construction progress data.

Field	Type	Property
PROG_CCJD	String	The mileage that represents the construction progress of the initial lining
PROG_ECJD	String	The mileage that represents the construction progress of the secondary lining
PROG_DATE	Date	The date of the construction progress
PROG_NAME	String	The bid section number

Table 3. Response content of geologic forecast data.

Field	Type	Property
GPRF_INTE	String	The interval of mileage
GPRF_LITH	String	Lithology
GPRF_WEA	String	Weathering degree
GPRF_WATE	String	Groundwater
GPRF_STAB	String	Stability
GPRF_FUAL	String	Special geological conditions
GPRF_STRE	String	Surrounding rock grade

Analysis Service Standard. Take the risk analysis of water inflow as an example. The request and response content of the risk analysis of water inflow are shown in Table 4. The request content includes ten properties: (1) carst scale, (2) carst distance, (3) advanced support, (4) support quality, (5) fill type, (6) emergence plan, (7) surrounding water, (8) geologic forecast, (9) excavation disturbance, (10) water inflow of tunnel face. Each property has its unique field, and its type is float. All the properties can be obtained from data service. The request content need to be sent to the corresponding analysis service interface, and then the response content will be returned, as shown in Table 4. The risk analysis of water inflow gives the risk value of water inflow in the present tunnel face according to the algorithms proposed by Liu [14].

Table 4. Request and response content of the risk analysis of water inflow.

	Field	Type	Property
Request	wfrisk_cs	String	Carst scale
	wfrisk_cd	Float	Carst distance
	wfrisk_ds	String	Advanced support
	wfrisk_sq	String	Support quality
	wfrisk_ft	String	Fill type
	wfrisk_ep	String	Emergence plan
	wfrisk_sw	String	Surrounding water
	wfrisk_gf	String	Geological forecast
	wfrisk_ed	String	Excavation disturbance
	wfrisk_wf	String	Water inflow of tunnel face
Response	wfrisk_value	Float	The risk value of water inflow

4.2 Unified Access to API Gateway

To save network resources, improve request efficiency and strengthen the reconstruction, API Gateway is established and provides microservice for external access. It encapsulates interfaces of all microservices and take charge of transmitting and synthesizing requests from users.

For example, if construction engineers need to know the risk of water inflow in the current tunnel excavation face, they only need to visit the API Gateway. After receiving the request, API Gateway will transmit requests to the data service and the analysis service, and the analysis results will be merged back to the construction engineer. Another advantage of using API Gateway is that users can request the same interface no matter how the algorithm changes in the analysis service. The microservice architecture using API Gateway is shown in Fig. 4.

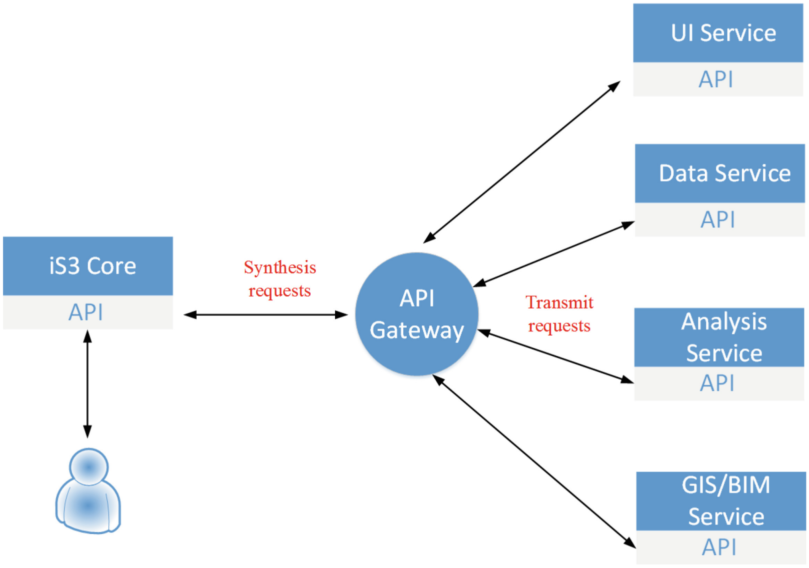


Fig. 4. Structure of microservice using API gateway.

5 Communication Mode of Service

5.1 Data Service Communication

The data requested is developed using Service-Oriented Architecture (SOA) [15], and implemented using REST architecture. REST requests treat all information as resources. There are four request types: (1) GET, meaning the operation of querying data; (2) POST, meaning the operation of adding data; (3) PUT, meaning the operation of updating data; and (4) DELETE, meaning the operation of deleting data. The target database is operated according to the URI and the request type. A self-increasing column which is the primary key named ID is added to improve index efficiency in the database. Examples of requests for construction progress data are listed in Table 5.

