

19/10

PROJECT
Decoder

Fundamentos Microservices

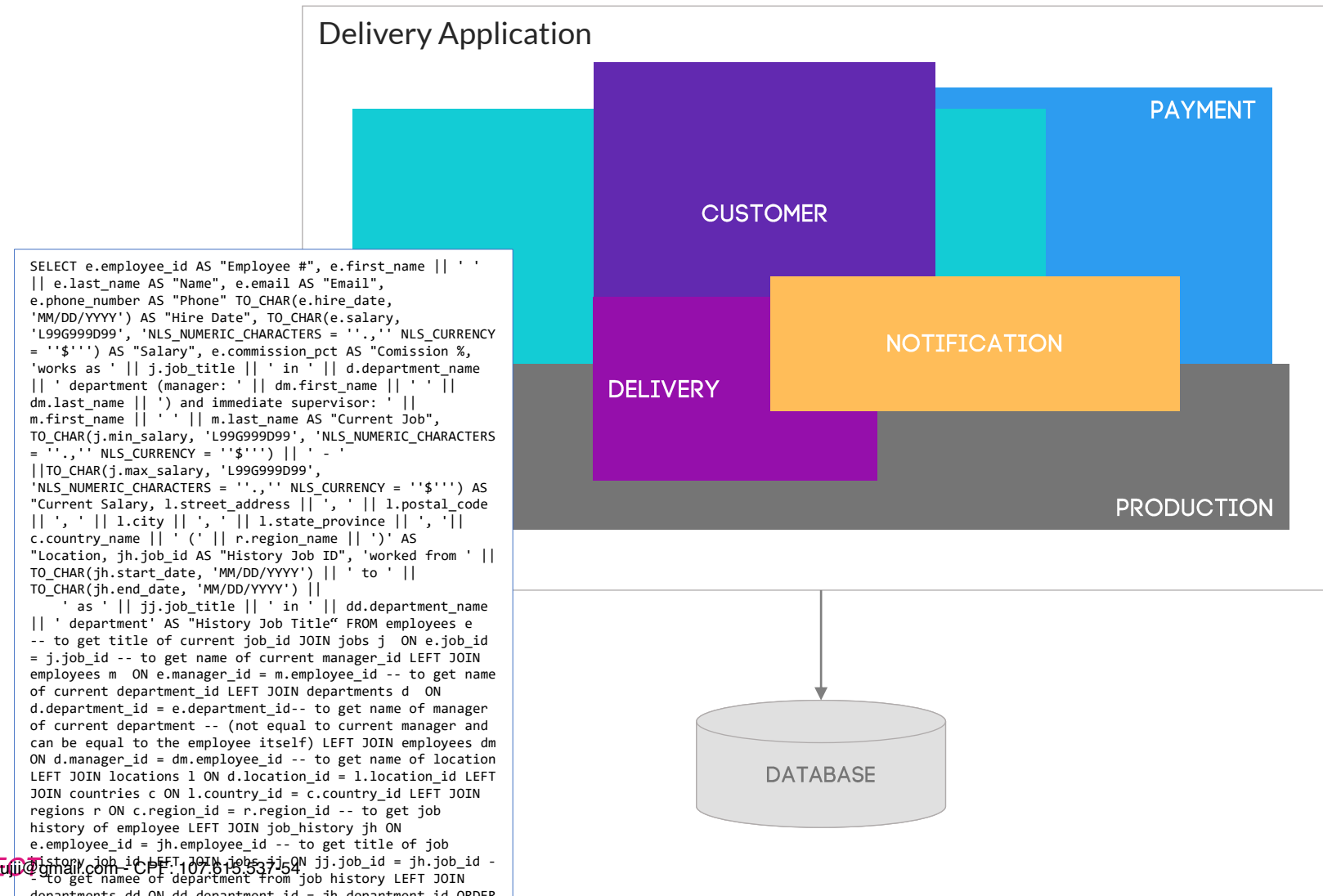
Desafios e Equívocos

O que você verá:

- Como a Arquitetura de Microservices veio para solucionar as complexidades dos negócios modernos;
- Os fundamentos de Microservices;
- Princípios e equívocos que envolvem esse modelo arquitetural;
- Sincronia de dados em sistemas distribuídos;
- Consistência Eventual e o Teorema CAP;
- Identificadores distribuídos;
- Questão do acoplamento e disponibilidade em Arquitetura de Microservices;
- Distribuição das base de dados em Microservices;
- Concepção e utilização de Microservices;
- Algumas premissas importantes sobre Microservices;
- E muito mais....



