

ARQUITETURA DE SW

Técnicas de Implementação MS - EAD



Análise e Desenvolvimento de Sistemas

Prof. Ms. Renato Leite
renato@sectras.edu.br

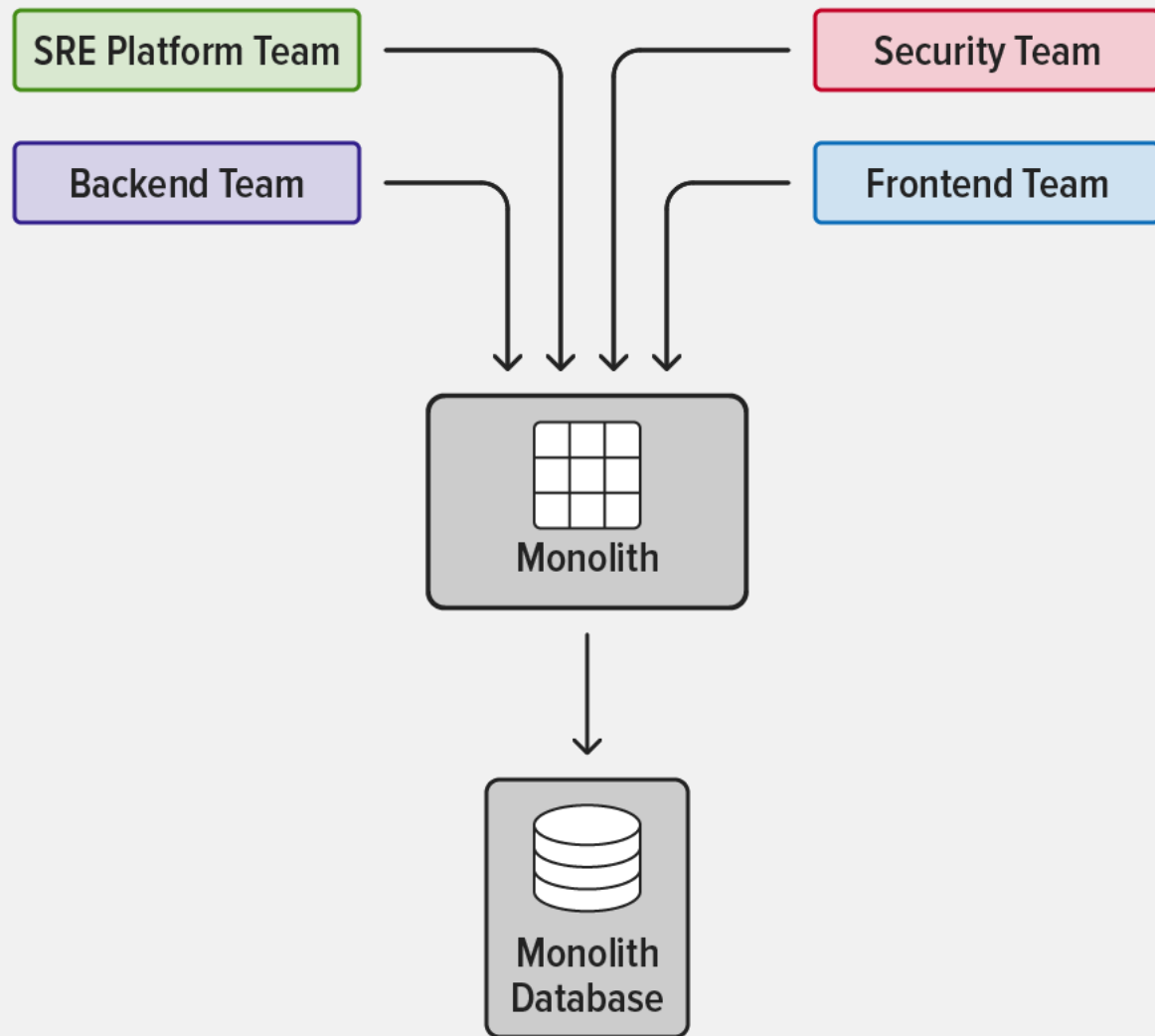
OBJETIVOS

- Definindo Limites de Serviços
- Estilos de Comunicação entre Microserviços (Orquestração vs. Coreografia)
- Armazenamento e Transações
- Infraestrutura e Deploy
- Monitoramento e Governança
- Integrações Legadas
- Desafios e Boas Práticas

