



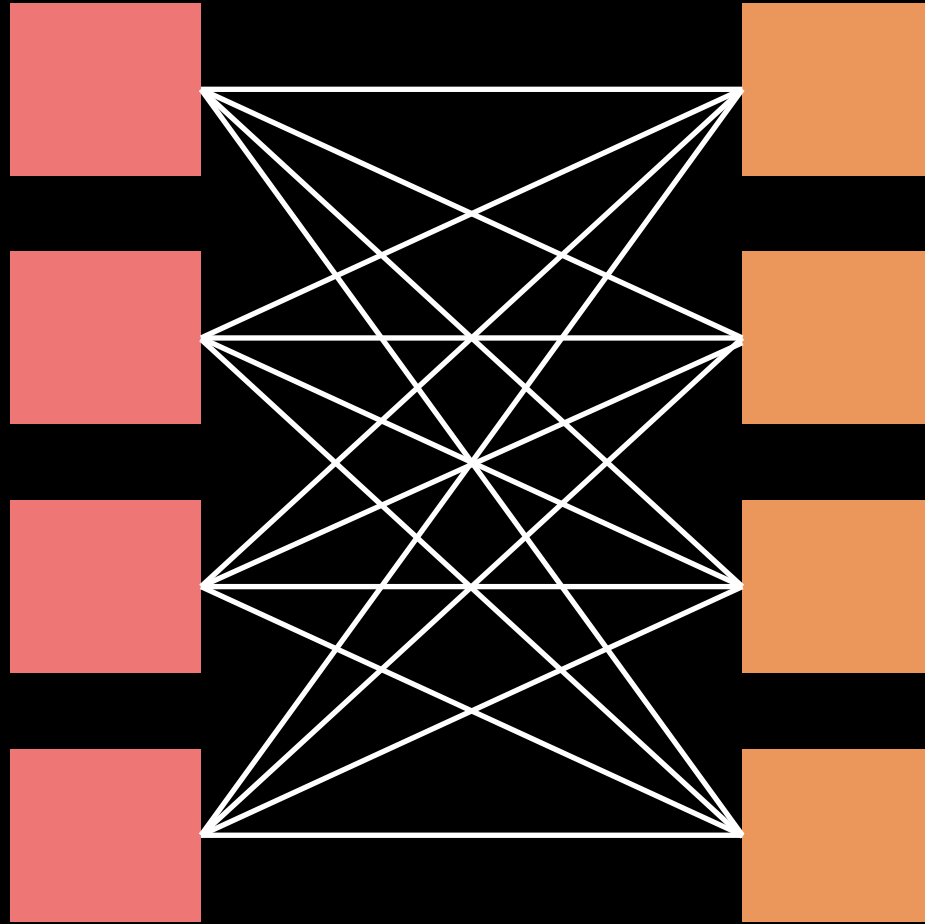
# SERVERLESS HAPPY HOURS

**ARQUITETURA BASEADA EM EVENTOS!**

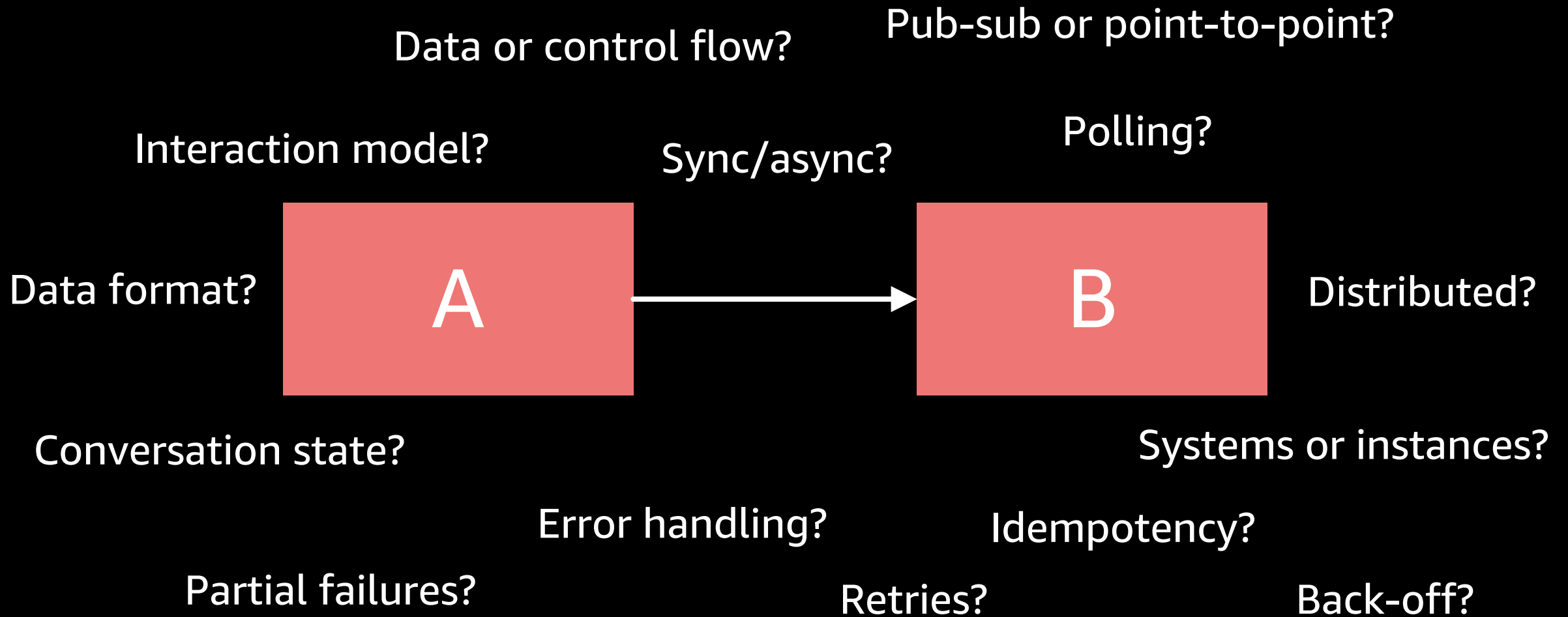
# Arquiteturas rigidamente acopladas são fáceis de começar com...



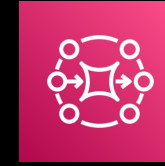
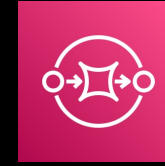
# ...mas a complexidade multiplica-se em escala



# Conectando dois sistemas. É difícil?



# Separe a sua arquitetura da sua escolha de produto



- Message-oriented
- Asynchronous
- Publish-subscribe
- Events
- Event-driven
- Distributed

X	X	X	X
X	X	X	X
X	X		
X	X		
?	?		
X	X	X	X

Note: for discussion purposes only. Not a product feature matrix.

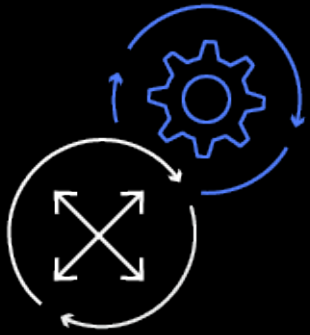


# A forma como os seus componentes estão Conectados define as propriedades essenciais do seu sistema

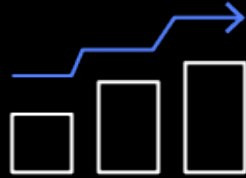
- Cada componente consegue escalar e falhar independentemente?
- Os sistemas estão lentos porque outros sistemas demoram em responder ?
- Será que as equipes de desenvolvimento precisam de uma coordenação rigorosa para lançar novas funcionalidades?
- Como vc trata com falhas e re-tentativas?
- Como vc integra em seu sistema outros Sistemas SaaS?



