

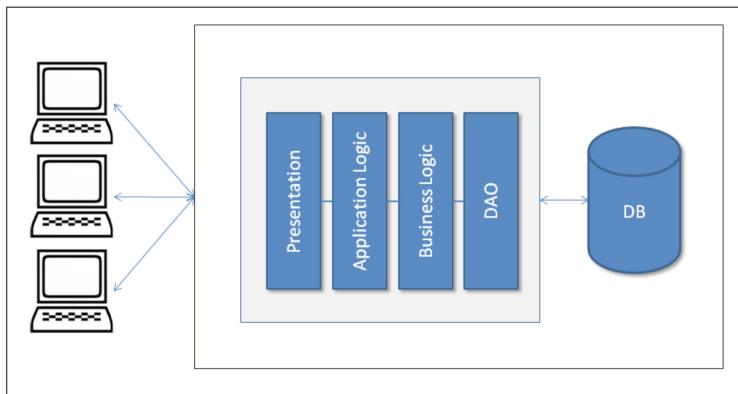
# Microserviços

Prof. Dr. Alexandre L'Erario

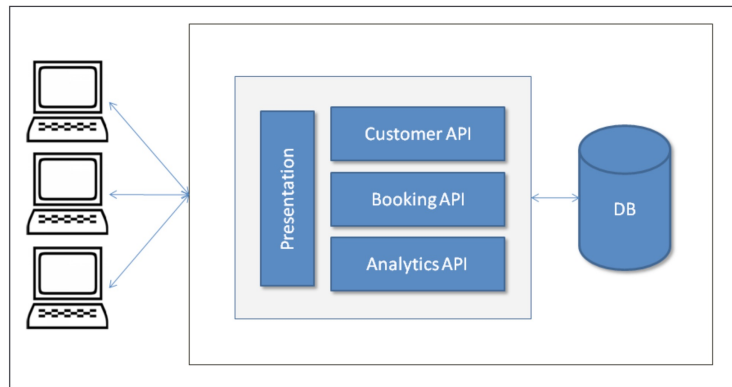


# μServices

- Estilo de arquitetura inspirado na computação orientada a serviços
- Se opõe principalmente aos modelos monolíticos



Traditional monolithic design



Monolithic design with services

# Definições

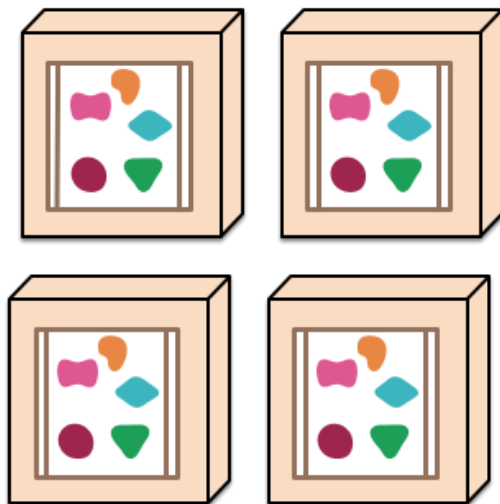
- **Um microsserviço** é um processo coeso e independente que interage por meio de mensagens.
  - Devem ser componentes independentes implantados conceitualmente de forma isolada e equipados com ferramentas de persistência de memória dedicada (por exemplo, bancos de dados).
  - Seu comportamento deriva da composição e coordenação de seus componentes por meio de mensagens.
- **Uma arquitetura de microsserviços** é uma aplicação distribuída onde todos os seus módulos são microsserviços.

# Comparação

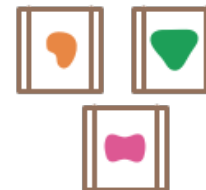
*A monolithic application puts all its functionality into a single process...*



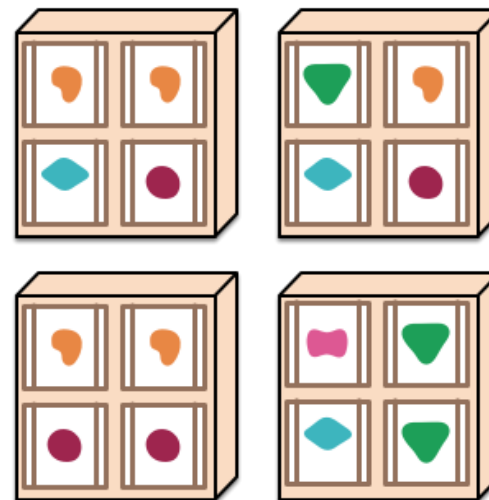
*... and scales by replicating the monolith on multiple servers*



*A microservices architecture puts each element of functionality into a separate service...*



*... and scales by distributing these services across servers, replicating as needed.*



# Microserviços vs Monolítico

- Os microserviços implementam uma quantidade limitada de funcionalidades, o que torna sua base de código pequena e limita o escopo de um bug.
- É possível planejar a transição gradual de uma versões de um microserviço.
- Alterar um módulo de uma arquitetura de microserviços não requer uma reinicialização completa de todo o sistema.
- Os microserviços naturalmente se prestam à containerização.
- Escalar uma arquitetura de microserviços não implica na duplicação de todos os seus componentes.
- A única restrição imposta a uma rede de microserviços interoperantes é a tecnologia usada para fazê-los se comunicar.