Arquitetura Monolítica



Expectativa



Realidade





Arquiteto(a)

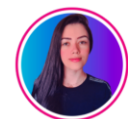
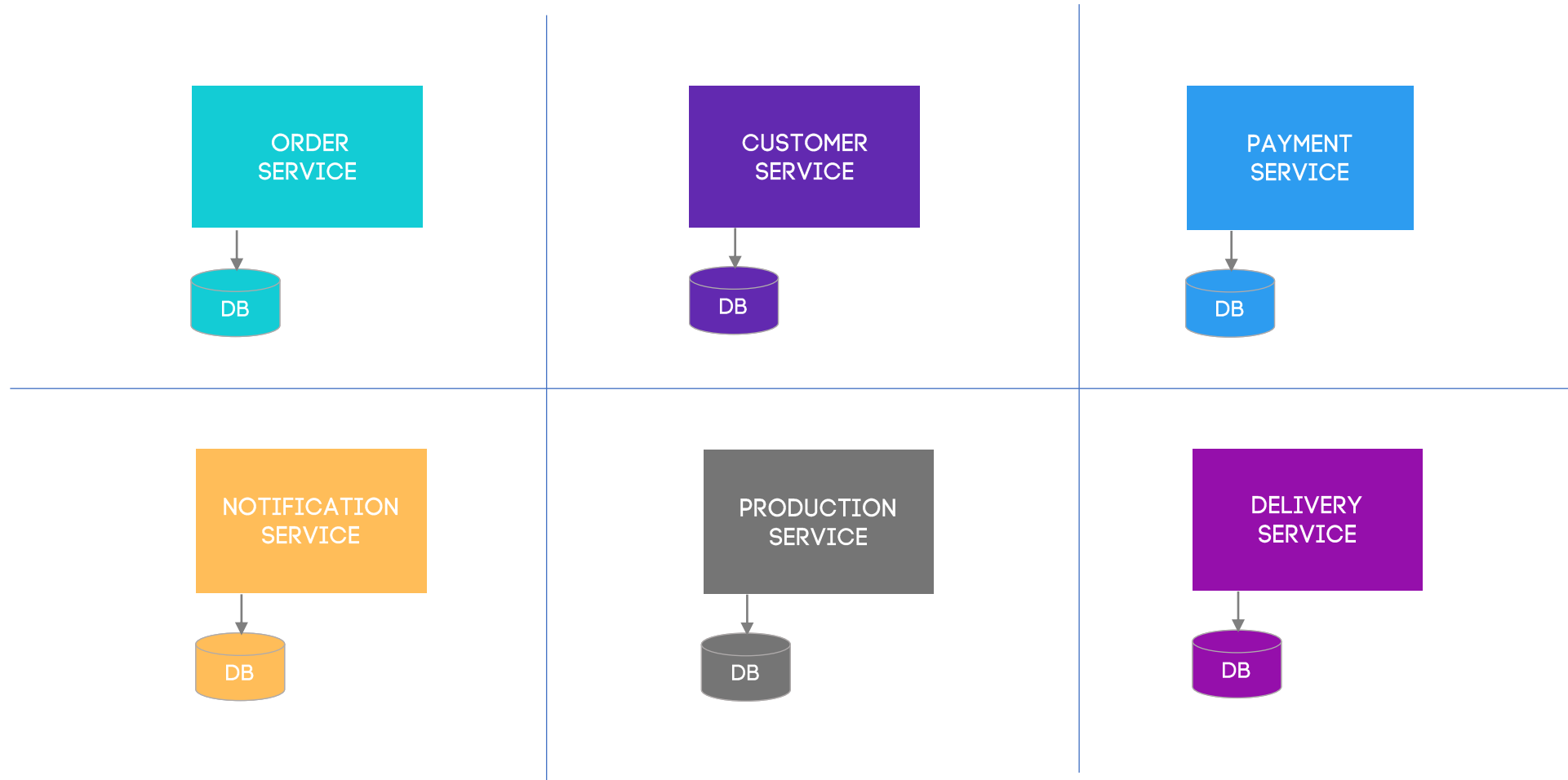
Monolito

