



Instituto Infnet

# Aula 3 - Arquitetura de Microsserviços e Mobile

## Aula 3

- Cenários Projeto Final;
- Modelo C4;
- Como projetar microsserviços;
- Como dividir a aplicação monolítica.

## Ementa proposta Aula 3

- Como dividir a aplicação monolítica;
- ~~Estilos de comunicação entre microsserviços;~~
- ~~Como implementar comunicação entre microsserviços;~~
- ~~Detalhes de workflow com microsserviços";~~

# Projeto final

Regras:

- O projeto é individual;
- O aluno deverá escolher, no mínimo, um cenário dos que serão apresentados a seguir;
- Assim que terminar, salve o seu arquivo PDF e poste no Moodle;
- Utilize o seu nome para nomear o arquivo, identificando também a disciplina no seguinte formato: "nomedoaluno\_nomedadisciplina\_pd.PDF".




# Projeto final

Cenário 1:

Cenário de Startup (investimentos limitados) de Marketplace Regional

Um marketplace de produtos artesanais e regionais do nordeste brasileiro busca uma solução tecnológica que:

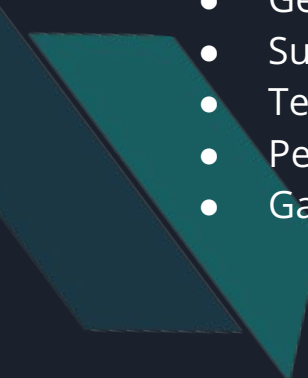
- Suporte o crescimento rápido de vendedores locais;
  - Permita integração com sistemas de pagamento locais e microempreendedores;
  - Tenha alta disponibilidade durante eventos sazonais (São João, Carnaval);
  - Possibilite escalonamento horizontal com baixo custo inicial.
- 

# Projeto final

Cenário 2:

Cenário de E-commerce Multinacional (sem problemas em investir) com Complexidade Regulatória

Uma empresa de moda com operações em Brasil, Argentina e Colômbia necessita de uma arquitetura que:

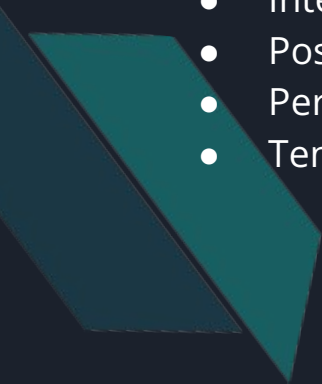
- Gerencie diferentes regras fiscais e tributárias por país;
  - Suporte múltiplas moedas e métodos de pagamento locais;
  - Tenha mecanismos robustos de segurança para proteção de dados pessoais;
  - Permita personalização de catálogo por região;
  - Garanta consistência de dados entre diferentes localizações.
- 

# Projeto final

Cenário 3:

Cenário de E-commerce para Produtos Tecnológicos com Alta Variabilidade

Uma empresa especializada em venda de componentes e equipamentos de tecnologia precisa de uma solução que:

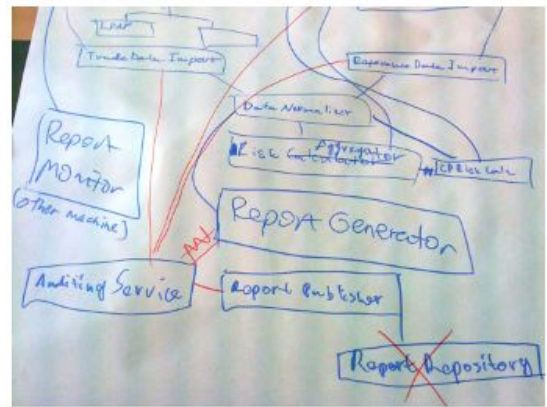
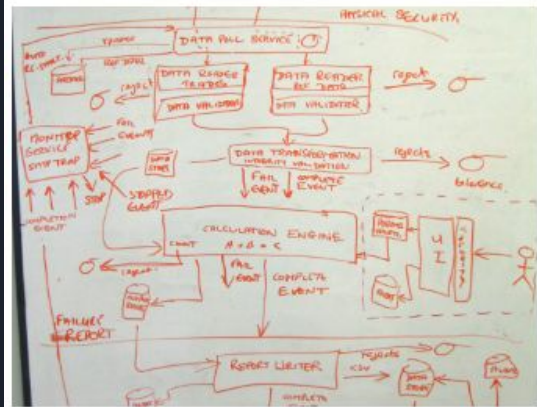
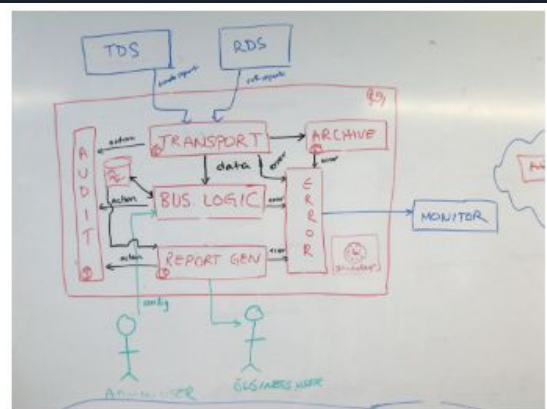
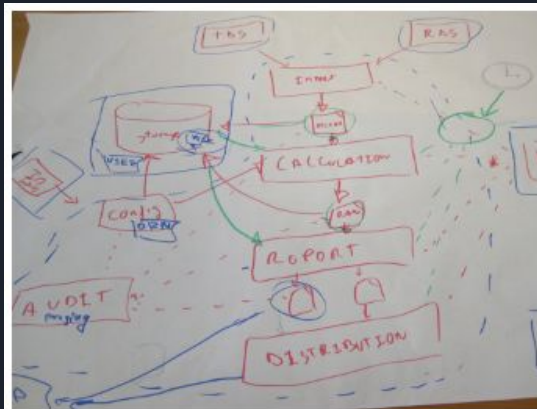
- Suporte catálogo dinâmico com produtos de alta rotatividade
  - Integre em tempo real com fornecedores para atualização de estoque
  - Possua sistema de recomendação personalizada
  - Permita cotações e comparações complexas entre produtos
  - Tenha mecanismo de precificação dinâmica baseado em disponibilidade e demanda
- 

# Modelo C4

Objetivo:

- Ajudar equipes de desenvolvimento de software a descrever e comunicar a arquitetura de software;

<https://c4model.com/>.



# Modelo C4

Objetivo:

- Ajudar equipes de desenvolvimento de software a descrever e comunicar a arquitetura de software;
- Criar “mapas(*Visuais*) do seu código”, em vários níveis de detalhes;
- Elevar o nível de maturidade associado aos diagramas de arquitetura de software;
- Ser simples.

<https://c4model.com/>.

Decorative teal geometric shapes, including a parallelogram and a triangle, located in the bottom-left corner of the slide.



# Modelo C4

Ferramentas de desenho:

- Drawio - Desenhos com mouse;
- PlantUML - Desenhos via código;

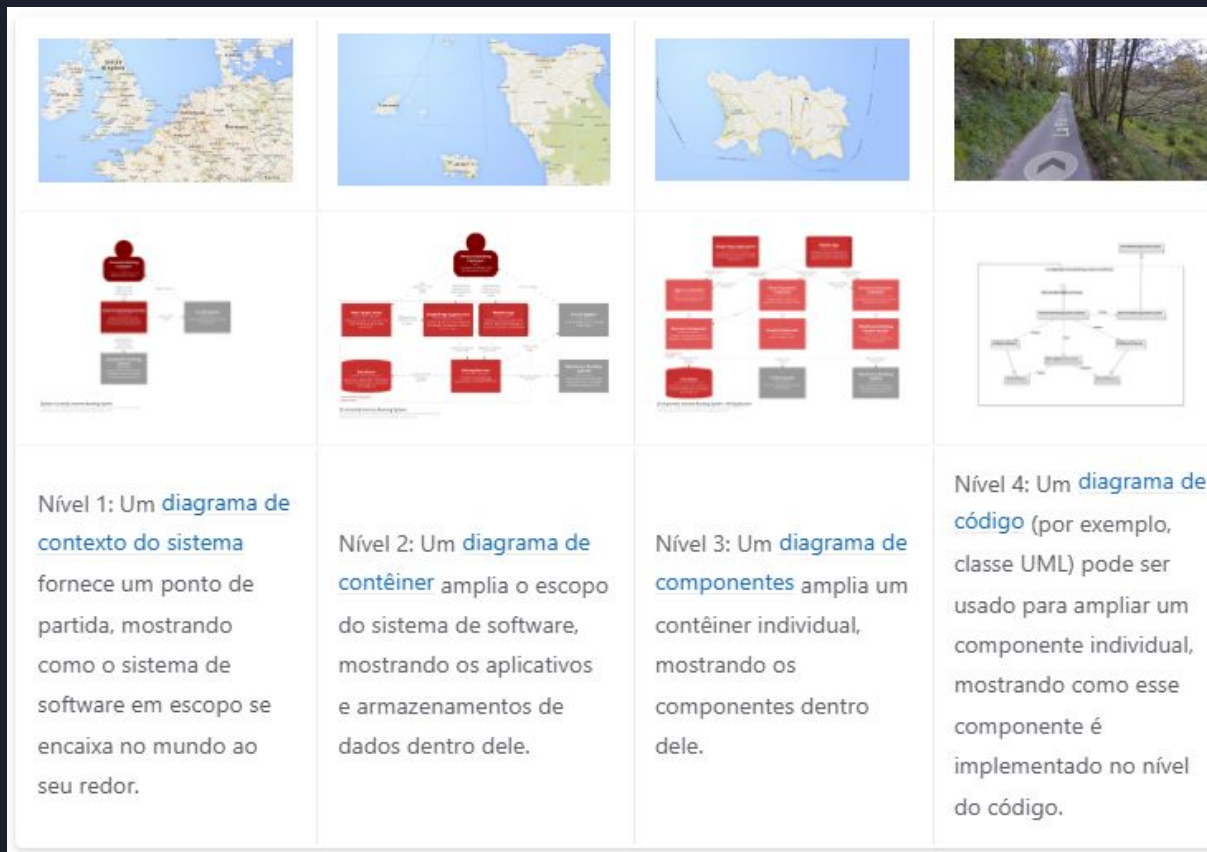


# Modelo C4

Objetivo:

- Criar “mapas(Visuais) do seu código”, em vários níveis de detalhes;

<https://c4model.com/>.



# Modelo C4

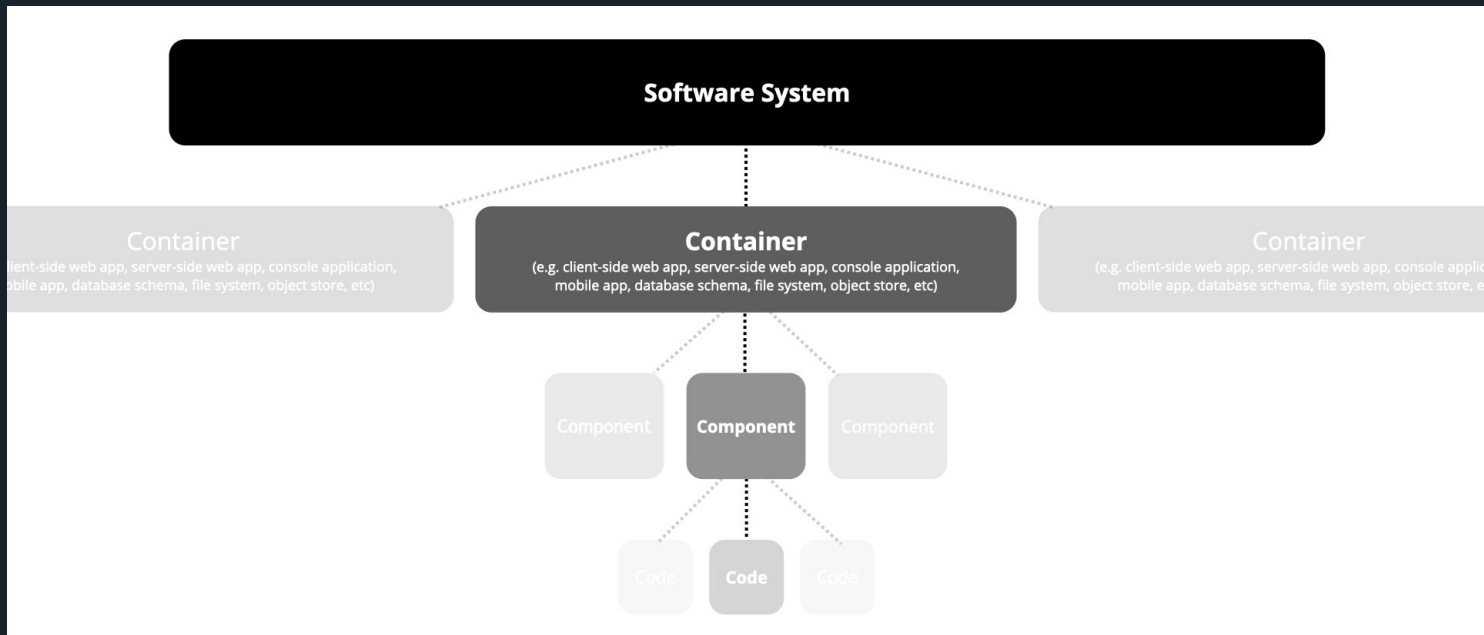
Objetivo:

- Elevar o nível de maturidade associado aos diagramas de arquitetura de software.