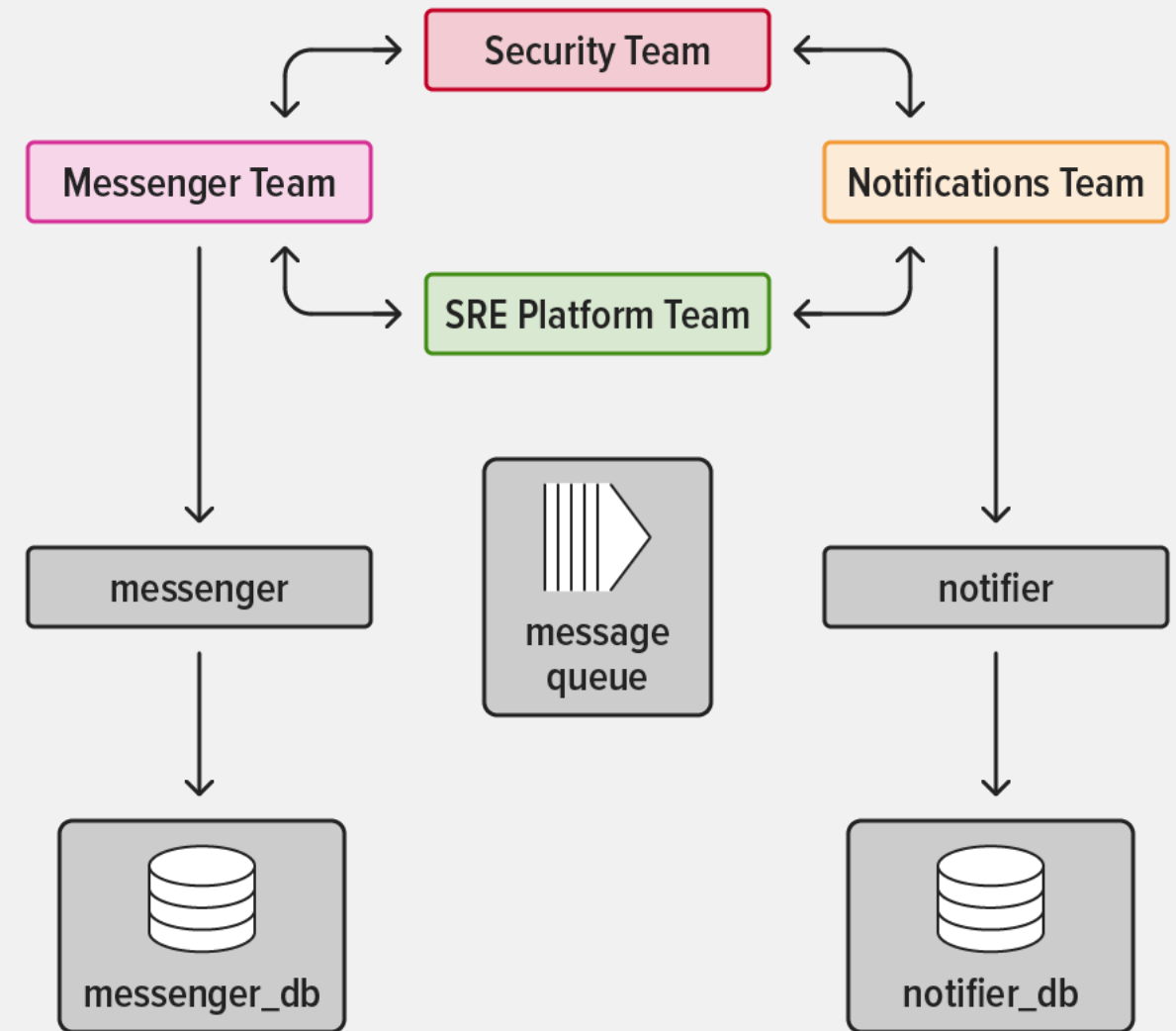
MICROSERVIÇOS

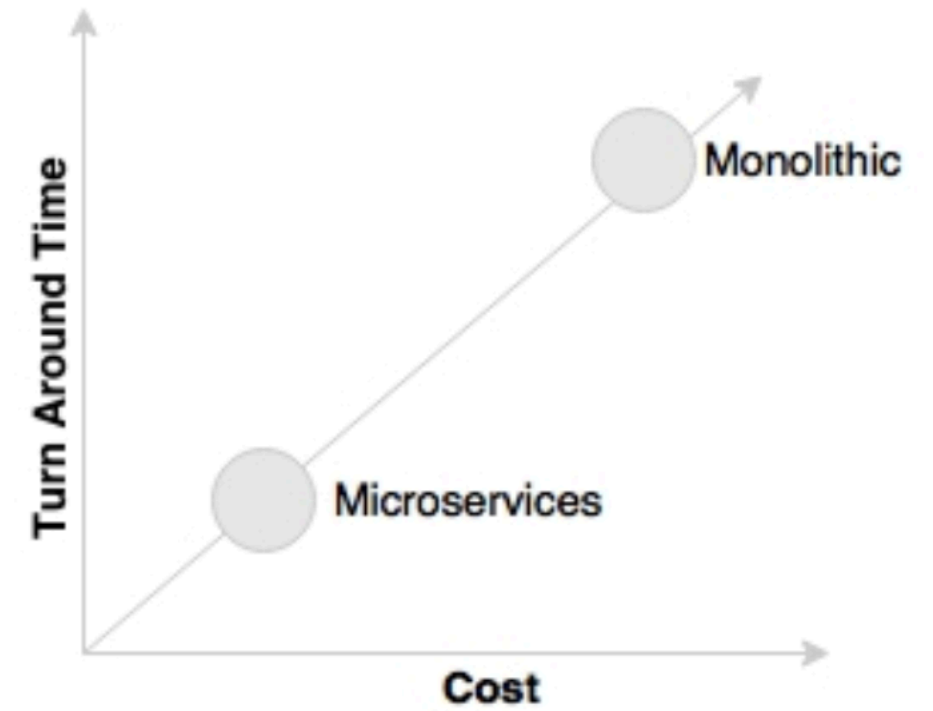
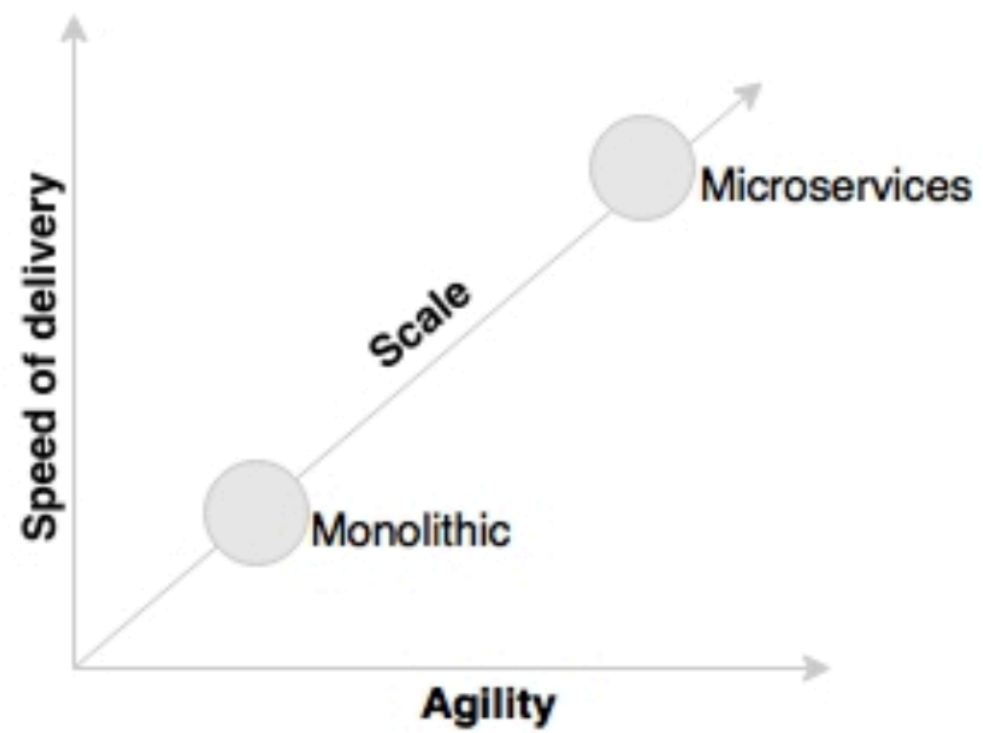
- Serviços pequenos, independentes e focados em um único propósito.
- Facilitam manutenção e evolução contínua.
- Adoção cuidadosa conforme o contexto.

