# Como DDD e principalmente Domain Model

contribuem na construção de microservices.

**Isaac Felisberto de Souza**
Engenheiro de Software

# JOGO RÁPIDO!

**RESPONDA APENAS**

**SE FOR**

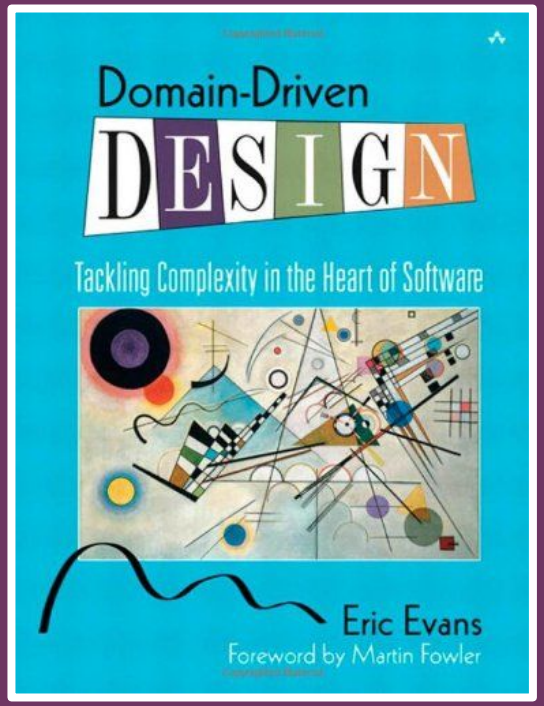
**SIM**

- ➔ Quem conhece DDD?
- ➔ Quem utiliza microservices?
- ➔ Quem aplica DDD em microservices?

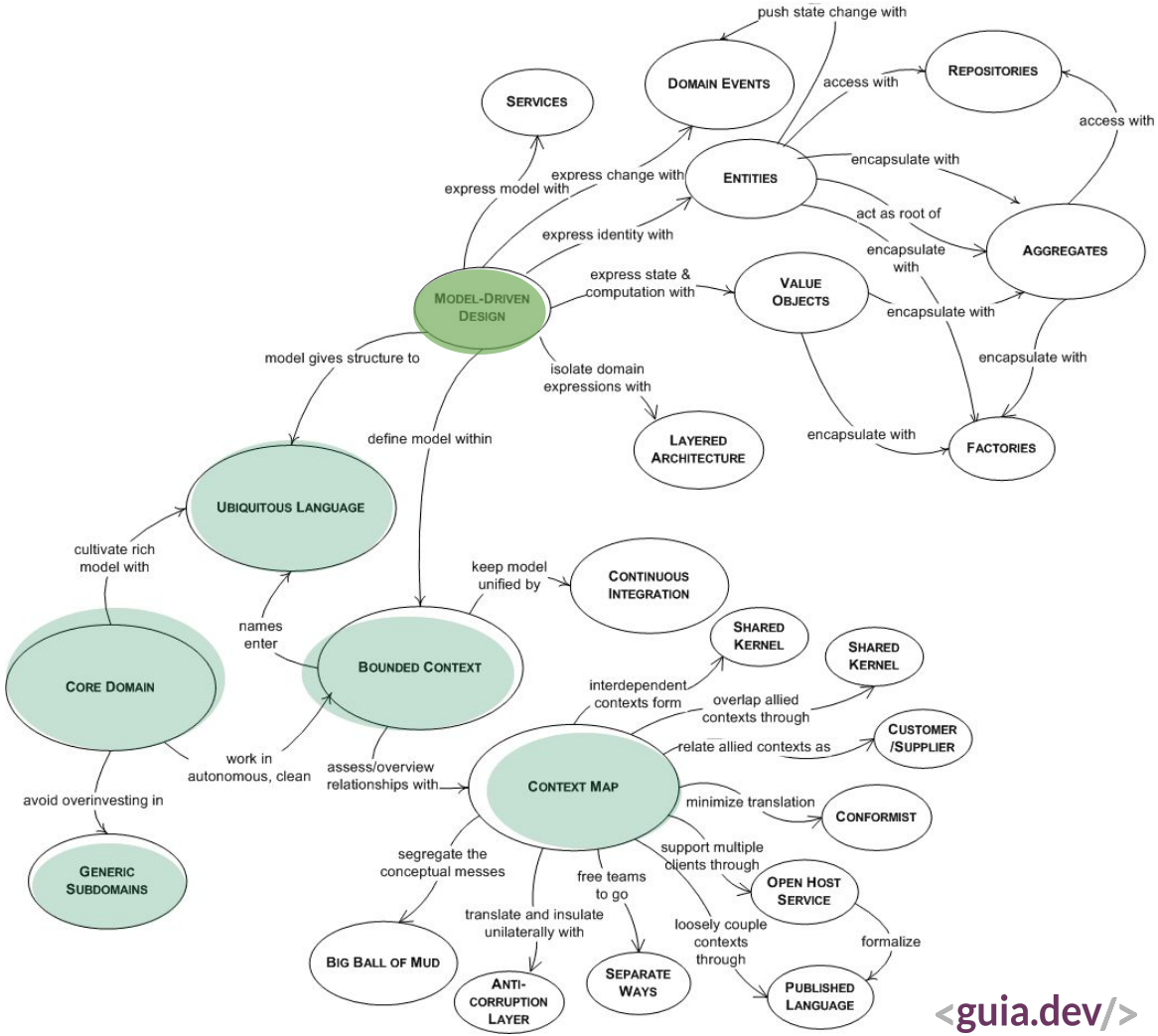
ROTEIRO
DE HOJE!