[https://doi.org/10.1007/978-3-319-67425-4\\_12](https://doi.org/10.1007/978-3-319-67425-4_12)

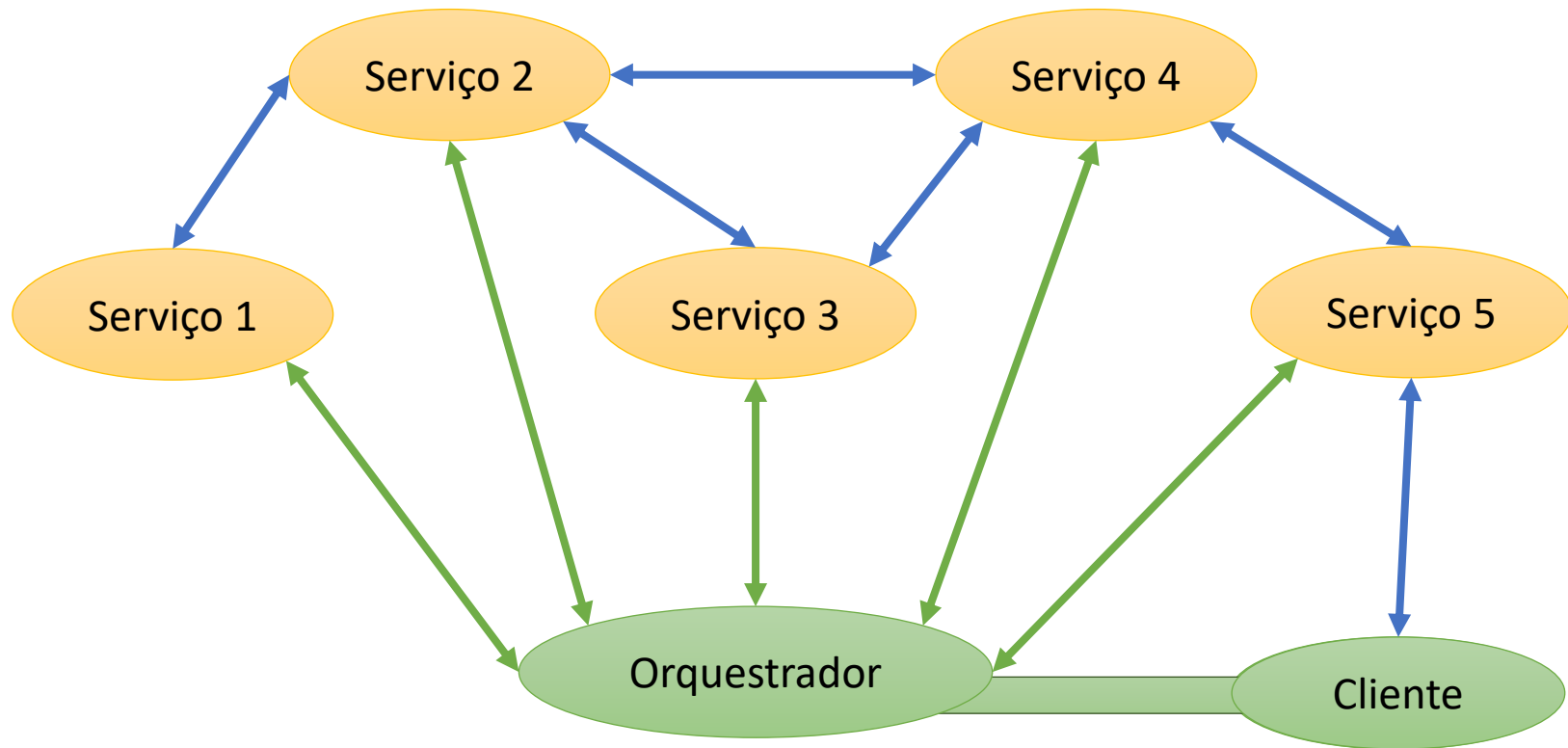
# Características – princípios básicos

- Baixo acoplamento
- Abstração
- Autonomia
- Reusabilidade
- Sem estado
- Descoberta
- Padronização de contrato
- Governança distribuída

# Orquestração e Coreografia

- A orquestração requer um condutor — um serviço central que enviará solicitações a outros serviços e supervisionará o processo recebendo respostas.
- A coreografia, por outro lado, não assume nenhuma centralização e usa eventos e mecanismos de publicação/assinatura para estabelecer a colaboração.

# Orquestração e Coreografia





# Demais atividades

- Autenticação / autorização
  - Cada microsserviço pode exigir um mecanismo diferente
- Monitoramento
  - O estado de um microsserviço pode influenciar na performance do sistema

# Desafios....

- Preparar sua equipe para trabalhar com  $\mu$ Services
  - Condições de negócio
  - Condições de engenharia: métricas, atividades/processos
  - Condições técnicas: implementação e consumo
- Leitura recomendada: [10.1109/MS.2021.3080335](https://labinov.org.br/10.1109/MS.2021.3080335)

# Considerações finais

- $\mu$ Services  $\rightarrow$  caminho sem volta!
- $\mu$ Services + estilos tradicionais