PG3402

Microservices

21.10.2023

Arbeidskrav

Høyskolen Kristiania

Høst 2023

Denne besvarelsen er gjennomført som en del av utdannelsen ved Høyskolen Kristiania. Høyskolen

Introduction

In this project, our goal is to develop a banking system utilizing microservices as the core architectural framework. The decision to create a bank application stems from two primary reasons:

1. **Rich Source of Inspiration**: Banks offer clear and easily understandable examples that serve as a rich source of inspiration for our application.
2. **Comprehensive Functionalities**: Modern banks have evolved into multifaceted entities, offering a wide array of services beyond traditional banking. This complexity aligns with our vision to create a microservice driven system filling various functionalities.

Given the magnitude of our project and the constraints of limited time, it's crucial to define a Minimal Viable Product (MVP) to focus our efforts effectively. This entails set out the essential components of our system that need to be developed first. Secondly, we must outline the supplementary features we plan to implement and establish a general design.

Software Architecture Overview

Microservices represent a structural software pattern, dictating the overarching organization of our application. Essentially, this approach involves split the inner workings of a program into numerous small, autonomous services that operate independently while remaining interconnected, forming a cohesive whole, hence the term "microservices."

The primary objective of this architectural design is to enhance the robustness of larger applications, especially when their complexity reaches a point where adding new features risks disrupting existing code. Traditional methods of scaling, like spinning up additional instances, become insufficient. The fundamental advantage lies in creating a resilient program where individual components can fail without compromising the functionality of other features. If a specific part of the program demands more capacity, increasing instances of that particular feature resolves the performance issue. Moreover, this approach streamlines development by enabling each team to concentrate on specific features within a dedicated service, utilizing technologies tailored to the service's requirements, independent of other features' dependencies.

While microservices offer substantial benefits, particularly for enterprise applications, they come with challenges. By simplifying the lower levels of code, complexities shift to the architectural level. The division of services necessitates the integration of various technologies to address the infrastructure challenges inherent to microservices.

In essence, microservices provide a scalable and resilient framework for building complex applications, empowering developers to focus on specialized features. However, managing the complexities at the architectural level and integrating diverse technologies are critical considerations in implementing this approach effectively.

