



Arquitetura

Para solucionar a questão de existir transações simultâneas, elaborei a seguinte arquitetura, explicarei cada componente desta.

Load Balancer: Faz a distribuição das solicitações de entrada entre as instâncias disponíveis do API Gateway e dos microsserviços, fornecendo assim alta disponibilidade e escalabilidade.

API Gateway: Roteia as solicitações dos clientes para os serviços apropriados. Implementa também autenticação e autorização.

Transaction Service: Principal responsável por orquestrar o processo de transação. Garante também a execução rápida e controla a concorrência utilizando o Redis para locks distribuídos.

Redis (Distributed Locking): Responsável por implementar bloqueios distribuídos para garantir exclusão mútua. Utiliza um lock por conta de cartão de crédito para garantir que apenas uma transação por conta seja processada de cada vez.

Account Service: Gerencia informações de conta e saldo. Interagindo com a camada de banco de dados para verificar e atualizar o saldo da conta.

Database: Gerencia os dados persistentes do sistema. Implementa transações ACID para garantir a consistência e isolamento dos dados.

Cache Layer (Redis): Armazena dados frequentemente acessados em memória para acesso rápido, reduzindo a latência de leitura/escrita frequente.

Logging & Monitoring(Prometheu e Grafana): Coleta métricas e logs de todos os componentes. Fornecendo visualizações e alertas para identificar e resolver problemas de desempenho e de concorrência.