

Universidade Presbiteriana Mackenzie
Faculdade de Computação e Informática
Ciência da Computação – 6º Semestre
Computação Distribuída
Profº Ismar Frango
Nome: Pedro Henrique Araujo Farias
RA: 10265432

Sobre Microsserviços e Computação Distribuída Tradicional

Não é de hoje que os microsserviços vêm tomando conta do mercado. A solução, normalmente baseada em REST ou gRPC, utiliza dos benefícios do baixo acoplamento entre diferentes serviços para oferecer não apenas uma melhor experiência de produto, mas também de desenvolvimento aos programadores.

Essas características, além da visível escalabilidade promovida por elas, fazem com que os microsserviços ganhem espaço e se estabeleçam como dominantes nas preferências de implementação. Entretanto, vale lembrar das qualidades dos outros métodos de distribuição.

Os SOAPs, embora condenados pelo *overhead* provocado pelos arquivos XML, são considerados muito seguros e ainda muito utilizados em empresas que já usam a alternativa há algum tempo. Ainda que pouco flexíveis e até mesmo limitados em performance, estes têm o seu valor. A utilização de XML no lugar de JSON é um fator que, sem dúvidas, faz com que microsserviços sejam preferíveis.

Normalmente, microsserviços são associados aos REST Web Services, mas estes não estão necessariamente atrelados. O padrão de arquitetura REST não é implementado, objetivamente, em microsserviços. Todavia, o *Representational State Transfer* é comumente utilizado pelos *microservices*, uma vez que a simplicidade e eficiência deste, somado à utilização do HTTP, oferece uma base sólida a implementação dos microsserviços.

Quanto a *Sockets*, é importante ter em mente que esta se trata de uma tecnologia mais rústica, de baixo nível, com transferência de *raw bytes*. Comparada a microsserviços, uma solução bem mais refinada e de alto nível, os *sockets* se destacam pela performance, apenas. Não é impossível implementar microsserviços a partir de *sockets*, mas são poucas as situações em que outras tecnologias não se adequam melhor.

Uma alternativa à implementação de microsserviços a partir de REST são as implementações que utilizam gRPC. Essa tecnologia, além de performar muito bem por se valer do HTTP/2, também apresenta a assincronicidade como ponto forte. Essa é, sem dúvidas, uma das melhores opções para se trabalhar com microsserviços.

A não ser que você esteja desenvolvendo um sistema totalmente em Java, não há motivo para utilizar RMI. Essa abordagem é limitada, obviamente, pela linguagem de programação, o que não permite heterogeneidade entre clientes e servidores de um microsserviço, por exemplo. O ideal é que microsserviços sejam independentes de linguagem e plataforma. Todavia, se ocorre homogeneidade na operação como um todo, RMI é um forte candidato para a implementação de microsserviços em razão da sua simplicidade e extensa documentação.

CORBA é horrível. Não se submeta a essa dor de cabeça. As alternativas supracitadas são muito mais simples, performáticas e escaláveis. Poupe-se.