Aula 14 - Arquitetura de Microsserviços

Disciplina: Arquitetura de Software

Prof. Me. João Paulo Biazotto

UniCesumar

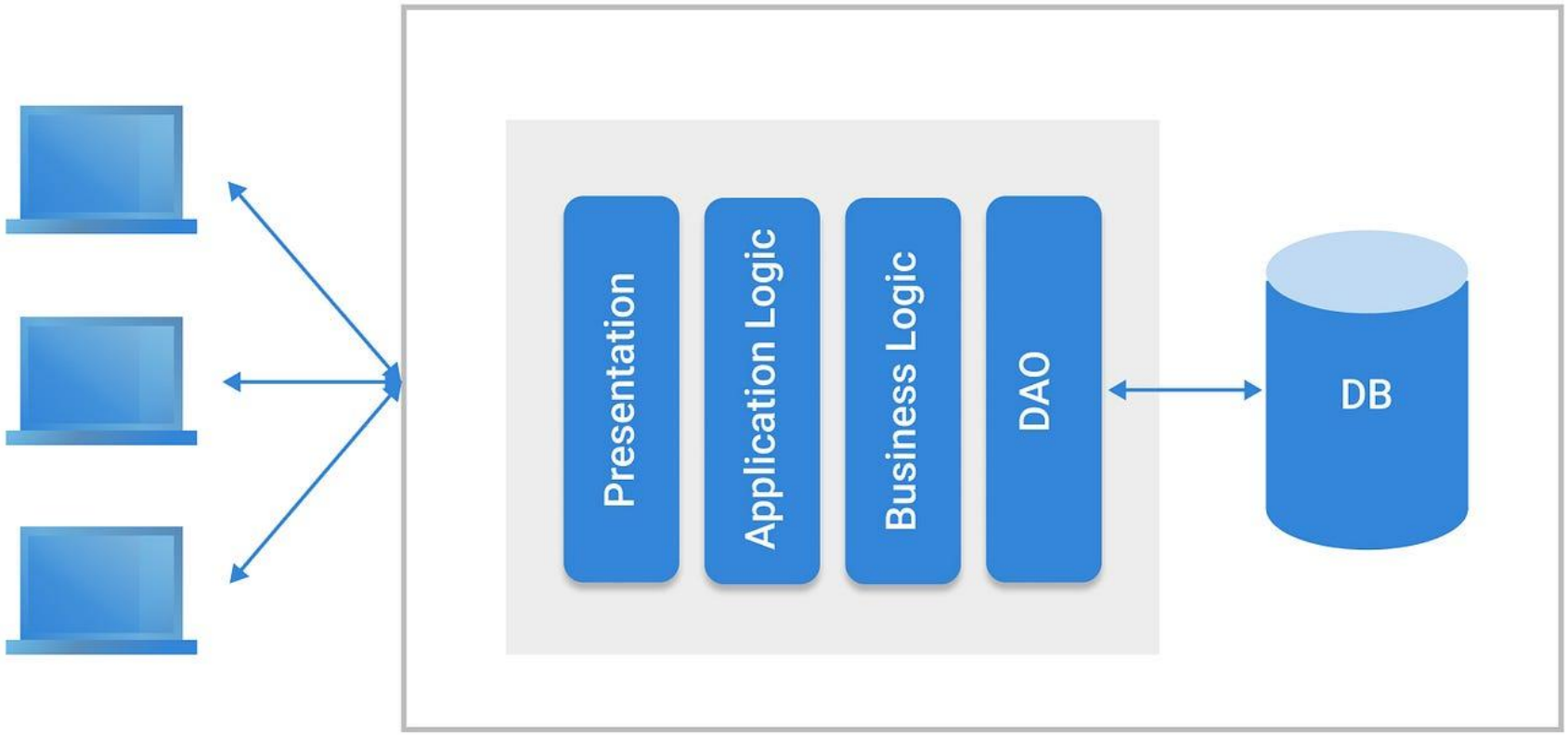
O que vimos até agora?

- Conceito de Arquitetura de Software
- Estilos Arquiteturais
  - Monolítica
  - Distribuída

Arquitetura Monolítica

- Sistema construído como uma única unidade coesa
- Componentes fortemente acoplados e executados no mesmo processo
- Exemplo: Aplicação web com frontend, backend e banco de dados em um único servidor.