Monolith Organization



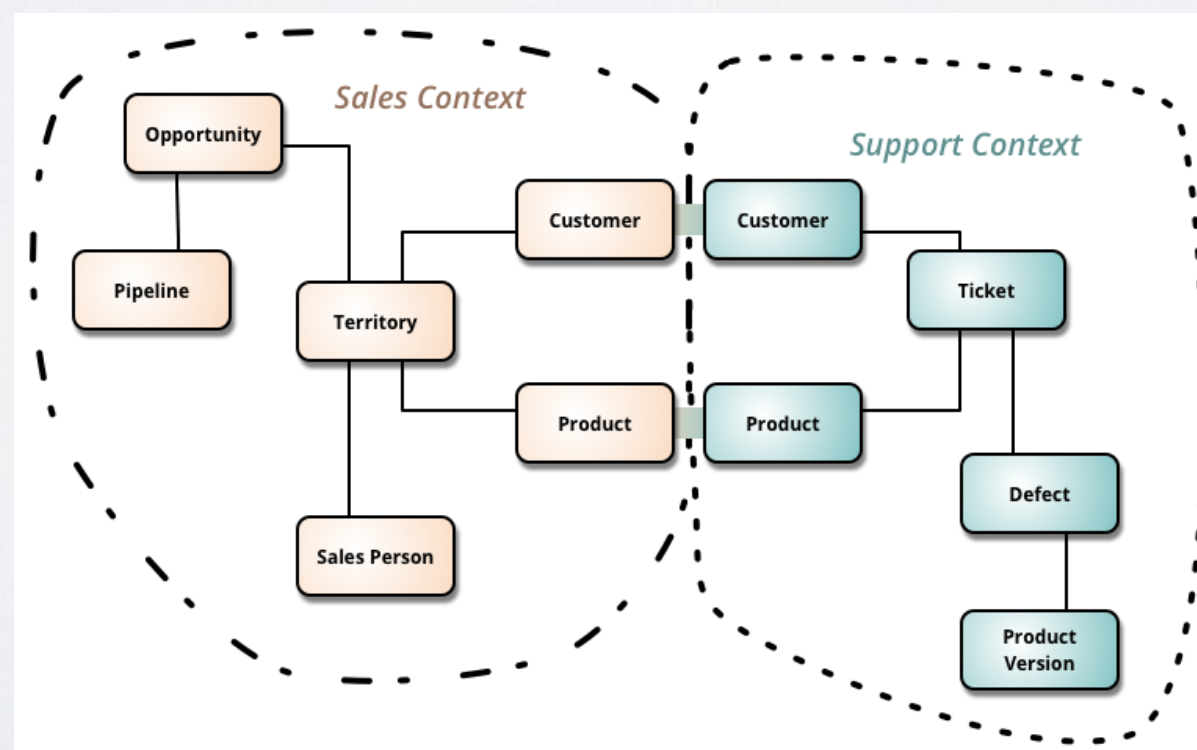
Microservices Organization





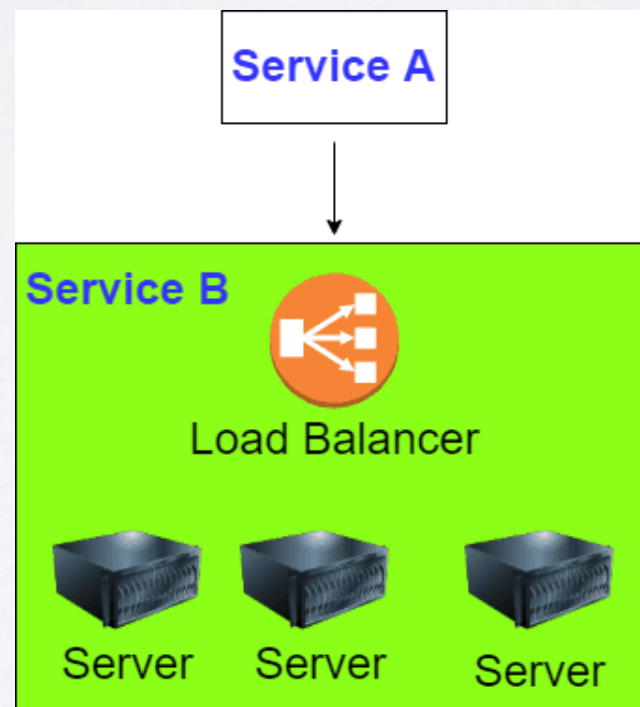
ATÉ ONDE VAI UM MS?

- Usar Bounded Contexts para separação lógica.
 - Agrupe: regras, terminologias e conceitos
- Evitar superdimensionamento ou excesso de fragmentação.
- Relacionar serviço a uma função de negócio clara.

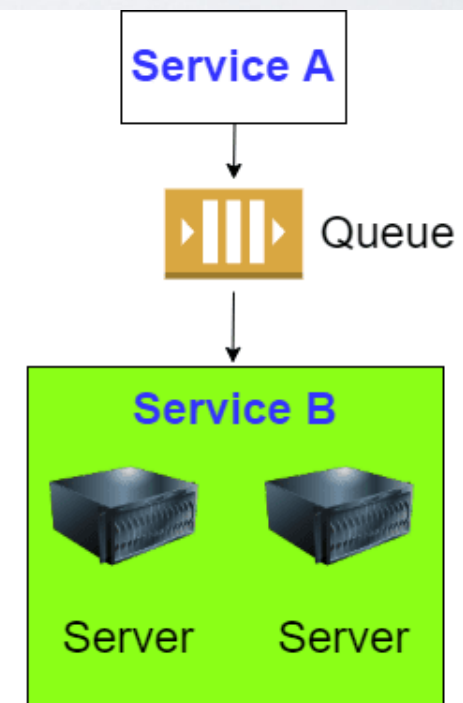


COMUNICAÇÃO

- Síncrona: chamada direta e resposta imediata.
- Assíncrona: comunicação baseada em eventos.
- Escolher conforme requisitos de acoplamento e desempenho.