1. VISÃO GERAL SOBRE DDD.
2. FALHAS COMUNS NO USO DE DDD.
3. APLICANDO DDD EM MICROSERVICES!

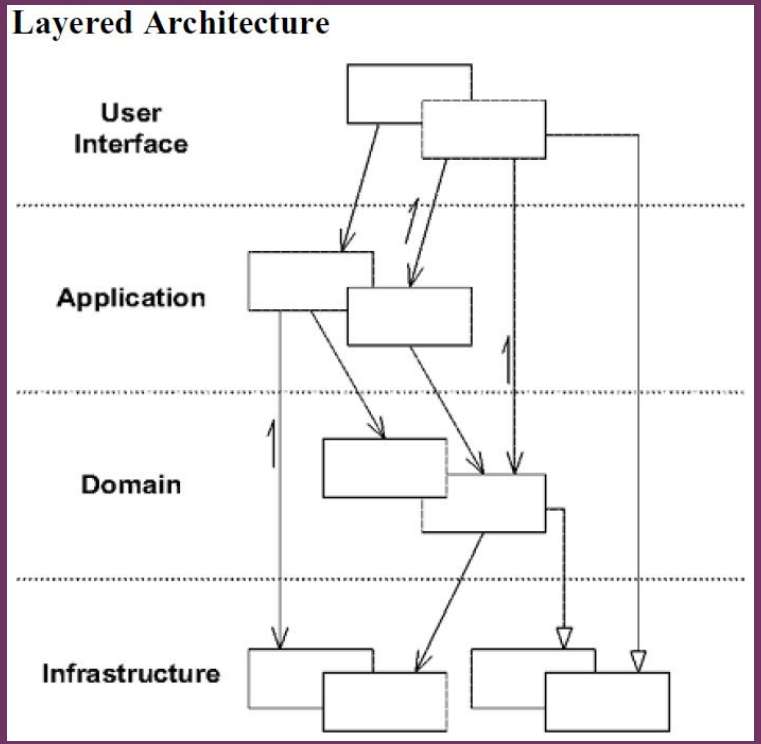
# VISÃO GERAL SOBRE DDD




- Eric Evans
- Publicado em 2004.
- Possui 450+ páginas.



# Organização por Camadas




- **User Interface:** Apresentar e interpretar comandos do usuário ou outro sistema.
- **Application:** Funções que são executadas pelo Software.
- **Domain:** Regras de negócio.
- **Infrastructure**: Recursos técnicos genéricos

Domain Model

Ubiquitous Language

Bounded Context

Polissemia

Designed by pch.vector / Freepik

Domain Model

Ubiquitous Language

Bounded Context

Polissemia

Eric Evans diz que:

> “O coração do software está na sua capacidade de resolver problemas relacionados ao domínio...”

> “... É preciso afiar sua capacidade de modelagem e dominar o design de domínios.”

➔ **Qual o domínio do negócio?**

➔ **Quais contextos e entidades?**

(entidades não são as tabelas de um banco de dados)

Domain Model

Ubiquitous Language

Bounded Context

Polissemia

## O uso de uma linguagem universal.

Quem?

