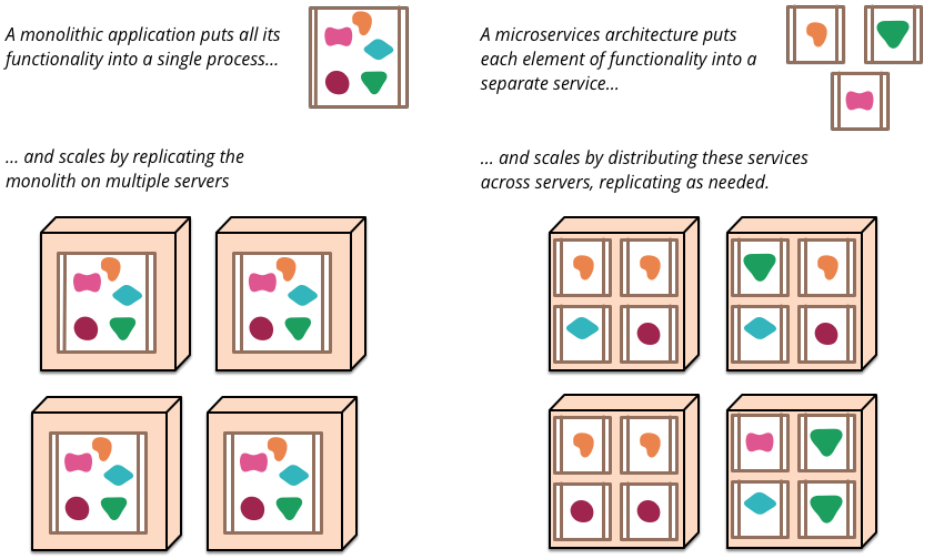
Microservices

Uma “especialização” de SOA e RESTFul, hoje mais comumente implementados usando RESTFul Api’s. Um dos criadores do termo: *Martin Fowler*.

Desenvolver uma aplicação como sendo um conjunto de vários serviços menores, cada um executando em seu próprio processo e recurso.

Podemos ter vários microserviços locais, ou distribuídos em outros servidores.



Há uma ***componentização*** dos serviços. O serviço terá sua especificação detalhada em uma API (exemplo: OpenAPI) ou algum outro formato, para poder ser chamado corretamente. DOCUMENTAÇÃO é importante!

A equipe pode ser dividida em especialidades relativas ao negócio, não necessariamente em especialistas em interface, middleware, DBAs, etc.

BAIXO ACOPLAMENTO, ALTA COESÃO

São pequenos produtos, não projetos complexos. Exemplo: criar um microserviço de localização que poderá ser usado em diversas partes. Ciclo de desenvolvimento rápido.

Cada um é responsável pelo seu próprio serviço. Permite escalabilidade, pois podem estar distribuídos.

Projeto “para falha” e evolução
Se 1 serviço falhar, OK, outro serviço pode ser chamado, ou a aplicação tem funcionalidades a menos, mas não para de executar.

Se eu incluir um novo serviço, a aplicação se adapta e fica mais rica.

Linguagem específica
É possível criar uma linguagem específica para chamar os serviços, não apenas REST. Ex.: GraphQL.

ENTRETANTO:
Dificulta a gestão. Como garantir que um serviço S manterá sempre a mesma interface? E se for modificado, como os usuários deste serviço precisarão adaptar?

Difícil de entender tudo o que uma aplicação que chama vários serviços fornece. E como ela está estruturada.

Com muitos serviços existentes, dificulta o teste, pois são “caixas pretas”. Testes do serviço único são simples, mas e de um sistema composto ?

Dificulta a parte operacional. Se tenho vários serviços, como sei qual está ou não sendo executado? Estão todos no ar ? Algum parou ?
Por isso, deve haver uma afinidade com a equipe operacional, sendo hoje comumente chamado de cadeia **DevOps.** Monitoramento é essencial.