

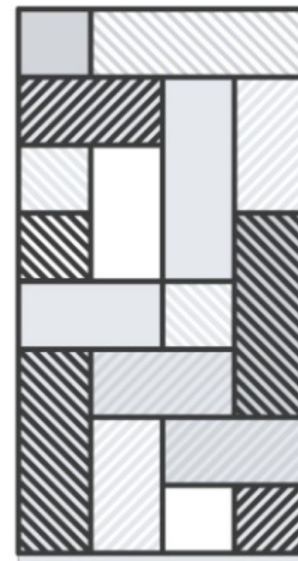
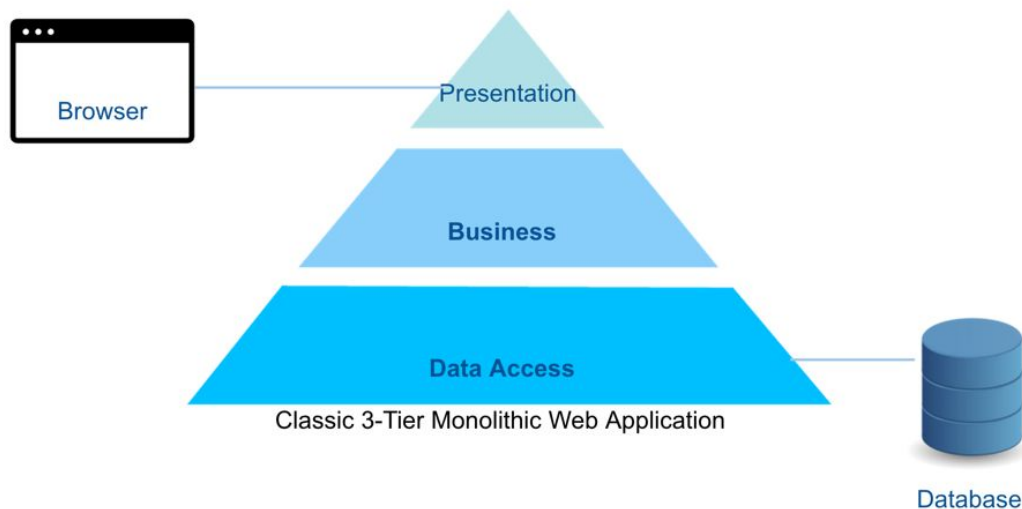


# Desenvolvimento de Software para Nuvem

Prof. Dr. Fernando Antonio Mota Trinta  
Prof. Dr. Paulo Antonio Leal Rego

# Microserviços

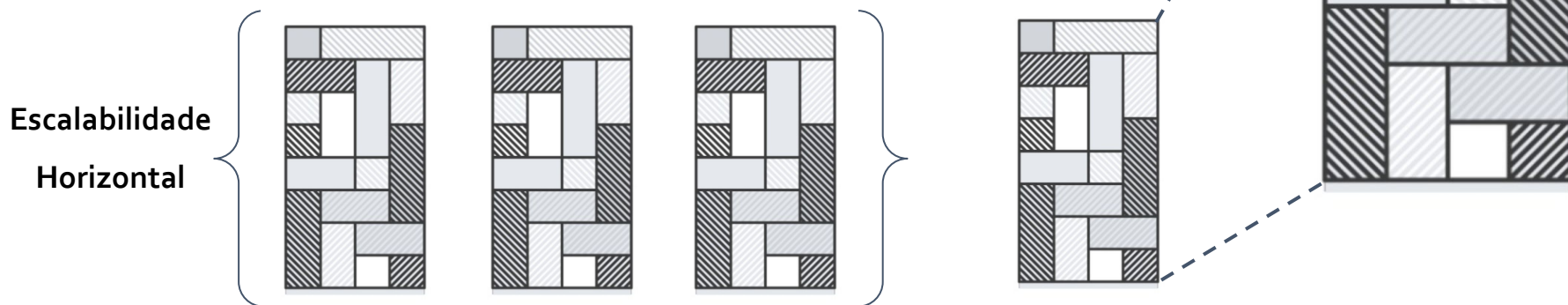
# Aplicações Monolíticas



# Aplicações Monolíticas

## ❑ Desafios

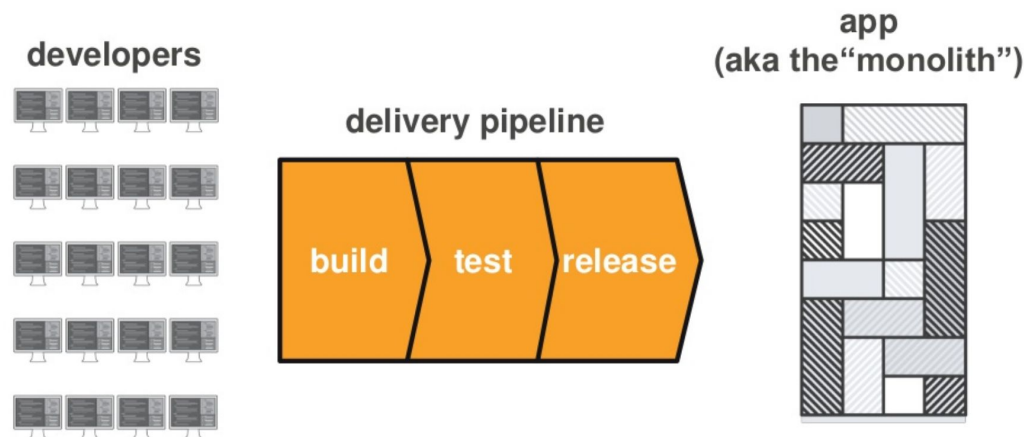
- ❑ Dificuldade de escalar
- ❑ Arquitetura difícil de manter e evoluir



# Aplicações Monolíticas

## ❑ Desafios

- ❑ Ciclos longos de build/test/release
- ❑ Longo tempo para adicionar novas funcionalidades
- ❑ Falta de agilidade, inovação, frustração de clientes



# Aplicações Monolíticas

- ❑ Tudo integrado



## Arquitetura de Microserviços

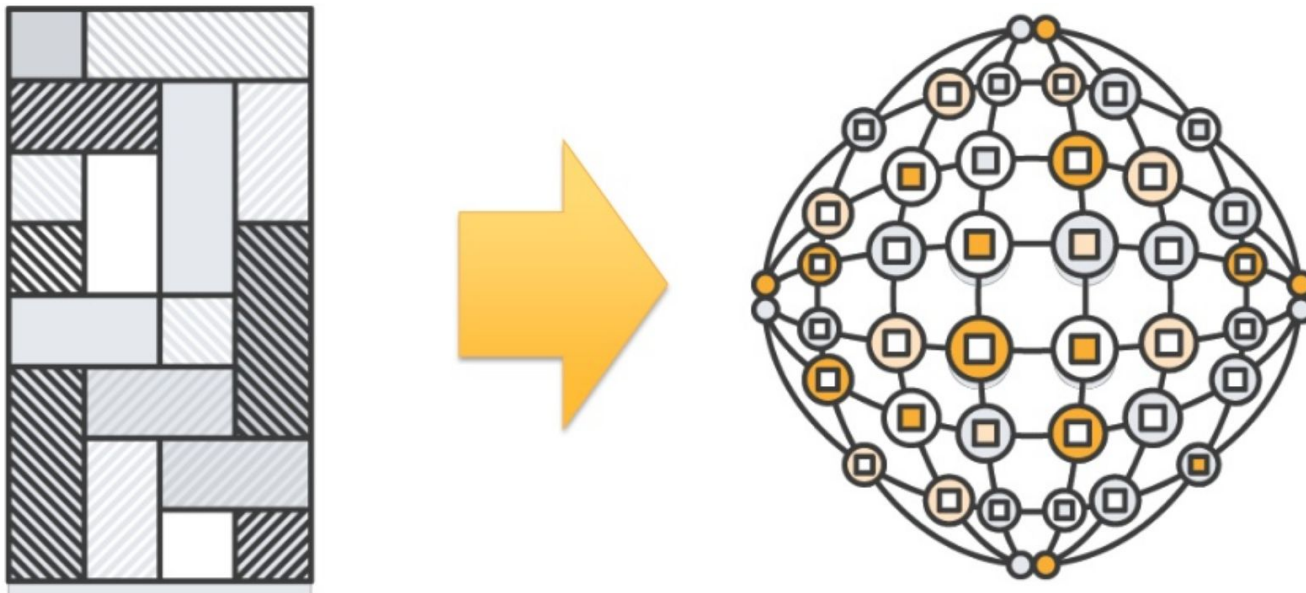
- ❑ É uma abordagem que desenvolve um aplicativo único como uma suíte de pequenos serviços, cada um executando seu próprio processo e se comunicando através de mecanismos leves, muitas vezes em uma API com recursos HTTP.
- ❑ Serviços funcionam através de mecanismos de deploy independentes totalmente automatizados.
- ❑ Há o mínimo possível de gerenciamento centralizado desses serviços, que podem ser escritos em diferentes linguagens de programação e utilizam diferentes tecnologias de armazenamento de dados.