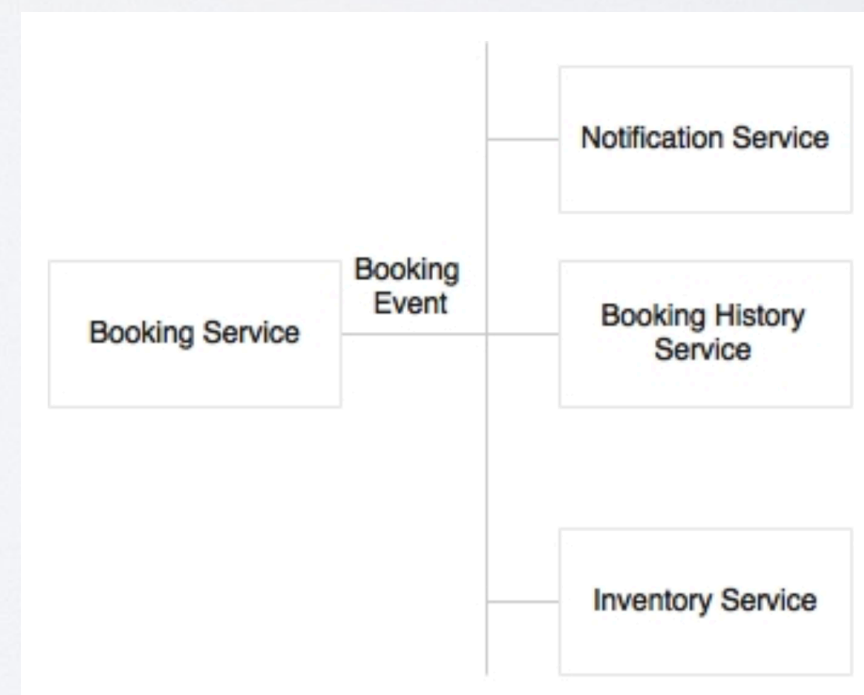
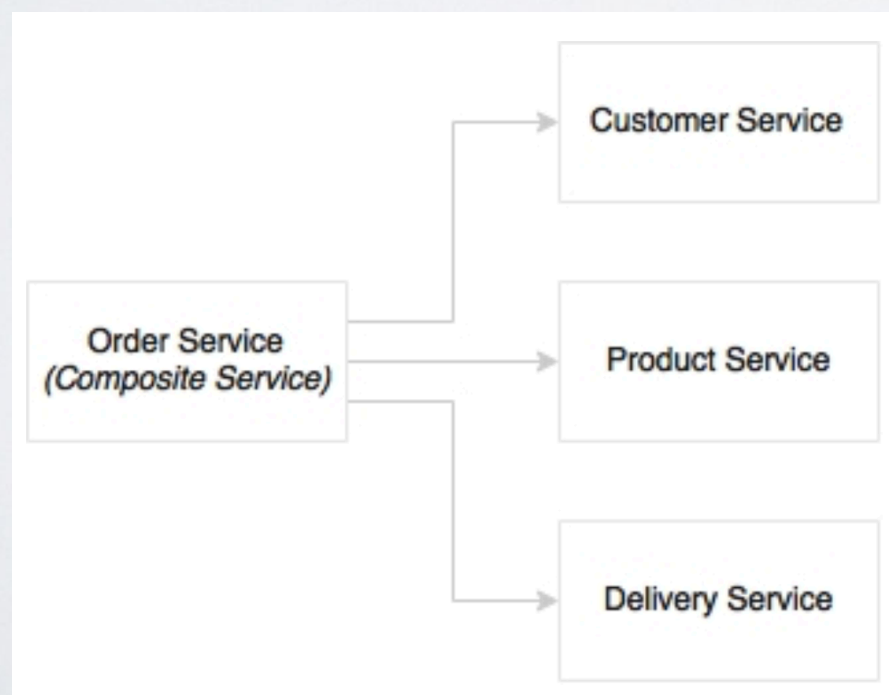
Synchronous Communication