# Porquê construir arquiteturas baseadas em eventos?



Tolerância a  
Falha



Escalabilidade

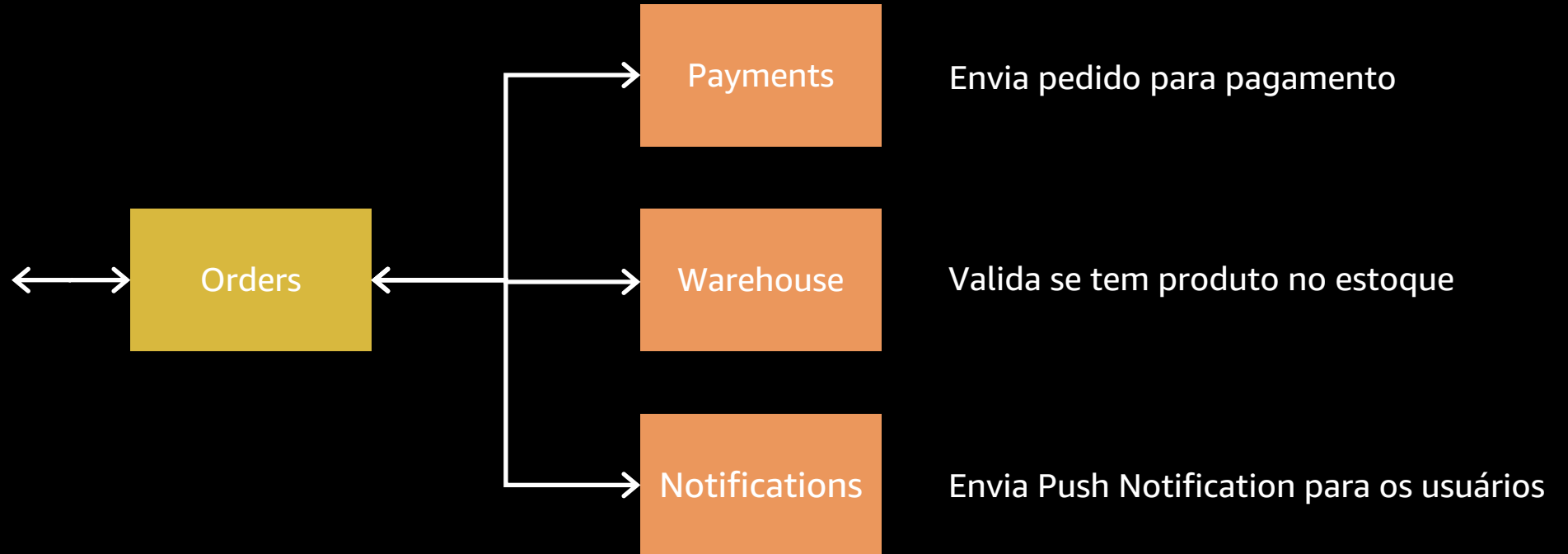