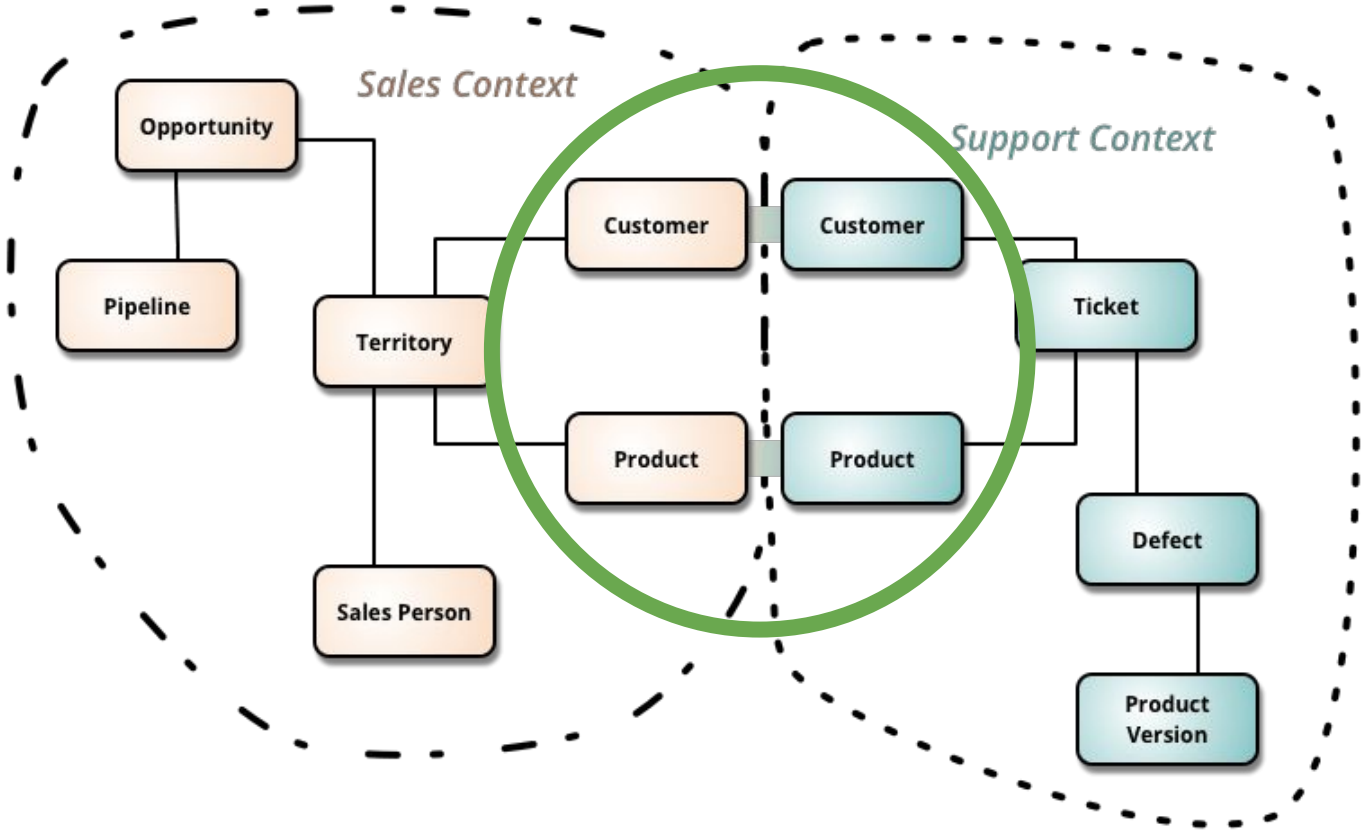
- ➔ Profissionais na gestão e negócio.
- ➔ Designers.
- ➔ Desenvolvedores(as).
- ➔ Times de Operação.
- ➔ Time de Suporte.

Domain Model

Ubiquitous Language

**Bounded Context**

Polissemia




martinfowler.com

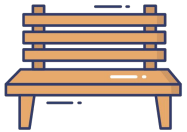
Domain Model

Ubiquitous Language

Bounded Context

Polissemia

Qual o significado da palavra **Banco**?

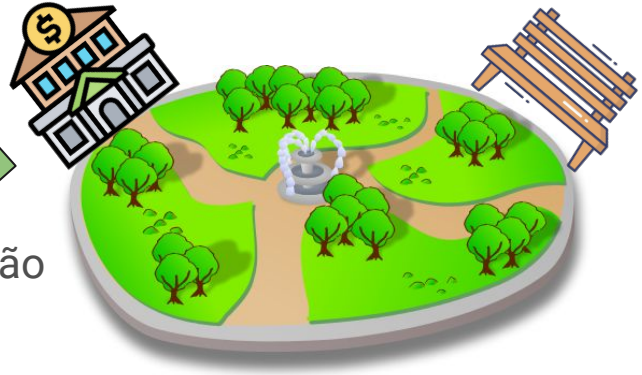


A porta do banco travou.

Chegando na praça, sente no banco e aguarde

Estou esperando em frente ao banco da praça.

Agência ou Instituição Bancária

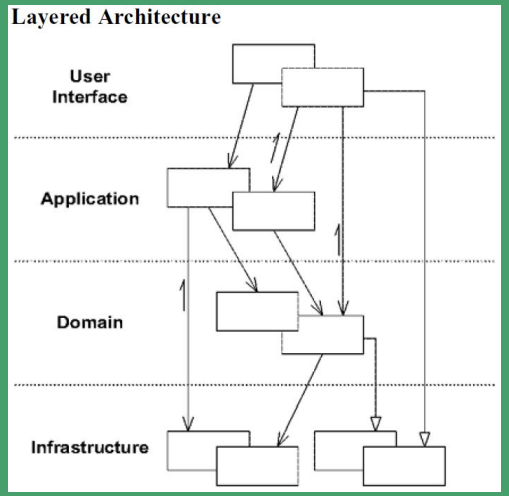


# Domain Model

Ubiquitous Language
+
Bounded Context
+
Polissemia

# FALHAS COMUNS NO USO DE DDD

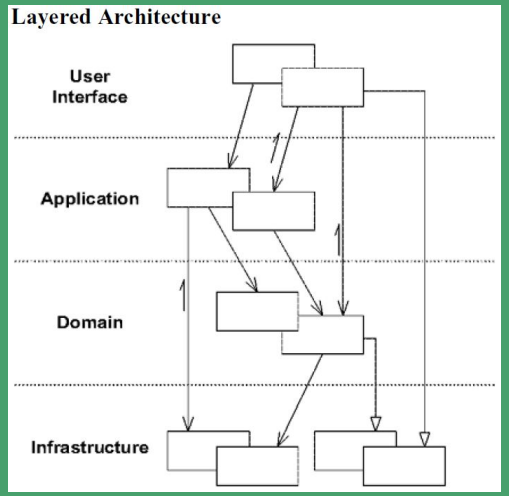
# EM MONÓLITOS




**Normalmente focam apenas nas camadas!**


- ➔ Regra de negócio não está apenas em Domain.
- ➔ Domain não organizados por contextos.
- ➔ Pouca clareza sobre conceitos do domínio.
- ➔ Múltiplos termos/palavras para coisas iguais.