## Arquitetura de Microserviços

- ❑ Decompõe a aplicação por funções básicas.
  - ❑ Cada função é denominada um serviço e pode ser criada e implantada de maneira independente.
  - ❑ Cada serviço individual pode funcionar ou falhar sem comprometer os demais.

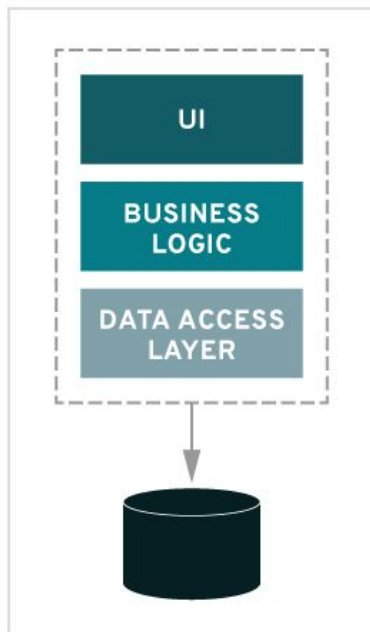


# Aplicações Monolíticas x Microserviços



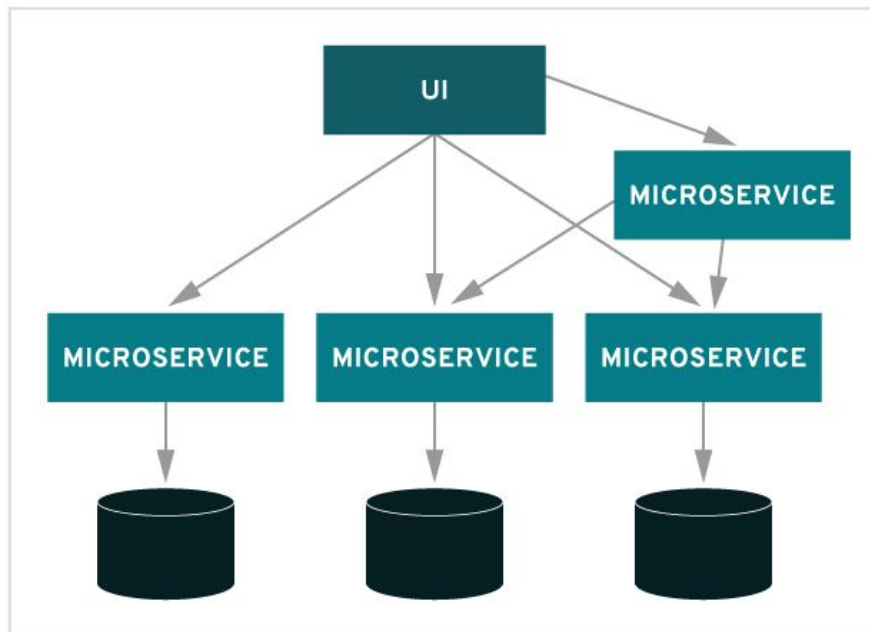
# Aplicações Monolíticas x Microserviços

## MONOLITHIC



VS.

## MICROSERVICES

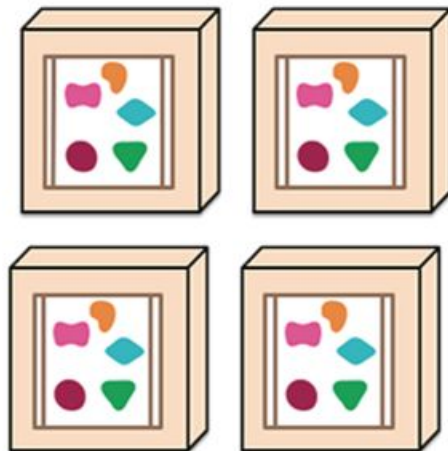


# Aplicações Monolíticas x Microserviços

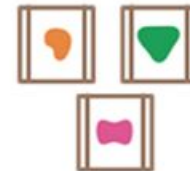
*Um aplicativo monolítico tem todas as suas funcionalidades em um único processo...*



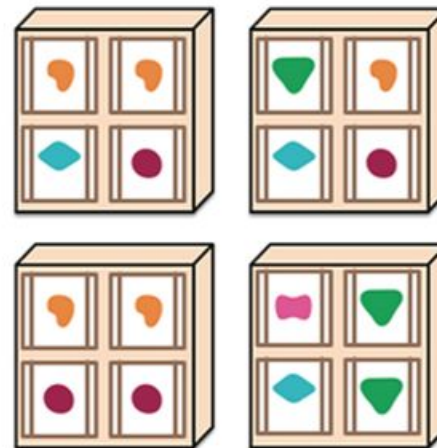
*...e escala replicando o monolito em servidores múltiplos*



*A arquitetura de micro-serviços coloca cada elemento de funcionalidade em um serviço separado...*



*...e escala distribuindo os serviços entre os servidores, replicando por demanda.*



## Assuntos relacionados

- ❑ Arquitetura Orientada a Serviços (SOA)
- ❑ Webservices

## Arquitetura Orientada a Serviços (SOA)

- ❑ É um paradigma para a realização e manutenção de processos de negócio em um grande ambiente de sistemas distribuídos que são controlados por diferentes proprietários.
- ❑ SOA é uma arquitetura conceitual onde funcionalidade do negócio, ou lógica da aplicação, é disponibilizada para usuários como serviços compartilhados e reutilizáveis.

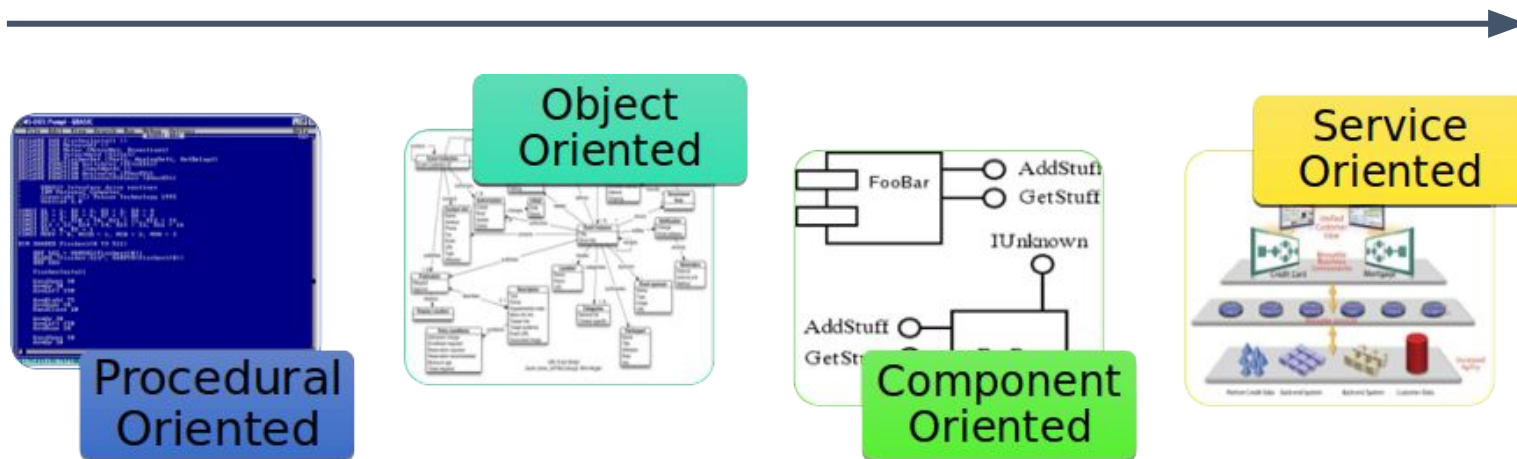
## Arquitetura Orientada a Serviços (SOA)

- ❑ Serviço corresponde a uma representação lógica de uma atividade do negócio que pode ser mapeada em entrada, processamento e saída.