In real engineering cases, monitoring data could be dynamic and real-time. It is inappropriate to use request/response mode for such data, because the data loses the significance of analysis without effectiveness. Therefore, the subscription/publishing mode is chosen to transmit data. Once the new data is generated, it needs to be transmitted as soon as possible. The subscription/publishing mode are implanted based on the AMOP (Advanced Message Queuing Protocol). Data producers send the messages they generate through channels, and data receivers read the messages from channels. The communication mode of data service is shown in Fig. 5.

Table 5. Examples of REST requests for construction progress data service.

Request type	URI	Description
GET	/data/construction/progress?project=YN& part_no=left	Obtain the construction progress of the left part of the project named YN
GET	/data/construction/progress/?project=YN&id=1	Obtain the construction progress record whose id is 1 in the project named YN
POST	/data/construction/progress	Insert a construction progress record
PUT	/data/construction/progress/?project=YN&id=1	Update the construction progress record whose id is 1 in the project named YN
DELETE	/data/construction/progress/?project=YN&id=1	Delete the construction progress record whose id is 1 in the project named YN

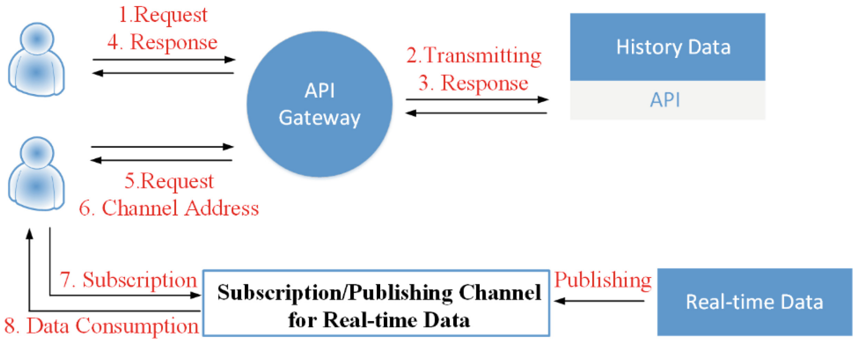


Fig. 5. Communication mode of data service.

5.2 Analysis Service Communication

Normally analytical services are suitable for request/response communication mode, such as the risk analysis of water inflow. The REST requests for the risk analysis service of water inflow are listed in Table 6.

Table 6. REST requests of the risk analysis service for water inflow.

Request type	URI	Description
GET	/analysis/tunnel/risk/waterflow	Requests for the risk analysis service for water inflow
POST		
PUT		
DELETE		

Each request is a new analysis of which the cost is low, and different operations are implemented according to the request type, including obtaining analysis results, adding analysis results, updating analysis results and removing analysis results. Take the risk analysis of water inflow as an example, the client make a request to API Gateway for analysis service, and API Gateway transmits the request to the REST API of the analysis service. When the risk analysis service is called, necessary parameters will be sent to the service. If analysis data is not loaded, the data service will be called by API Gateway automatically.

6 Service Discovery Mechanism

With the continuous expansion of the platform, the number of microservices may reach hundreds, so a service discovery mechanism is needed. The principle of service discovery is shown in Fig. 6. Platform users firstly query the network location of microservices from the service registry, then find the required services from all available service instances, and finally make requests. The service instance registers the network location to the service registry at startup and deletes it from the registry at service termination.