# EM MICROSERVICES




**Microservices abandonam a organização de camadas.**

- ➔ Não há visão de contextos.
- ➔ Cada microservice utiliza suas nomenclaturas.
- ➔ Não há clareza sobre dependências.
- ➔ Frontend acessa direto todos microservices.
- ➔ Inconsistências nos Contratos de API’s.

# APLICANDO DDD EM MICROSERVICES

Domain Model

Ubiquitous Language

Bounded Context

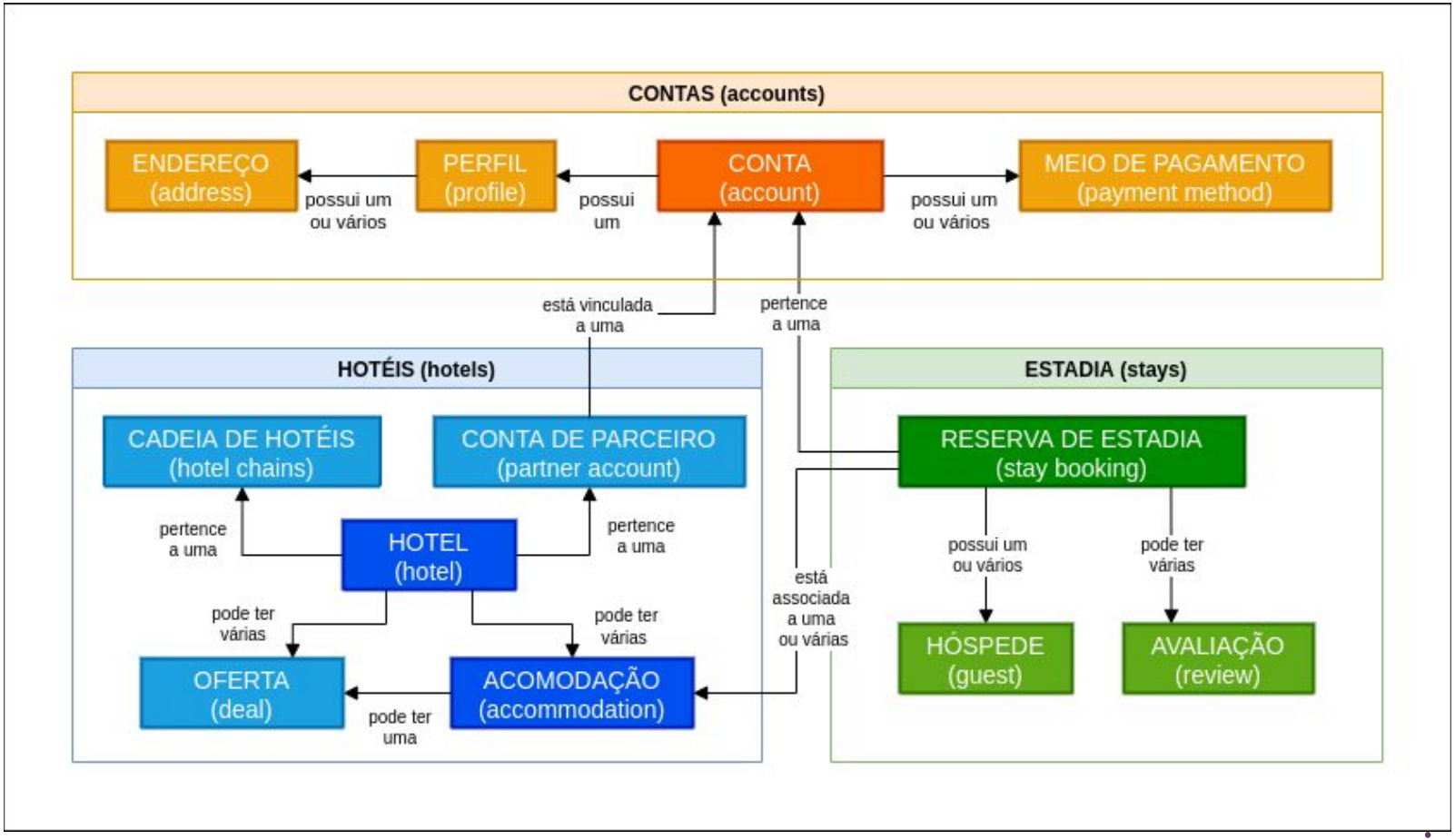
Polissemia

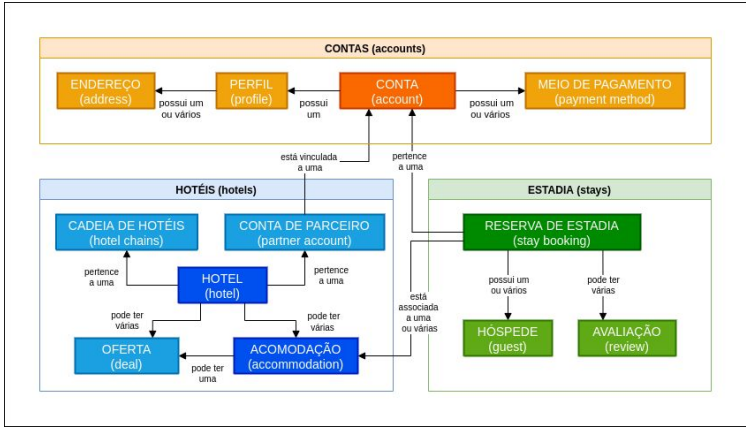
## A solução deve possuir um modelo de domínio!

> “O coração do software...”