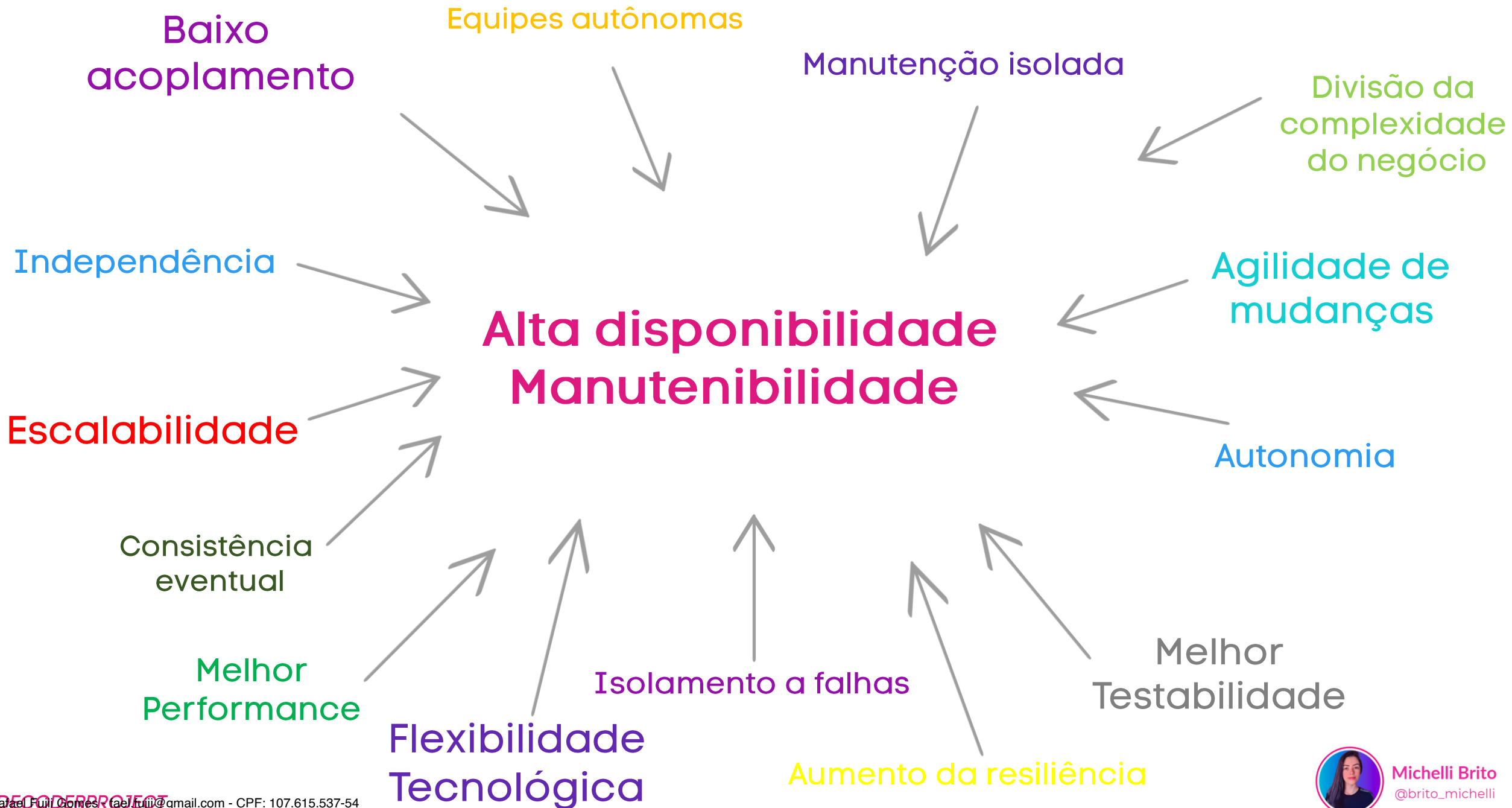
Só
precisamos
de um
ajuste!



Contrata-se Desenvolvedor(a)

Arquitetura de Microservices





Fundamentos de Microservices

Arquitetura de Microservices envolve desde conceitos técnicos, financeiros e gerenciais moldados principalmente sobre a necessidade do negócio.

A mudança de paradigma para Microservices requer também uma mudança equivalente na estrutura da organização. Não é só uma questão técnica. É também cultural e organizacional.