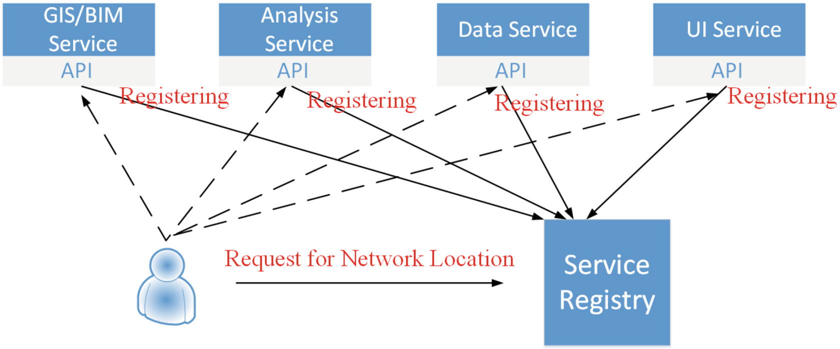
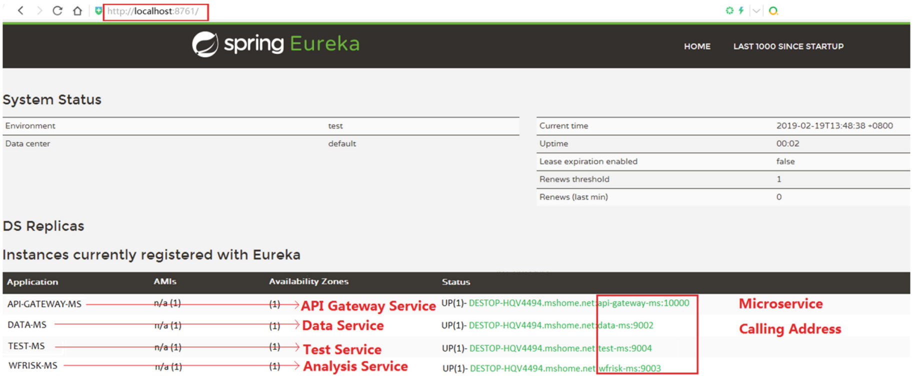


Fig. 6. Microservice discovery principle.

7 Validation of Microservices

7.1 Microservice Management Interface

In this paper, Sprint Boot is used as the framework of microservice implementation. Spring Cloud Gateway is used to implement the interface gateway of microservice. Kafka is used in the subscription/publishing message system. The management interface of microservice is shown in Fig. 7 using Eureka to register, manage and query the microservice instances. Four microservices have been implemented: data service, analysis service, test service and API Gateway service. Each microservice corresponds to a separate calling address.



System Status

Environment	test
Data center	default

Current time	2019-02-19T13:48:38 +0800
Uptime	00:02
Lease expiration enabled	false
Renews threshold	1
Renews (last min)	0

DS Replicas

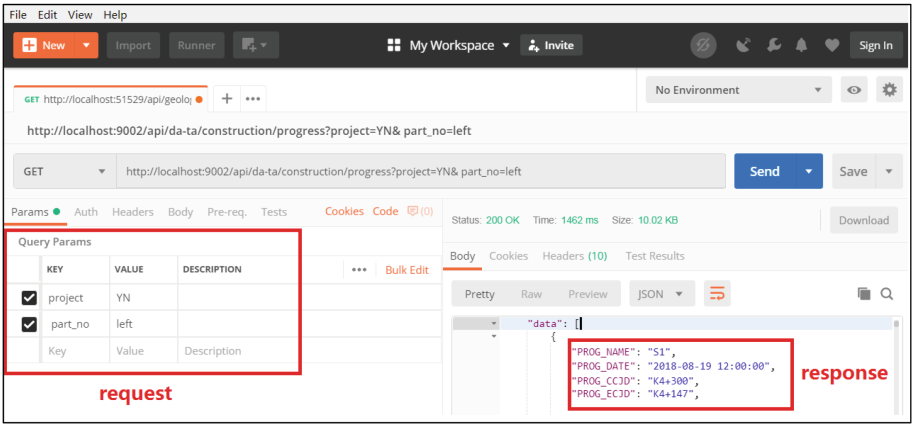
Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
API-GATEWAY-MS	n/a (1)	(1)	UP(1)- DESTOP-HQV4494.mshome.net:api-gateway-ms:10000
DATA-MS	n/a (1)	(1)	UP(1)- DESTOP-HQV4494.mshome.net:data-ms:9002
TEST-MS	n/a (1)	(1)	UP(1)- DESTOP-HQV4494.mshome.net:test-ms:9004
WFRISK-MS	n/a (1)	(1)	UP(1)- DESTOP-HQV4494.mshome.net:wfrisk-ms:9003

Fig. 7. Microservice management interface.

7.2 Data Microservice Instance

Take the data service of construction progress as a microservice instance. The request data contains the basic information of construction progress, such as the project's name, the part of the tunnel and the construction date. The data service returns the data according to the request, including the mileage of the construction progress of the initial lining, the mileage of the construction progress of the secondary lining and the bid section number, as shown in Fig. 8.



