
Java – Microservices

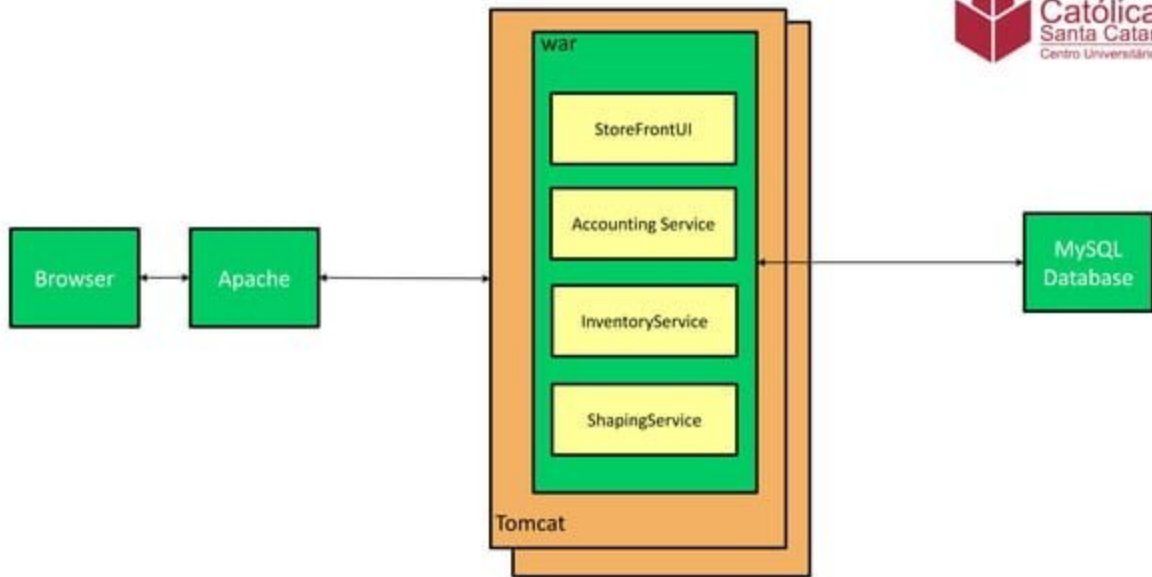
PROFESSOR: FABIO TAVARES DIPPOLD

ALUNOS: LUIS MACHADO, OSMAR PETRY, E SERGIO ROIKO

A arquitetura monolítica

Uma aplicação monolítica é auto-suficiente, e independente de outras aplicações.

A filosofia dessa arquitetura é que o aplicativo seja responsável não apenas para executar uma determinada tarefa, mas sim para executar todos os passos necessários para completar uma função específica.



A arquitetura microservices

Tem em seu design o objetivo fazer com que o código seja menos frágil e, fazer com que softwares complexos e de larga escala mudem.

História

Uma workshop de arquitetos de software em maio de 2011 em Venice utilizou o termo "microservice" para descrever o que os participantes viram como um estilo arquitetônico comum que muitos deles estiveram recentemente explorando.

Em 2012, Adrian Cockcroft com a Netflix foi pioneiro neste estilo em uma escala web.

Vantagens

1. Base de código menor
2. Implementação continua
3. Escalado de forma independente
4. Facilita escalação do desenvolvimento
5. Melhor isolamento de falhas
6. Elimina compromisso de longo prazo com a stack de tecnologia
7. Mais aberta a experimentações

Desvantagens

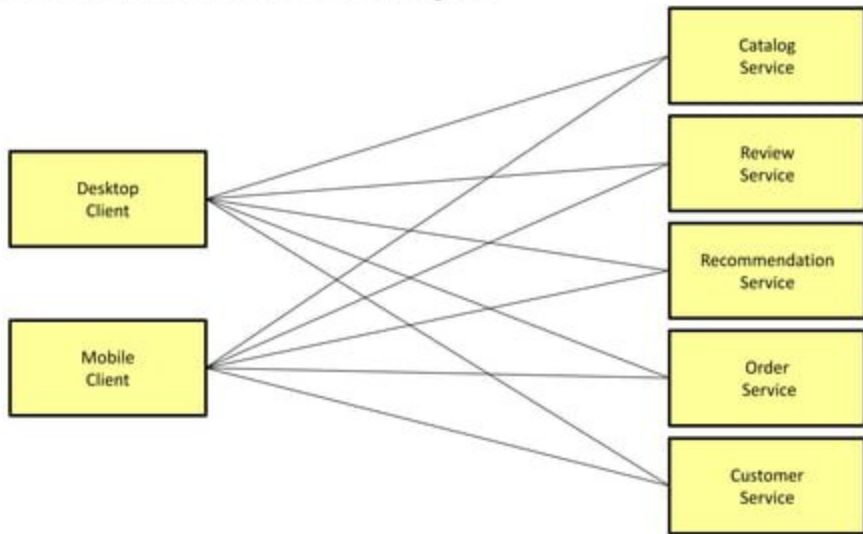
1. Maior complexidade do código em sistemas distribuídos
2. Complexidade operacional significativa
3. Requer plano ordenado de implementação de serviços
4. É necessário alto nível de automação
5. Momento deve-se adotar a arquitetura