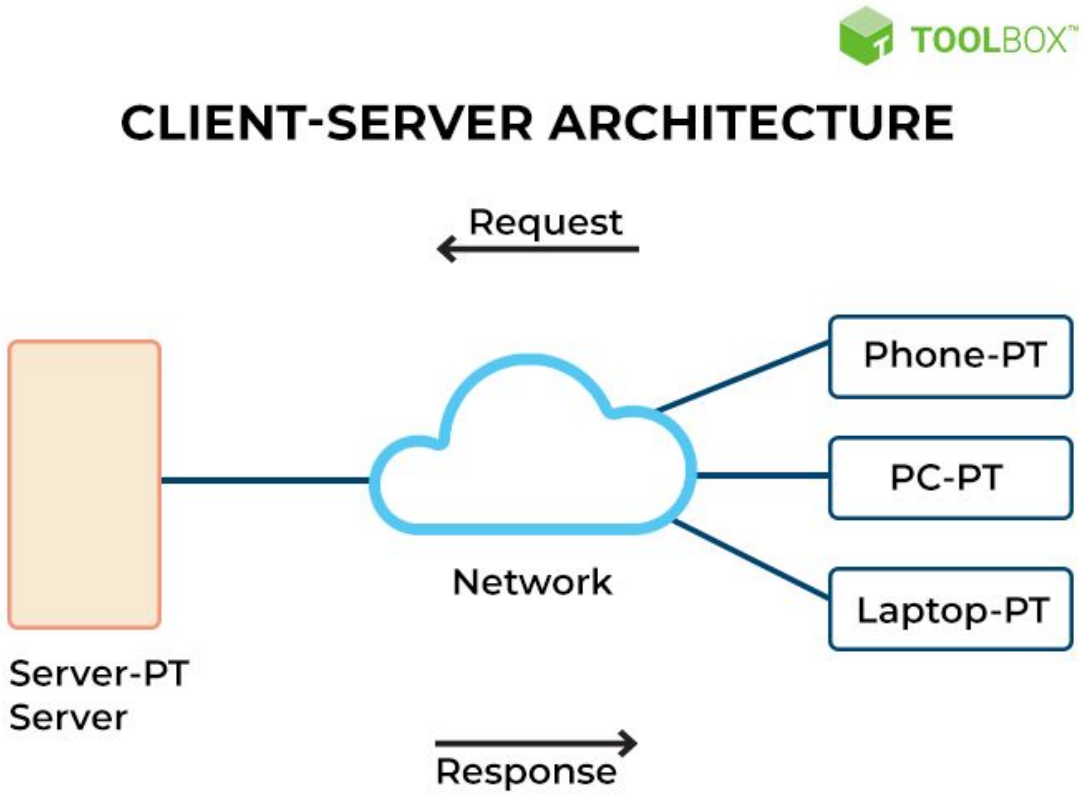
Arquitetura Monolítica



Arquitetura Distribuída

- Sistema dividido em múltiplos componentes independentes, comunicando-se por rede
- Maior escalabilidade, mas exige mecanismos de comunicação e coordenação;
- Exemplo: Aplicação web onde o frontend roda no navegador, backend em múltiplos serviços e dados em servidores distintos