GET http://localhost:51529/api/geolo;

http://localhost:9002/api/da-ta/construction/progress?project=YN& part_no=left

GET http://localhost:9002/api/da-ta/construction/progress?project=YN& part_no=left

Status: 200 OK Time: 1462 ms Size: 10.02 KB

Query Params

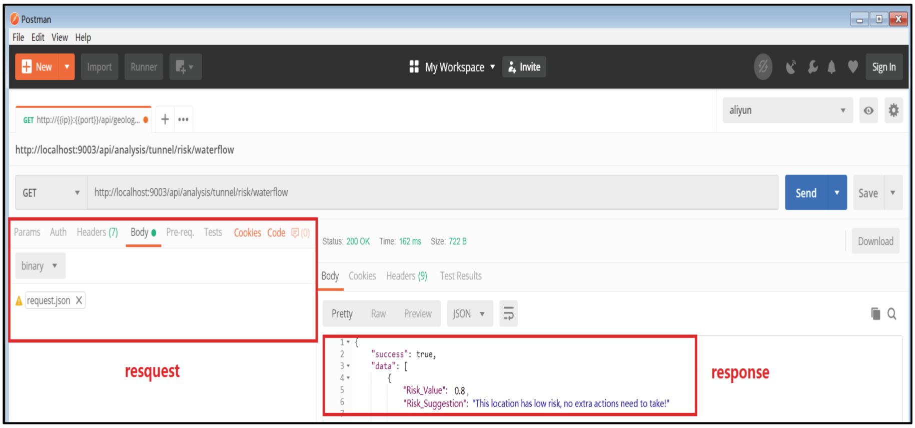
	KEY	VALUE	DESCRIPTION
☑	project	YN	
☑	part_no	left	
	Key	Value	Description

```
"data": [
    {
        "PROG_NAME": "S1",
        "PROG_DATE": "2018-08-19 12:00:00",
        "PROG_CCJD": "K4+300",
        "PROG_ECJD": "K4+147",
```

Fig. 8. Request and response of the construction progress data.

7.3 Analysis Microservice Instance

Take the analysis service of water flow risk as another microservice instance. The request content contains several analytical parameters, such as the carst scale, the carst distance, the advanced support, the support quality, the fill type, the emergence plan, the surrounding water, the geological forecast, the excavation disturbance and the water inflow of the tunnel face. The analysis service returns the result according to the request, including the risk value and the suggestion against the risk, as shown in Fig. 9.



GET http://{{ip}}:{{port}}/api/geolog...

http://localhost:9003/api/analysis/tunnel/risk/waterflow

GET http://localhost:9003/api/analysis/tunnel/risk/waterflow

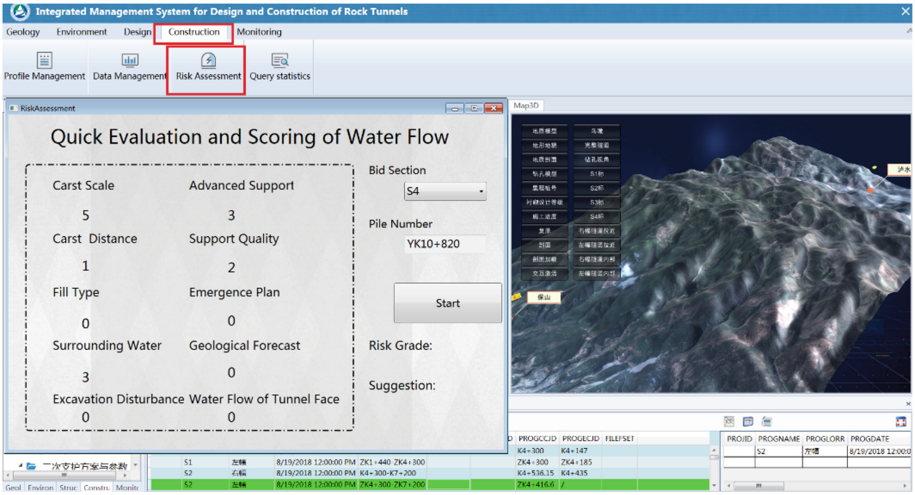
Status: 200 OK Time: 162 ms Size: 722 B

```
{
    "success": true,
    "data": [
        {
            "Risk_Value":  0.8 ,
            "Risk_Suggestion":  "This location has low risk, no extra actions need to take!"
```

Fig. 9. Request and response of the water flow risk.

7.4 Application Based on Microservice

The microservice framework designed in this paper is used as the service layer of iS3 to support the applications based on iS3. A construction management software of rock tunnels is developed based on iS3, and it provides the analysis function of the water inflow's risk, as shown in Fig. 10.