# Arquitetura Orientada a Serviços (SOA)

## ❏ Caminho que levou à SOA



## Arquitetura Orientada a Serviços (SOA)

- ❑ Algumas características (ou princípios) de serviços são:
  - ❑ Deve ser operacionalmente independente, garantindo alta coesão e acoplamento fraco;
  - ❑ Deve ser sem estado (stateless);
  - ❑ Deve permitir composição;
  - ❑ Deve ser atômico/auto-contido;
  - ❑ Devem garantir interoperabilidade (i.e., poder se comunicar com consumidores ou consumir outros serviços desenvolvidos empregando diversificadas tecnologias;
  - ❑ Deve permitir reuso elevado.

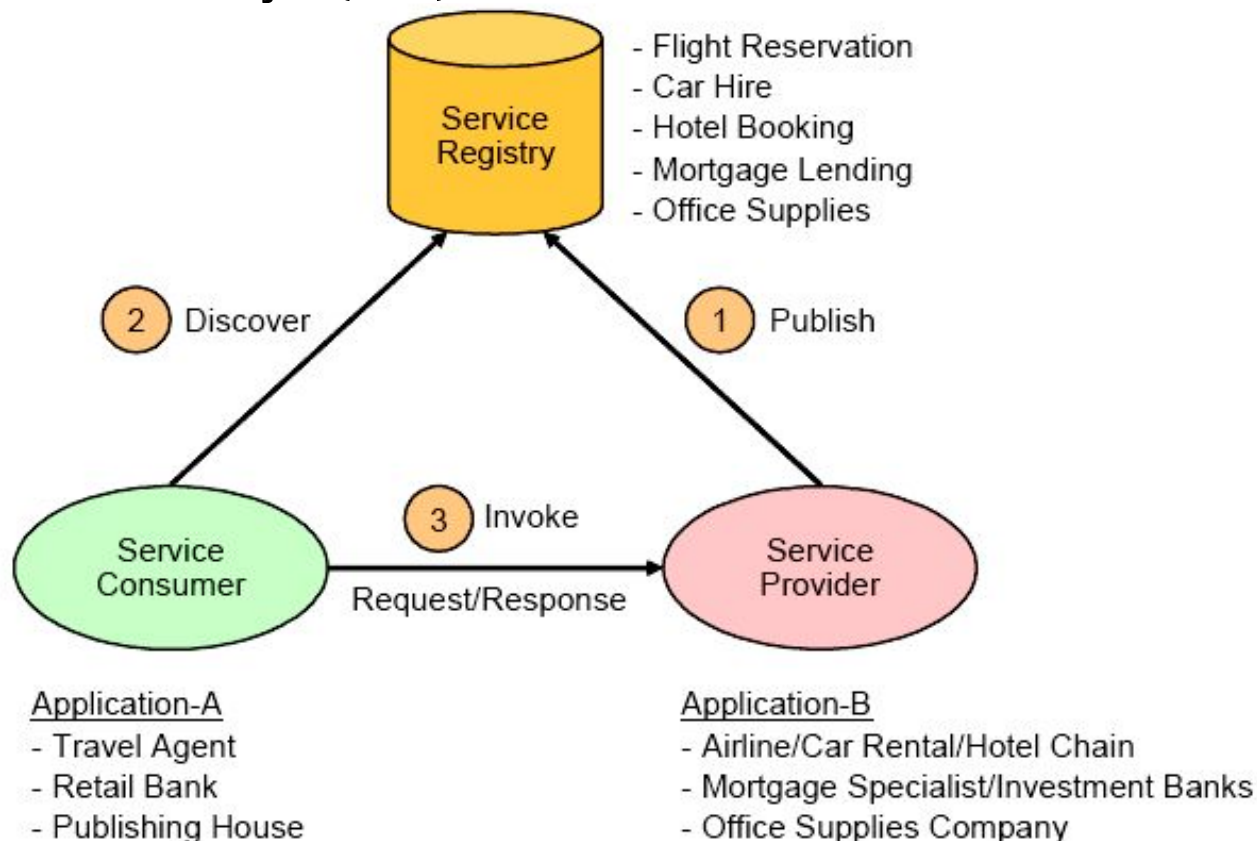


## Arquitetura Orientada a Serviços (SOA)

- ❑ Onde comprar um livro?
  - ❑ Classificados? Mercado Livre? Amazon?
- ❑ Encontrar alguém que ofereça o serviço de acordo com minhas necessidades
- ❑ Comprar o livro



# Arquitetura Orientada a Serviços (SOA)



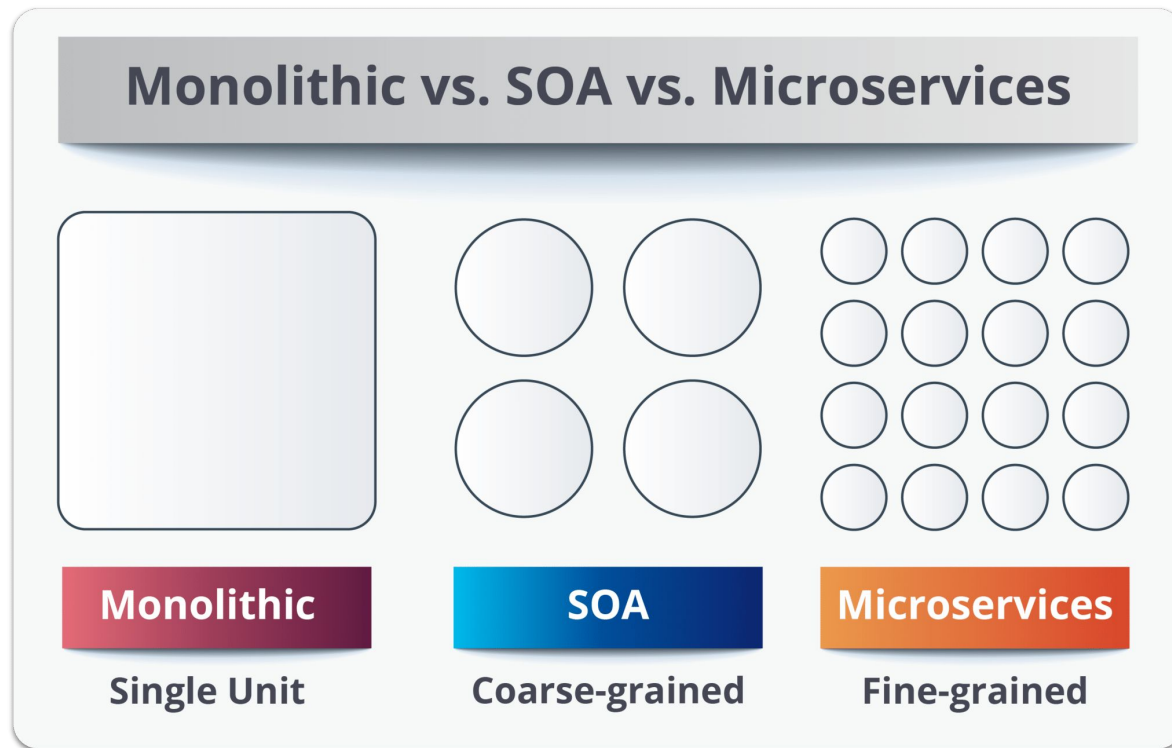
## SOA x Microserviços

- ❑ Os microserviços podem se comunicar entre si, normalmente de maneira stateless.
  - ❑ Por meio de APIs independentes de linguagem.
- ❑ Microserviços não são uma ideia completamente nova. Porém, eles se tornaram mais viáveis graças aos avanços nas tecnologias de containers.
  - ❑ Partes de uma aplicação podem ser executadas de maneira independente no mesmo hardware e com um controle muito maior sobre os componentes individuais e ciclos de vida.