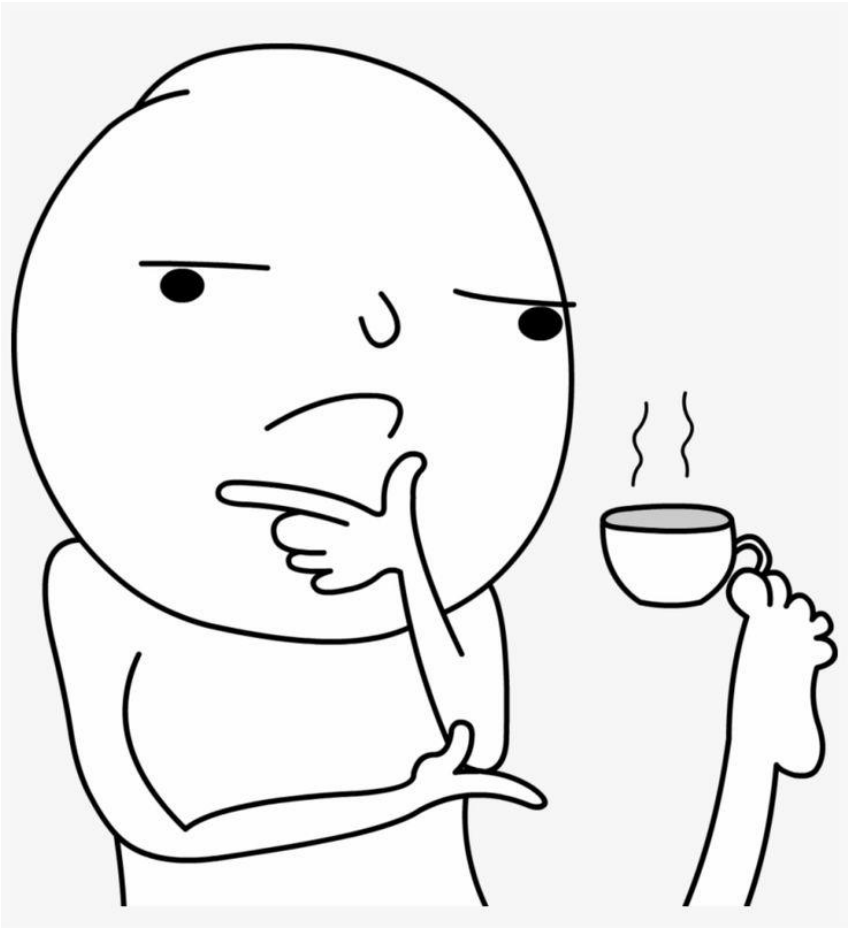
Arquitetura Distribuída



E se...

- A se SOA fosse melhorado, de forma a melhorar a **modularidade**?

- A se SOA fosse melhorado, de forma a melhorar a **baixo acoplamento**?



O que são Microsserviços?

Microsserviços são uma abordagem de arquitetura de software onde um sistema é dividido em pequenos serviços independentes, que se comunicam entre si e executam funções específicas.

Exemplo de Microsserviços

A **Netflix** usa microsserviços para dividir suas funcionalidades em serviços como "Catálogo de Vídeos", "Perfil do Usuário" e "Recomendações". Cada serviço é **implantado** e **escalado independentemente**, permitindo maior resiliência e velocidade de desenvolvimento.

Por que surgiram os Microsserviços?

- Complexidade de sistemas monolíticos
- Necessidade de escalabilidade
- Aumento da agilidade no desenvolvimento
- Independência entre equipes

Princípios de Design - *Independência de Serviços*

Serviços devem ser autônomos e isolados.

Exemplo Netflix: O serviço de *"Legendas"* pode ser atualizado e implantado separadamente do serviço de "Player de Vídeo".

Princípios de Design - *Modelagem orientada a Domínio*

Basear os serviços em contextos delimitados do domínio de negócio.

Exemplo Netflix: Serviços como *"Faturamento"*, *"Recomendações"* e *"Gerenciamento de Conta"* são baseados em contextos claros de domínio.

Princípios de Design - *Tolerância a falhas*

Serviços devem ser resilientes e degradar de forma controlada.

Exemplo Netflix: A Netflix usa o *Hystrix* para manter o sistema funcionando mesmo se o serviço de *recomendações* falhar.

Princípios de Design - *Escalabilidade horizontal*

Capacidade de escalar instâncias de serviços conforme demanda.

Exemplo Netflix: Durante picos de acesso, a Netflix escala automaticamente o serviço de *"Streaming de Vídeo"* sem afetar os demais.

Princípios de Design - *Automação de Deploy*

Serviços devem ser implantados automaticamente com CI/CD.

Exemplo Netflix: A Netflix utiliza a ferramenta *Spinnaker* para automação de pipelines de deploy.

Princípios de Design - *Observabilidade*

Logs, métricas e rastreamento distribuído são essenciais.

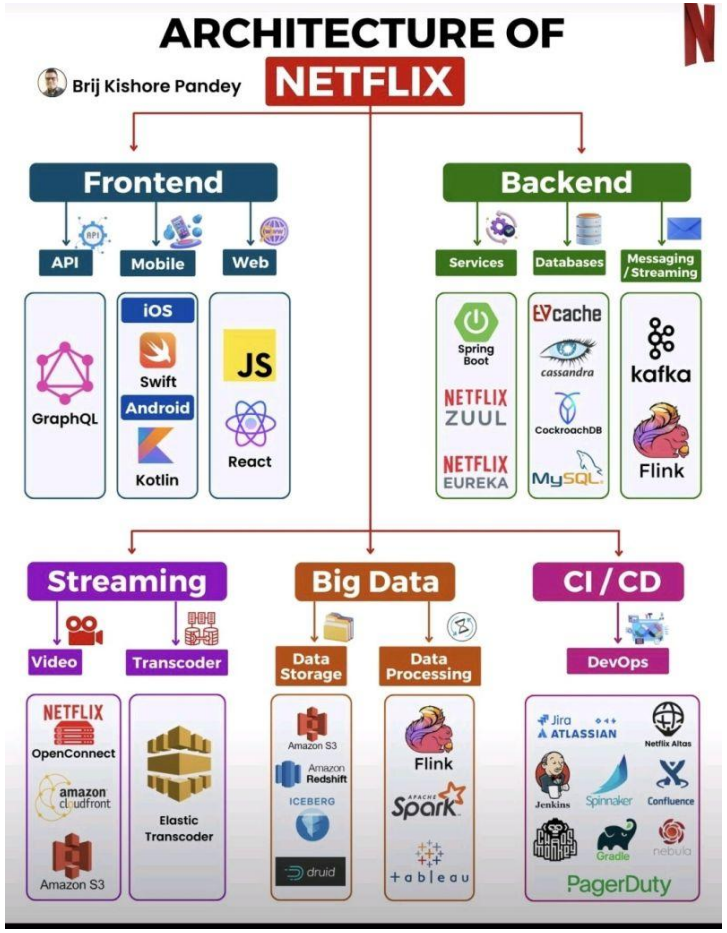
Exemplo Netflix: A Netflix usa o sistema *Atlas* para monitoramento e alertas de desempenho.