# Modelo C4

Abstrações:



“Um **SISTEMA** de software é composto de um ou mais **CONTÊNER**(aplicativos e armazenamentos de dados), cada um dos quais contém um ou mais **COMPONENTES**, que por sua vez são implementados por um ou mais elementos de **CÓDIGO**(classes, interfaces, objetos, funções, etc.). E as pessoas (atores, papéis, personas, indivíduos nomeados, etc.) usam os sistemas de software que construímos.”

<https://c4model.com/>.

# Modelo C4

Diagramas:

- Contexto do Sistema;
- Contêiner;
- Componente;
- Código.



# Modelo C4

Diagrama de Contexto do Sistema:

“Um sistema de software é o mais alto nível de abstração e descreve algo que entrega valor aos seus usuários, sejam eles humanos ou não. Isso inclui o sistema de software que você está modelando e os outros sistemas de software dos quais seu sistema de software depende (ou vice-versa)”

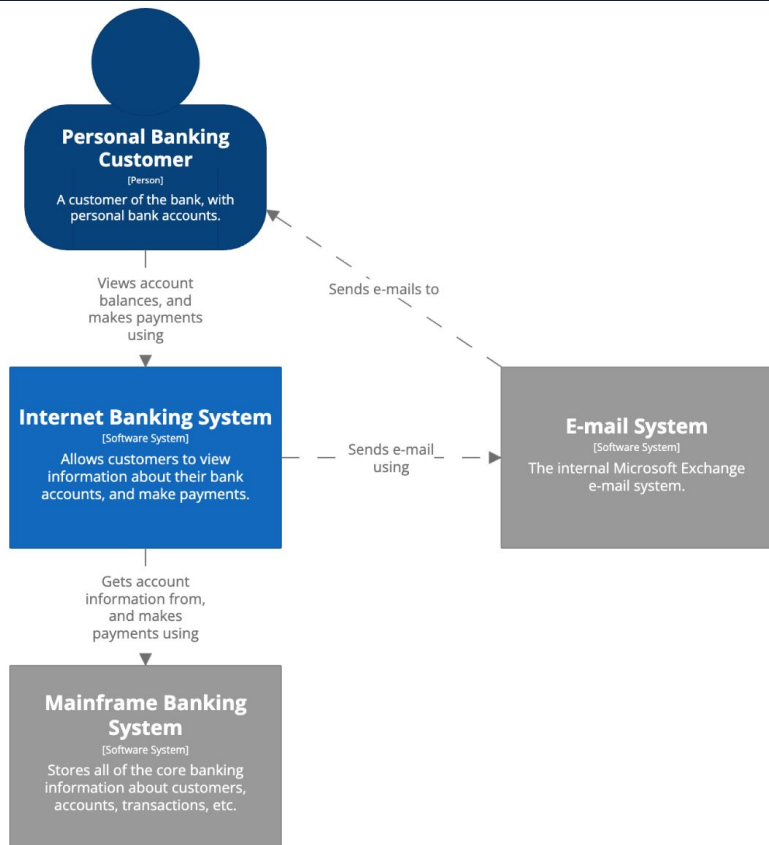
Ex.: “aplicativo”, “produto”, “serviço”, etc.



# Modelo C4

Diagrama de Contexto do Sistema.

Ex.:



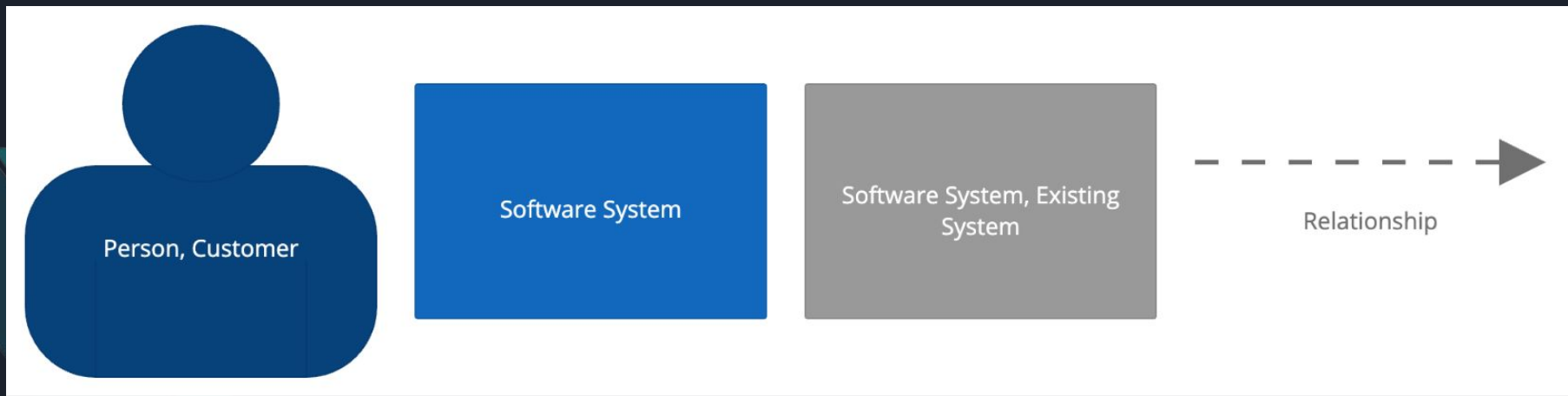
## [System Context] Internet Banking System

The system context diagram for the Internet Banking System - diagram created with Structurizr.  
Saturday, 11 November 2023 at 09:04 Greenwich Mean Time

# Modelo C4

## Diagrama de Contexto do Sistema

Elementos do diagrama:





# Modelo C4


## Diagrama de Contexto do Sistema

- Público-alvo:

Todos, tanto técnicos quanto não técnicos, dentro e fora da equipe de desenvolvimento de software.

- Recomendado?

Sim, um diagrama de contexto do sistema é recomendado para todas as equipes de desenvolvimento de software.



# Modelo C4

## Diagrama de Contêineres

“Não é Docker! No modelo C4, um contêiner representa um aplicativo ou um armazenamento de dados. Um contêiner é algo que precisa estar em execução para que o sistema de software geral funcione. ”

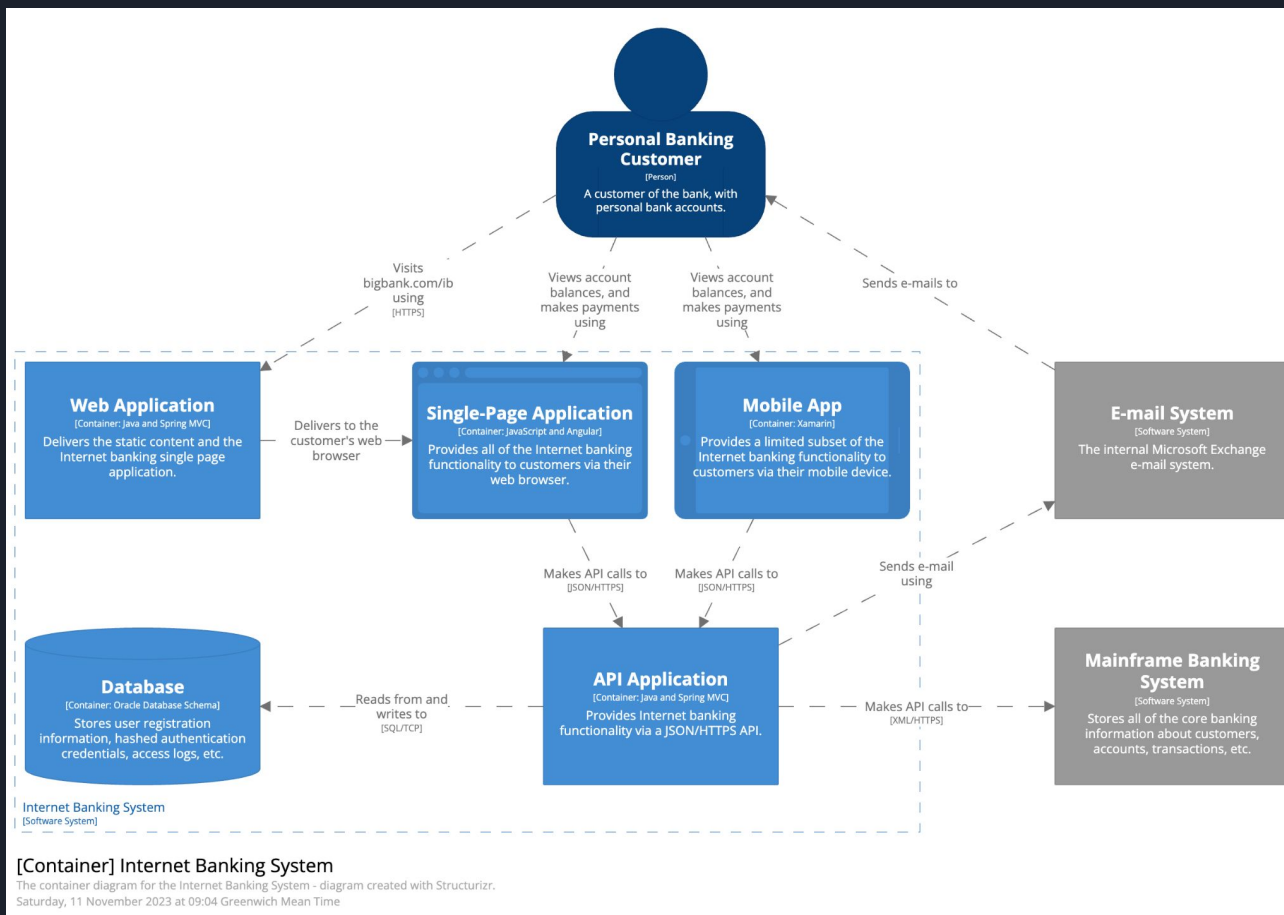
Ex.: Aplicação web do lado do servidor, Aplicação desktop do lado do cliente, Aplicativo móvel, Aplicações serverless, etc...