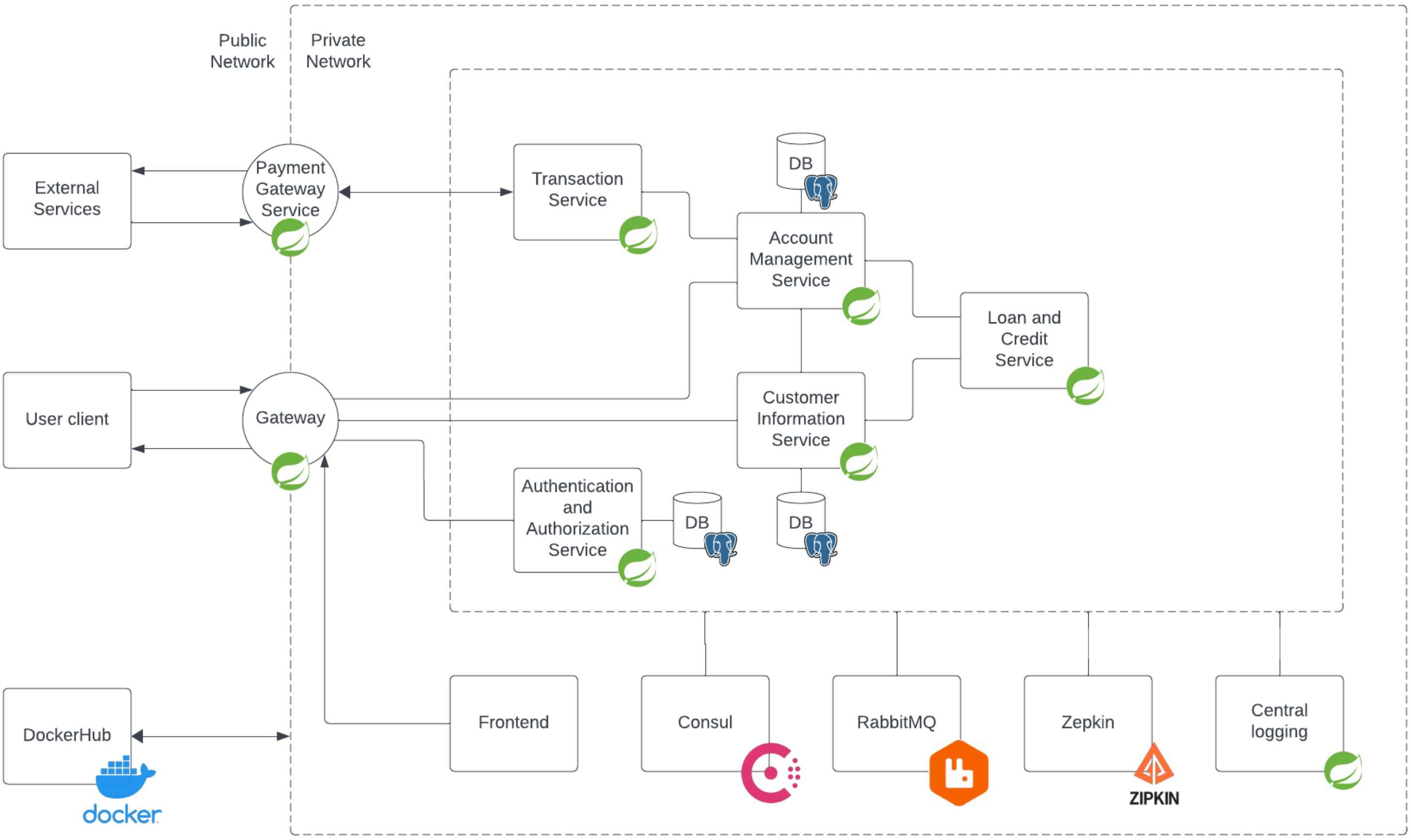
User Stories

As a user, envisioning the interaction flow within the application becomes effortless. With well-defined user stories, understanding the implemented functionalities provides a clear insight into the backend operations required. These stories effectively outline the blueprint of what the application aims to achieve.

In crafting the Minimum Viable Product (MVP), the primary focus is on catering to the needs of the everyday private individual using the bank's services. If time permits, the aspiration is to extend and enrich the feature set tailored to the bank's private customers. Furthermore, as we explore the expansion of features, there is an exciting opportunity to interconnect certain functionalities with a colleague's application, potentially enhancing the user experience and creating synergy between both applications.

- As a private customer, I want to sign in to my bank and see what my account value is.
- As a private customer, I want to make an account to this bank.
- As a private customer, I want to transfer money between my personal account and others.
- As a private customer, I want to pay my bills.
- As a private customer, I want to apply for loan and / or credit card.
- As a private customer, I want to apply for insurance (collaboration with college).
- As a private customer, I want to connect my account to a money transfer service (collaboration with college).

Microservices Descriptions



- **Consul:** A service that has multiple purposes. Firstly, it serves out stored configurations for services that connect to it. Secondly, it works as a look-up catalog for other services to tell them how to contact the needed service. Thirdly it does regular health checks on services to make sure they are available for others to connect to.
- **RabbitMQ:** A message broker who receives messages and distributes them to other services using queues. Gives the advantage of being asynchronous and are not dependent on the original sender for the receiving service to execute its job.
- **Zipkin:** A tracing tool used to help and track the lifetime of an incoming request to the system. Provides graphical data from tracing ID that can be retrieved from the centralized logging service.
- **Central logging:** A service that receives logs from all the services through RabbitMQ. Depending on configuration these logs can potentially be stored.
- **Frontend:** A server that sends out the frontend part of the application to the regulare user client.

- **Gateway and Payment Gateway:** Works as a gate that receives incoming requests from the public network, checks if they meet requirements. Then the messages will be marked with a tracing Id and forwarded to the correct service for further processing.
- **Authentication and Authorization Service:** Takes in login request from the client and approves access to use rest of the application.
- **Customer Information Service:** Manages customer profiles and personal information.
- **Account Management Service:** Responsible for managing customer accounts. Handles account creation, deletion and updating account information.
- **Transaction Service:** Manages financial transactions and processes transfers, payments, deposits, and withdrawals.
- **Loan and Credit Service:** Manages loan and credit applications. Calculates loan eligibility, interest rates and repayments schedules.