

Resenha do artigo Microservices

Aluno: João Vitor Romero Sales

Disciplina: Projeto de Software

Professor: João Paulo Aramuni

O artigo Microservices, elaborado por James Lewis e Martin Fowler, explora o emergente estilo arquitetural de microsserviços, contrastando-o com a abordagem monolítica tradicional que dominou o desenvolvimento de software por muitos anos. Em essência, a arquitetura de microsserviços propõe a construção de aplicações como um conjunto de pequenos serviços independentes, cada um executando em seu próprio processo e comunicando-se através de mecanismos leves, (geralmente uma API RESTful). Essa abordagem se destaca por sua flexibilidade, escalabilidade e capacidade de implantação independente, características cada vez mais requisitadas no mundo do desenvolvimento de software moderno, e que se contrastam com a rigidez e as limitações dos monolíticos.

Os autores iniciam comparando microsserviços com aplicações monolíticas, delineando as vantagens e desvantagens de cada modelo. Enquanto monólitos podem ser mais simples e fáceis de se construir inicialmente, esta forma de arquitetura torna-se uma barreira para a escalabilidade, atualizações e gerenciamento de código à medida que a aplicação cresce em complexidade. Os microsserviços, por outro lado, são modulares, possibilitando que cada serviço seja desenvolvido, implantado e escalado de forma independente, proporcionando agilidade para equipes de desenvolvimento, facilitando a adoção de novas tecnologias e acelerando o ritmo de inovação.

O artigo então aborda as características definidoras da arquitetura de microsserviços. A componentização por meio de serviços é fundamental, transformando cada serviço em uma unidade substituível e atualizável. A organização em torno do “business capability” (capacidades do negócio) é outro pilar, garantindo que os serviços reflitam as lógicas de negócio de forma coesa e sejam gerenciados por equipes multifuncionais. Essa estrutura promove a autonomia e a responsabilidade do time, alinhadas com o conceito de “produtos, não projetos”.

A comunicação entre microsserviços é um aspecto crucial neste tipo de arquitetura, e o artigo defende a filosofia de “smart endpoints and dumb pipes” (endpoints inteligentes e canais burros). Os serviços encapsulam a lógica de negócio e se comunicam através de protocolos como HTTP/REST, evitando complexidade desnecessária e facilitando a interoperabilidade. Essa abordagem contrasta com a tradicional Arquitetura Orientada a Serviços (SOA), que se apoiava

em “Enterprise Service Buses” (ESBs) complexos e centralizados, ocasionando falhas.

Os autores colocam a governança descentralizada como um princípio central dos microsserviços, incentivando os times a escolherem as tecnologias e soluções mais adequadas para cada serviço, sem a necessidade de conformidade com padrões rígidos impostos por um superior. Essa liberdade estimula a experimentação e a inovação, permitindo que equipes selecionem as melhores ferramentas para o produto, sejam elas linguagens de programação, bancos de dados ou frameworks. Um exemplo desta flexibilidade é o “Polyglot Persistence” (persistência poliglota), onde diferentes serviços podem usar diferentes tecnologias de banco de dados.

A automação da infraestrutura é outro elemento fundamental da arquitetura de microsserviços, essencial para gerenciar o ciclo de vida de múltiplos serviços em ambientes distribuídos. Ferramentas e práticas de Continuous Integration/Continuous Delivery (CI/CD) são amplamente utilizadas para automatizar os processos de desenvolvimento, testes e implantação, garantindo a confiabilidade e a agilidade necessárias para escalar os serviços.

O design para falhas é uma consideração primordial em arquiteturas distribuídas, e o artigo destaca a importância de construir microsserviços resilientes a falhas, implementar padrões como o Circuit Breaker, que interrompe a comunicação com um serviço indisponível para evitar falhas em cascata, é essencial. Visando não apenas resolver problemas, mas evitá-los/prevê-los é essencial, e neste tópico o monitoramento em tempo real de indicadores de saúde e desempenho, tanto a nível de infraestrutura quanto de negócio, são cruciais para detectar e possivelmente solucionar problemas rapidamente, minimizando o impacto para os usuários e garantindo a disponibilidade do sistema.

O artigo reconhece a complexidade relacionada à gestão de um sistema de microsserviços, destacando a necessidade de expertise técnica, automação robusta e uma cultura organizacional que valorize a autonomia e a responsabilidade das equipes. Os autores argumentam que microsserviços não são uma bala de prata, ou seja, uma solução universal, e que a decisão de adotar essa arquitetura deve ser cuidadosamente ponderada, levando em consideração os custos, vantagens e desvantagens.

Apesar do potencial dos microsserviços apresentado, os autores reconhecem que a maturidade dessa abordagem ainda está em desenvolvimento, sendo necessário mais tempo e experiência para avaliar o impacto a longo prazo dos microsserviços em relação à sua manutenibilidade, evolução e gerenciamento de complexidade.

O artigo de Lewis e Fowler serve como um guia abrangente e inicial para a arquitetura de microsserviços, analisando seus fundamentos, benefícios, desafios e o papel que desempenha na evolução da arquitetura de software. A mensagem principal é clara: os microsserviços são uma ferramenta poderosa, mas sua aplicação requer planejamento e uma profunda compreensão de suas vantagens e desafios.