# Modelo C4

## Diagrama de Contêineres

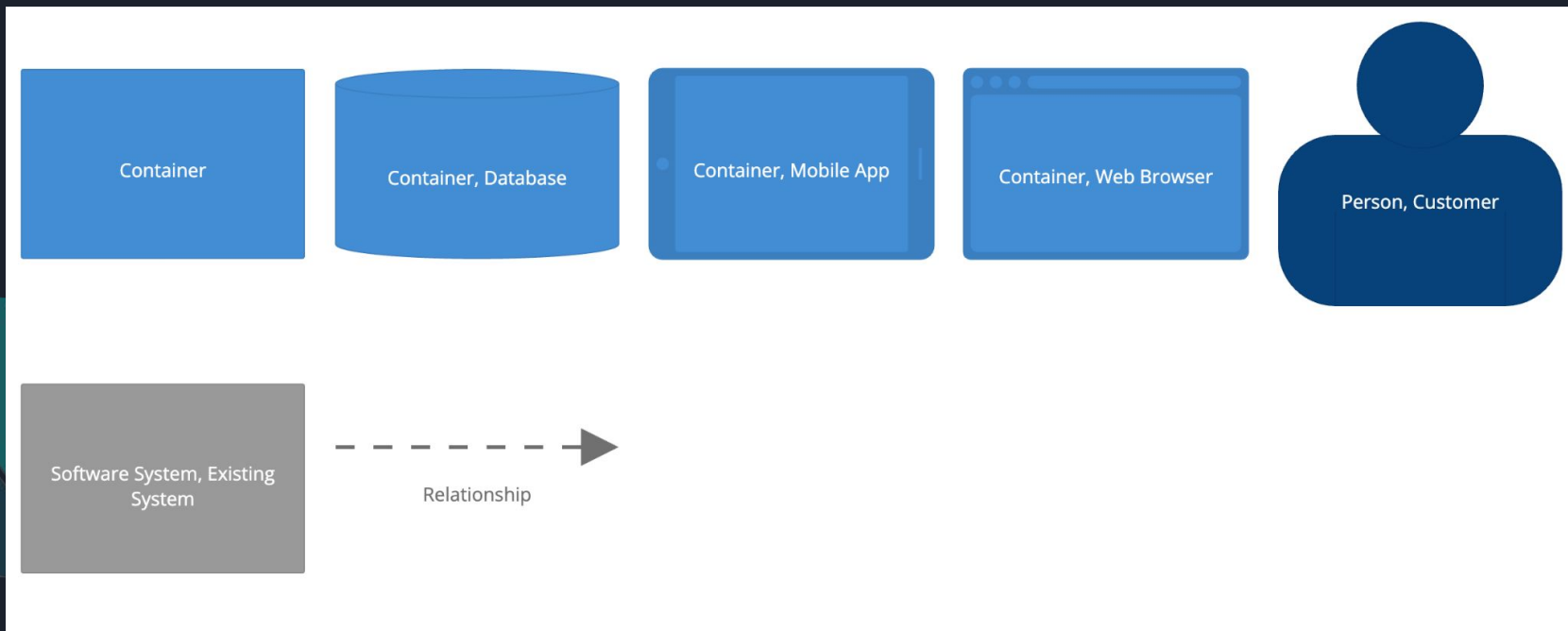
Ex.:



# Modelo C4

## Diagrama de Contêineres

Elementos do diagrama:



# Modelo C4

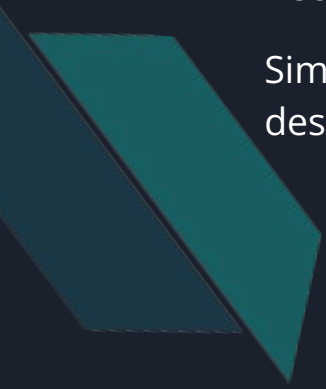
## Diagrama de Contêineres

- Público-alvo

Pessoas técnicas dentro e fora da equipe de desenvolvimento de software; incluindo arquitetos de software, desenvolvedores e equipe de operações/suporte.

- Recomendado?

Sim, um diagrama de contêiner é recomendado para todas as equipes de desenvolvimento de software.



# Modelo C4

## Diagrama de Componente

“...no modelo C4, um componente é um agrupamento de funcionalidades relacionadas encapsuladas por trás de uma interface bem definida. Se você estiver usando uma linguagem como Java ou C#, a maneira mais simples de pensar em um componente é que ele é uma coleção de classes de implementação por trás de uma interface.”

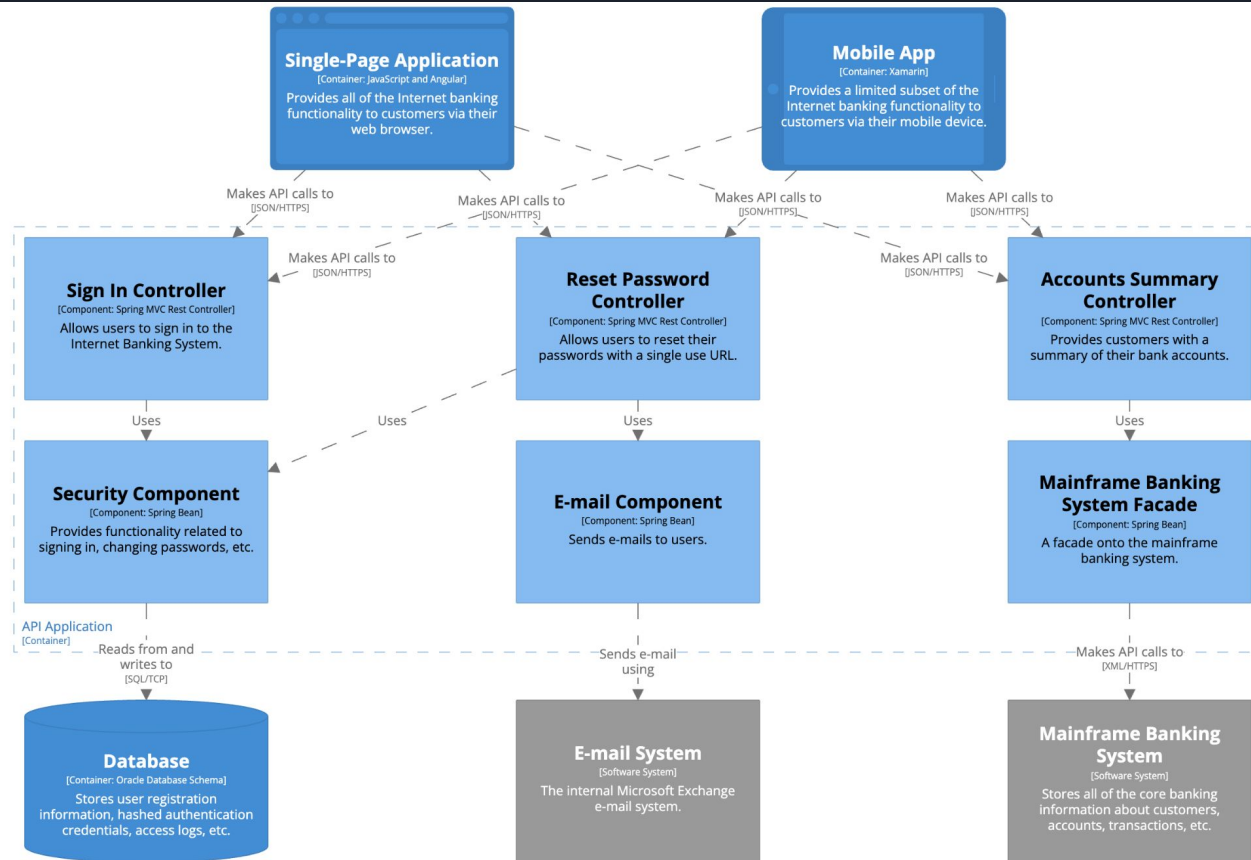
Ex.: Em OO: composto de classes e interfaces, Em procedimental: ,componente pode ser composto de vários arquivos C em um diretório específico, Js módulo JavaScript, que é composto de vários objetos e funções, etc...



# Modelo C4

Ex.

Diagrama de componentes



[Component] Internet Banking System - API Application

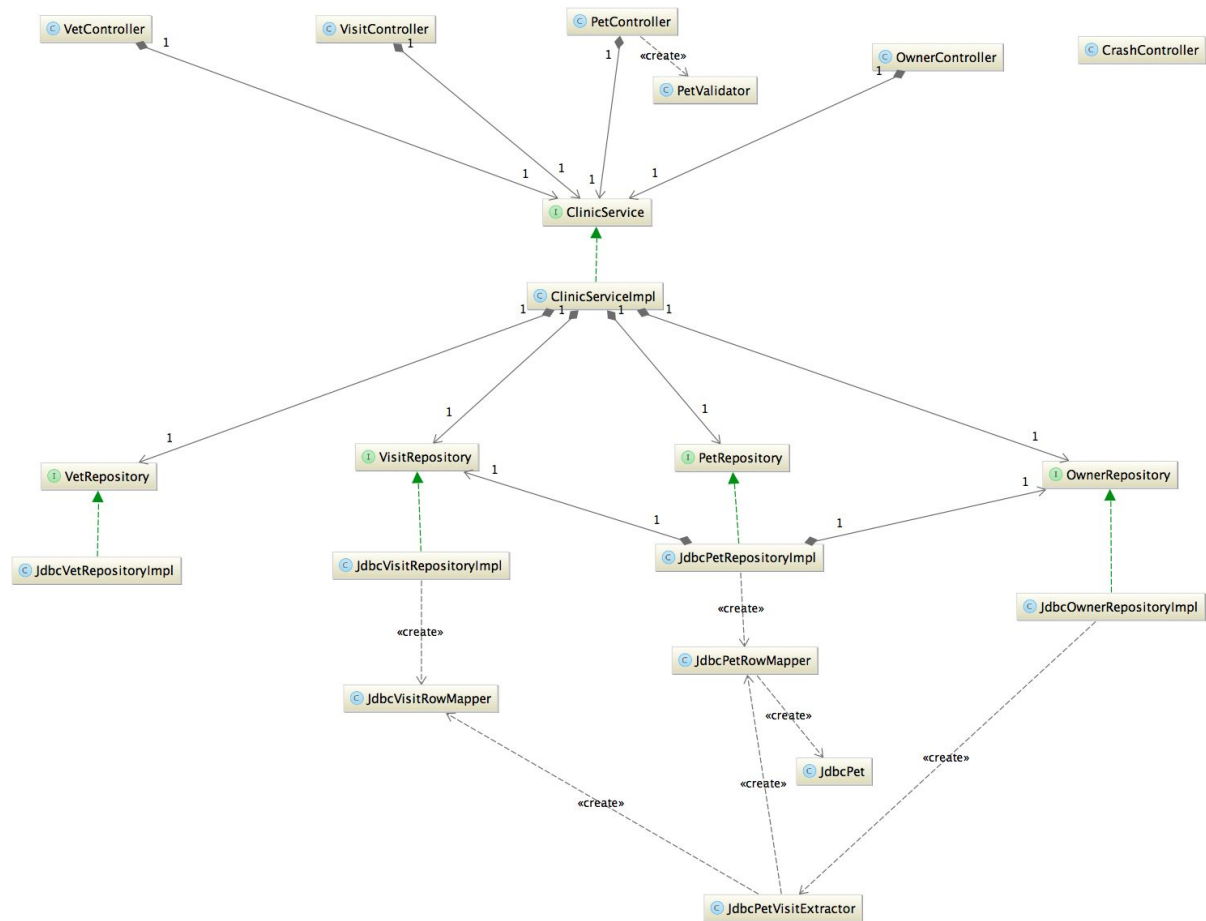
The component diagram for the API Application - diagram created with Structurizr.