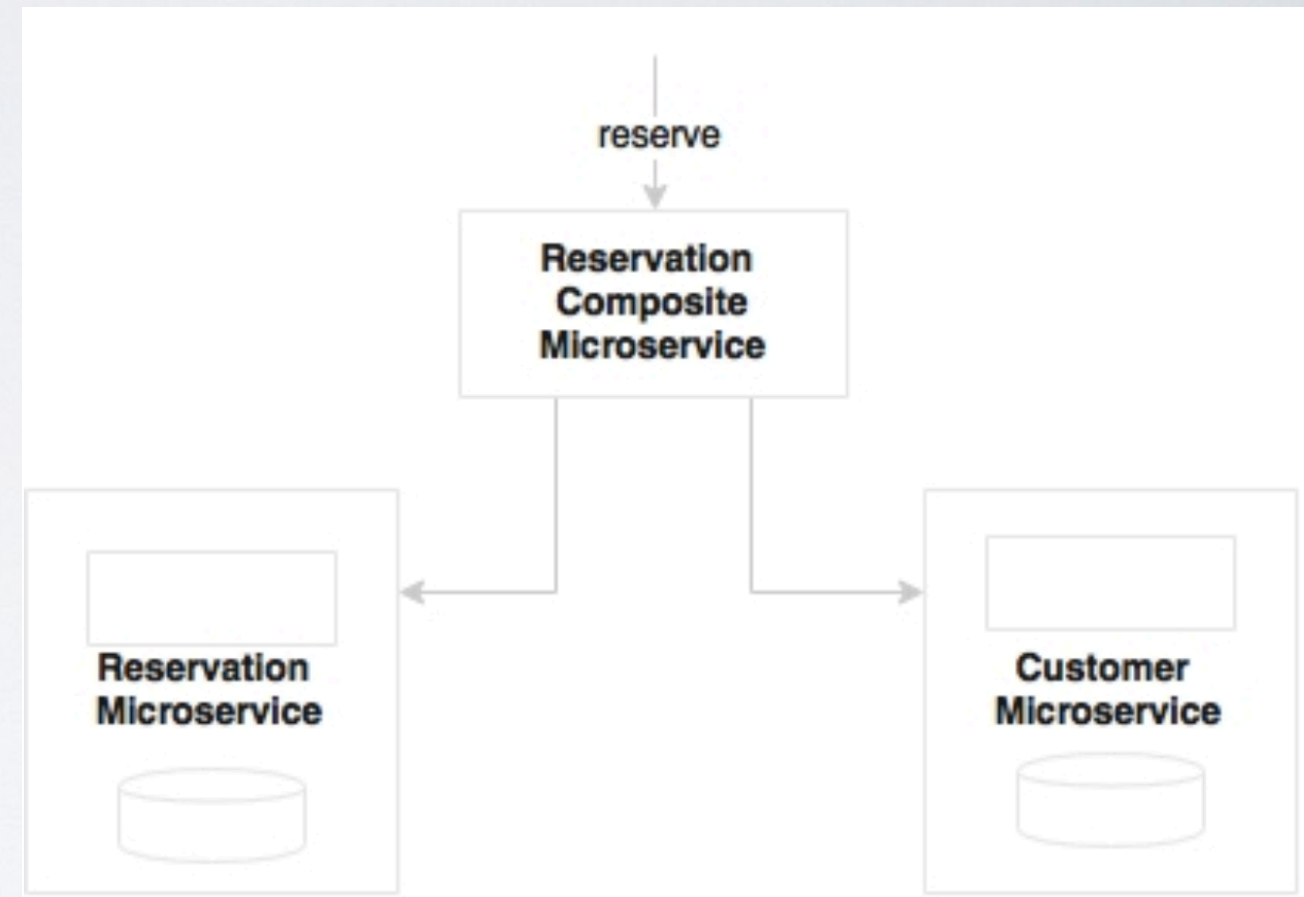
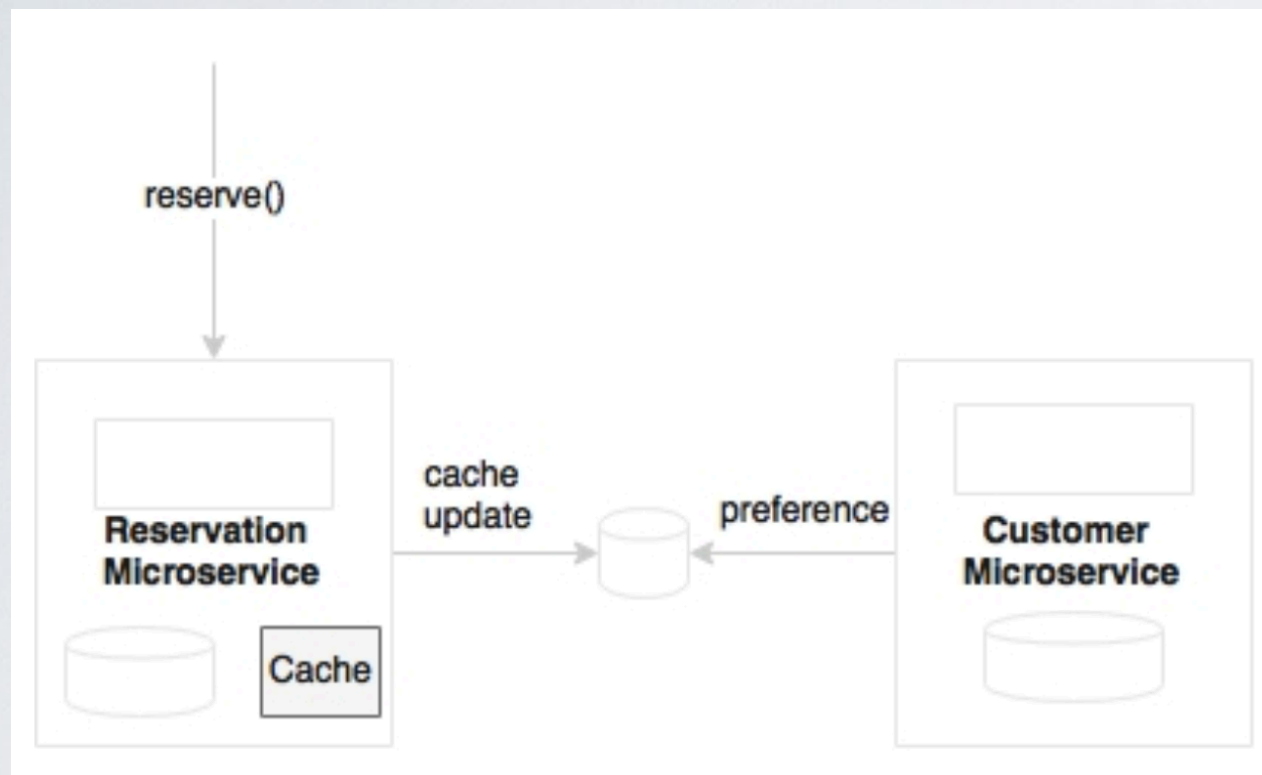
Asynchronous Communication

COORDENAÇÃO ENTRE MS

- **Orquestração:** controle central do fluxo.
- **Coreografia:** serviços interagem reagindo a eventos.
- Equilibrar controle e autonomia dos serviços.



QUAL MELHOR ESCOLHA?