Extensibilidade



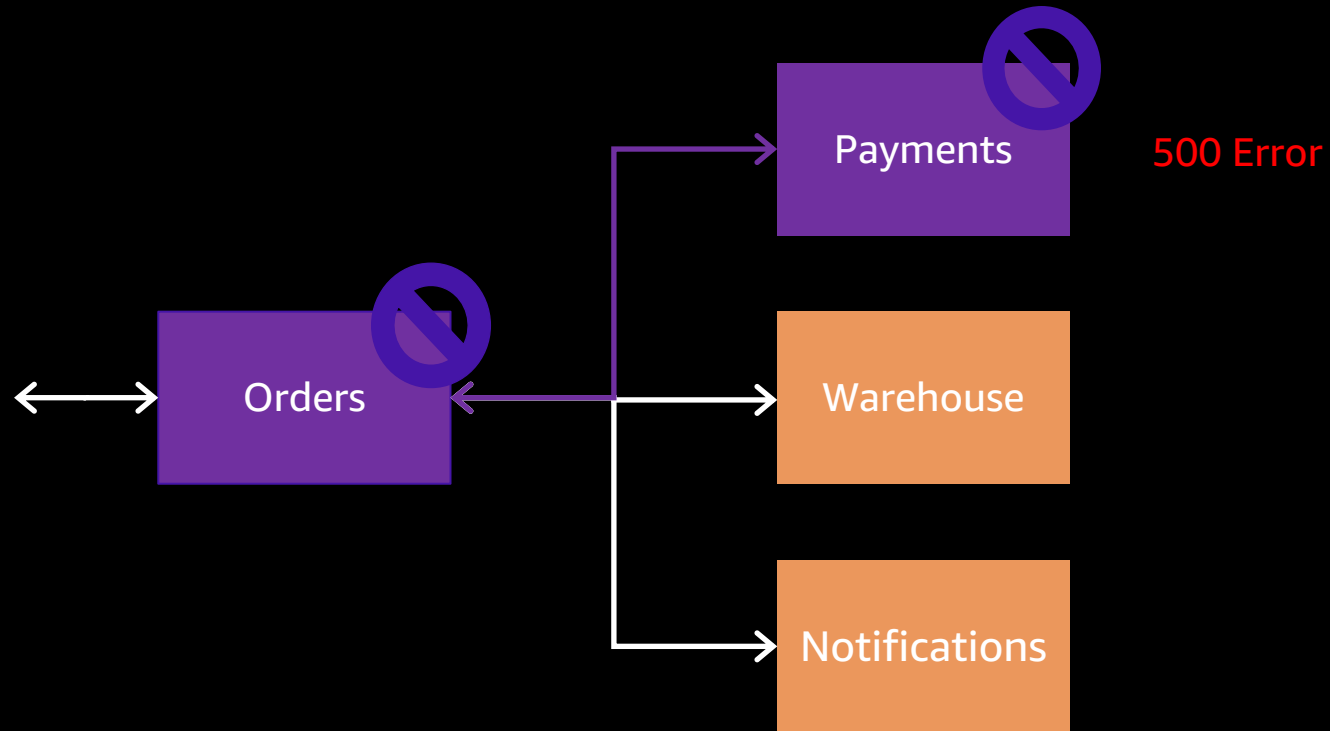
Agilidade

# Arquitetura acoplada

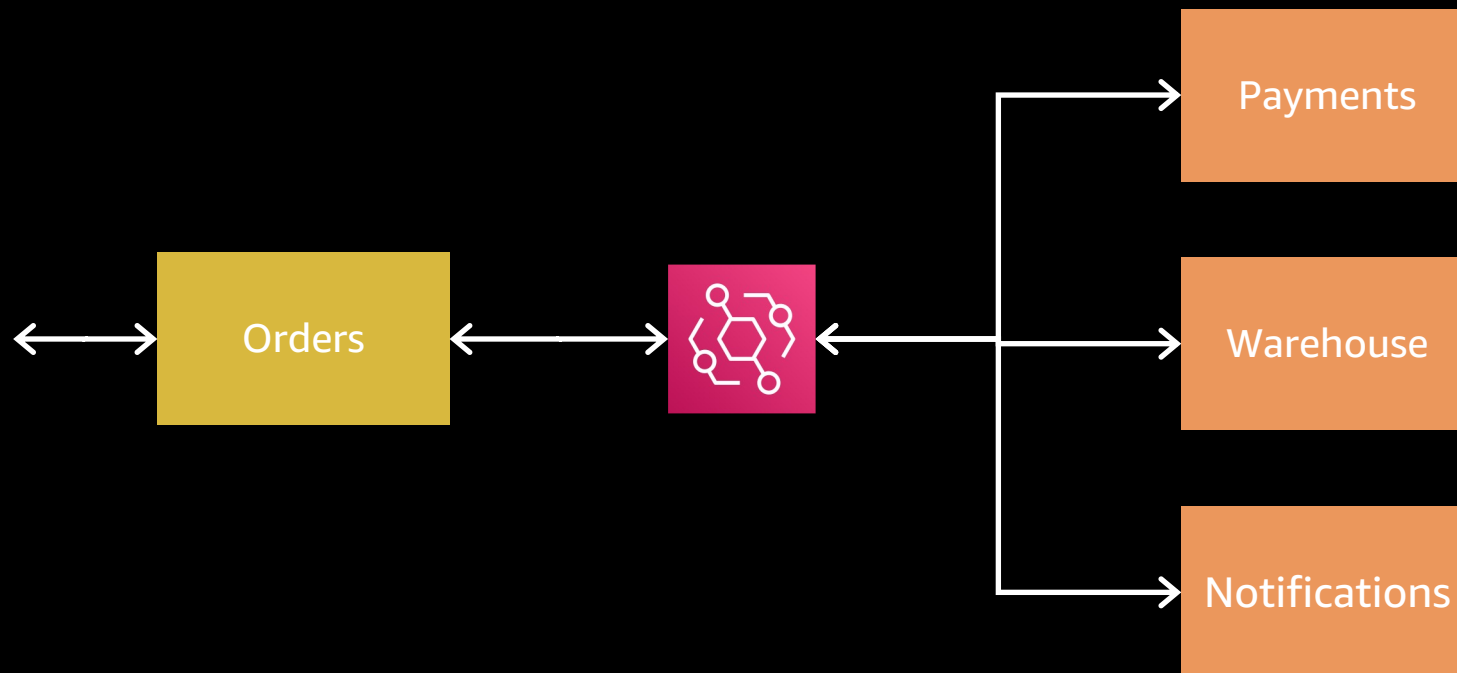




