**RESENHA CRITICA DO ARTIGO "Microservices"**

**Resenha da obra:** LEWIS, James; FOWLER, Martin. Microservices , 25 de Março de 2014.

A presente resenha se presta a analisar o artigo Microservices de James Lewis e Martin Fowler.

O artigo se divide em Resumo, introdução, nos capitulos: *Characteristics of a Microservice Architecture, Componentization via Services, Organized around Business Capabilities, Products not Projects, Smart endpoints and dumb pipes, Decentralized Governance, Decentralized Data Management, Infrastructure Automation, Design for failure, Evolutionary Design, Are Microservices the Future?*

A proposta do artigo é abordar as principais ideias e caractertisticas da arquitetura de microserviços. Os autores reconhecem que por mais que tenha se tornado muito popular, existe uma lacuna nas informações a respeito do que são e como implementar.

Nessa via, os autores definem microserviços como uma abordagem para desenvolver uma aplicação com uma junção de pequenos serviços que se comunicam ente si, geralmente por HTTP, cada um podendo ser desenvolvido de uma maneira e com suas propiás tecnologias. No artigo é dito que essa arquitetura vem se tornando popular devido a os códigos monolíticos (somente um serviço grande) se tornarem um problema dependendo do uso, precisando de "deploys"do código inteiro a cada mudança e de uma grande atenção para manter a modularidade e organização.

O artigo ressalta que não existe uma arquitetura fixa para microsserviços, mas que existem características em comum presentes na maioria deles, sendo elas:

Componentização via Serviços: Os microserviços usam "serviços como componentes"em vez de bibliotecas. Serviços são "componentes fora do processo que se comunicam através de mecanismos como requisições de web service ou chamadas de procedimento remoto", permitindo deployment independente.

Organizados em Torno de Capacidades de Negócio: Diferentemente da divisão por camadas tecnológicas, os microserviços são "organizados em torno de capacidade de negócio". Cada serviço "implementa software para essa área de negócio, incluindo interface de usuário, armazenamento persistente e colaborações externas".

Produtos, não projetos: Ao contrario de uma abordagem por projetos, que depois de acabados o time é dissolvido, a arquitetura de microserviços usa uma abordagem de produto, onde o time deve ser responsável por aquele produto durante todo o ciclo de vida dele.

Endpoints inteligentes e Pipes simples: As aplicações devem ser coesas e desacopladas, fazendo uso de protocolos simples e se mantendo dentro de sua logica interna, não fazendo uso de trocas de mensagens complexas.

Governança Descentralizada: Preferem "usar a ferramenta certa para o trabalho"em vez de padronizar em uma única plataforma tecnológica. As equipes "preferem produzir ferramentas úteis que outros desenvolvedores possam usar para resolver problemas similares".

Gerenciamento de Dados Descentralizado: Os microserviços "descentralizam as decisões de armazenamento de dados"e "preferem deixar cada serviço gerenciar seu próprio banco de dados", seja através de diferentes instâncias da mesma tecnologia ou sistemas completamente diferentes.

Automação de Infraestrutura: Fazem "uso extensivo de técnicas de automação de infraestrutura"e geralmente são construídos por equipes com experiência em "Continuous Delivery"e "Continuous Integration".

Design para Falha: Cada serviço deve ser desenvolvido para suportar falhas, sendo importante agilidade na correção delas.

Design Evolutivo: Veem "a decomposição de serviços como uma ferramenta adicional para permitir que desenvolvedores controlem mudanças em sua aplicação sem desacelerar a mudança". Muitos grupos "esperam explicitamente que muitos serviços sejam descartados em vez de evoluídos a longo prazo".

Por fim, o artigo discute sobre o futuro, questionando se os microserviços são ou não o futuro do desenvolvimento de software. É destacado que, por mais que seja uma arquitetura promissora que vem demonstrando bons resultados, ela apresenta um nível de complexidade superior quando comparada a sistemas monolíticos e pode apresentar outros problemas com o passar do tempo que ainda não puderam ser observados. Portanto, a consideração final é de que um sistema não deve ser inicialmente desenvolvido em microserviços, mas sim evoluir de um sistema monolítico para microserviços.

O artigo é extremamente relevante e foi pioneiro em abordar a arquitetura de microserviços de forma sistemática e equilibrada. Após 10 anos de sua publicação, podemos observar como os microserviços continuam em crescimento, com mais empresas e desenvolvedores optando por essa arquitetura. A abordagem cautelosa dos autores se mostrou relevante, pois a indústria de fato vivenciou tanto sucessos quanto fracassos significativos na implementação de microserviços. O artigo permanece como uma referência fundamental não apenas por ter definido as características centrais dessa arquitetura, mas também por ter antecipado seus principais desafios, como a complexidade de gerenciar limites entre serviços e a necessidade de equipes mais experientes. Sua recomendação de "monolith first"tornou-se um consenso na comunidade, validando a sabedoria de uma evolução gradual ao inves uma adoção precipitada dessa arquitetura.