Saturday, 11 November 2023 at 09:04 Greenwich Mean Time

# Modelo C4

Ex2. Spring PetClinic:

Engenharia reversa

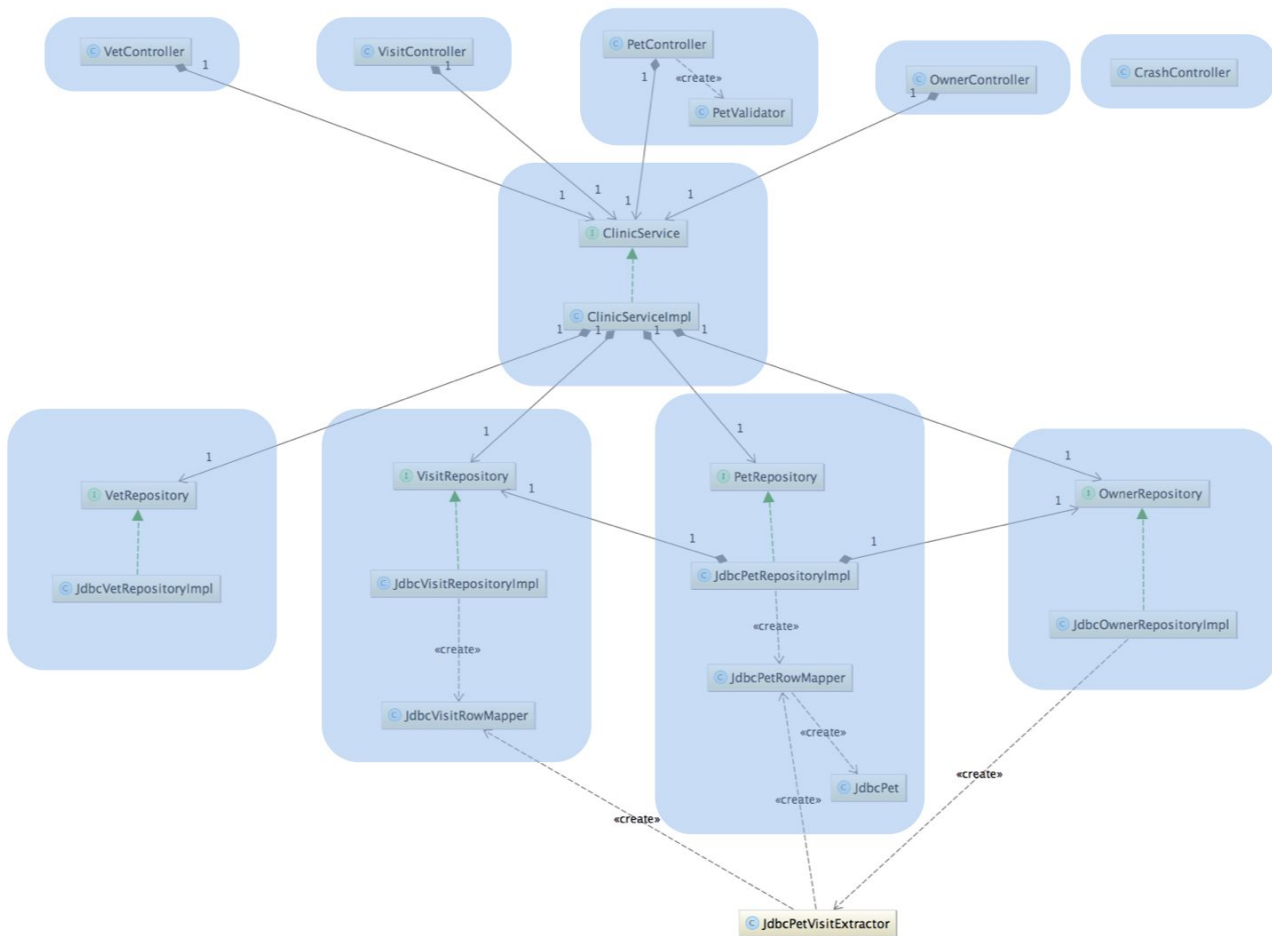




# Modelo C4

Ex2. Spring PetClinic:

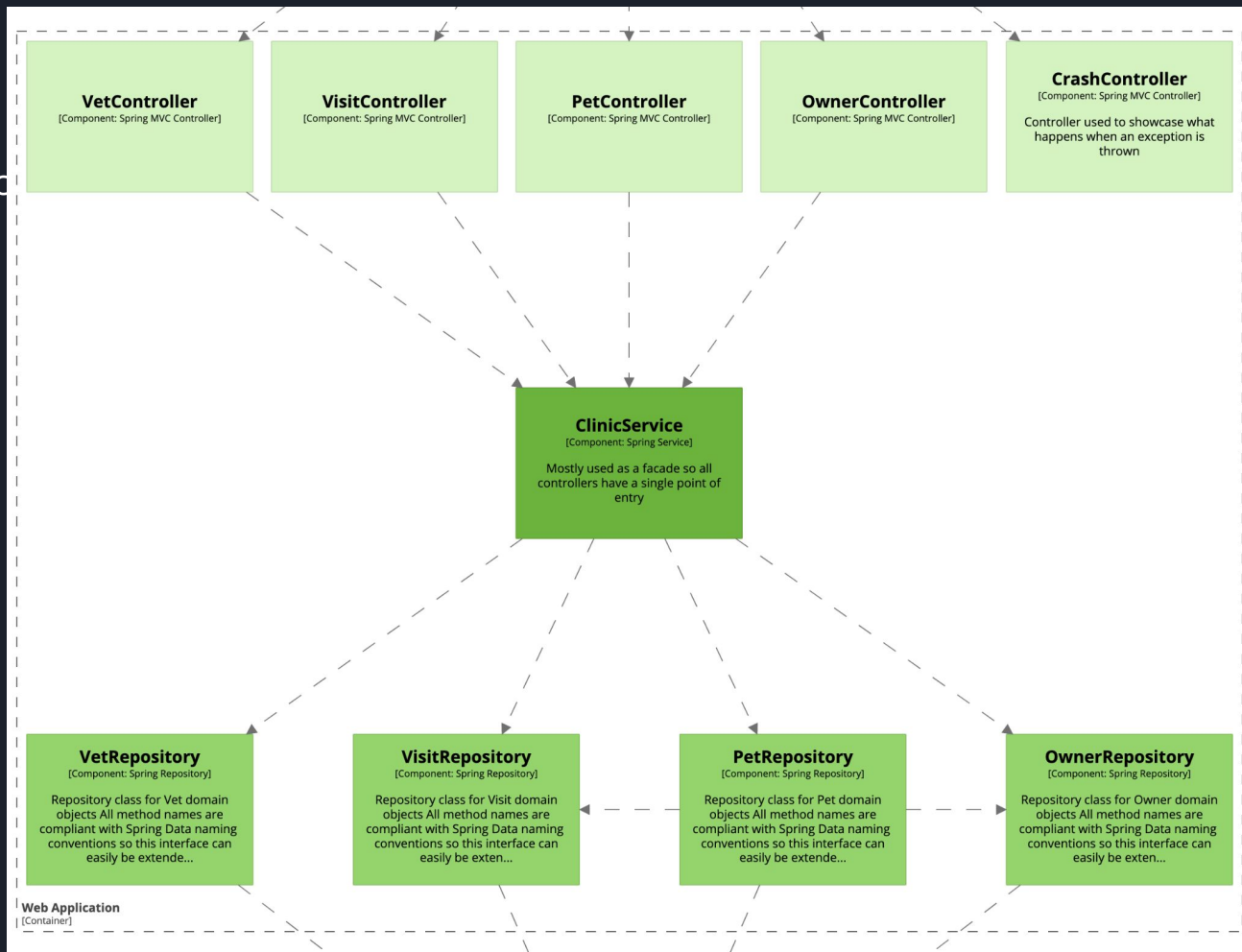
Agrupando  
componentes



# Modelo C4

Ex2. Spring PetClinic

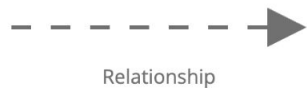
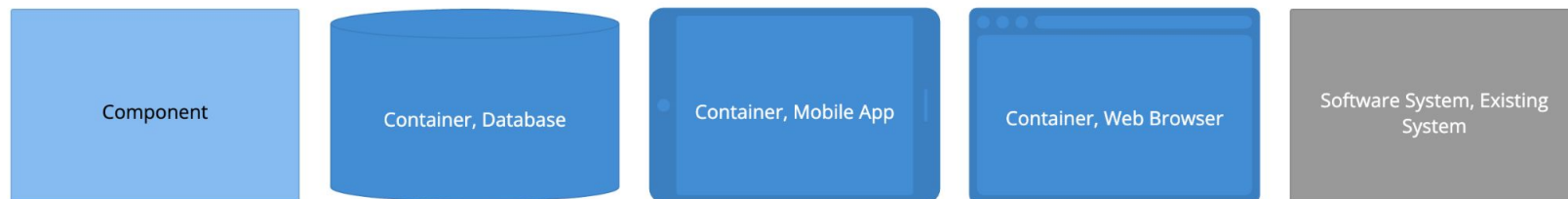
Diagrama de  
componentes



# Modelo C4

## Diagrama de Componentes

Elementos do diagrama:



# Modelo C4

## Diagrama de Componentes

- Público-alvo

Arquitetos e desenvolvedores de software.

- Recomendado?

Não, crie diagramas de componentes somente se você achar que eles agregam valor e considere automatizar sua criação para uma documentação de longa duração.



# Modelo C4

Diagrama de Código:

“os componentes são compostos por um ou mais elementos de código construídos com os blocos de construção básicos da linguagem de programação que você está usando.”

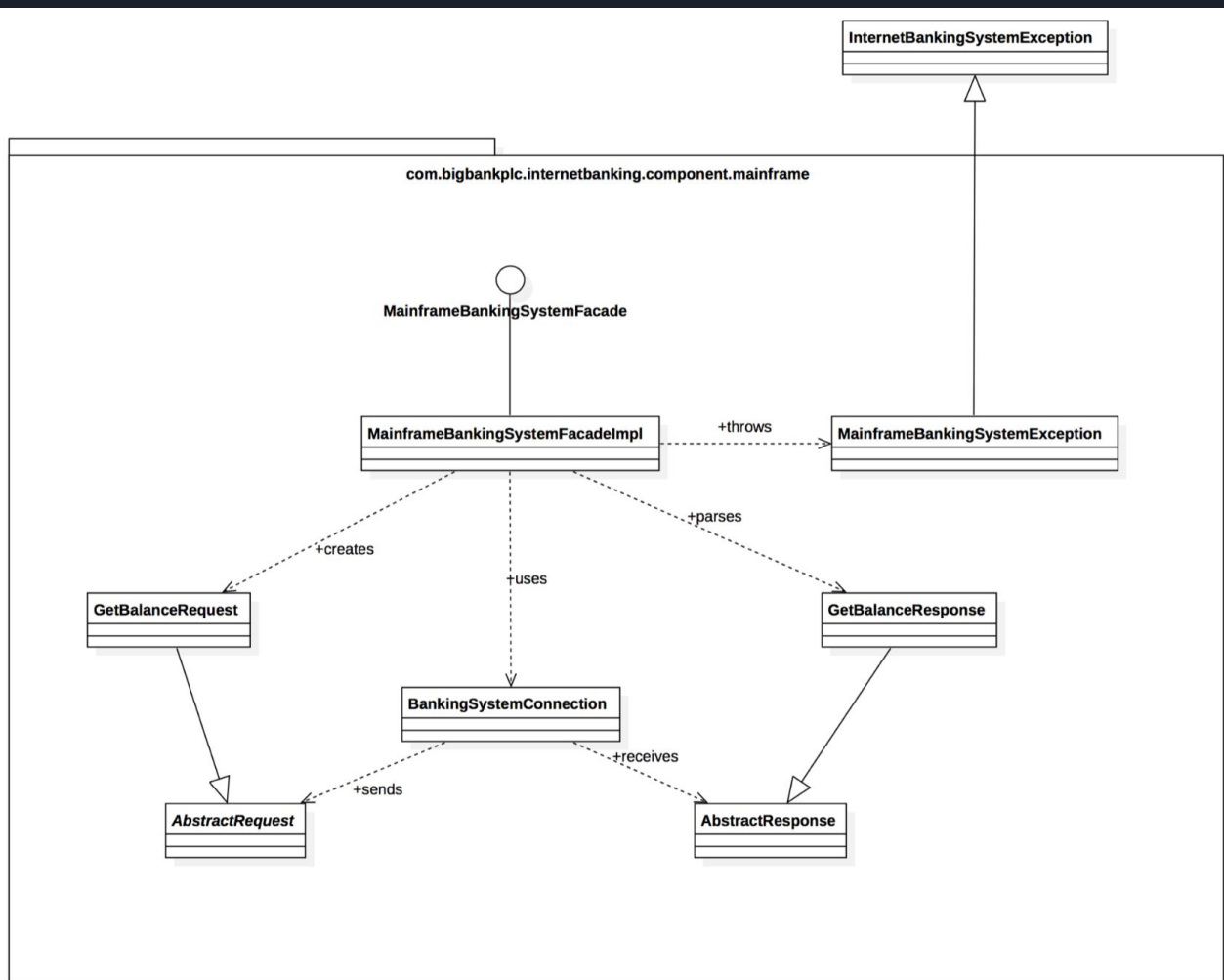
Ex.: classes, interfaces, enumerações, funções, objetos, etc.



# Modelo C4

## Diagrama de Código

ex.:



# Modelo C4

## Diagrama de Código

- Elementos primários do diagrama

Elementos de código (por exemplo, classes, interfaces, objetos, funções, tabelas de banco de dados, etc.) dentro do componente em escopo.

- Público-alvo

Arquitetos e desenvolvedores de software.