Princípios de Design - *Desacoplamento tecnológico*

Cada serviço pode ser desenvolvido com diferentes linguagens e frameworks.

Exemplo Netflix: A Netflix possui serviços escritos em Java, Python, Node.js, de acordo com o que for mais adequado para cada contexto.

Princípios de Design - *Desacoplamento tecnológico*



Princípios de Design - *Comunicação entre Serviços*

REST, gRPC, filas de mensagens (e.g., *RabbitMQ*).

Exemplo Netflix: A comunicação entre serviços da Netflix é feita principalmente via REST, com suporte a mensageria via *Apache Kafka*.

Principais Características de Microsserviços

- Modularidade
- Independência de deploy
- Escalabilidade individual
- Baixo acoplamento
- Alta coesão

Como decompor microsserviços?

Estudo de Caso - Sistema de Streaming

Vamos aplicar os conceitos em um estudo de caso fictício baseado em um sistema semelhante à Netflix.

Requisitos do Sistema

- Cadastro de usuários
- Catálogo de vídeos
- Reproduzir vídeos
- Recomendações personalizadas
- Controle de assinaturas e pagamentos
- Suporte multilíngue (legendas e dublagem)

Requisitos do Sistema

- Usuários
- Catálogo
- Reprodutor
- Recomendações
- Pagamentos
- Internacionalização

Quebrando em Microsserviços

- Serviço de Usuários
- Serviço de Catálogo
- Serviço de Player
- Serviço de Recomendações
- Serviço de Faturamento
- Serviço de Legendas

Decompondo em Microsserviços

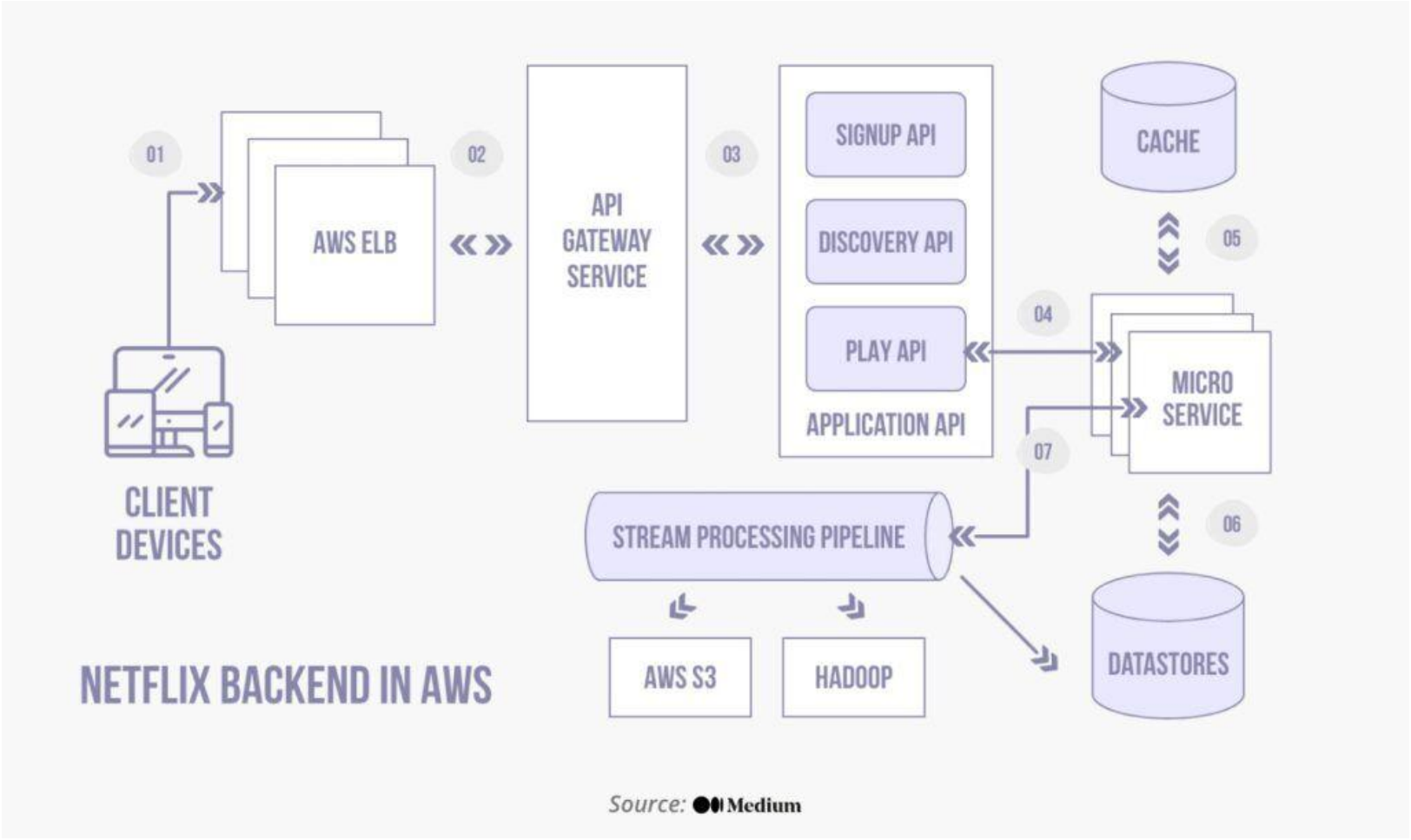
- Cada serviço em um container Docker.
- Orquestração com Kubernetes.
- Comunicação: REST + RabbitMQ
- Observabilidade: Prometheus + Grafana