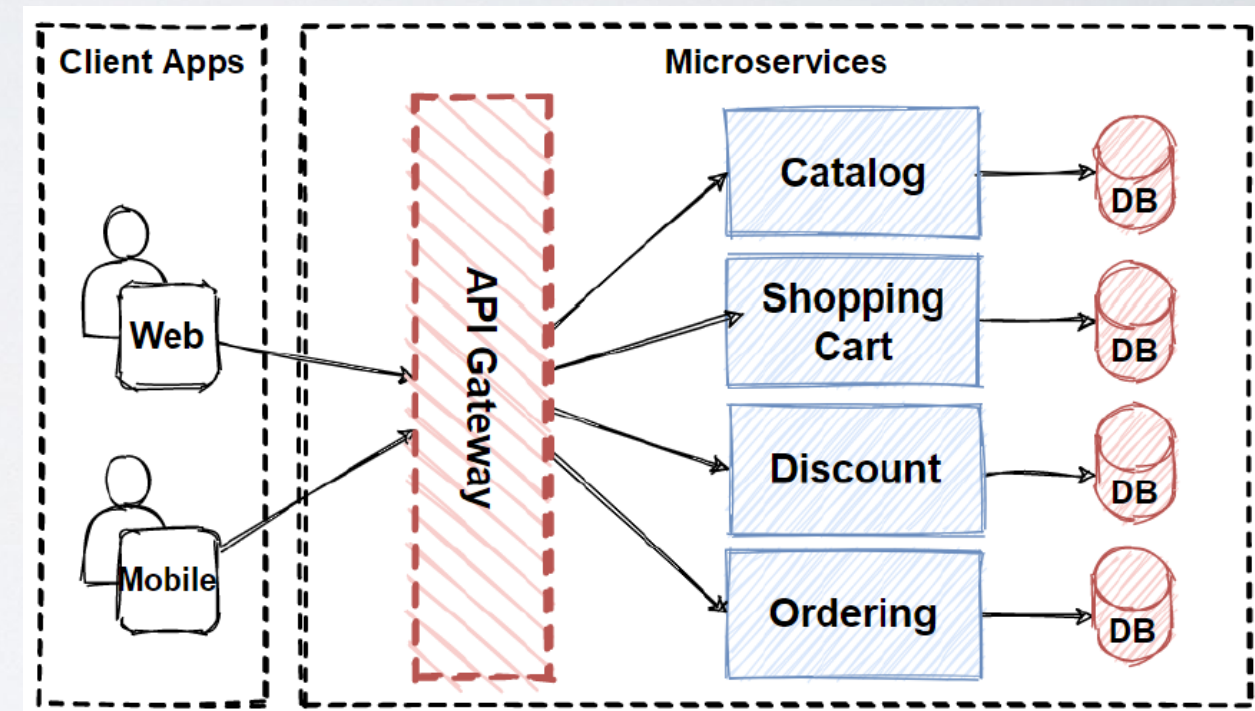


COORDENAÇÃO ENTRE MS

- Nem sempre orquestrar será bom, tampouco utilizar coreografia para coordenar, será ideal.
- Depende do negócio
- No exemplo anterior, não fazia sentido criar um orquestrador. Por quê?

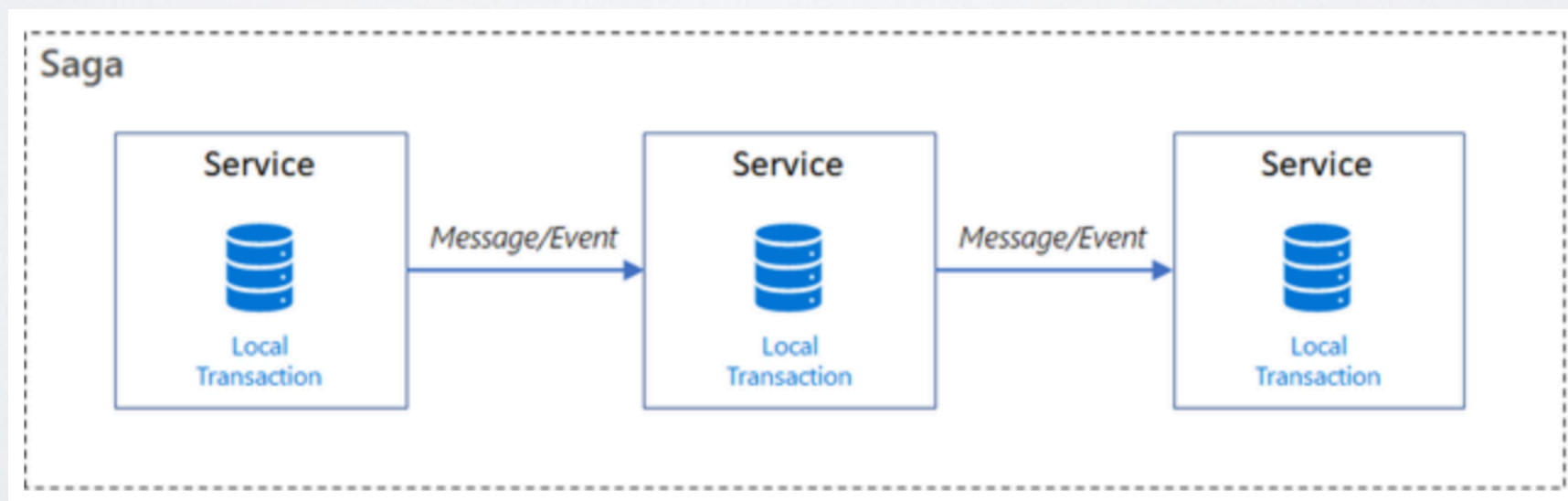
EXPOSIÇÃO DE SERVIÇOS

- Contratos bem definidos e versionados.
- Seguir princípios como KISS e YAGNI
 - Keep It Simple, Stupid
 - You Aren't Gonna Need It
- Facilitar entendimento e consumo de APIs.



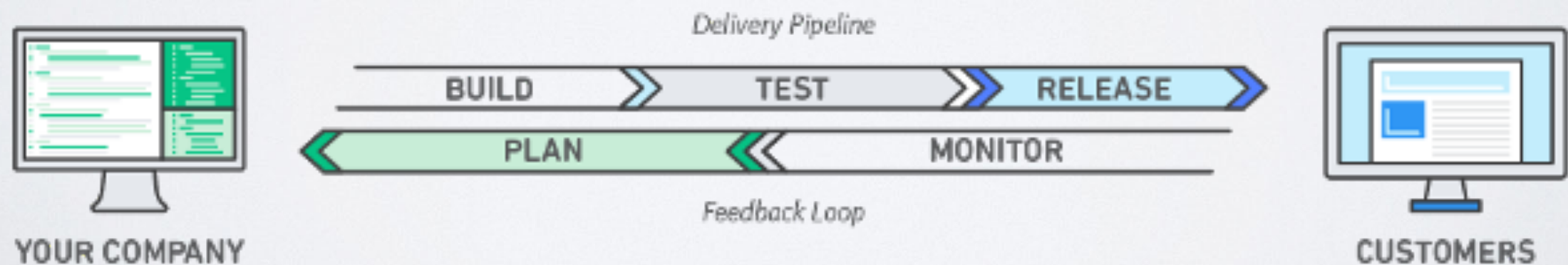
DADOS

- Cada serviço com seu banco de dados.
- Evitar transações distribuídas.
- Usar sagas para consistência eventual - mecanismo de compensação (sonda e cancelamentos)!