## SOA x Microserviços

- ❑ Os microserviços são considerados uma versão mais moderna da SOA.
  - ❑ São menores (fine grained), escopo claro e bem definido, gerenciados de forma independente, etc
- ❑ Novo mundo:
  - ❑ Velocidade de entrega / time-to-market
  - ❑ Inovação / experimentação
  - ❑ Escalabilidade
  - ❑ Cloud / DevOps

# SOA x Microserviços

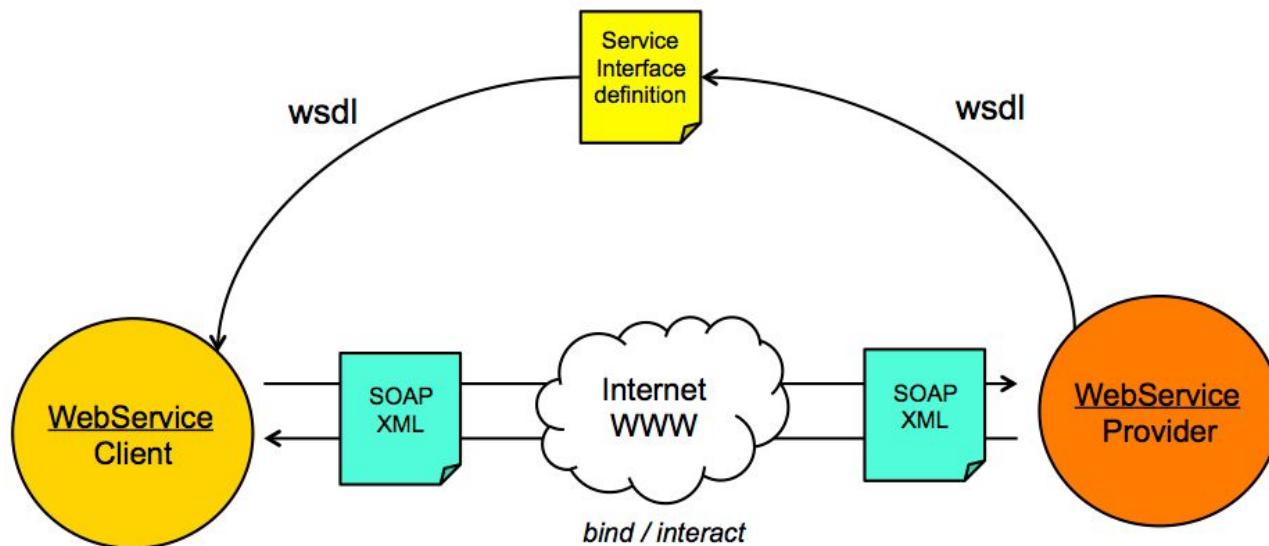


## Webservices

- ❑ Sistema de software projetado para apoiar interação interoperável máquina-a-máquina através de uma rede.
  - ❑ Tem uma interface descrita em um formato processável por máquina.
  - ❑ Outros sistemas interagem com o web service empregando um protocolo de comunicação (tipicamente HTTP)
- ❑ Existem dois tipos de web services:
  - ❑ Web service SOAP
  - ❑ Web service RESTful

## Webservices SOAP

- Empregam um conjunto de tecnologias baseadas em XML, tais como SOAP, WSDL, XSD e UDDI.

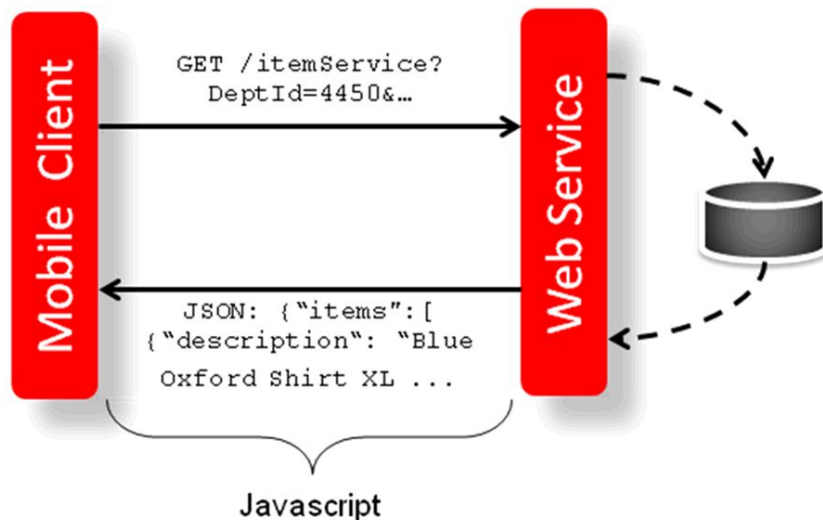


## Webservices RESTful

- ❑ Alternativa mais simples a Web services SOAP.
  - ❑ Mais fácil de utilizar.
  - ❑ Serviços são vistos como recursos e podem ser identificados unicamente por suas URLs.
  - ❑ Não impõe restrições no formato da mensagem.
  - ❑ Formatos mais comuns: JSON, XML e texto puro, mas em teoria qualquer formato pode ser usado.
  - ❑ Principal característica: usar explicitamente os métodos HTTP (POST, GET, PUT, DELETE) para denotar a invocação de diferentes operações.

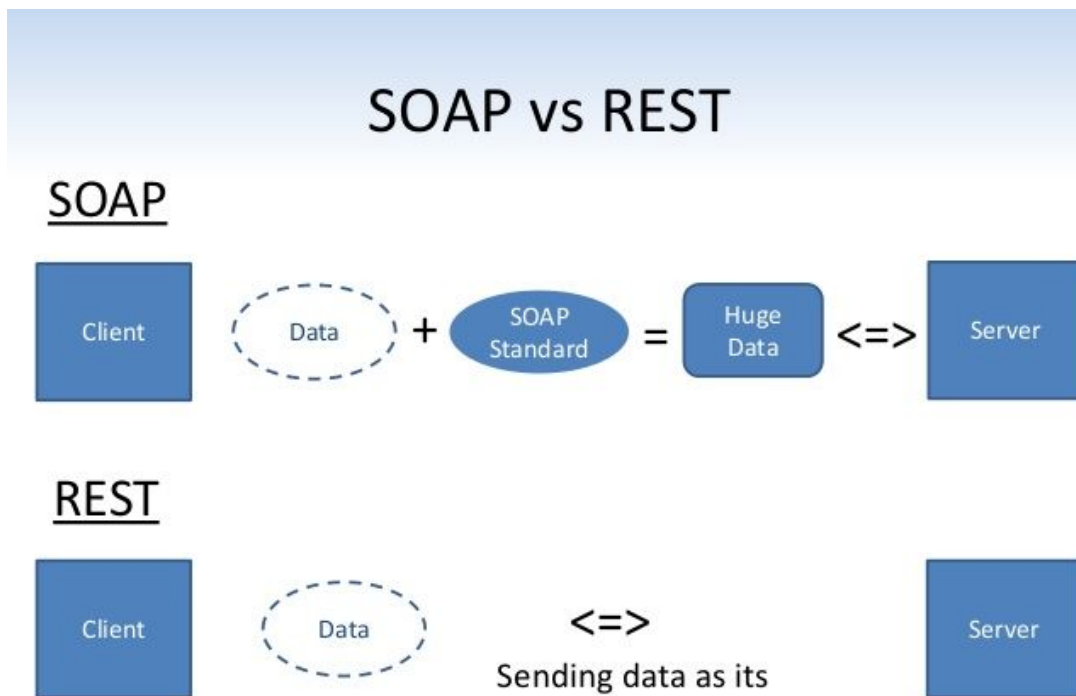


# Webservices RESTful



GET	/movies	Get list of movies
GET	/movies/:id	Find a movie by its ID
POST	/movies	Create a new movie
PUT	/movies	Update an existing movie
DELETE	/movies	Delete an existing movie

# SOAP x RESTful



\*REST are mostly used in industry

## Princípios da arquitetura de microsserviços

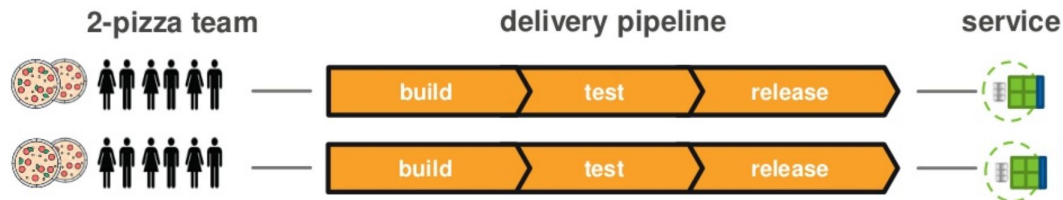
1. Serviços com responsabilidades específicas e independentes que encapsulam tanto lógica de processamento quanto dados.
  - a. Dados são expostos através de APIs Web ou filas de mensagens assíncronas - não devem ser acessados diretamente.