Uma solução fictícia, chamada Guia Hóspede, o domínio é: RESERVA DE HOTÉIS.

# Modelo de domínio do Guia Hóspede

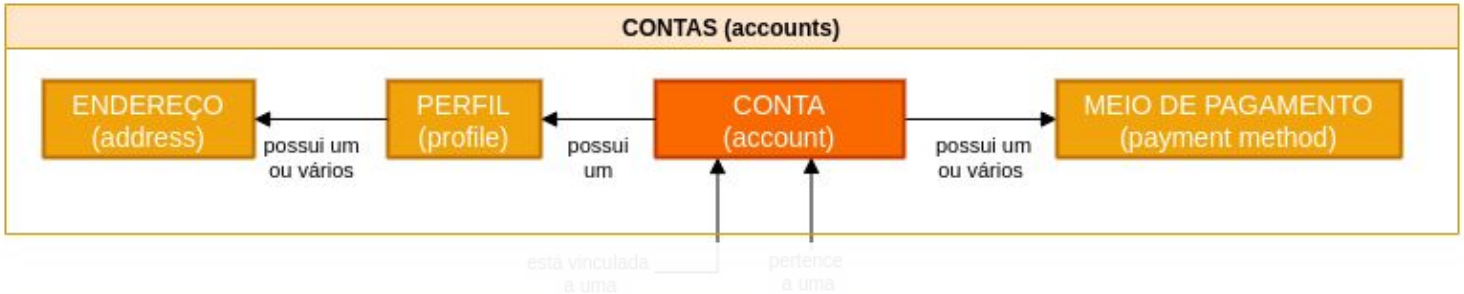





- ➔ As entidades determinam os contratos das API's.
- ➔ O modelo indica as dependências.
- ➔ Contextos indicam a fragmentação inicial de microservices.

## QUAL O TAMANHO

## DE UM MICROSERVICE?

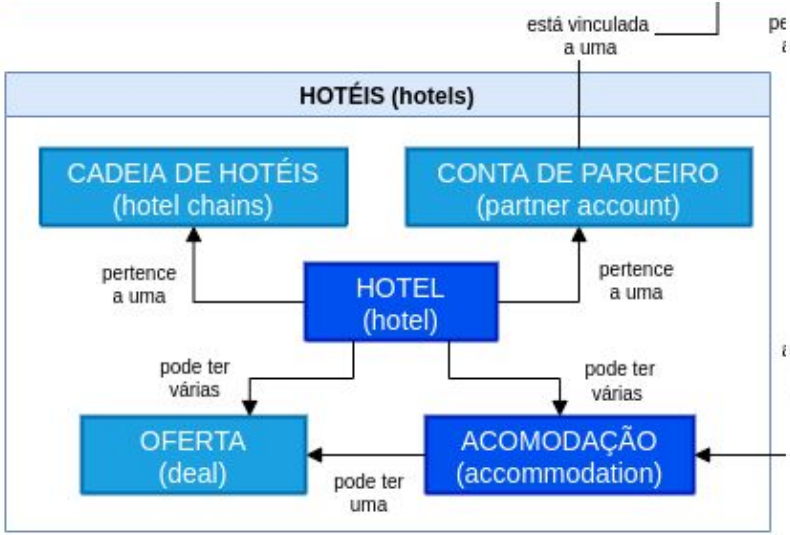



➔ Entidades fracas não precisam ser isoladas em um microservice.

⚠️ Cuidado com excesso de fragmentação. ⚠️

➔ Um microservice para esse contexto é aceitável.