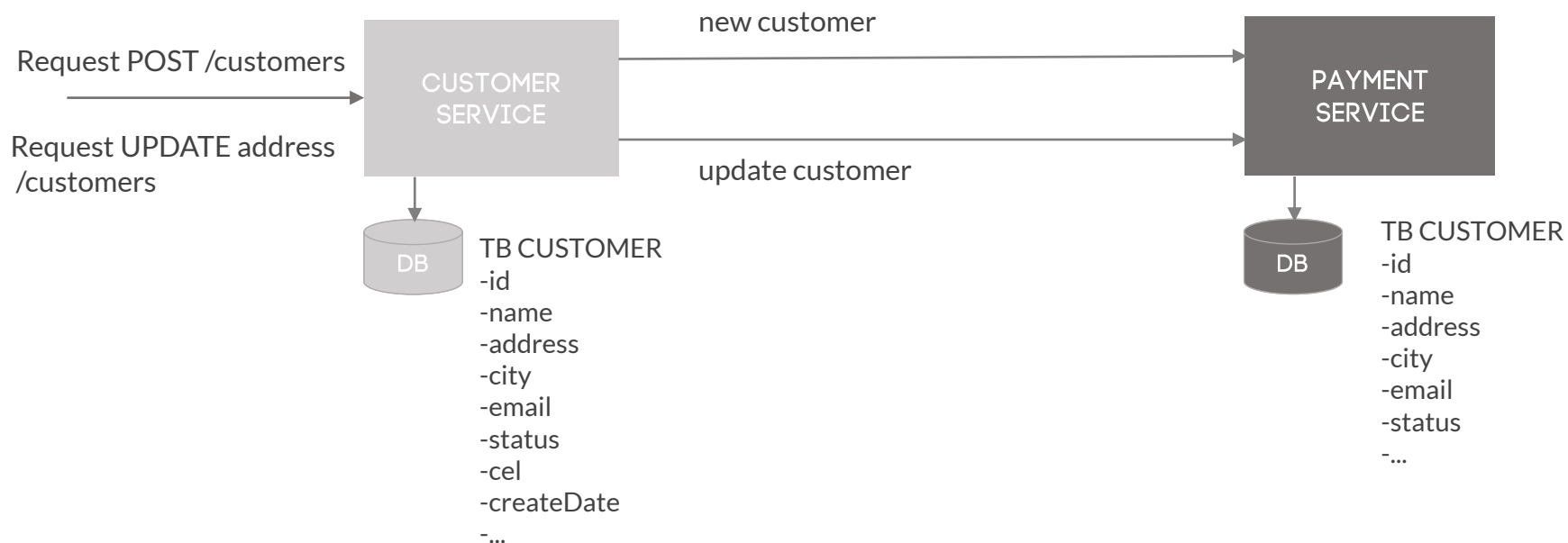
Arquitetura de Microservices é um subconjunto dos conceitos de sistemas distribuídos modernos.



Michelli Brito
@brito_michelli

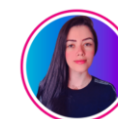
Sincronia de Dados em Sistemas Distribuídos

Não há como garantir alta disponibilidade com consistência forte ao mesmo tempo em uma arquitetura de Microservices com dados distribuídos.



CONSISTÊNCIA EVENTUAL

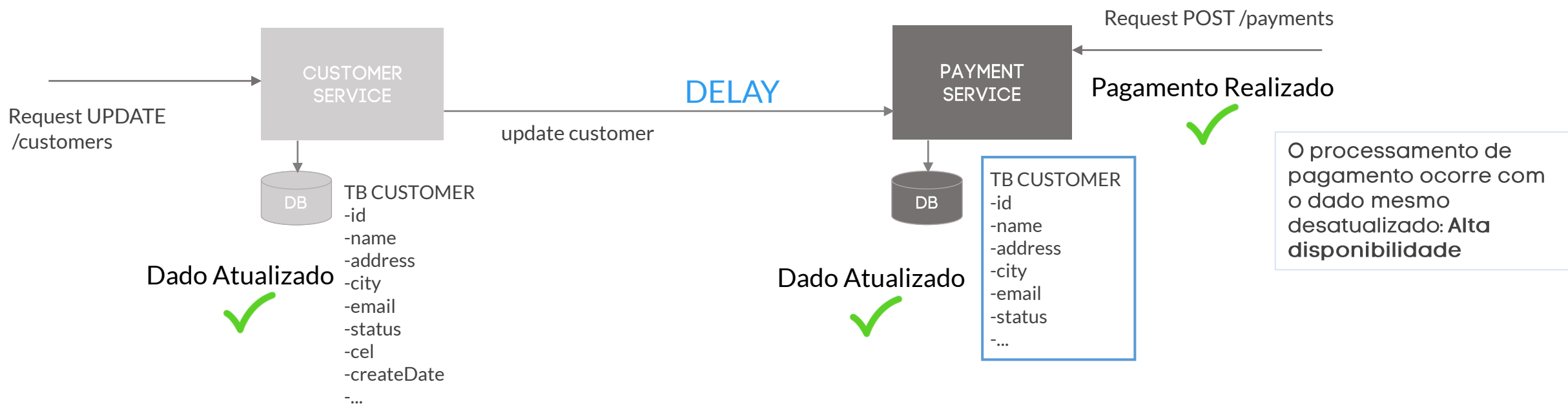
Dado do cliente replicado no microservice Payment pela necessidade do negócio.



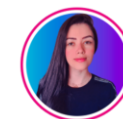
Michelli Brito
@brito_michelli

Disponibilidade vs Consistência Forte : Teorema CAP

Para garantir a alta disponibilidade em sistemas distribuídos com dados compartilhados precisamos que a sincronia de dados ocorra de maneira assíncrona e não bloqueante, causando assim uma consistência eventual, onde os dados podem estar momentaneamente desatualizados ou ainda não replicados quando consultados.