2. Práticas de desenvolvimento orientadas ao negócio: Técnicas e princípios para identificar e conceituar serviços, e.g., dividir grandes domínios em contextos menores.

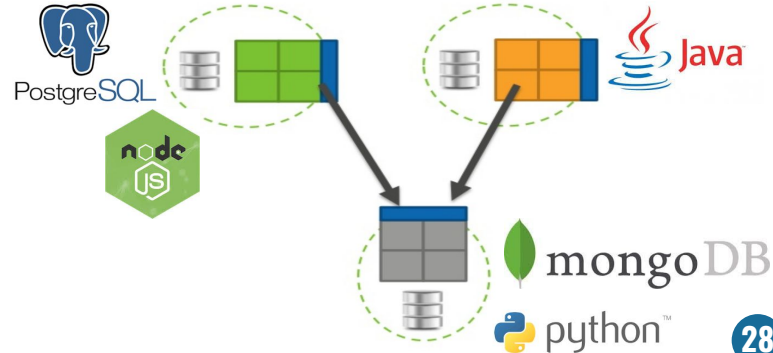
## Princípios da arquitetura de microsserviços

3. Princípios de design baseados em computação em nuvem, tais como distribuição, elasticidade e baixo acoplamento.
4. Entrega contínua descentralizada: Promove um alto nível de automação e autonomia. Demanda maturidade para construir artefatos de forma automatizada, aliada a uma suíte de testes.
5. Contêineres leves: Uso de metodologias de containerização de componentes de software e processos de entrega contínua.

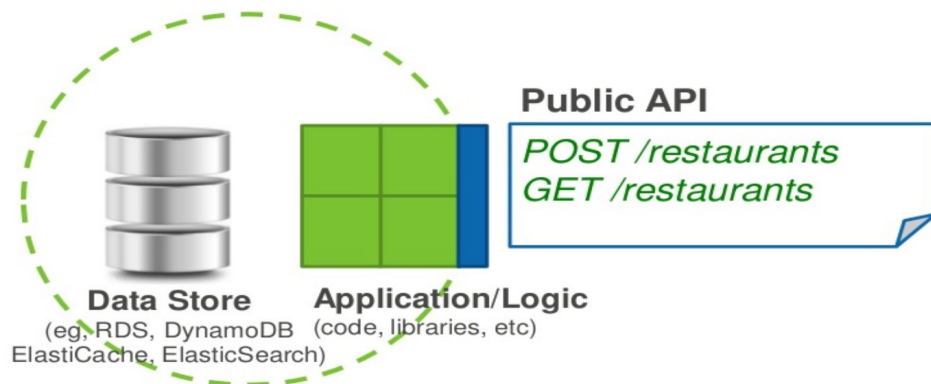


## Princípios da arquitetura de microsserviços

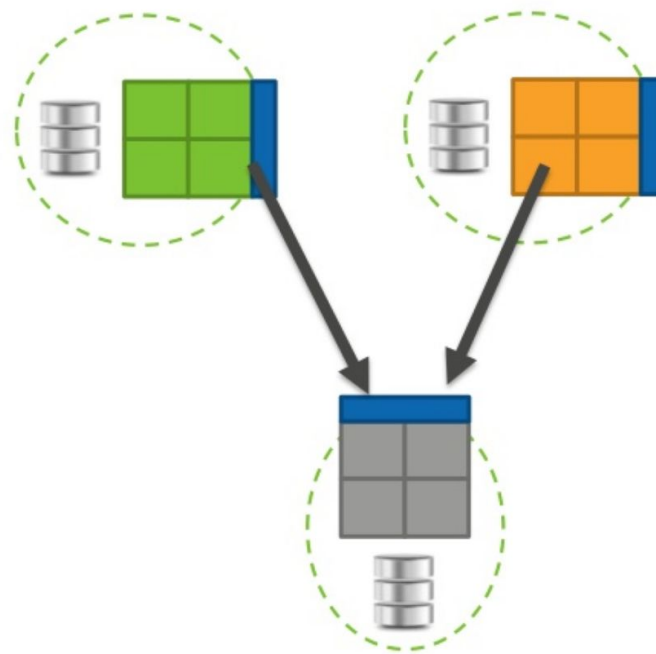
6. DevOps: Uso de técnicas e automatização de configuração, desempenho e gerenciamento de falhas, estendendo práticas ágeis para o monitoramento de serviços.
7. Uso da ferramenta correta: Combina ganhos de diferentes abordagens de computação, linguagens de programação e de tipos distintos de armazenamento.



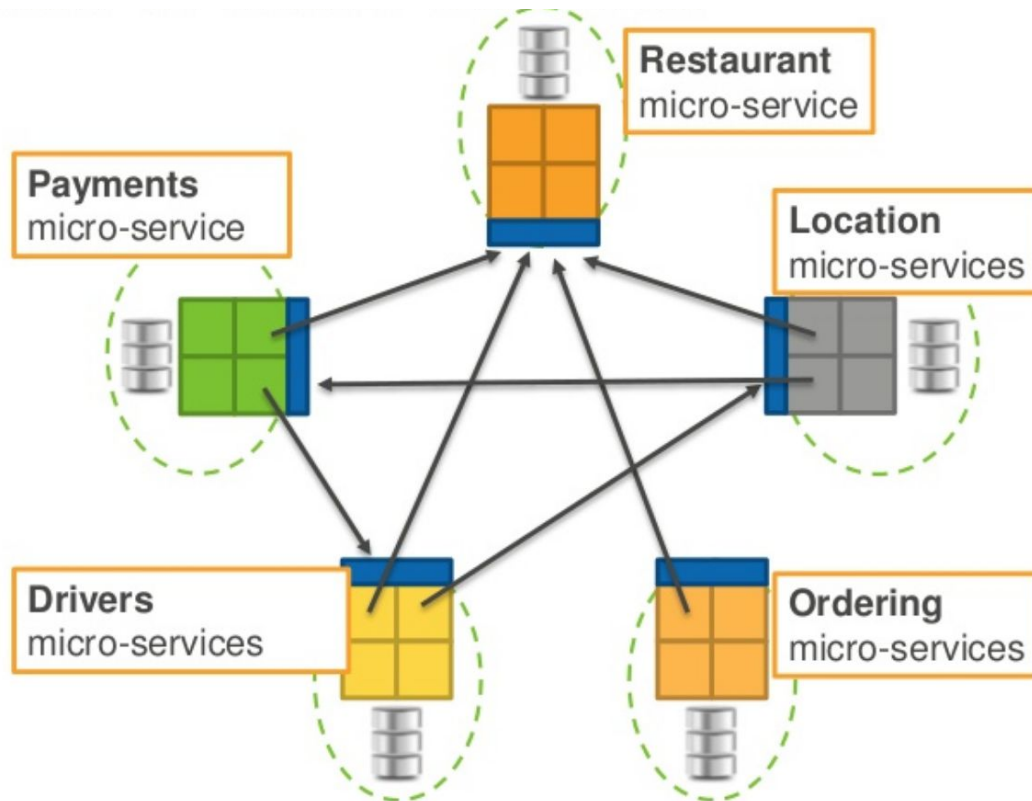
# Anatomia de um microserviço



# Reduzir o acoplamento de software



# Ecosystem of microservices



## Benefícios da arquitetura de microsserviços

- ❑ Mais open source: a adoção de novos frameworks/tecnologias pode ser aplicada em um microsserviço de cada vez.
- ❑ Os microsserviços são pequenos
  - ❑ Melhor isolamento de falhas; iniciam mais rápido.
- ❑ Lançamento no mercado com mais rapidez
- ❑ Altamente escalável
- ❑ Resiliente
- ❑ “Fácil” de implementar e implantar
- ❑ Acessível