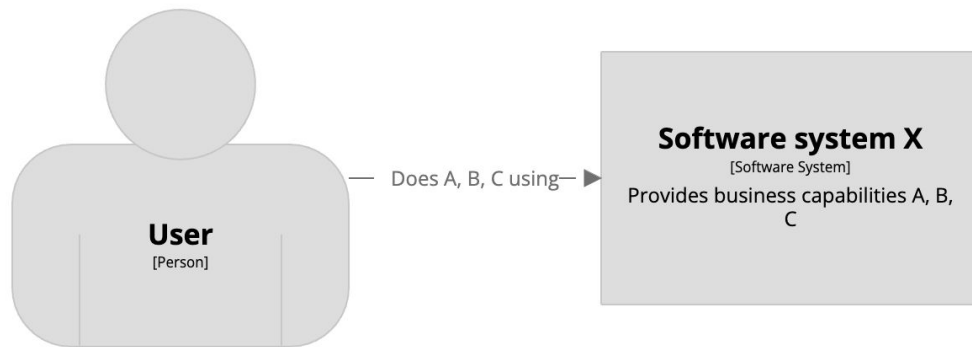
- Recomendado?

Não, especialmente para documentação de longa duração, porque a maioria dos IDEs pode gerar esse nível de detalhe sob demanda.

# Modelo C4 Aplicado a Microsserviços

## Etapa 1

- Estilo arquitetônico monolítico



[System Context] Software system X



# Modelo C4 Aplicado a Microsserviços

## Etapa 1

- Estilo arquitetônico inicial (monolítico)



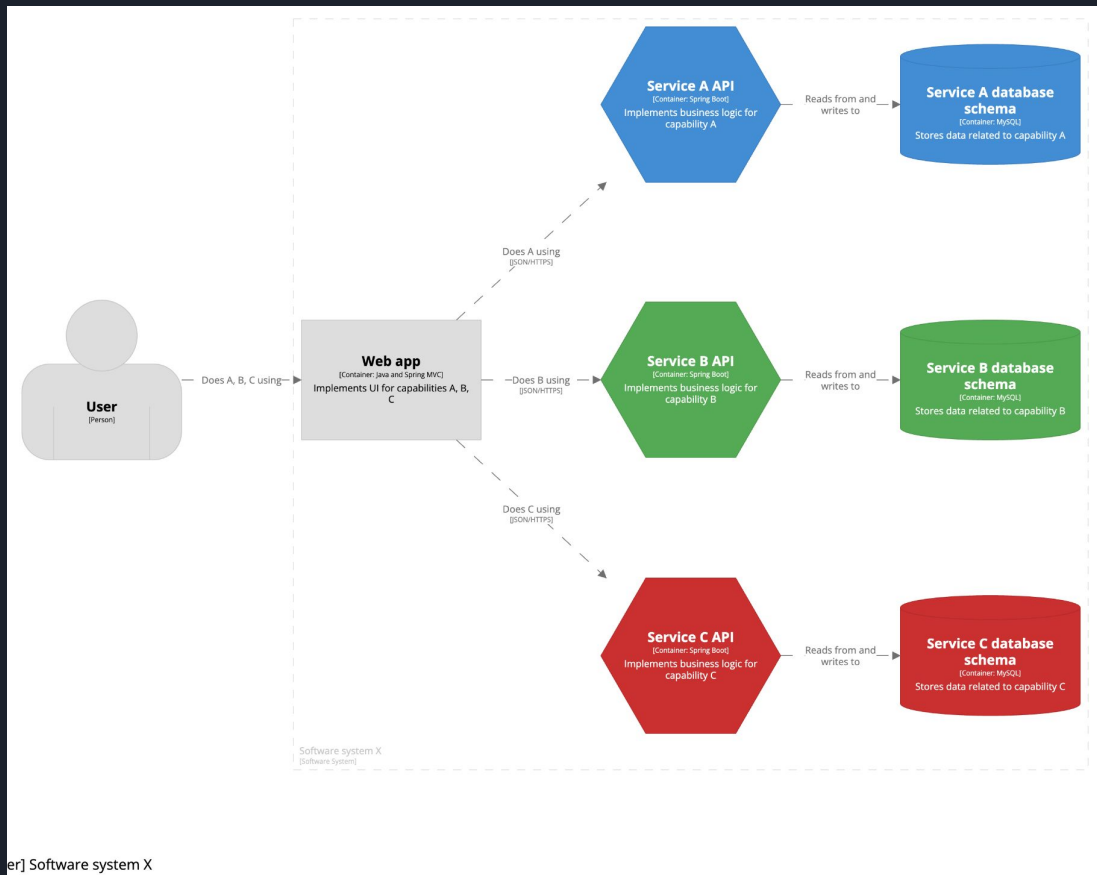
[Container] Software system X

# Modelo C4 Aplicado a Microsserviços

## Etapa 2

- Alterando o Estilo arquitetônico (Microsserviços)

\*Ainda com equipe única de engenharia

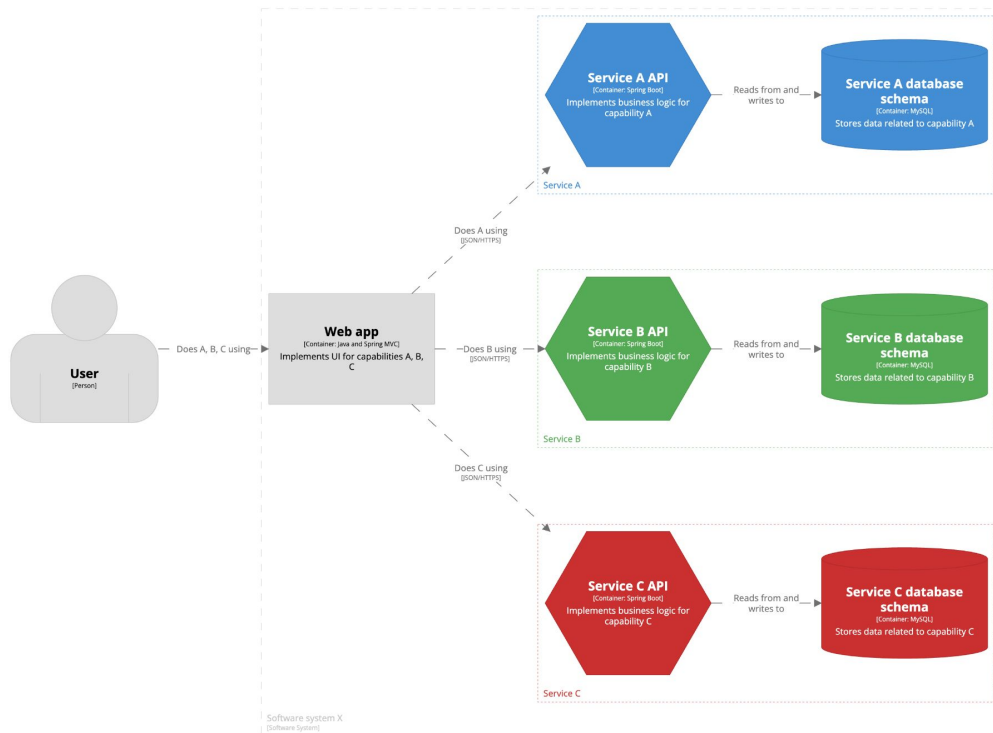


# Modelo C4 Aplicado a Microsserviços

## Etapa 2

- Alterando o Estilo arquitetônico (Microsserviços)

\*Ainda com equipe única de engenharia

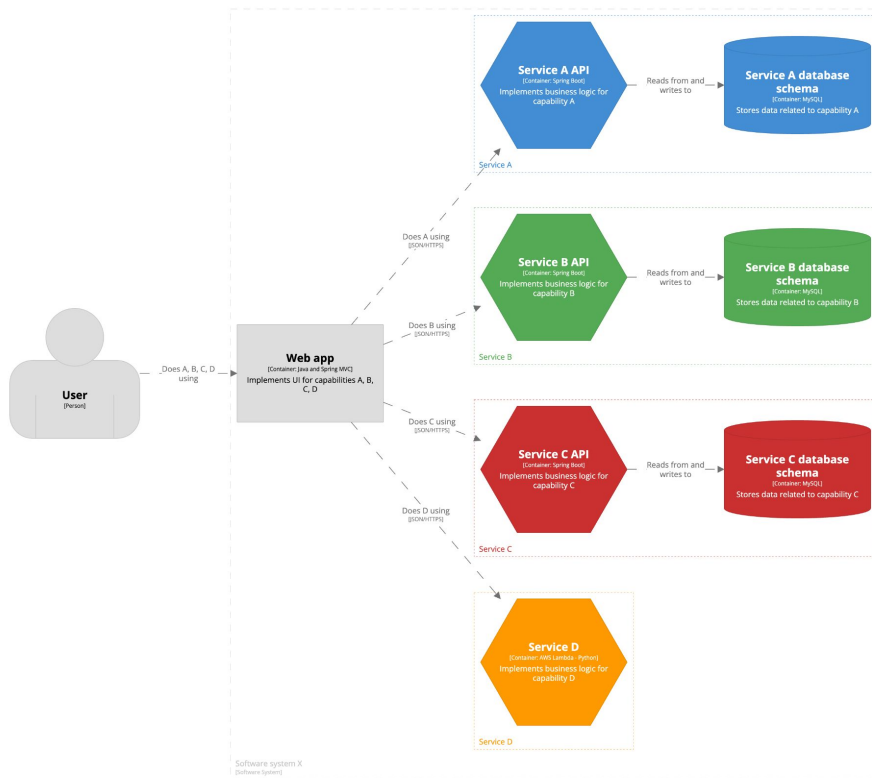


# Modelo C4 Aplicado a Microsserviços

## Etapa 2

- Alterando o Estilo arquitetônico (Microsserviços)

\*Adicionando aplicação sem estado

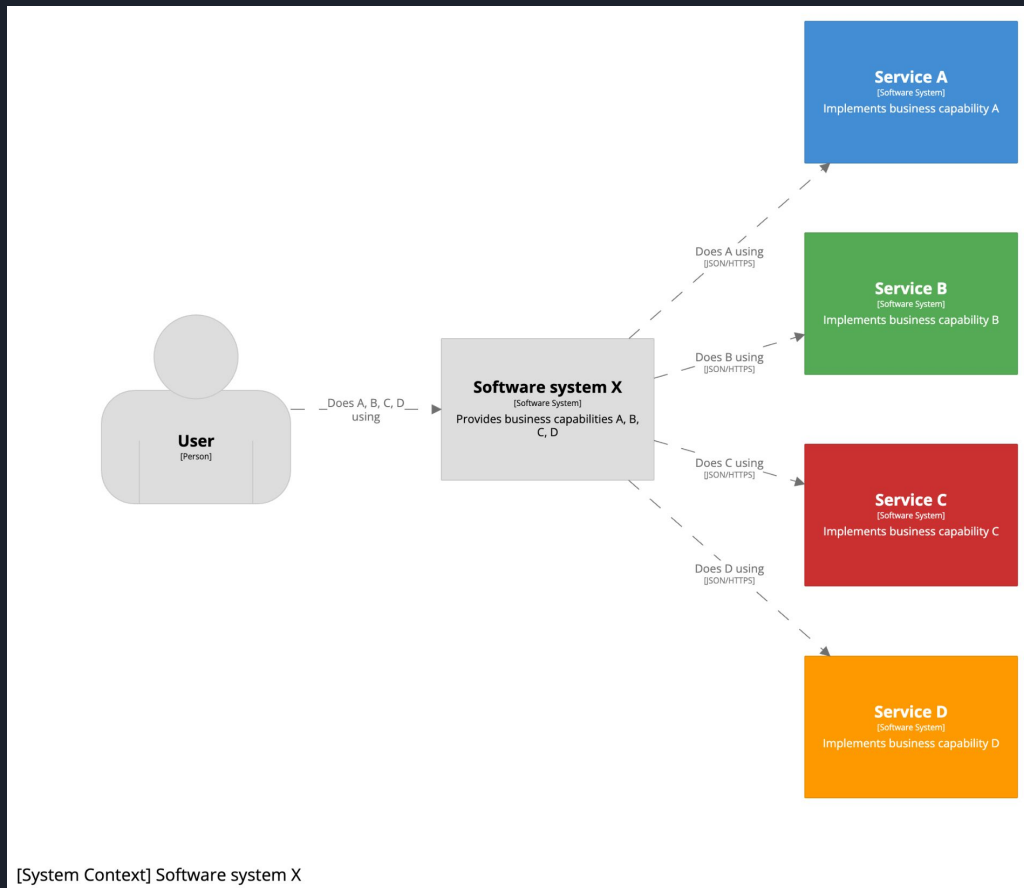


# Modelo C4 Aplicado a Microserviços

## Etapa 3

- Alterando o Estilo arquitetônico (Microserviços)

