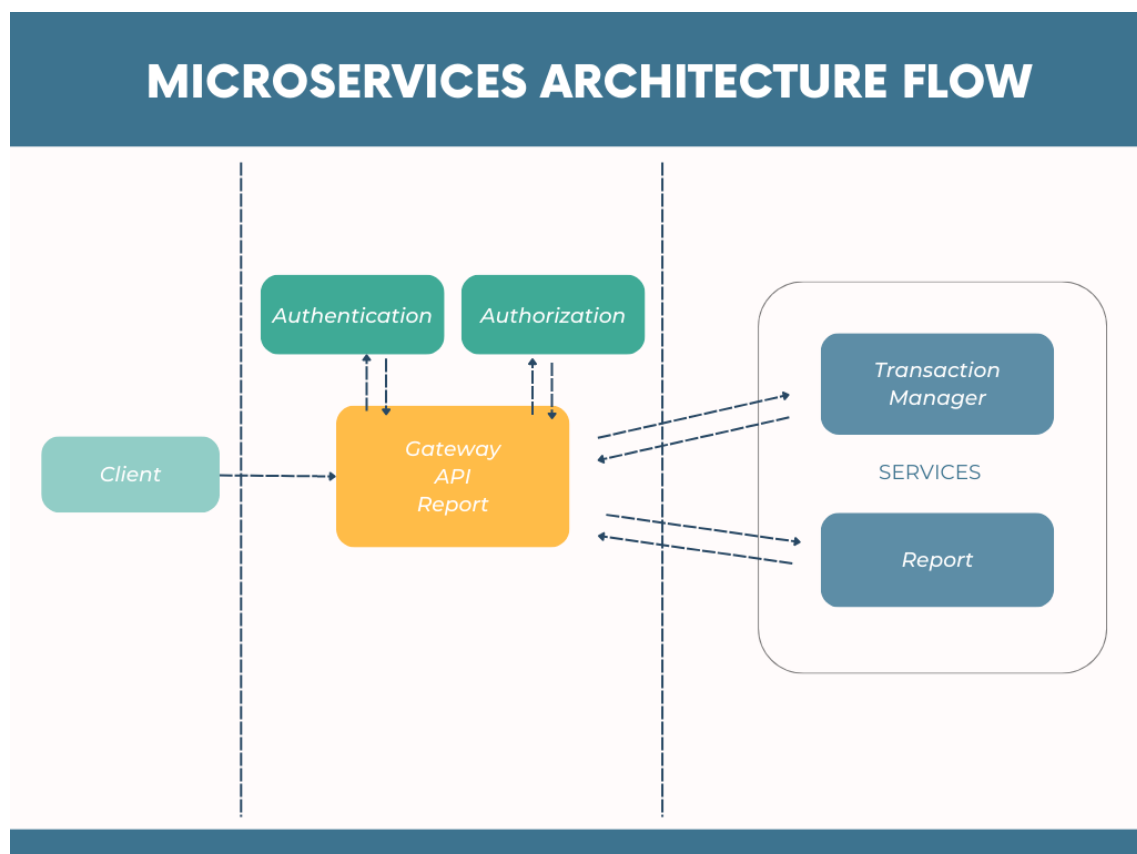


Documento de Arquitetura de Software- Sistema de Fluxo de Caixa

Introdução

O presente documento descreve a arquitetura de software para um sistema de fluxo de caixa, que foi dividido em quatro microserviços distintos. Esses microserviços são: Autorização e Autenticação, Gateway, Transaction Manager e Report. Cada microserviço desempenha um papel específico no sistema, visando garantir a segurança, o controle de rotas, a validação e persistência das transações, bem como a geração de relatórios consolidados diários.



Visão Geral da Arquitetura

A arquitetura do sistema é baseada em uma abordagem de microsserviços, onde cada componente é responsável por uma funcionalidade bem definida. A comunicação entre os microserviços é realizada por meio de APIs RESTful. A autenticação e autorização são realizadas utilizando o Spring Security, OAuth 2.0 e JWT. O Spring Cloud Gateway é utilizado para o gerenciamento de rotas.

2.1 Microserviço de Autorização e Autenticação

O microserviço de Autorização e Autenticação é responsável por garantir a segurança do sistema. Ele recebe todas as solicitações e valida a autenticidade do token JWT junto com o serviço de autorização. Caso a autenticação seja bem-sucedida, a solicitação é redirecionada para o microserviço correspondente. Esse microserviço é implementado utilizando o Spring Security e é mapeado como uma rota no gateway.

2.2 Microserviço de Gateway

O microserviço de Gateway é responsável pelo controle de rotas do sistema. Todas as solicitações são enviadas para o Gateway, que as encaminha para o microserviço apropriado com base na rota especificada. O Spring Cloud Gateway é utilizado para o gerenciamento de rotas e o roteamento das solicitações.

2.3 Microserviço de Transaction Manager

O microserviço de Transaction Manager é responsável pelas validações de recebimento e pagamento, bem como pela persistência dos dados no banco de dados. Ele é mapeado como a rota "/api" no gateway. Uma consideração importante para esse microserviço é a necessidade de manter a integridade do saldo remanescente após cada transação, mesmo em situações de transações concorrentes. Para atingir esse objetivo, foi aplicado o padrão de projeto Transaction Script, seguindo os princípios ACID (Atomicidade, Consistência, Isolamento e Durabilidade). Além disso, foram utilizadas transações otimistas para melhorar a performance do método "executetransaction". O banco de dados PostgreSQL foi escolhido devido à sua natureza relacional, que facilita a implementação do padrão Transaction Script.

2.4 Microserviço de Report

O microserviço de Report é responsável pela geração de relatórios consolidados diários. Para cada transação executada pelo Transaction Manager, o Report deve ser atualizado. Para permitir essa comunicação entre os microserviços, foi utilizada uma abordagem baseada em mensageria com Kafka e Apache Camel. O Transaction Manager atua como produtor (producer) enviando as informações relevantes para o Kafka, enquanto o Report atua como consumidor (consumer), recebendo as mensagens do Kafka e atualizando os relatórios. O banco de dados MongoDB foi escolhido para esse microserviço devido à sua estrutura flexível e à baixa latência de bancos de dados não relacionais.

Conclusão

A arquitetura proposta para o sistema de fluxo de caixa, dividido em quatro microserviços distintos, proporciona uma abordagem escalável, modular e segura. A separação das responsabilidades em cada microserviço permite um desenvolvimento mais ágil e facilita a manutenção do sistema. Além disso, a utilização de tecnologias como Spring Security, OAuth 2.0, JWT, Spring Cloud Gateway, PostgreSQL, MongoDB, Kafka e Apache Camel contribuem para a implementação eficiente e confiável do sistema.