## Desvantagens da arquitetura de microsserviços

- ❑ Complexidades adicionais do desenvolvimento de sistemas distribuídos:
  - ❑ Aplicações muito mais complexas e constituídas por mais elementos;
  - ❑ Uso de transações distribuídas ou consistência eventual;
  - ❑ Gerenciamento de um número maior de aplicações.
- ❑ Para ser utilizada de forma eficaz, a arquitetura de microsserviços exige um alto nível de automação.

## Desafios ao migrar para arquitetura de microsserviços

- ❑ Mudanças não somente nas aplicações, mas também no modo como as pessoas trabalham.
  - ❑ Mudanças organizacionais e culturais
- ❑ Complexidade e eficiência
- ❑ Identificação das dependências entre os serviços.
- ❑ Testes de integração e testes end-to-end podem ser mais difíceis e importantes como jamais foram, pois uma falha pode reverberar em outros serviços.

## Desafios ao migrar para arquitetura de microsserviços

- ❑ Controle de versão: ao atualizar para versões novas, a compatibilidade com versões anteriores pode ser rompida.
- ❑ Implantação (Configuração inicial): é necessário investir em automação, pois a complexidade dos microsserviços é demais para a implantação manual.
- ❑ Geração de logs: é necessário ter logs centralizados para unificar e facilitar o acompanhamento dos mesmos.

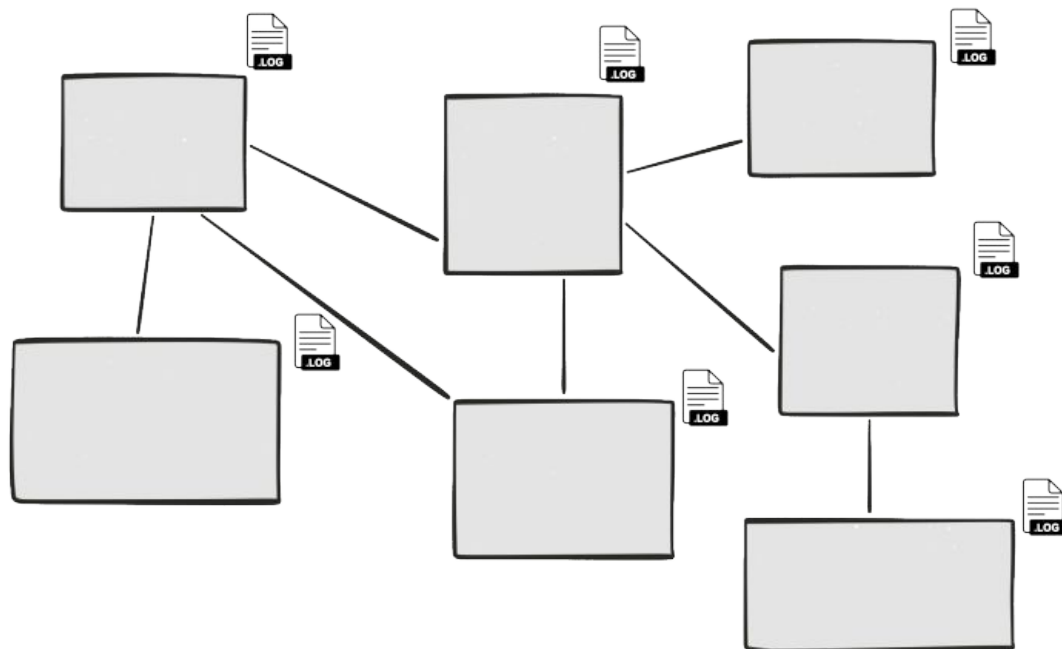
## Desafios ao migrar para arquitetura de microsserviços

- ❑ Monitoramento: é importante uma visão centralizada do sistema para identificar as fontes de problemas.
- ❑ Depuração: a depuração remota não é uma opção e não funciona com centenas de serviços.
- ❑ Conectividade: considere a detecção de serviços, seja de maneira centralizada ou integrada.

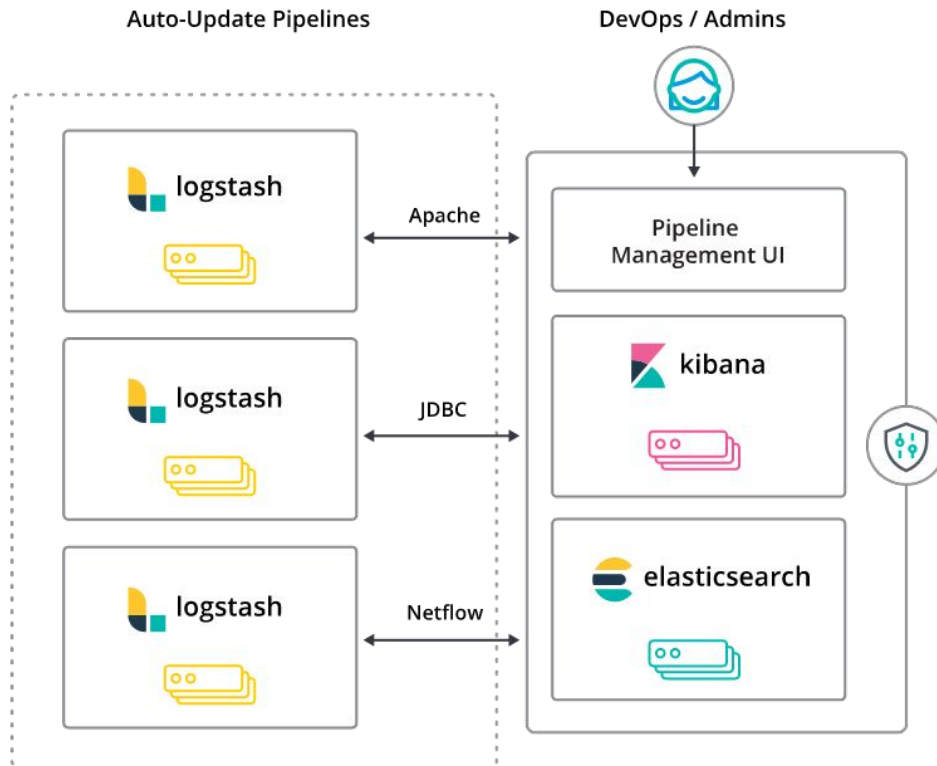
## Desafios ao migrar para arquitetura de microsserviços

- ❑ Monitoramento: é importante uma visão centralizada do sistema para identificar as fontes de problemas.
- ❑ Depuração: a depuração remota não é uma opção e não funciona com centenas de serviços.
- ❑ Conectividade: considere a detecção de serviços, seja de maneira centralizada ou integrada.

