Université de Versailles/Saint-Quentin

Service-Oriented Architecture Project Report

« Implementing a House Loan Application Evaluation Composite Web Service »

Student: Zhao JIN
Project Supervisor: Yehia Taher

Entity: UVSQ (Versailles, 78)

Contents

1 Introduction **3**
- 1.1 Context . . . 3
- 1.2 Objectives . . . 3

2 Preliminaries **3**
- 2.1 Service-Oriented Architecture . . . 3
- 2.2 Representational state transfer[2] . . . 4

3 Implementation **4**
- 3.1 Architecture . . . 4
- 3.2 Application . . . 5
- 3.3 Information Extraction Service . . . 5
- 3.4 Resolvability Verification Service . . . 5
- 3.5 Property Evaluation Service . . . 5
- 3.6 Approval Decision Service . . . 6

4 Conclusion **6**
- 4.1 Perspectives . . . 6
- 4.2 Benefits . . . 6
- 4.3 Future work . . . 7

1 Introduction

In this section, we will give the context of the project in Section 1.1 and the objectives in Section 1.2.

1.1 Context

The Home Loan Application Evaluation Composite Web Service is designed to automate the evaluation of home loan applications using specialized web services. It allows customers to submit home loan applications expressed in natural language. The service integrates application text business information extraction, credit check, property valuation, and approval decision components to provide a complete and accurate assessment of loan applications. We will use REST API for this implementation.

1.2 Objectives

This service aims to provide a single interface for customers who wish to apply for a home loan. The service coordinates the web services necessary to evaluate the customer's request, ensuring a smooth and automated process.

2 Preliminaries

In this Section, we will give the preliminaries about the SOA (Section 2.1) and the REST (Section 2.2).

2.1 Service-Oriented Architecture

The service-oriented architecture (SOA) is an architectural style that focuses on discrete services instead of a monolithic design. A service[1]:

- is a logical representation of a repeatable business activity that has a specified outcome (e.g., check customer credit, provide weather data, consolidate drilling reports),
- is self-contained,
- may be composed of other services,
- is a "black box" to consumers of the service.

The SOA architectural style has the following distinctive features[1]:

- It is based on the design of the services – which mirror real-world business activities – comprising the enterprise (or inter-enterprise) business processes.
- Service representation utilizes business descriptions to provide context (i.e., business process, goal, rule, policy, service interface, and service component) and implements services using service orchestration.
- It places unique requirements on the infrastructure – it is recommended that implementations use open standards to realize interoperability and location transparency.

- Implementations are environment-specific – constrained or enabled by context and must be described within that context.
- It requires strong governance of service representation and implementation.
- It requires a "Litmus Test", which determines a "good service".

2.2 Representational state transfer[2]

Representational state transfer is a software architectural style that was created to guide the design and development of the architecture for the World Wide Web. REST defines a set of constraints for how the architecture of a distributed, Internet-scale hypermedia system, such as the Web, should behave. The REST architectural style emphasizes uniform interfaces, independent deployment of components, the scalability of interactions between them, and creating a layered architecture to promote caching to reduce user-perceived latency, enforce security, and encapsulate legacy systems.

3 Implementation

In this Section, we will give the architecture of the system in Section 3.1. Then, we talk about the details of the Information Extraction Service in Section 3.3, the Resolvability Verification Service in Section 3.4, the Property Evaluation Service in Section 3.5 and the Approval Decision Service in Section 3.6.

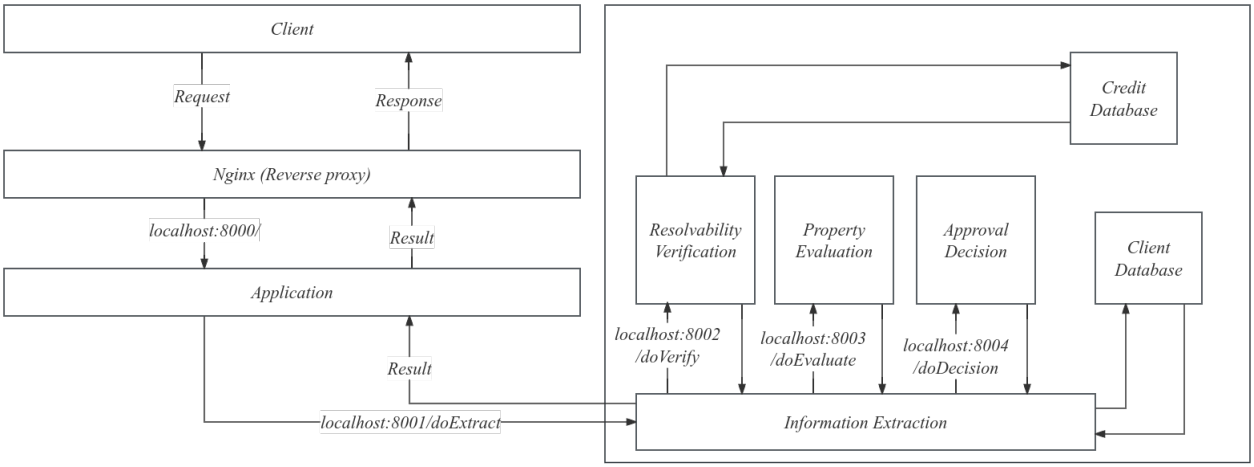

Figure 1: Architecture

3.1 Architecture

Figure 1 shows the architecture of the RESTful API implementation. We use the reverse proxy technique to expose the application with the port of 8000. The other services can know each other in the internal network. The client can only have access to the Nginx.