Mecanismos de comunicação

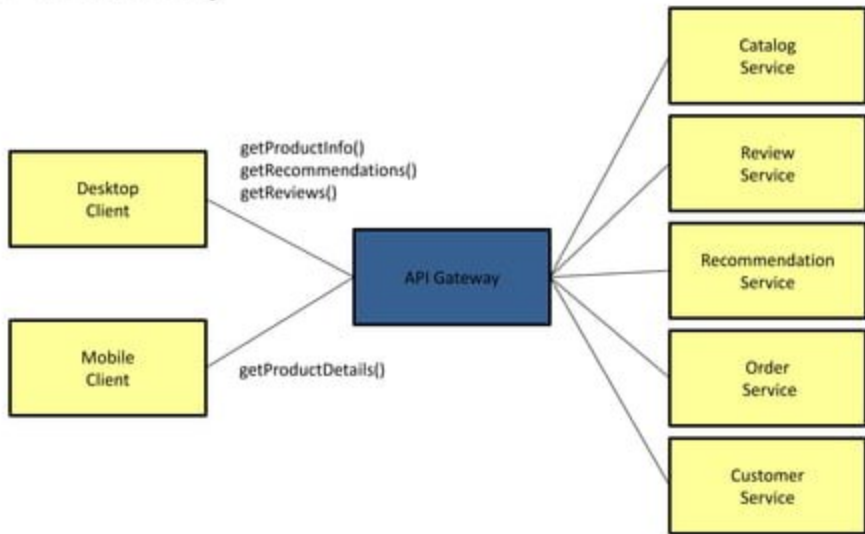
Em uma arquitetura de microservices, os padrões de comunicação entre clientes e a aplicação, bem como entre os próprios componentes da aplicação, são diferentes de uma aplicação monolítica.

Primeiramente, são abordadas as questões de interação entre clientes e os microservices. Em seguida, são abordados os mecanismos de comunicação entre microservices

Chamada direta aos serviços



API Gateway



Mecanismos de comunicação inter-service

Outra grande diferença da arquitetura de microservices é a forma de interação entre os diferentes componentes da aplicação.

Em uma aplicação monolítica, componentes comunicam-se através de chamadas de métodos

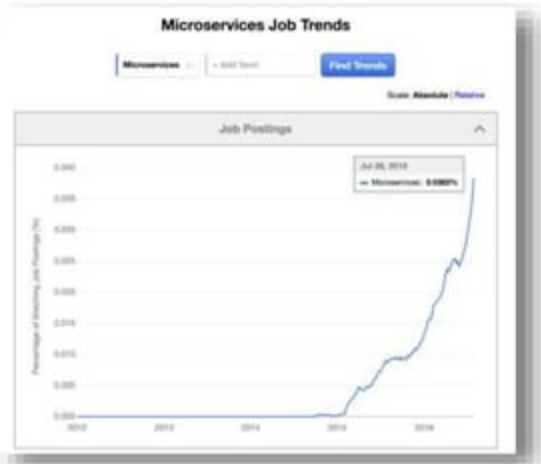
Mas em uma arquitetura de microservices, diferentes serviços são executados em diferentes processos.

Consequentemente, os serviços devem utilizar uma comunicação entre processos (IPC - inter-process communication)

HTTP síncrono

Mensagens Assíncronas

Procura por profissionais



Casos de sucesso



Referências

Microservices: Decomposing Applications for Deployability and Scalability -

<https://www.infoq.com/articles/microservices-intro>

Pattern: Monolithic Architecture - <http://microservices.io/patterns/monolithic.html>

Pattern: Microservices Architecture - <http://microservices.io/patterns/microservices.html>

Microservices with Spring - <https://spring.io/blog/2015/07/14/microservices-with-spring>

Microservices - <https://en.wikipedia.org/wiki/Microservices#History>

Monolithic application - https://en.wikipedia.org/wiki/Monolithic_application