\*Dividindo os serviços entre equipes

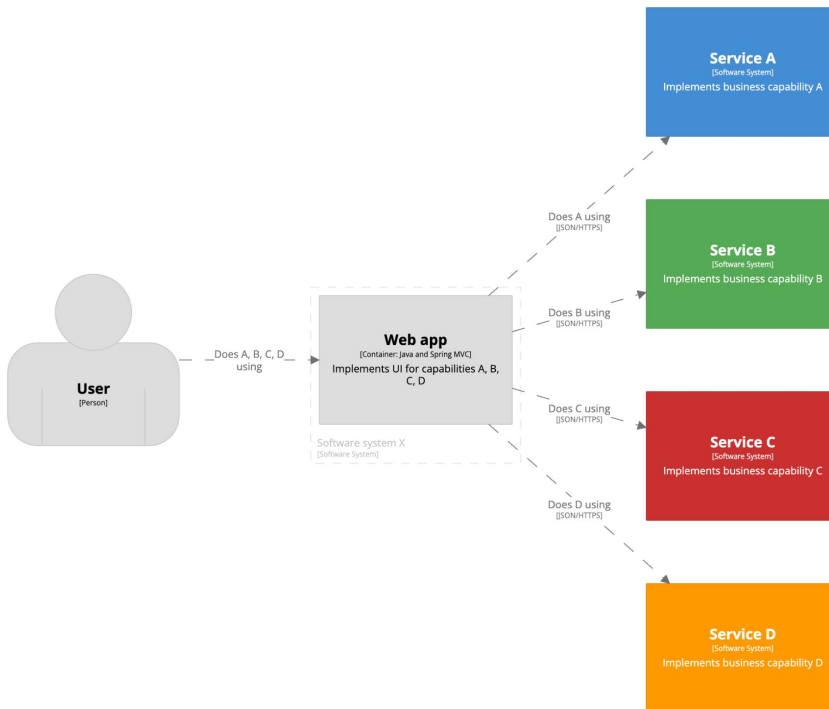


# Modelo C4 Aplicado a Microsserviços

## Etapa 3

- Alterando o Estilo arquitetônico (Microsserviços)

\*Dividindo os serviços entre equipes

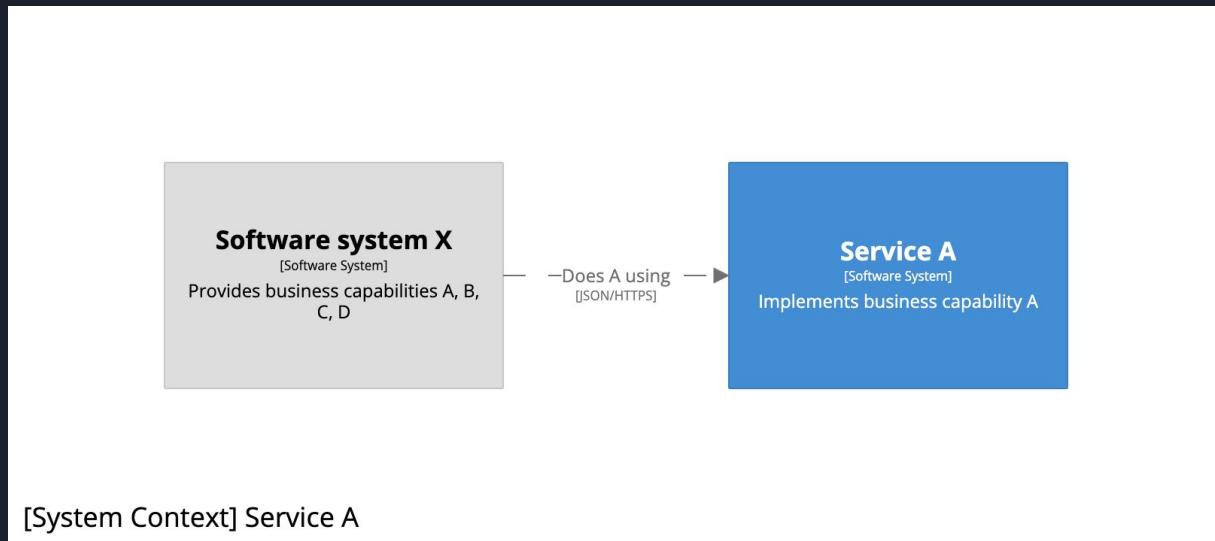


# Modelo C4 Aplicado a Microsserviços

## Etapa 3

- Alterando o Estilo arquitetônico (Microsserviços)

\*Perspectiva do Service A

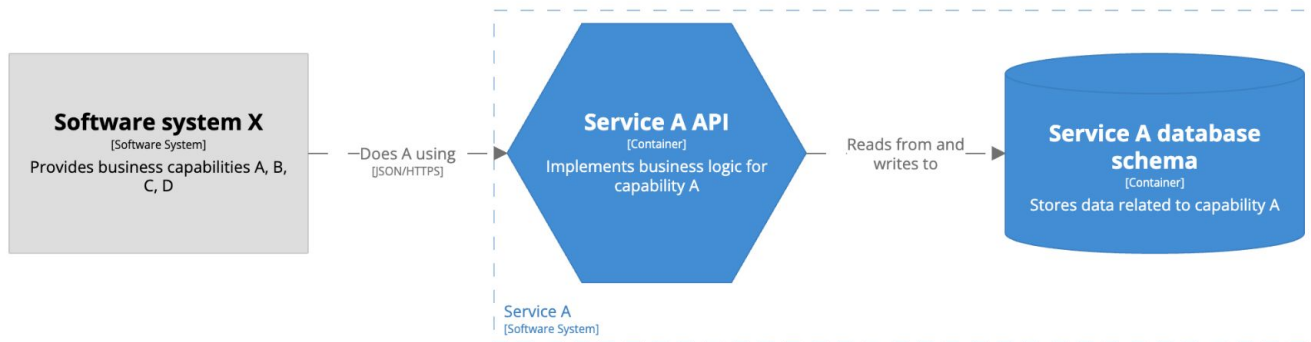


# Modelo C4 Aplicado a Microsserviços

## Etapa 3

- Alterando o Estilo arquitetônico (Microsserviços)

### \*Perspectiva do Service A



[Container] Service A



# Como modelar Microserviços

Foque em determinar os limites dos serviços de modo que:

- Oculte o máximo de informações do microserviço;
- Gere alta coesão entre as responsabilidades;
- Gere baixo acoplamento entre o microserviços.



# Limites dos Microserviços

- Ocultação de informações;
  - módulos devem teoricamente proporcionar:
    - Tempo de desenvolvimento melhorado (paralelismo);  
*"As conexões entre os módulos são as suposições que os módulos fazem uns sobre os outros."*
    - Compreensibilidade;
    - Flexibilidade.



# Limites dos Microserviços

- Ocultação de informações;
- Coesão - “Código que muda junto, permanece junto”;
- Acoplamento.



# Limites dos Microserviços

- Ocultação de informações;
- Coesão;
- Acoplamento - “uma mudança em um serviço não deve exigir uma mudança em outro”.



# Interação entre acoplamento e coesão

*"Coesão se aplica ao relacionamento entre as coisas dentro de um limite (um microserviço em nosso contexto), enquanto acoplamento descreve o relacionamento entre as coisas através de um limite"*



# Interação entre acoplamento e coesão

Lei de Constantine

- *"Uma estrutura é estável se a coesão for forte e o acoplamento for baixo".*



# Interação entre acoplamento e coesão

Lei de Constantine

- *"Uma estrutura é estável se a coesão for forte e o acoplamento for baixo".*

\* Lembrar:

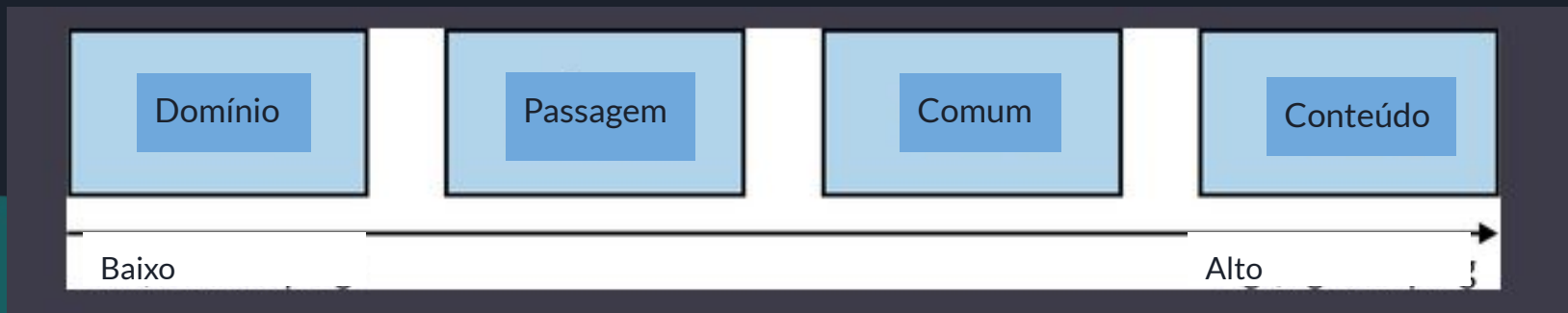
- O mundo não é estático;
- Podem haver impossibilidades temporárias de estabilidade.



# Tipos de acoplamento em microserviços

Acoplamentos não é uma verdade pura:

- Tipos de acoplamento no contexto de microserviços:

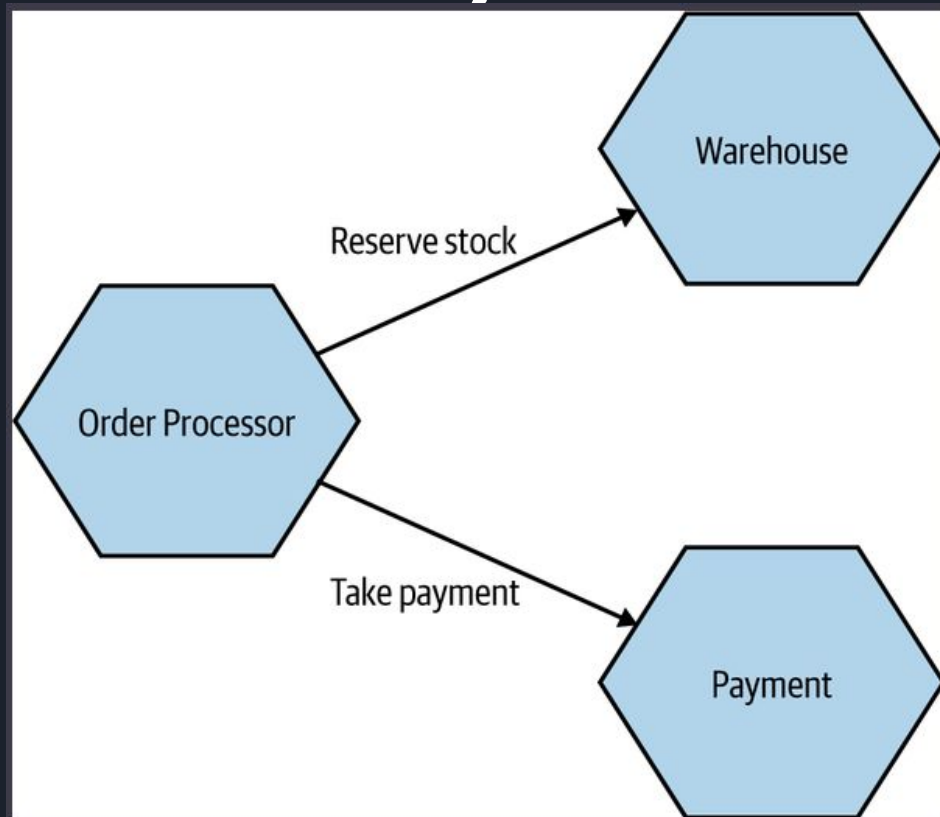




# Tipos de acoplamento em microserviços

## Acoplamento de Domínio:

- Através do uso de funcionalidades necessárias ao contexto do app.
- Inevitável mas evite precisar de muitos destes acoplamentos.