# Arquitetura acoplada



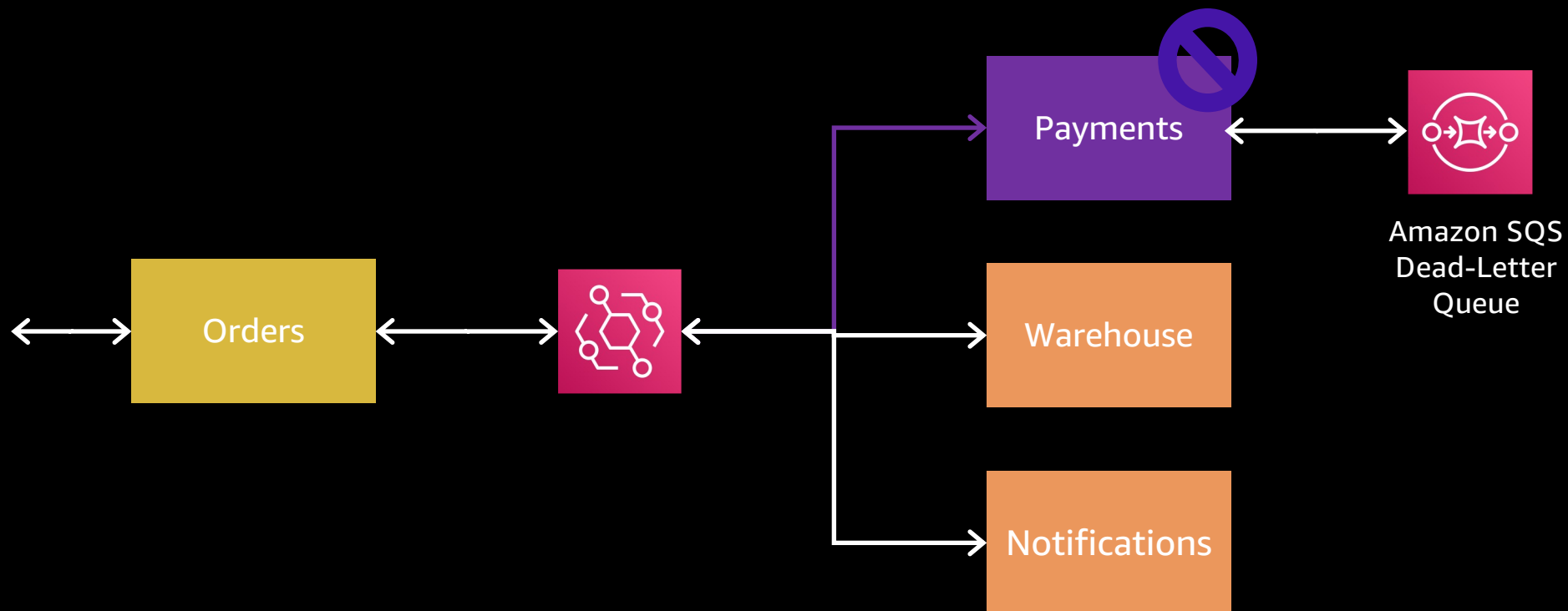
# Exemplo de Desacoplamento



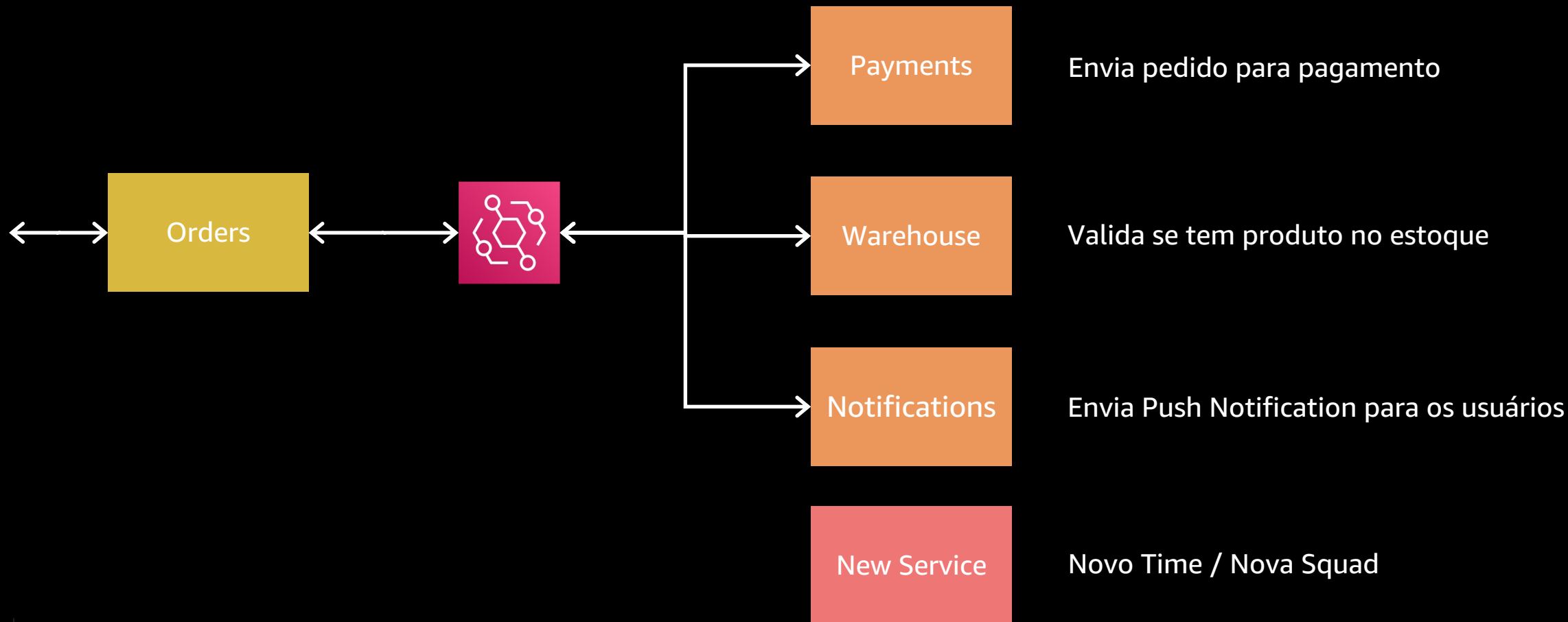
O serviço Order não precisa mais esperar os demais serviços responder para dar continuidade a outras interações:

