

Engenharia de Software

Software Concorrente e Distribuído

Universidade Federal de Goiás

Marcus Mitra Muniz Inácio - 201508741

O que são microsserviços? (Pontos chaves do vídeo do podcast Hipsters.tech sobre microsserviços)

SOAP e WSDL

São respectivamente um protocolo de acesso para troca de informações e um XML que descreve um web service, sendo considerados “anciãos” nos protocolos de acesso mas ainda sim relevantes, ainda pode ser aplicado em microsserviços apesar de hoje em dia ser mais comum ouvir falar em REST.

Web Services

Web Services são basicamente os microsserviços, o contraste é que Web Services é uma tecnologia, enquanto microsserviços são uma arquitetura de software que podem ser implementados com web services.

Implantação e integração mais fáceis

Com a utilização de microsserviços, fica mais fácil implementar e integrar novas funcionalidades, com equipes menores e mais fáceis de serem geridas. Gerando muitas aplicações independentes e, conseqüentemente, um sistema distribuído.

Testes mais rápidos

Ao invés de ter que realizar testes extensos, ao desenvolver em microsserviços é possível ter testes mais enxutos e que ainda cobrem o escopo total. Facilitando e

agilizando os testes. Novamente, entra nas características de sistemas distribuídos pois são aplicações independentes.

Unidades Independentes

Unidades Independentes se trata da questão base de sistemas distribuídos, “nós” de um sistema que são independentes.

Camada de Dados

Camada de conjunto de informações(geralmente banco de dados) que fornece um acesso simplificado a estes dados. O parênteses com SD(Sistemas Distribuídos) é que geralmente os microserviços(“nós”) contém seu próprio banco de dados, pois eles precisam ser independentes uns dos outros, ao contrário do que seria uma arquitetura com camada de dados.

Monolitos

Aplicações não distribuídas, é um sistema único que roda em um único processo, exatamente ao contrário de um SD.

Seu próprio banco de dados

Na arquitetura de SD e microserviços, geralmente cada uma dessas aplicações tem seus dados autocontidos, seu próprio banco de dados.

Request e Response

É uma maneira de comunicação entre “nós” de um SD, ou seja, microserviços, que consiste em requisições e respostas.

JSON

Um padrão de troca de dados simples e rápida entre sistemas, que é o que os microserviços de um SD utilizam para transferir requisições e respostas.

Serviços de Mensageria

É um serviço utilizado como intermédio de troca de mensagens entre sistemas, com o objetivo de propiciar a integração destes serviços, o que é essencial em um sistema distribuído.

HTTP

Um protocolo de redes, utilizado em SDs para a comunicação de “nós” de um sistema.

GRPC e Performance

É um sistema open source de chamada de procedimentos desenvolvido pela Google, atualmente se tornando predominante no mercado e sendo cada vez mais utilizado como padrão em SDs. A tentativa é prover uma alternativa ao JSON com maior performance.

Apache Thrift

Outro modelo de chamada de procedimentos, porém não tão predominante quanto o GRPC.

Service Bus e Apache Kafka

Service Bus é uma arquitetura de construção de software que geralmente é implementado em software “middleware” para controle de “streams” de dados. Já o Apache Kafka é uma tecnologia similar mas para tratar especificamente de SDs.

Desenvolver Novas Funcionalidades

Aqui os anfitriões do podcast entram na discussão sobre a migração de aplicações “monolitos” para microsserviços, pontuando que é quase impossível fazer essa migração de uma vez só, tendo em vista que a arquitetura de sistemas distribuídos é muito diferente da arquitetura de monolitos, que é mais fácil desenvolver novas funcionalidades pensando em microsserviços e aos poucos ir fazendo essa transição.