# Tipos de acoplamento em microserviços

## Acoplamento de Passagem:

- Quando é preciso enviar dados à microserviços, mas que esses dados só serão utilizados por microserviços bem abaixo da cadeia.

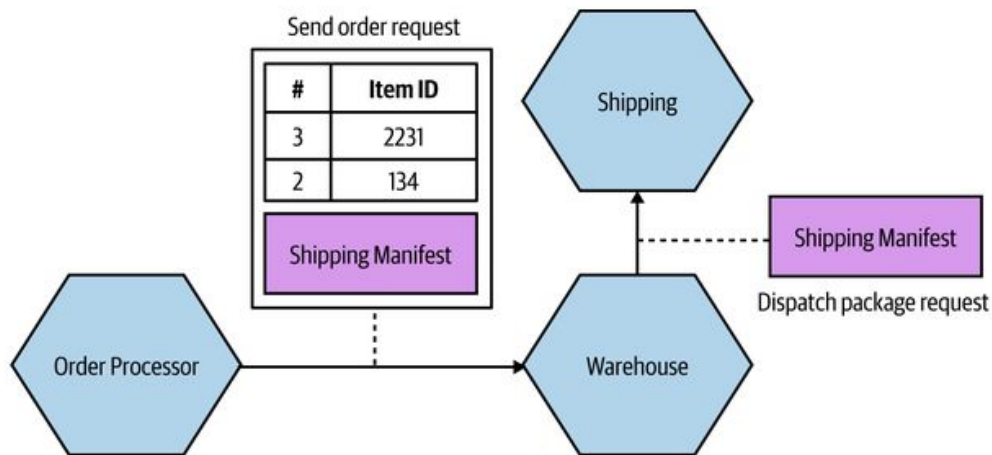


Figura 2-4. Acoplamento pass-through, no qual os dados são passados para um microserviço simplesmente porque outro serviço downstream precisa deles

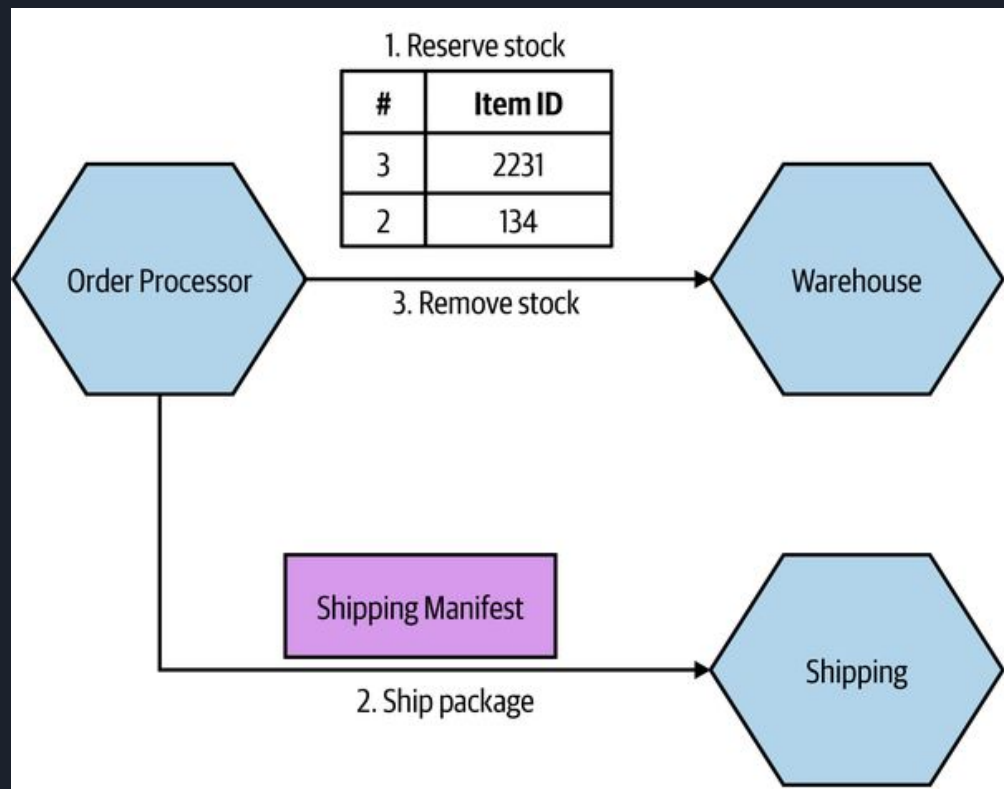
# Tipos de acoplamento em microserviços

Opção 1 ao acoplamento de Passagem:

- Comunicação direta com microserviço que utilizará os dados.

Concessões:

- Aumenta o acoplamento de domínio;
- Sem controle direto da ordem de execução.



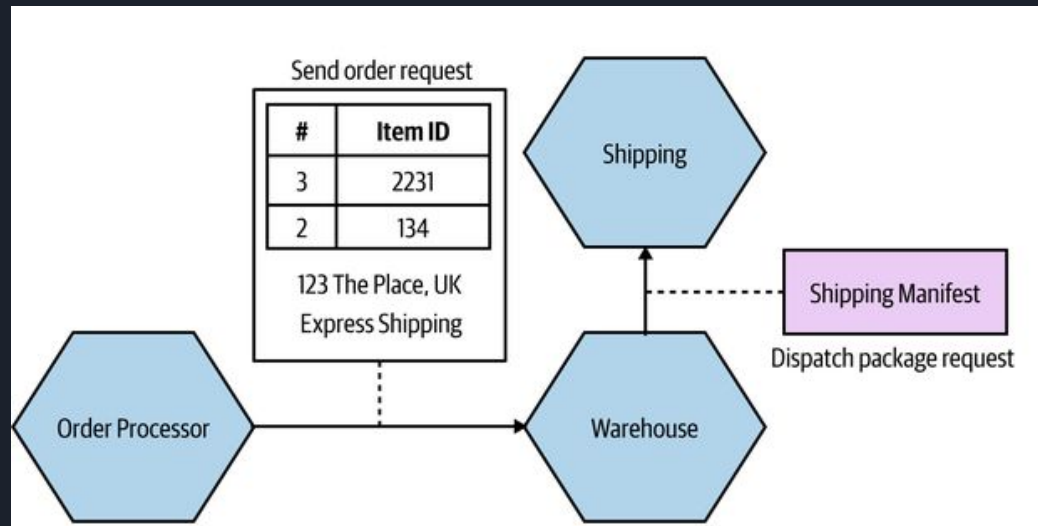
# Tipos de acoplamento em microserviços

Opção 2 ao acoplamento de Passagem:

- Construir o objeto de envio no microserviço mais próximo.

Concessões:

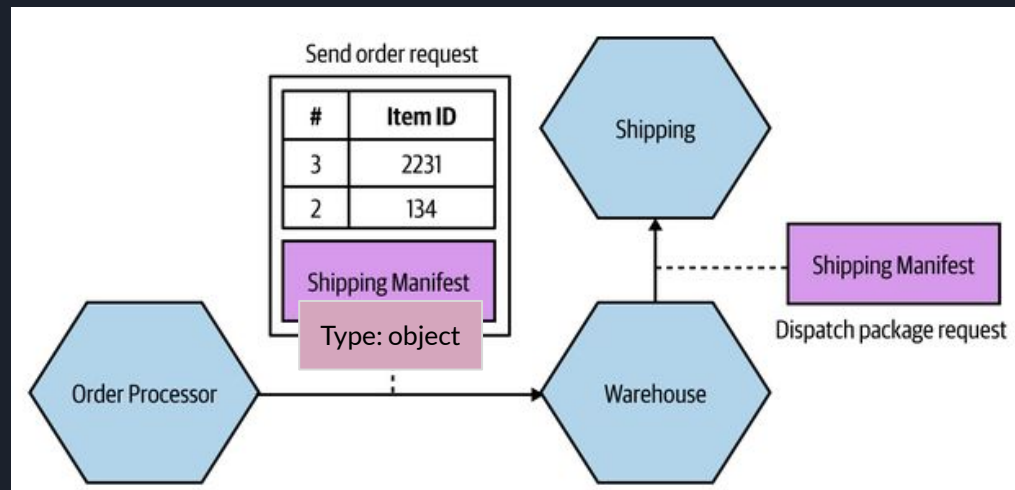
- O Microserviço construtor precisa receber os dados necessário para a construção;
- Alterações ainda podem exigir mudanças nos três serviços.



# Tipos de acoplamento em microserviços

Opção 3 ao acoplamento de Passagem:

- Fazer o envio de passagem de forma abstrata/oculta:
  - Ainda pode gerar conhecimento desnecessário entre MS.



# Tipos de acoplamento em microserviços

Acoplamento comum:

- Acoplamento comum ocorre quando dois ou mais microserviços fazem uso de um conjunto comum de dados.

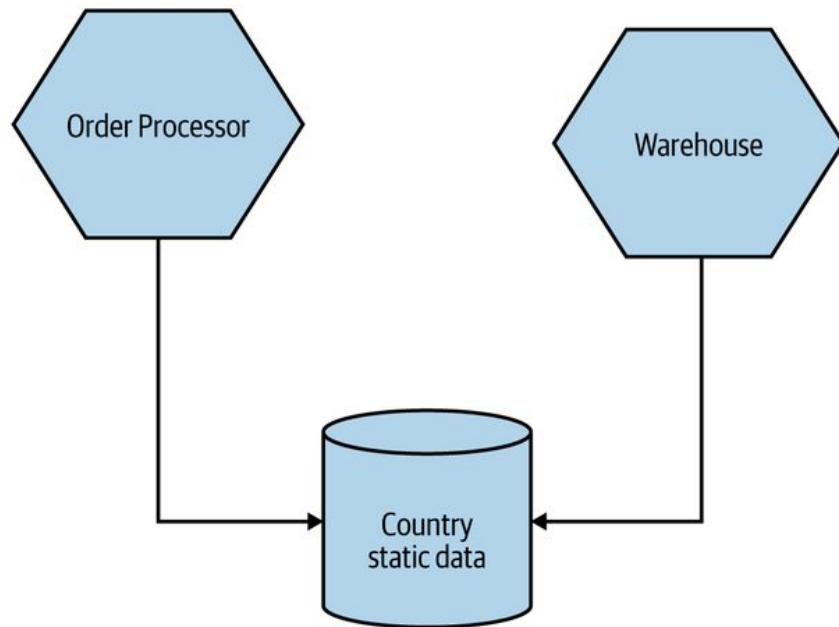


Figura 2-7. Vários serviços acessando dados de referência estáticos compartilhados relacionados a países do mesmo banco de dados

# Tipos de acoplamento em microserviços

Acoplamento comum é problemático em situações quando:

- os dados estáticos variam muito;
- vários microserviços podem ler e gravar ao mesmo tempo;
- Sobrecargas no recurso compartilhado.

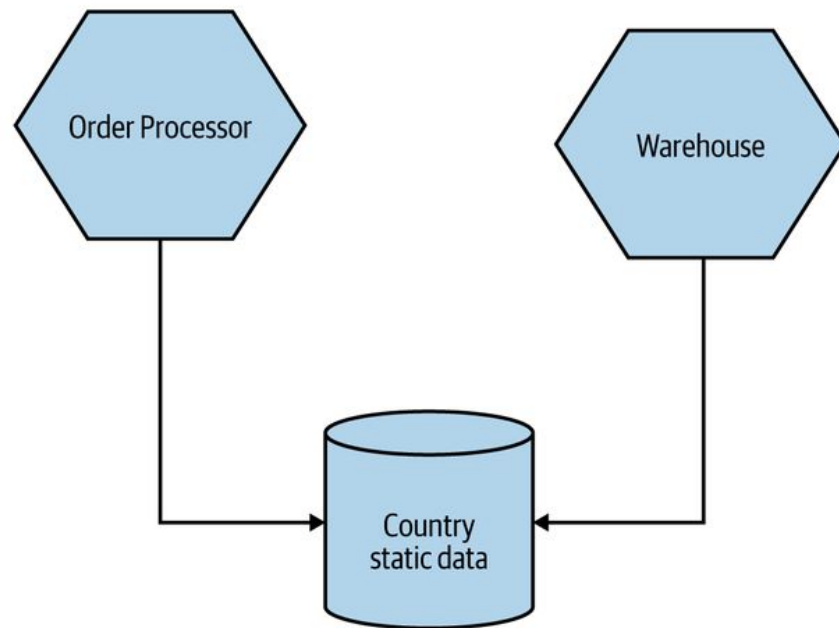


Figura 2-7. Vários serviços acessando dados de referência estáticos compartilhados relacionados a países do mesmo banco de dados

# Tipos de acoplamento em microserviços

Opção 1 ao acoplamento comum:

- Máquina de estados

(Possibilidade de um microserviço responsável pelo gerenciamento estado)

Concessões:

- Pode gerar centralização excessiva de responsabilidades e acoplamentos neste microserviço.