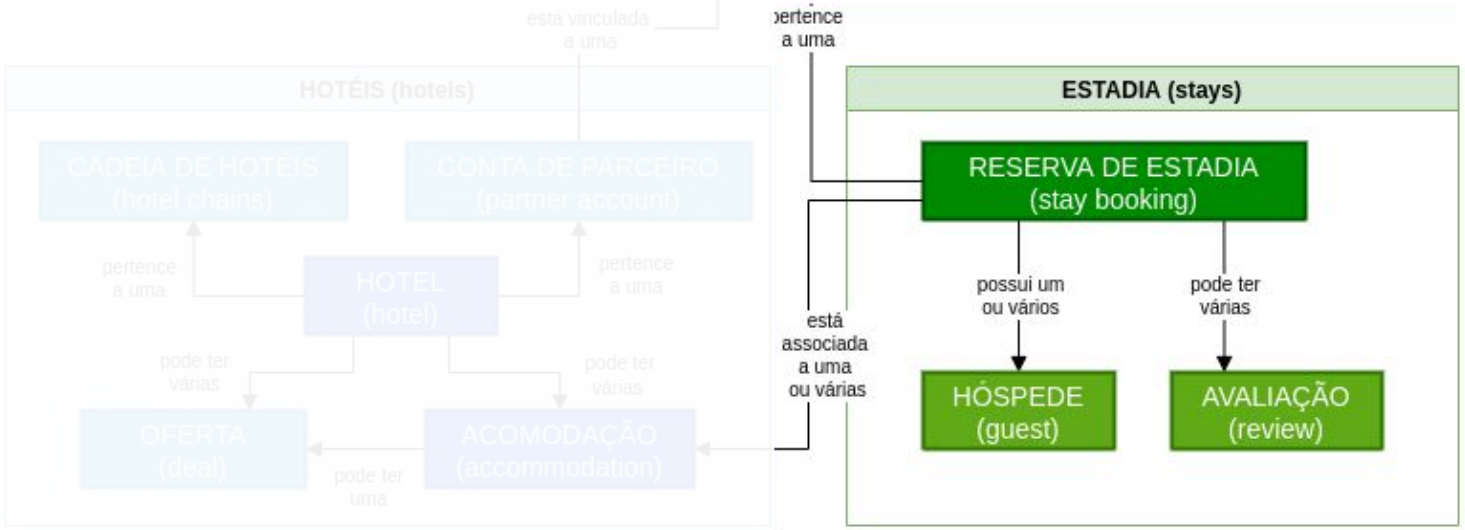
- ➔ Hotel é a entidade forte, terá um microservice.
- ➔ Acomodação também é uma entidade forte.

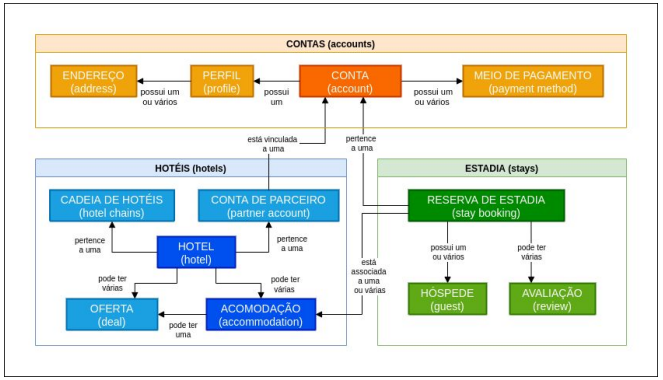


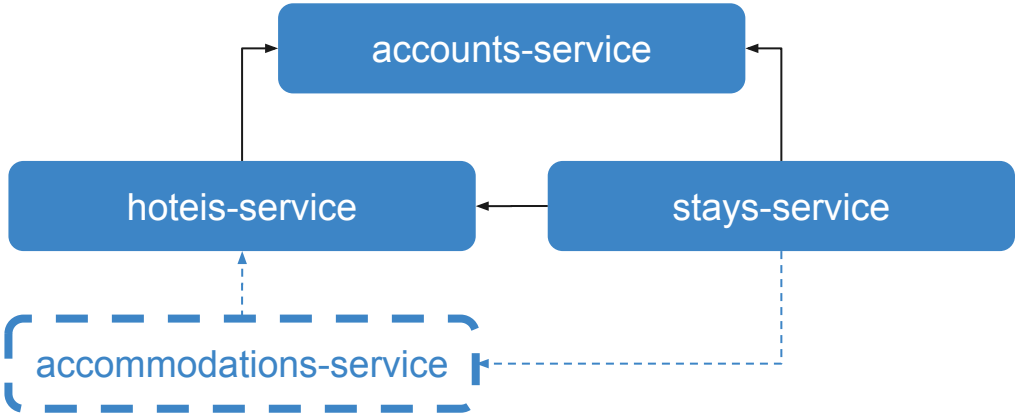

Acomodação...

- ➔ Inicia no microservice de hotéis.
- ➔ Quando precisar de escala, será segregada.

- ➔ Hóspede parece uma entidade forte, mas...
- ➔ O contexto foca na estadia, um microservice para o contexto é aceitável.






- ➔ Mais importante que o tamanho do microservice, é o contexto.
- ➔ As dependências entre microservices seguem o modelo de domínio.
- ➔ O modelo ajuda a compreender possíveis evoluções.
- ➔ Os microservices explicitam melhor o domínio de negócio.

Domain Model

Ubiquitous Language

Bounded Context

Polissemia

## Todos utilizam uma linguagem universal!

O nome de contextos, entidades e atributos é utilizada em:

- ➔ Nome dos microservices e Contrato das API's.
- ➔ Modelagem da persistência de dados.
- ➔ Artefatos distribuídos.
- ➔ Endereço dos Endpoints.
- ➔ Mapeamentos de monitoramento.
- ➔ No diálogo entre integrantes do projeto/empresa.