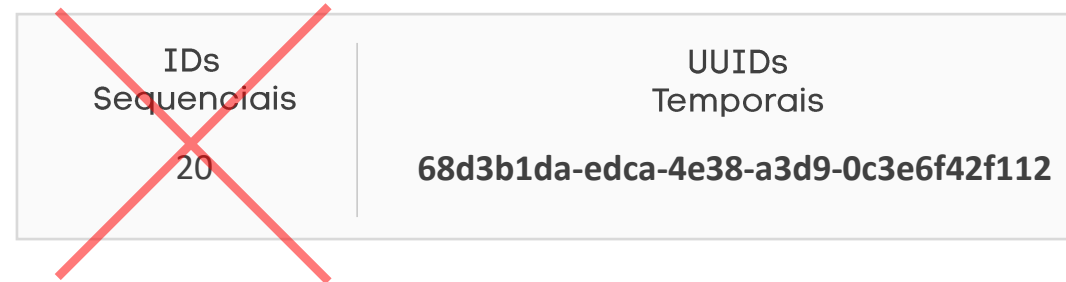


Priorizando a disponibilidade : consistência eventual.



Identificadores UUIDs

IDs do tipo UUID são identificadores temporais universalmente exclusivos e essenciais para sincronia de dados distribuídos.



podem ser gerados em qualquer lugar

garantem maior manutibilidade

facilita replicação de dados

únicos em qualquer base de dados



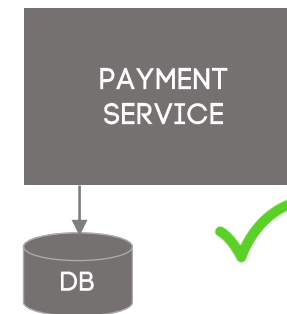
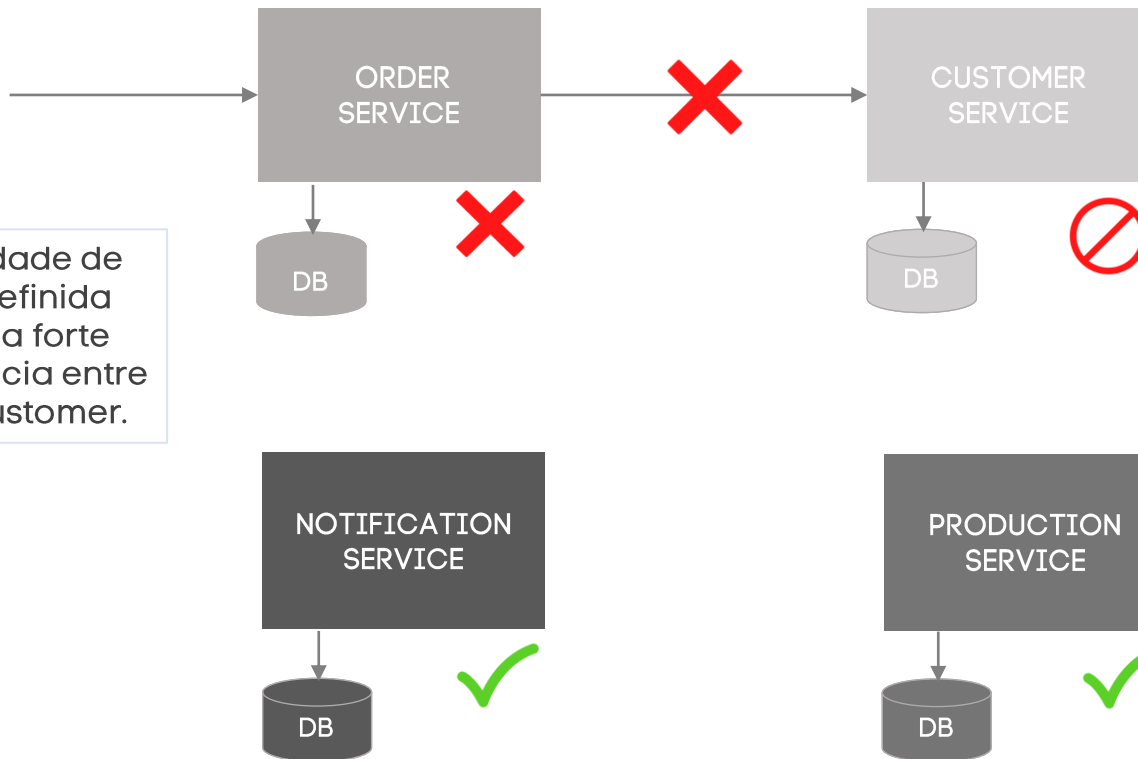
Michelli Brito
@brito_michelli

Acoplamento em Microservices

Não existe desacoplamento absoluto entre Microservices.