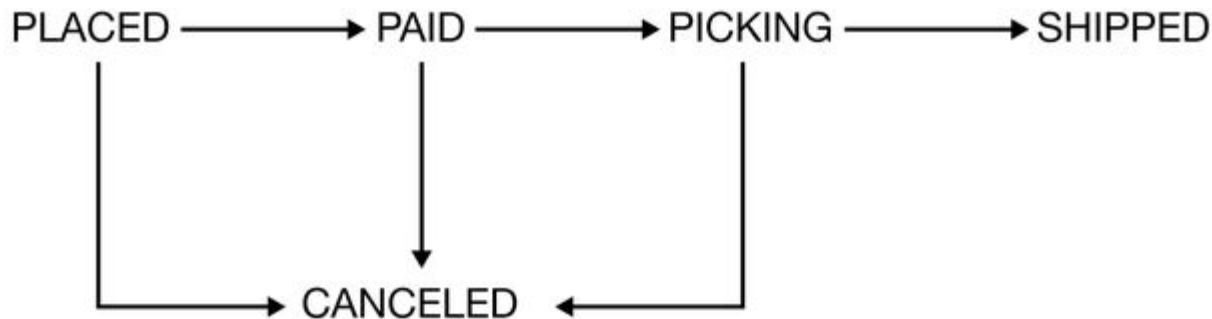


Figura 2-9. Uma visão geral das transições de estado permitidas para uma ordem no MusicCorp



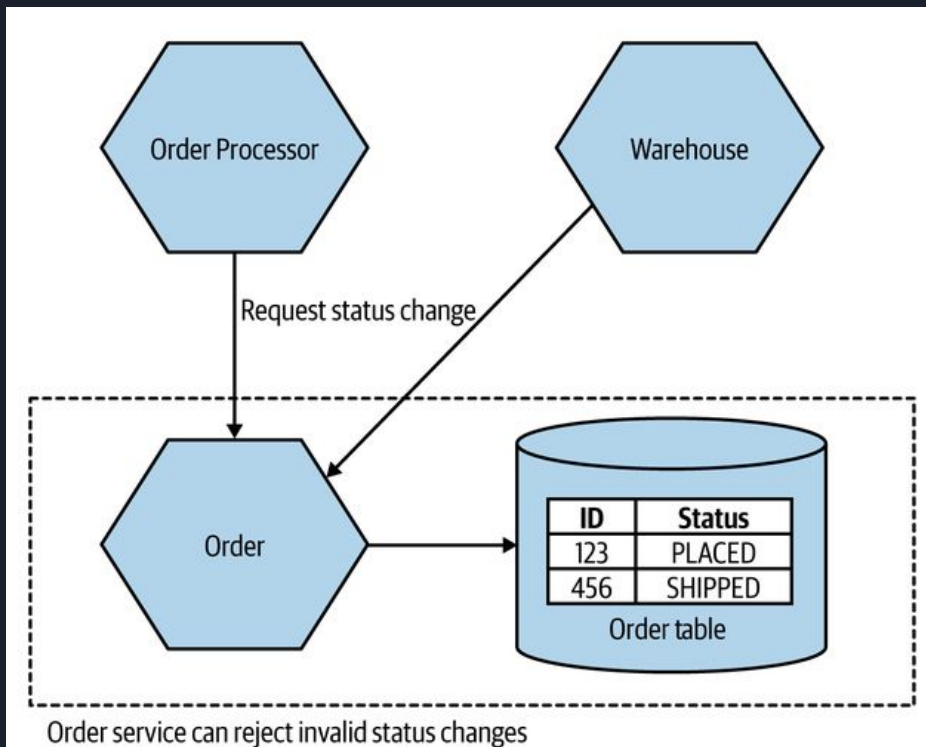
# Tipos de acoplamento em microserviços

Opção 2 ao acoplamento comum:

- Microserviço Wrapper (Quase um Repository)

Concessões:

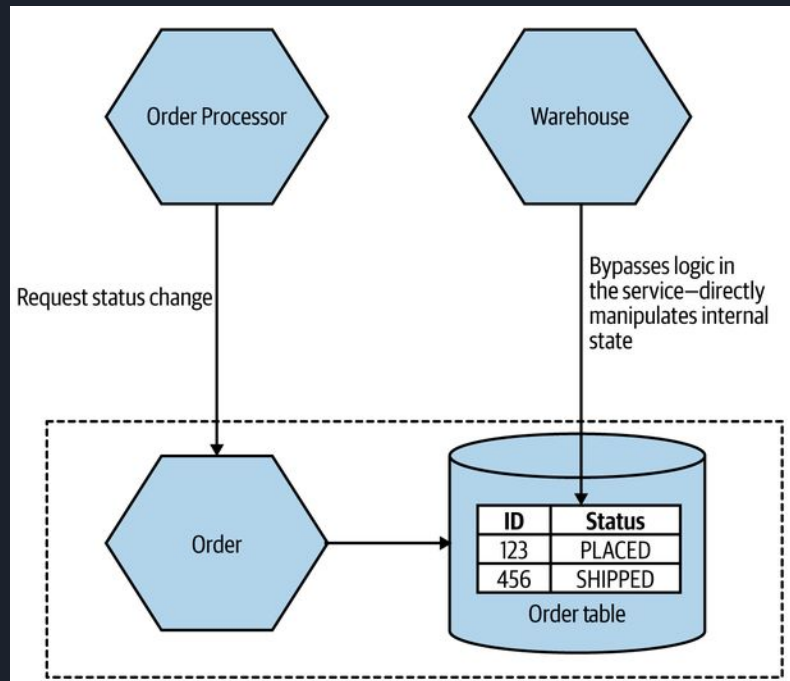
- A informação de comportamentos fica mais exposta;
- Baixa coesão: deixa a lógica de ordenamento distribuída no restante dos MS.



# Tipos de acoplamento em microserviços

Acoplamento de conteúdo:

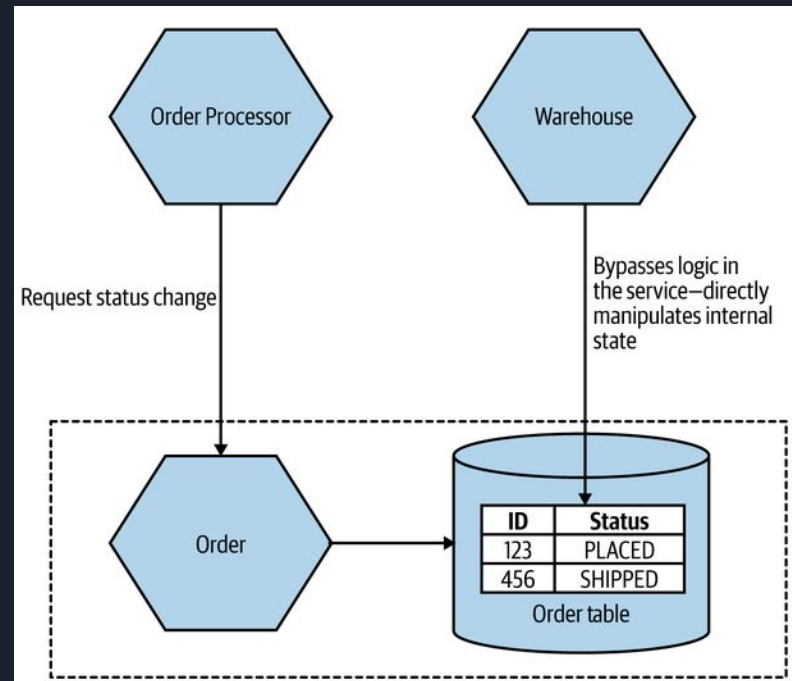
- “Acoplamento de conteúdo descreve uma situação na qual um serviço upstream alcança os internos de um serviço downstream e altera seu estado interno.”



# Tipos de acoplamento em microserviços

Acoplamento de conteúdo vs acoplamento comum :

- Acoplamento comum: O serviço altera apenas os dados sobre sua responsabilidade.
- Acoplamento de conteúdo: O serviço altera dados que são de responsabilidade de outro.



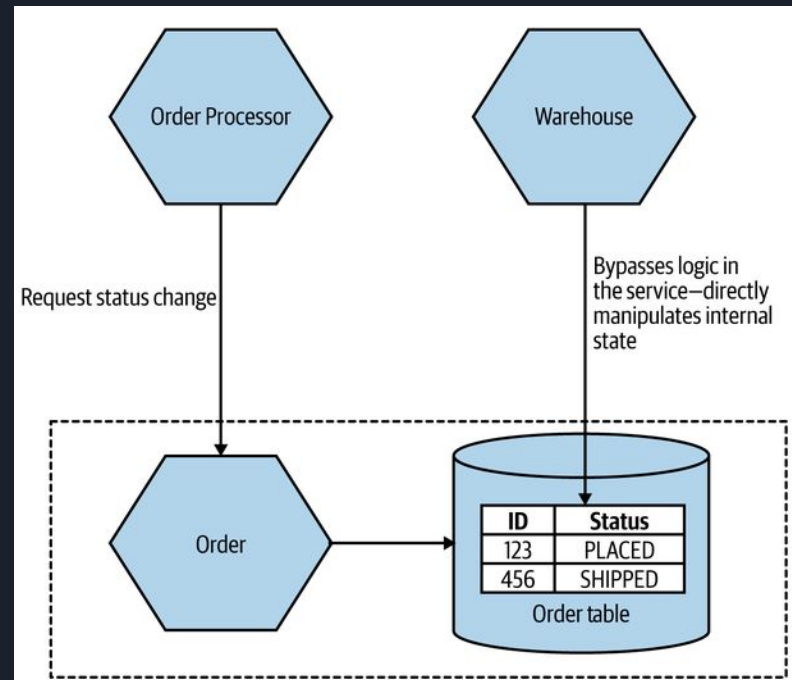
# Tipos de acoplamento em microserviços

Acoplamento de conteúdo:

- **Sempre evite este acoplamento!**

Problemáticas:

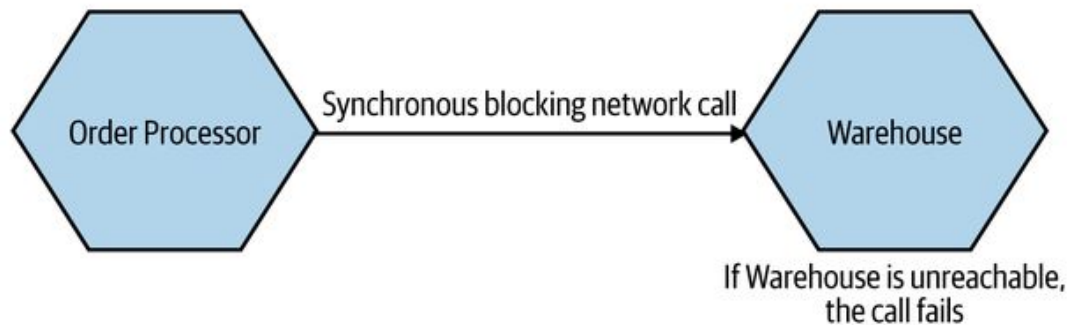
- No melhor dos casos, gera repetição de código (códigos de verificação);
- No pior, o segundo serviço altera dados sem saber as regras do serviço principal.
- Engessa alterações na tabela;



# Tipos de acoplamento Extra

## Acoplamento Temporal:

- Quando a funcionalidade de um microserviço depende da execução de outro ms para terminar.



**Figura 2-3. Um exemplo de acoplamento temporal, no qual** Order Processor **faz uma chamada HTTP síncrona para o** Warehouse **microserviço**



Instituto Infnet

# Aula 3 - Arquitetura de Microsserviços e Mobile