Benefícios de Microsserviços

- Escalabilidade eficiente
- Manutenção facilitada
- Flexibilidade tecnológica
- Alta resiliência

A Netflix consegue escalar serviços como "Streaming de Vídeo" separadamente dos serviços de "Busca" ou "Sugestões", otimizando recursos e garantindo alta disponibilidade mesmo em horários de pico.

Benefícios de Microsserviços



Desvantagens de Microsserviços

- Complexidade de comunicação
- Gerenciamento de dados distribuídos
- Monitoramento mais complexo
- Necessidade de cultura DevOps

Exemplo real: A Netflix desenvolveu uma infraestrutura robusta para lidar com falhas entre serviços, como o *Hystrix*, que implementa circuit breakers para manter a resiliência diante de falhas em comunicações distribuídas.

Desvantagens de Microsserviços

Decompondo em Microsserviços

Top highlight

Rebuilding Netflix Video Processing Pipeline with Microservices

Netflix Technology Blog · Follow · 9 min read · Jan 10, 2024

1.6K · 22

Liwei Guo, Anush Moorthy, Li-Heng Chen, Vinicius Carvalho, Aditya Mavlankar, Agata Opalach, Adithya Prakash, Kyle Swanson, Jessica Tweneboah, Subbu Venkatrav, Lishan Zhu

Referência: https://netflixtechblog.com/rebuilding-netflix-video-processing-pipeline-with-microservices-4e5e6310e359

SOA vs Microsserviços

- SOA (Service-Oriented Architecture) define um conjunto de serviços que se comunicam por meio de um barramento central (ESB).
- Microsserviços distribuem funcionalidades em serviços independentes, com comunicação direta ou por mensagens.

SOA vs Microsserviços

Característica	SOA	Microsserviços
Comunicação	ESB	REST/gRPC/Kafka
Tamanho dos serviços	Grandes	Pequenos
Acoplamento	Médio	Baixo
Escalabilidade	Parcial	Alta
Independência	Limitada	Alta

Vantagens SOA

- Integração com sistemas legados
- Governança centralizada
- Reutilização de serviços

Vantagens Microsserviços

- Deploys independentes
- Escalabilidade granular
- Liberdade tecnológica

Quando usar cada uma?

- SOA: Integração de sistemas corporativos legados.

- Microsserviços: Sistemas escaláveis, ágeis, com múltiplas equipes.

Quando usar cada uma?

- Sistema de e-commerce:
  - SOA: Um serviço de pagamento centralizado para todo o sistema.
  - Microsserviços: Serviços separados para "Carrinho", "Pagamento", "Entrega" e "Recomendações".

DÚVIDAS?

UniCesumar