O forte acoplamento nem sempre é um problema de modelagem arquitetural, mas sim pode ser uma necessidade do próprio negócio.

A necessidade de negócio definida obriga uma forte dependência entre Order e Customer.



O entendimento do negócio é essencial na definição da arquitetura.

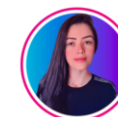
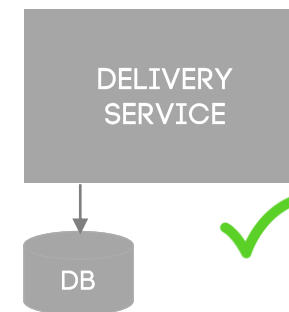
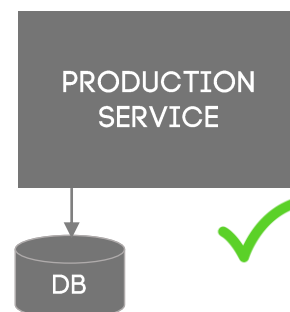
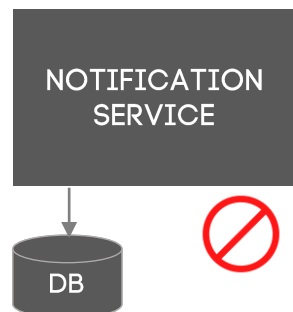
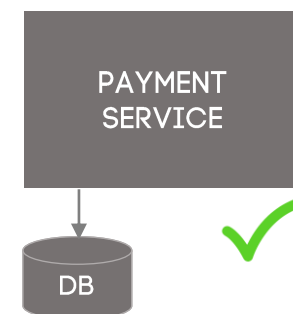
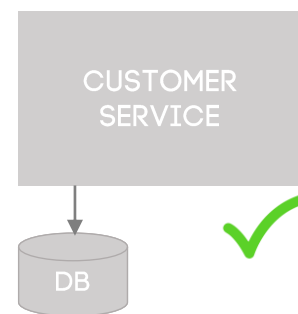
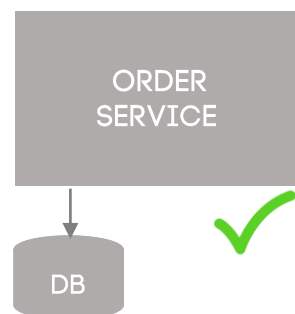


Michelli Brito
@brito_michelli

Disponibilidade em Microservices

Não criamos Microservices para que qualquer um possa parar em algum momento sem afetar os demais, mas sim para que alguns possam parar eventualmente e o sistema continuar disponível. E isso já é muito melhor do que se nenhum pudesse parar.

A busca é sempre pela maior disponibilidade, mesmo não existindo na prática uma disponibilidade de 100%