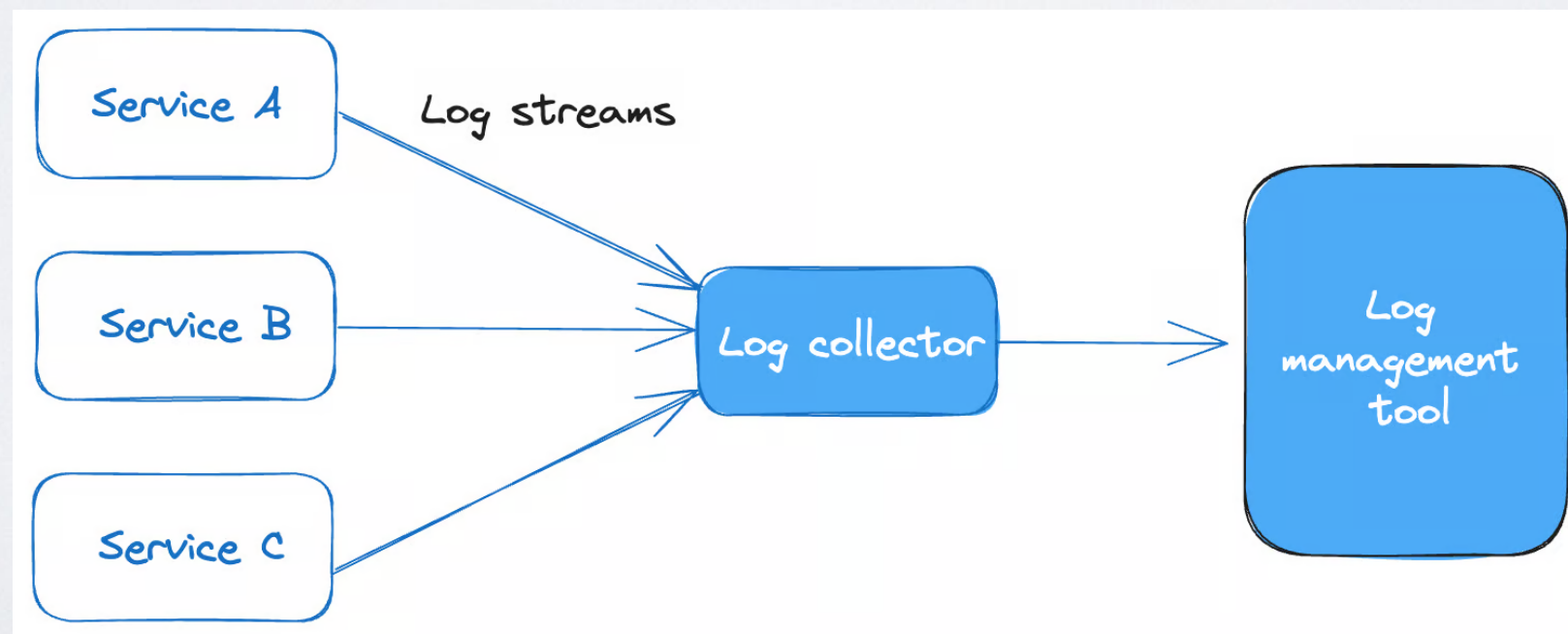
DEPLOY

- Automatizar entregas com CI/CD.
- Containers facilitam padronização e escala.
- Escalabilidade independente entre serviços.