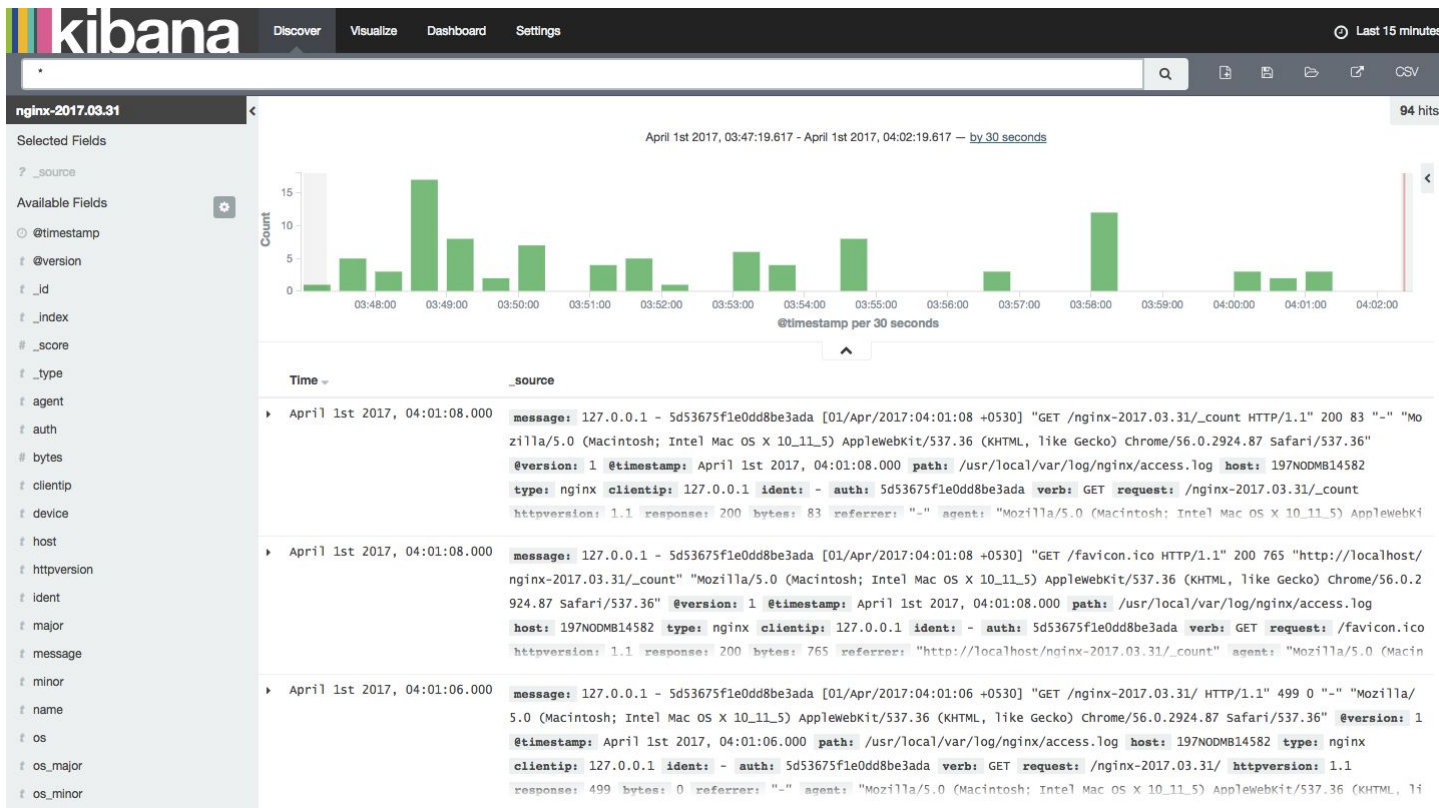
# Logging Centralizado



# Logging Centralizado

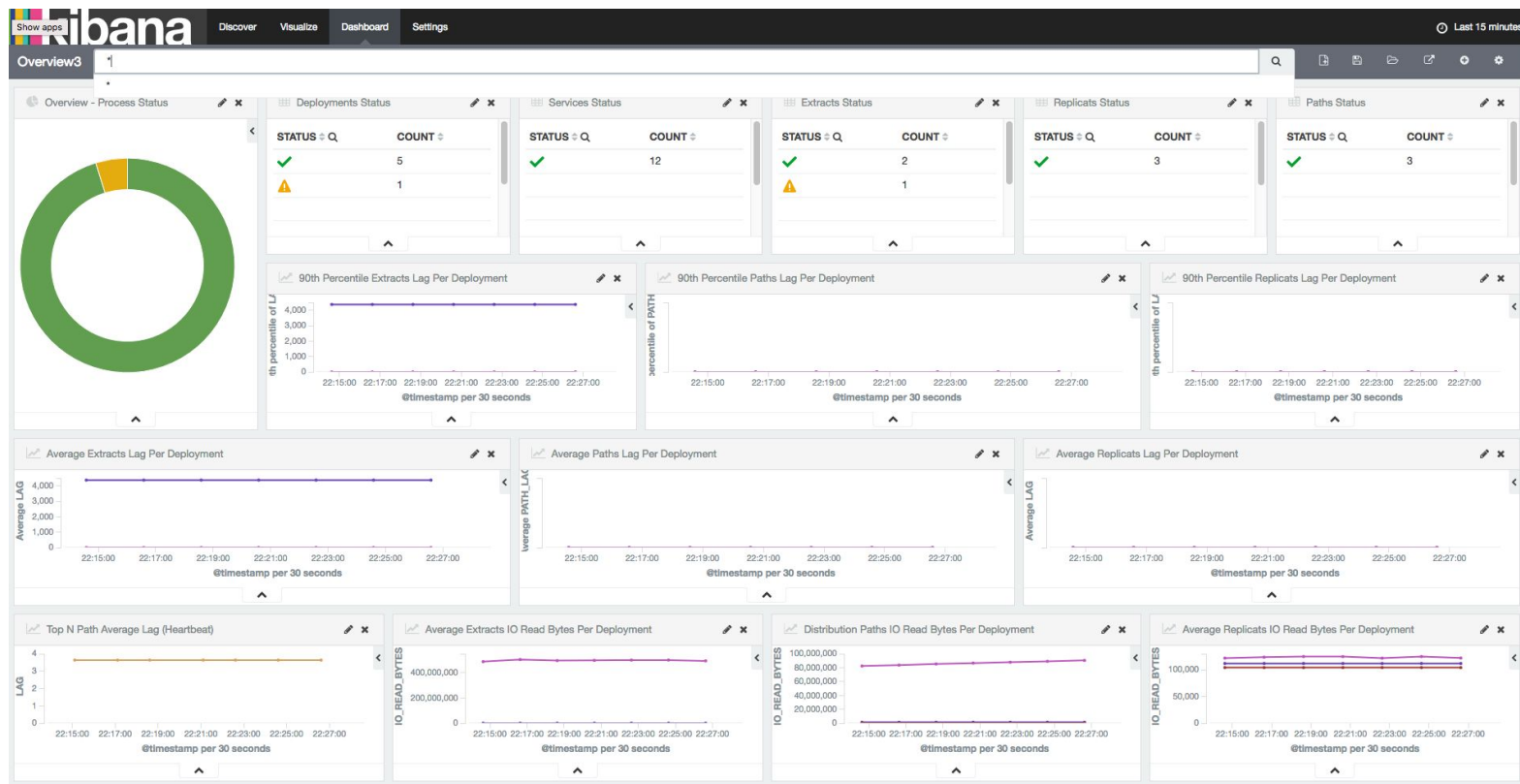


# Logging Centralizado

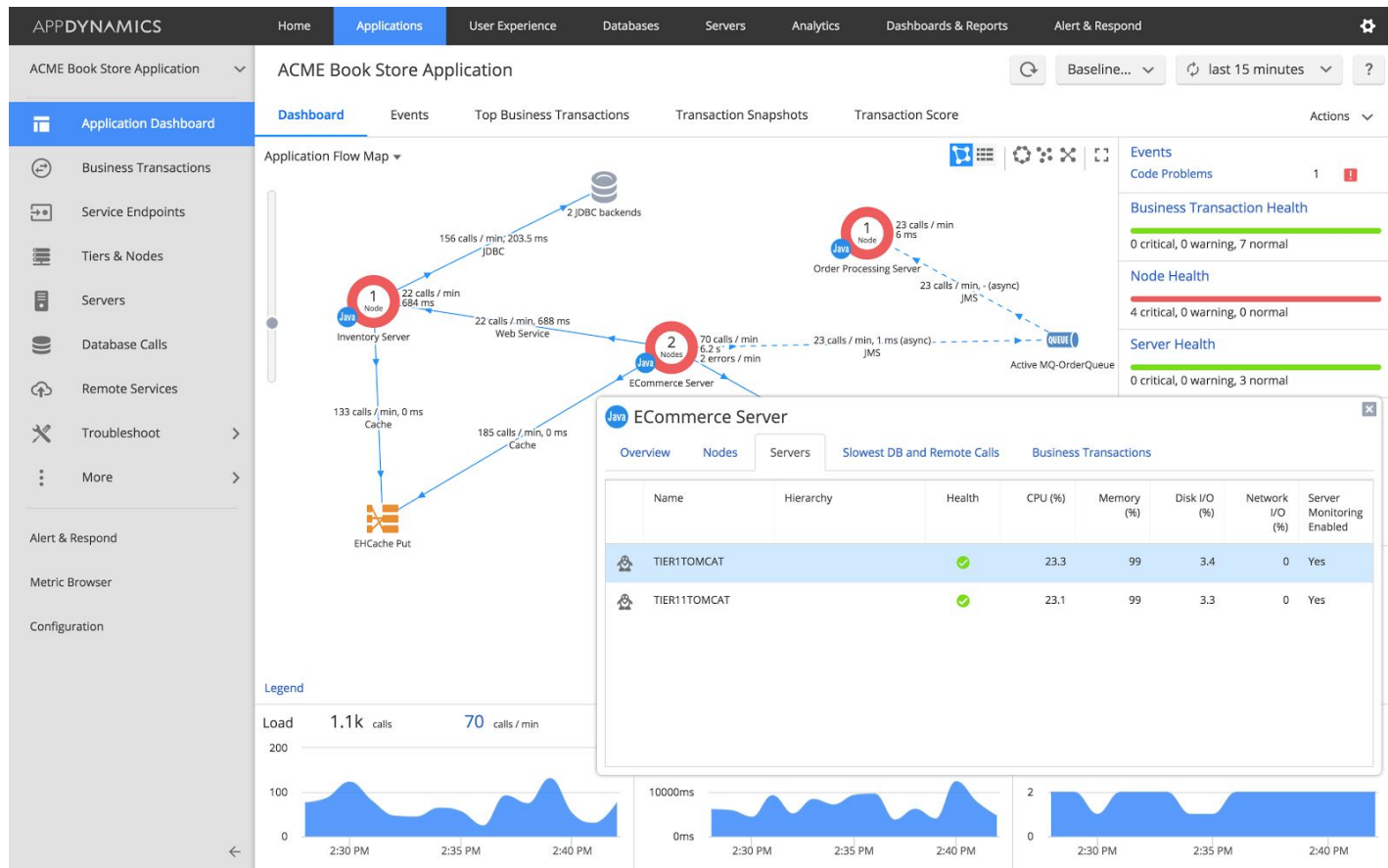




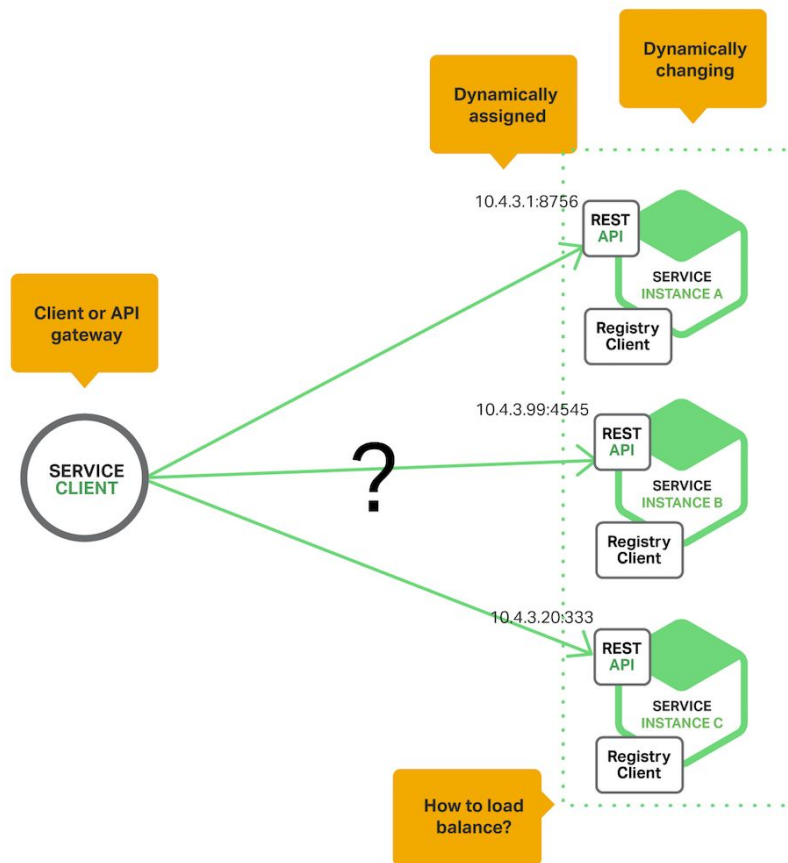
# Monitoramento



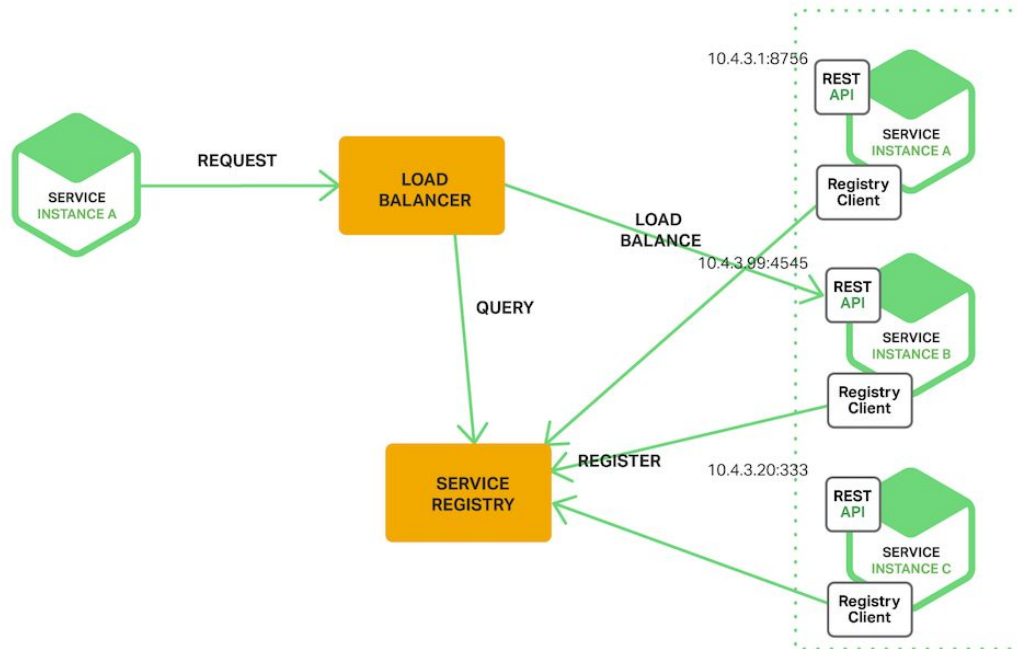
# Monitoramento



# Descoberta de serviços



# Descoberta de serviços



## Referências

1. Luís H. N. Villaça *et al.* Construindo Aplicações Distribuídas com Microserviços. Minicursos do SBSI 2018. URL: <<https://www.ucs.br/site/midia/arquivos/topicos-sistema-informacao.pdf>>
2. AWS. Introduction to Microservices. URL: <<https://www.slideshare.net/AmazonWebServices/introduction-to-microservices-78575696>>.
3. James Lewis e Martin Fowler. Microservices: a definition of this new architectural term. URL: <<https://martinfowler.com/articles/microservices.html>>.
4. Nicola Dragoni *et al.* Microservices: yesterday, today, and tomorrow. In: Present and ulterior software engineering. Springer, Cham, 2017. p. 195-216.
5. Sam Newman. Building microservices: designing fine-grained systems. "O'Reilly Media, Inc.", 2015.
6. Red Hat. O que são microserviços? URL: <<https://www.redhat.com/pt-br/topics/microservices/what-are-microservices>>