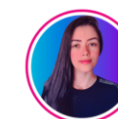
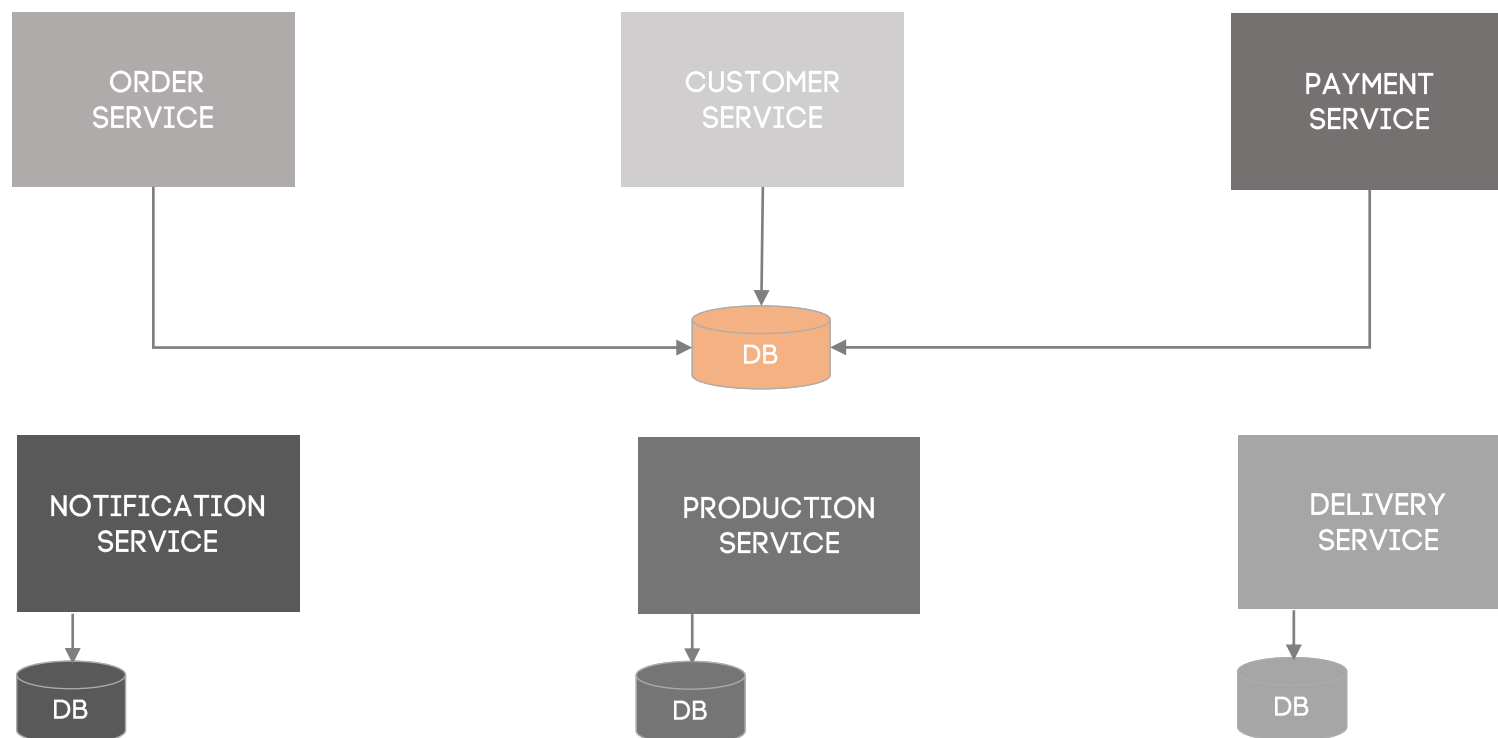


Michelli Brito
@brito_michelli

Distribuição das Bases de Dados em Microservices

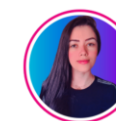
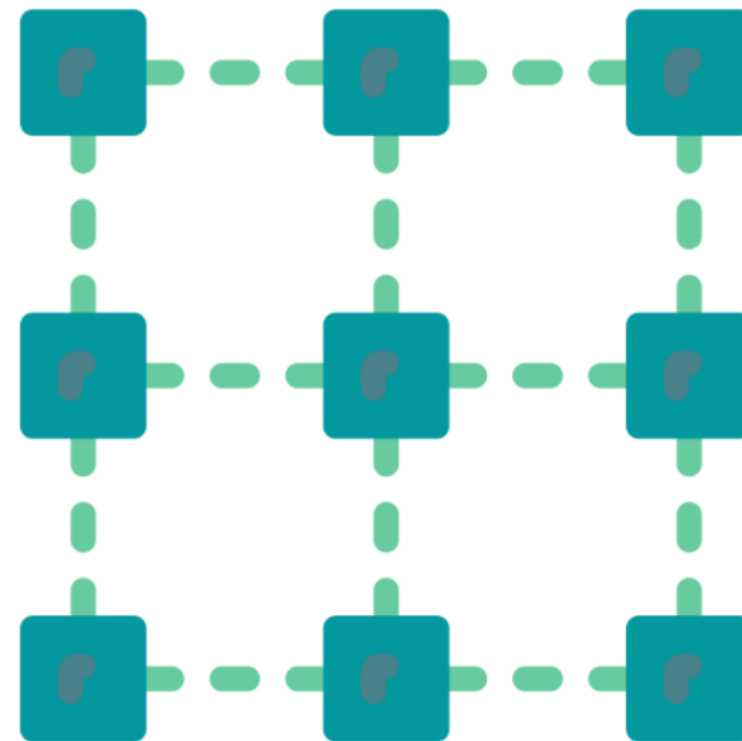
Sim! Ainda temos uma Arquitetura de Microservices com base de dados compartilhada.

Na migração de Monolítico para Microservices o uso de base de dados compartilhada é comum no início.