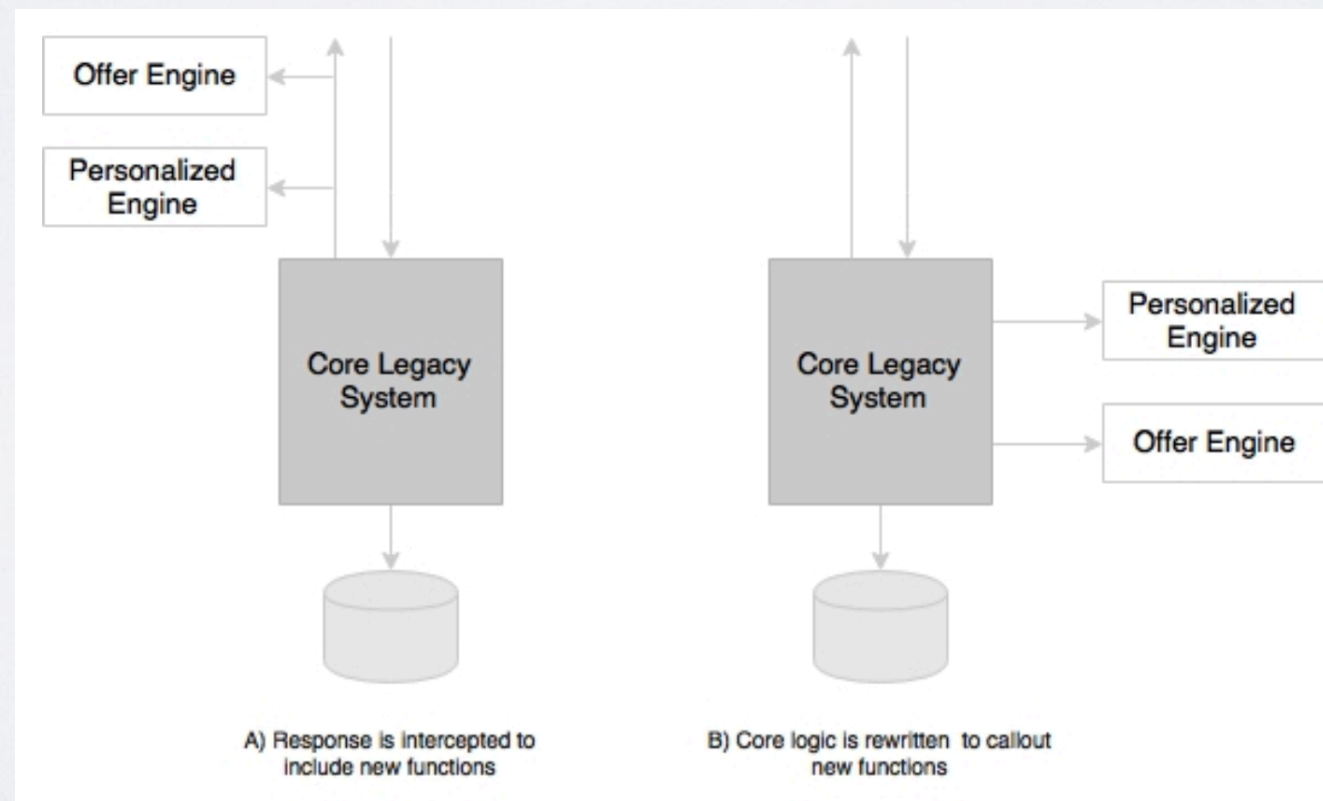
MONITORAMENTO E GOVERNANÇA

- Centralizar logs e métricas.
- Implementar rastreamento distribuído.
- Controlar versões e dependências entre serviços.



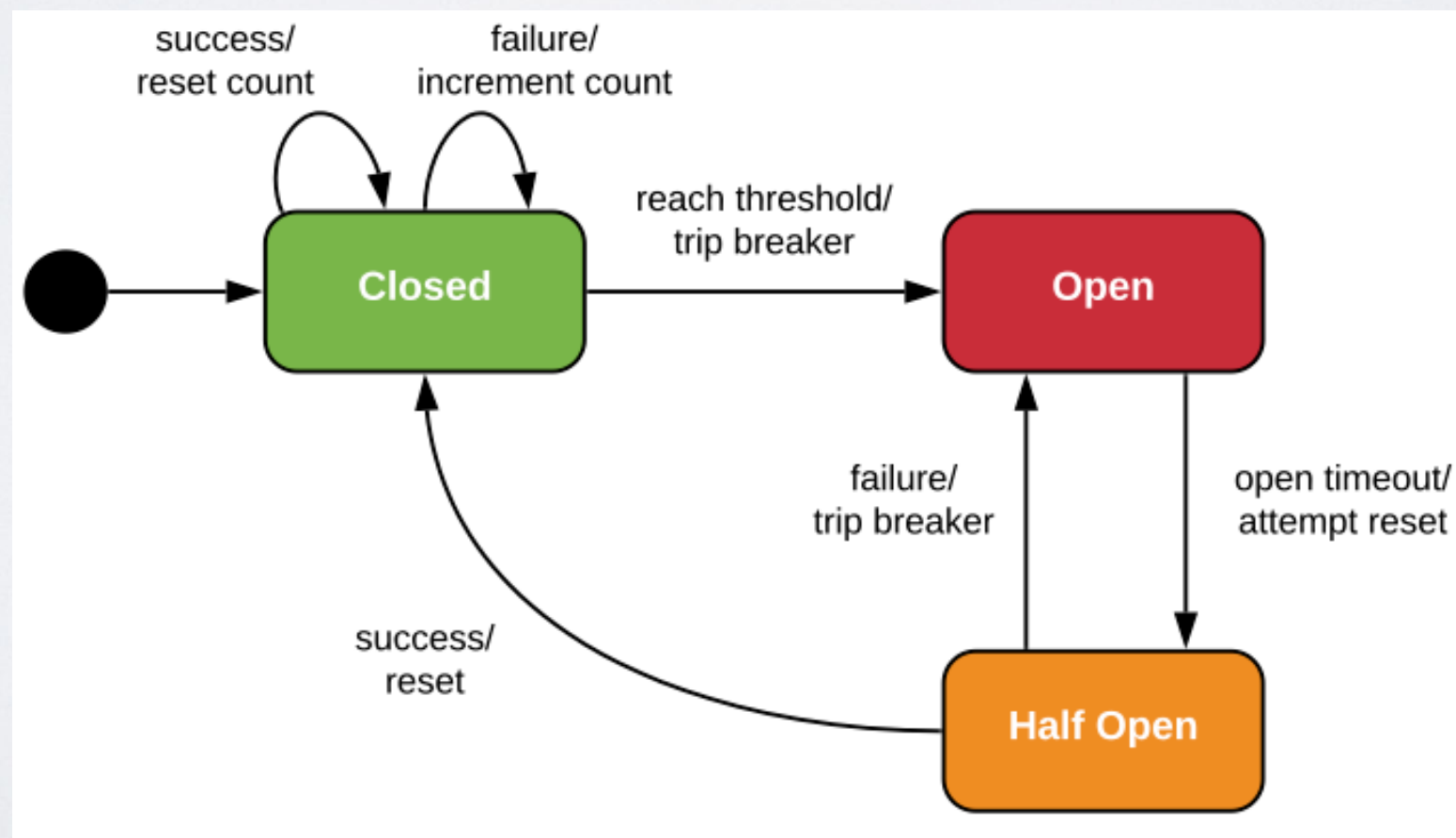
SISTEMAS LEGADOS

- Evitar acoplamento com sistemas antigos - então como migrar?
- Usar API Gateway para abstração.
- Adotar integrações modernas e escaláveis.



DESAFIOS E BOAS PRÁTICAS

- Implementar testes automatizados.
- Gerenciar falhas com circuit breakers.
 - Instrumentação é necessário
- Adotar estratégias de rollout controlado.



OFFER OF THE DAY Buy 10 socks, get a pet human for free!

Login



HOME

CATALOGUE ▾

0 items in cart



WE LOVE SOCKS!

Fun fact: Socks were invented by woolly
creatures to keep warm. They died out because

BEST PRICES

We price check our socks with trained monkeys
back at the office

100% SATISFACTION GUARANTEED

BIBLIOGRAFIA

- GALLOTTI, G. M. A. Arquitetura de Software. 1ª Ed. São Paulo: Pearson, 2016. [acervo digital]
- JUNIOR, A. K. Computação em Nuvem. 1ª Ed. Curitiba: Contentus, 2020. [acervo digital]
- ZANETTI, M. Arquitetura de TI e modelos de negócios. 1ª Ed. Curitiba: Contentus, 2020. [acervo digital]
- Baseado nos materiais de aula de Vinícius Garcia - CIn UFPE

DÚVIDAS?