3.2 Application

The application part is a spring-boot application that receives the client's application in the natural language and returns the final decision on the loan. It is also a client part of REST to call the internal services.

3.3 Information Extraction Service

The information extraction service contains the following stages:

1. Normalization of the application, e.g., remove the illegal characters and turn the characters to lowercase.

2. Extraction of the information. We use a pattern matcher to resolve the critical information such as the name, address, telephone number, income, spending, and the amount and duration of the loan. For future work, we can replace this stage through a natural language processing (NLP) model to do the linguistic analysis and identification of different entities.

3. If there are some errors in data, we will inform the client immediately.

4. Save the client data into the database.

5. Call Resolvability Verification Service and Property Evaluation Service, then call the Approval Decision Service to give the final decision with the client information and previous results of the two services.

3.4 Resolvability Verification Service

The resolvability verification aims to do the creditable analysis. It contains the following elements:

1. Calculate the score of the credit. We normalize the time and the amount of the loan and make the proportion with 40% time and 60% amount. With the increasing time or amount, the score will be less. If the score does not pass 50%, we think this application is not reasonable.

2. Analyze the income and spending. This part is to compare if the client can refund the current loan.

3. Find the history data of the client. We try to find all loan applications of the client and analyze the ability to refund. If the client already has a loan to repay and cannot refund after this credit, we will refuse the application.

3.5 Property Evaluation Service

This part evaluates the properties of a house. The goal is to analyze the reasonability for applying for a loan. Since this part should integrate professional analysis and the human being to inspect the house, we give only the following interfaces. The others can implement them in their own way.

- Analysis of the value of a house to verify if the loan is reasonable and enough or not too much.

- Inspection of the house to collect information on the quality of the house and give feedback on the house if the client can buy it.
- Analysis of the legal and regulatory compliance. This part is to verify if the client has the authorization to buy the house.

3.6 Approval Decision Service

This service takes the information of the client and all results of the resolvability verification and property evaluation. The same reason as the Property Evaluation Service. It contains some particular parts to implement in the future in a personalized way.

- Risk analysis. The risk analysis uses the bank's information to analyze the history of pay information to draw a virtual profile of the client to analyze if the client has a high or low possibility of refunding the loan.
- Politics analysis. This part is to analyze the validation of the current credit. It will vary between different institutions. To simplify the processes, we give the $true$ as a return to allow all types of credit.
- Prediction. We use machine learning with multiple factors to simulate the future refunding of the client.
- Making decision. This part will give the final decision depending on the previous results of verification and evaluation, as well as the risk analysis, politics analysis, and prediction to give the final decision. After calculating the final decision, it will return the result to the Information Extraction Service and give it to the Application. Then, the client will receive the result.

4 Conclusion

We have constructed a system of a house loan application evaluation composite web service with a SOAP implementation. This section will give the perspectives for this system (Section 4.1). Finally, we conclude what we learned from this project (Section 4.2) and future work in Section (4.3).

4.1 Perspectives

We have constructed the system. However, there are many things to improve, for example, the NLP model, the machine learning algorithm, the dataset to train, etc. With a limited time, we cannot do everything perfectly. But we have the entire structure. For the future work, we can implement them directly.

4.2 Benefits

Thanks to this project, we learned the usage of the service-oriented architecture. With the SOA, I find that whatever the need, we separate each module into a single service. We find even extend the service we want to meet with the increasing demand. For example, if the approval decision service needs more capacity, we can extend it horizontally without modifying other services.

4.3 Future work

The Spring organization proposed a chain of frameworks for web services. For example, the SpringCloud-Netflix. We can register the services with Eureka, keep load balancing with Ribbon and Feign, and use Zuul for the API gateway.

References

[1] SOA Work Group. *SOA Source Book*. Van Haren, 2009.

[2] Carlos Rodríguez, Marcos Báez, Florian Daniel, Fabio Casati, Juan Carlos Trabucco, Luigi Canali, and Gianraffaele Percannella. "REST APIs: A Large-Scale Analysis of Compliance with Principles and Best Practices". In: *Web Engineering - 16th International Conference, ICWE 2016, Lugano, Switzerland, June 6-9, 2016. Proceedings*. Ed. by Alessandro Bozzon, Philippe Cudré-Mauroux, and Cesare Pautasso. Vol. 9671. Lecture Notes in Computer Science. Springer, 2016, pp. 21–39. DOI: 10.1007/978-3-319-38791-8_2. URL: https://doi.org/10.1007/978-3-319-38791-8%5C_2.