Domain Model

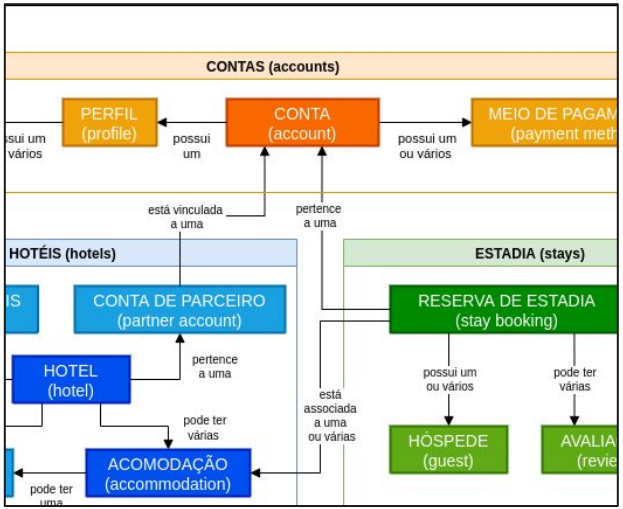
Ubiquitous Language

Bounded Context

Polissemia

# Microservices respeitam os limites de cada contexto!

- ➔ Informação adequada ao o contexto.
- ➔ Dependência fraca entre entidades.




- ➔ Conta é uma entidade genérica. (Hotéis e usuários).
- ➔ Dados específicos para conta de hotéis no contexto de hotéis.
- ➔ A chave entre entidades é um UUID, ou “strings humanizadas”.

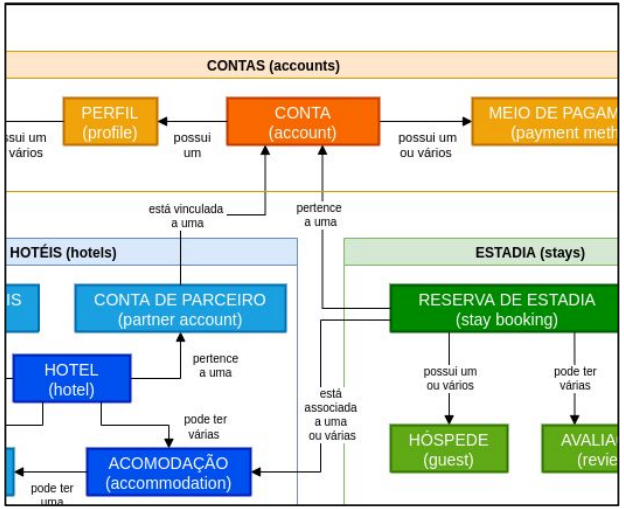
Domain Model

Ubiquitous Language

Bounded Context

Polissemia

# Nomenclaturas não geram dúvidas!

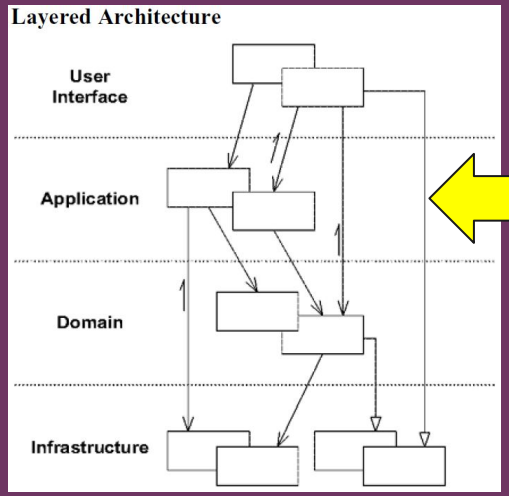



➔ Conta, é a representação genérica.

➔ Conta de Parceiros, dados adicionais de contas de hotéis.

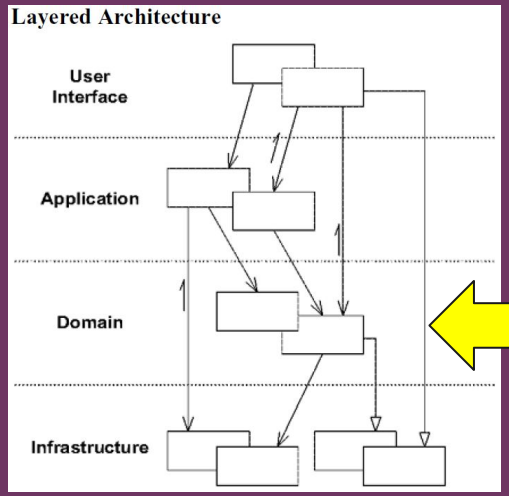
➔ Reserva, é Reserva de Estadias.

# CAMADAS DO DDD NO CONJUNTO DE MICROSERVICES




**Representação da Application Layer**

- ➔ Adoção de API Gateway.
- ➔ Frontend ou sistemas externos acessam o API Gateway.
- ➔ API’s mapeadas no gateway respeitam o modelo de domínio.
- ➔ **O API Gateway -> Application Layer**




**Os microservices representam a Domain Layer**

- ➔ Agrupados por domínio/contexto.
- ➔ Uso de um Core Domain.