Concepção e Utilização da Arquitetura de Microservices

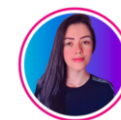
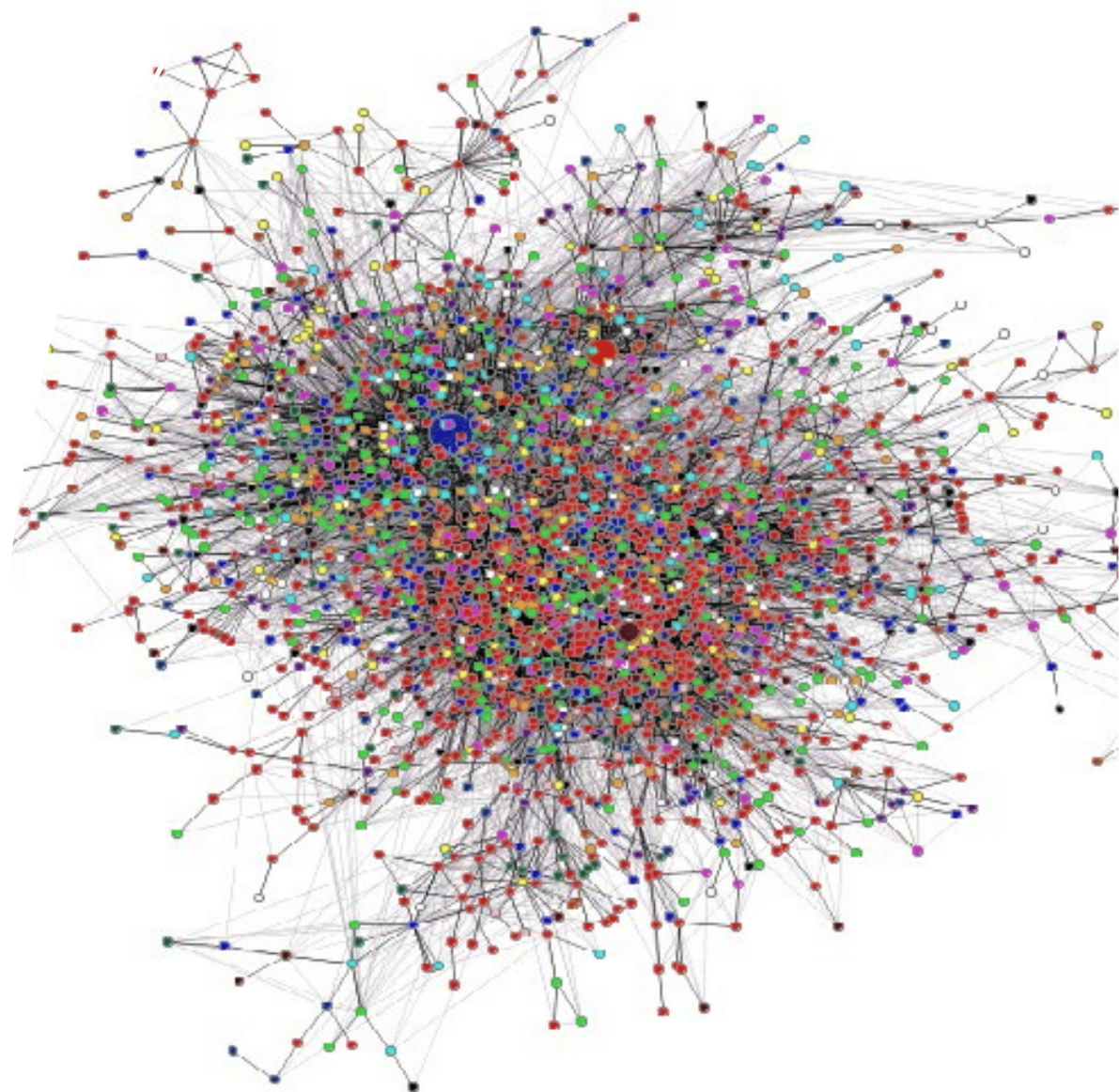
Arquitetura de Microservices são conceitos e princípios que devem ser sempre adaptados ao negócio, não é um modelo e nem um padrão pronto e replicável.



Padrões de Microservices

O desconhecimento sobre os principais Microservices Patterns é um grande risco na elaboração de Arquiteturas de Microservices.

Base de dados por Microservice diminui o acoplamento, mas é necessário saber lidar corretamente com a sincronia de dados e com consistência eventual por os dados estarem distribuídos, por exemplo.



Michelli Brito
@brito_michelli

Premissas Importantes

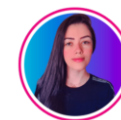
O entendimento do negócio juntamente com a experiência e conhecimentos adquiridos, como o de Microservices Patterns, são essenciais para você fazer as melhores escolhas e definir as melhores soluções e assim ser um profissional diferenciado no mercado.

A discussão de Microservices não deve ser apenas sobre tamanho ou complexidade do negócio, mas sim sobre projetar sistemas que respondam facilmente a mudanças e modernizações tecnológicas.



Obrigada!

Não perca o próximo dia: **Microservices Patterns e Spring Projects**



Michelli Brito
@brito_michelli