Integrated Management System for Design and Construction of Rock Tunnels
Geology Environment Design Construction Monitoring
Profile Management Data Management Risk Assessment Query statistics
RiskAssessment
Map3D
Start
二次支护方案与参数
Geol Environ Struc Constru Monitc

Quick Evaluation and Scoring of Water Flow

Carst Scale: 5
Advanced Support: 3
Carst Distance: 1
Support Quality: 2
Fill Type: 0
Emergence Plan: 0
Surrounding Water: 3
Geological Forecast: 0
Excavation Disturbance: 0
Water Flow of Tunnel Face: 0

Bid Section: S4
Pile Number: YK10+820
Risk Grade:
Suggestion:

					PROGCCJD	PROGECJD	FILEFSET
					K4+300	K4+147	
S1	左幅	8/19/2018 12:00:00 PM	ZK1+440-ZK4+300		ZK4+300	ZK4+185	
S2	右幅	8/19/2018 12:00:00 PM	K4+300-K7+200		K4+536.15	K4+435	
S2	左幅	8/19/2018 12:00:00 PM	ZK4+300-ZK7+200		ZK4+416.6	/	

PROJID	PROGNAME	PROGLORR	PROGDATE
	S2	左幅	8/19/2018 12:00:0

Fig. 10. Construction management software of rock tunnel based on iS3.

8 Conclusion

A distributed microservice architecture for infrastructure information management is proposed in this paper. The following three key aspects of the microservice architecture are identified:

(1) Standardizing the request and response data for the services, and providing unified access to API Gateway.
(2) The request/response mode is implemented using REST. Sprint Boot is used as the framework of microservice implementation, and Spring Cloud Gateway is used to implement the interface gateway of microservice.
(3) Aiming at the management of a large number of dynamic microservices, service discovery mechanism is implemented using Eureka.

Using above techniques, a complex standalone application, such as iS3, is decomposed into multiple services. Each service can be independently developed choosing the latest development technology, and provides corresponding API. The deployment of each microservice is independent, which makes continuous deployment much easier.

References

1. Yoo, C., Kim, J.M.: Tunneling performance prediction using an integrated GIS and neural network. Comput. Geotech. **34**(1), 19–30 (2007)
2. Li, X., Zhu, H.: Development of a web-based information system for shield tunnel construction projects. Tunn. Under. Space Technol. **37**, 146–156 (2013)
3. Zhang, W., Sun, K., Lei, C., Zhang, Y., Li, H., Spencer Jr, B.F.: Fuzzy analytic hierarchy process synthetic evaluation models for the health monitoring of shield tunnels. Comput.-Aided Civil Infrastruct. Eng. **29**(9), 676–688 (2014)
4. Yu, J., Lin, M.: BIM applications in subway installation. J. Inf. Technol. Civil Eng. Archit. **2** (2013). (in Chinese)
5. Bradley, A., Li, H., Lark, R., Dunn, S.: BIM for infrastructure: an overall review and constructor perspective. Autom. Constr. **71**, 139–152 (2016)
6. Chong, H.Y., Lopez, R., Wang, J., Wang, X., Zhao, Z.: Comparative analysis on the adoption and use of BIM in road infrastructure projects. J. Manag. Eng. **32**(6), 05016021 (2016)
7. Shou, W., Wang, J., Wang, X., Chong, H.Y.: A comparative review of building information modelling implementation in building and infrastructure industries. Arch. Comput. Methods Eng. **22**(2), 291–308 (2015)
8. Zhu, H., Li, X.: Digital underground space and engineering. J. Rock Mech. Geotech. Eng. **190**(11), 2277–2288 (2007). (in Chinese)
9. Li, X., Zhu, H., Zheng, L.: Digitalization and application of shield tunnels. Chin. J. Geotech. Eng. **31**(9), 1456–1461 (2009). (in Chinese)
10. Zhu, H., Li, X., Chen, X.: Digitization techniques for integrated construction and maintenance of infrastructure—part 1: theory and methodology. China Civil Eng. J. **48**(4), 99–110, 123 (2015). (in Chinese)
11. Zhu, H., Li, X., Lin, X.: Infrastructure Smart Service System (iS3) and its application. China Civil Eng. J. **51**(1), 1–12 (2018). (in Chinese)

12. Zhu, H., Li, X., Chen, X.: Digitization techniques for integrated construction and maintenance of infrastructure—part 2: engineering application. China Civil Eng. J. **48**(6), 114–121 (2015). (in Chinese)
13. Zhu, H., Wu, W., Li, X.: High-precision acquisition, analysis and service of rock tunnel information based on iS3 platform. Chin. J. Rock Mech. Eng. **36**(10), 2350–2364 (2017). (in Chinese)
14. Liu, X.: Dynamic Assessment and Control Technology of Safety Risk in Highway Tunnel Construction. China Communications Press Co., Ltd. (2015)
15. Newcomer, E., Lomow, G.: Understanding SOA with Web Services. Addison-Wesley (2005)