# Arquitetura Orientada a Eventos

possibilita uso de DOMAIN EVENTS

- ➔ Schemas seguem o modelo de domínio.
- ➔ Nomenclatura de filas seguem as entidades.
- ➔ Agrupamentos lógicos seguem os contextos.

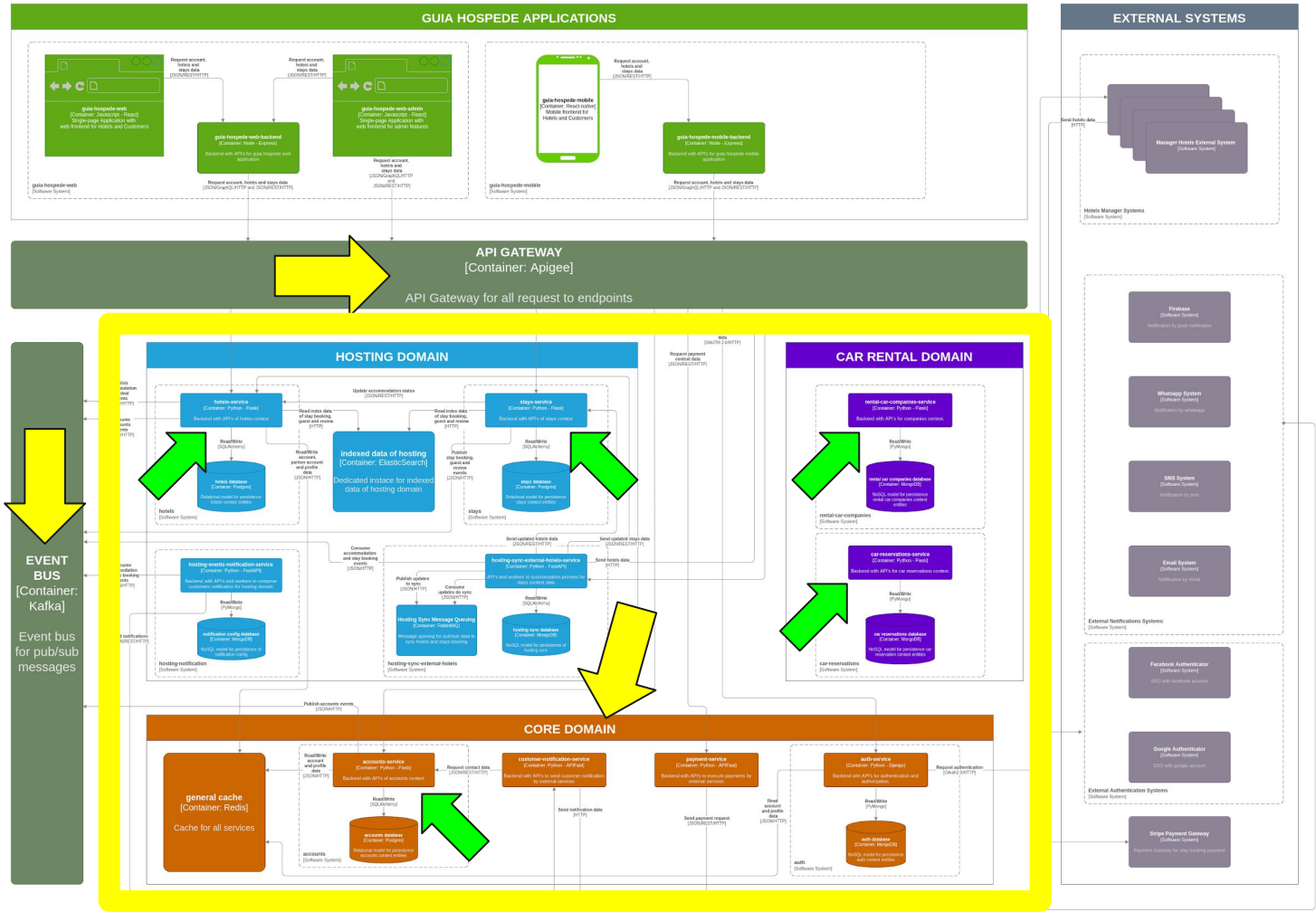
Jan/24/2021

# Guia Hóspede - Diagram with general solution architecture

Vision of products, services and relationships

PRODUCT LAYER

SERVICE LAYER



Domain Model
Ubiquitous Language
Bounded Context
Polissemia

Organizados com:
**Application**
**Domains**

MICROSERVICES ORGANIZADOS,
COM MENOR ACOPLAMENTO, MAIOR COESÃO
E FOCADOS NO NEGÓCIO.

# OBRIGADO!

**Isaac Felisberto de Souza**
Engenheiro de Software

**isaacsouza@gmail.com**

**linkedin.com/in/isaacfsouza**

# <guia.dev/>

**Um guia para auxiliar o Desenvolvimento de Software.**

Através de conteúdo para desenvolvedores(as).

www.guia.dev

Guia Dev (linkedin.com/company/guiadev)

@guia_dev

**Siga! e compartilhe ;-)**