- Aumento da resiliência do Serviço de Order
- Menor Latência aos usuários
- Fim das quedas em cascata

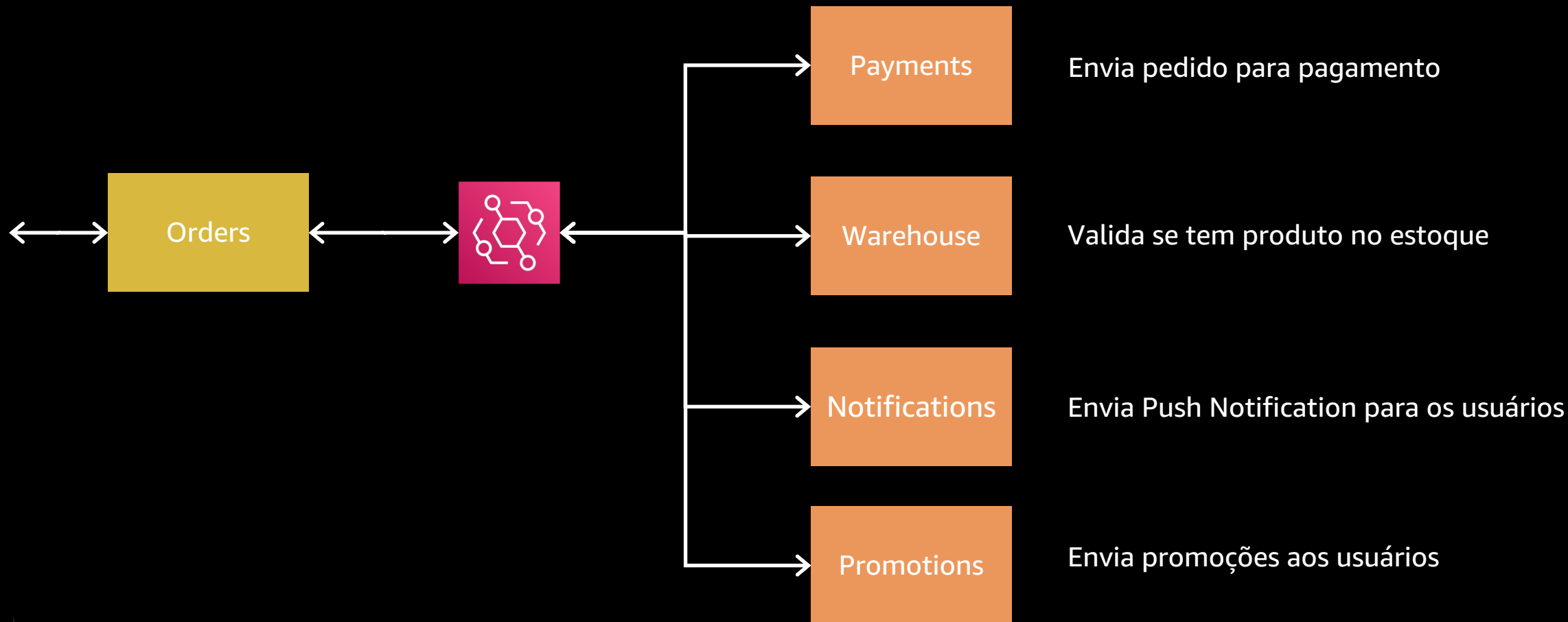
# Exemplo de Desacoplamento



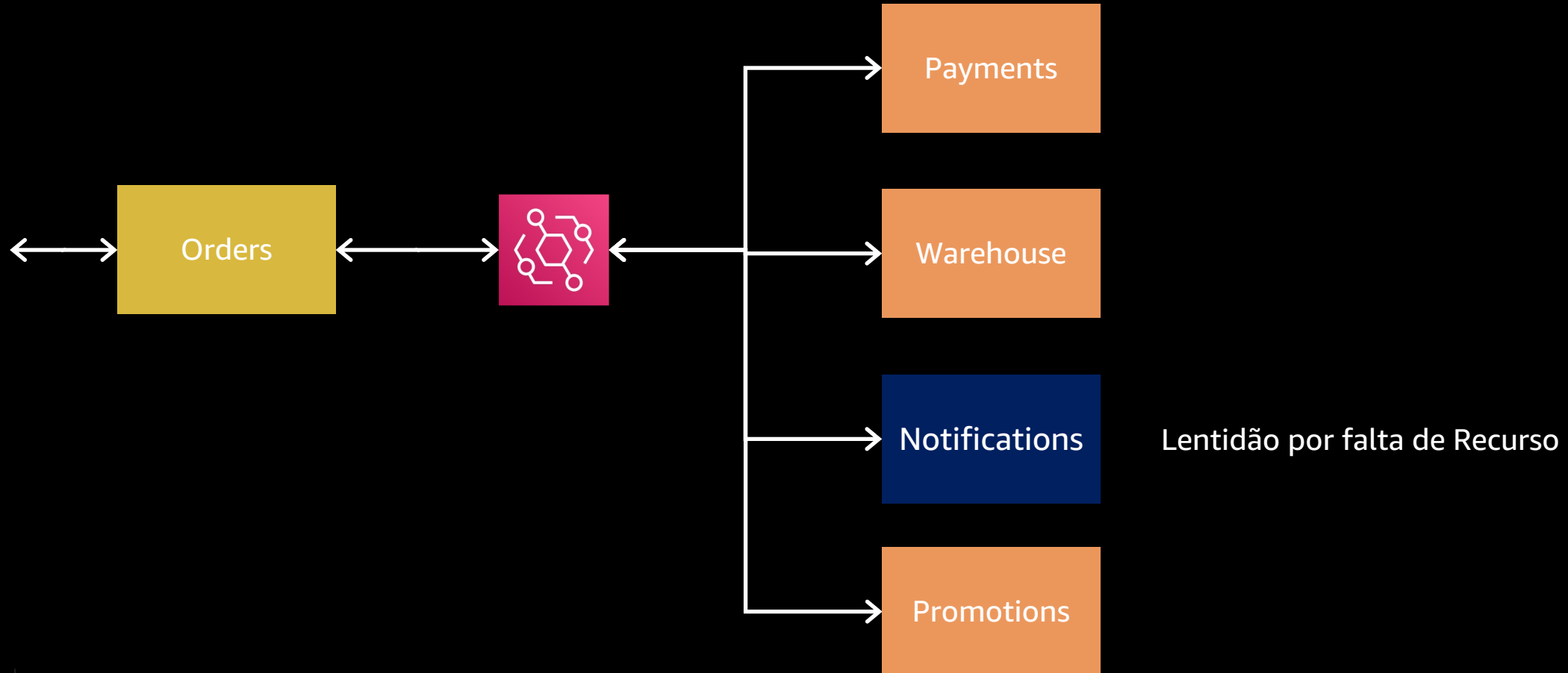
# Exemplo de Desacoplamento



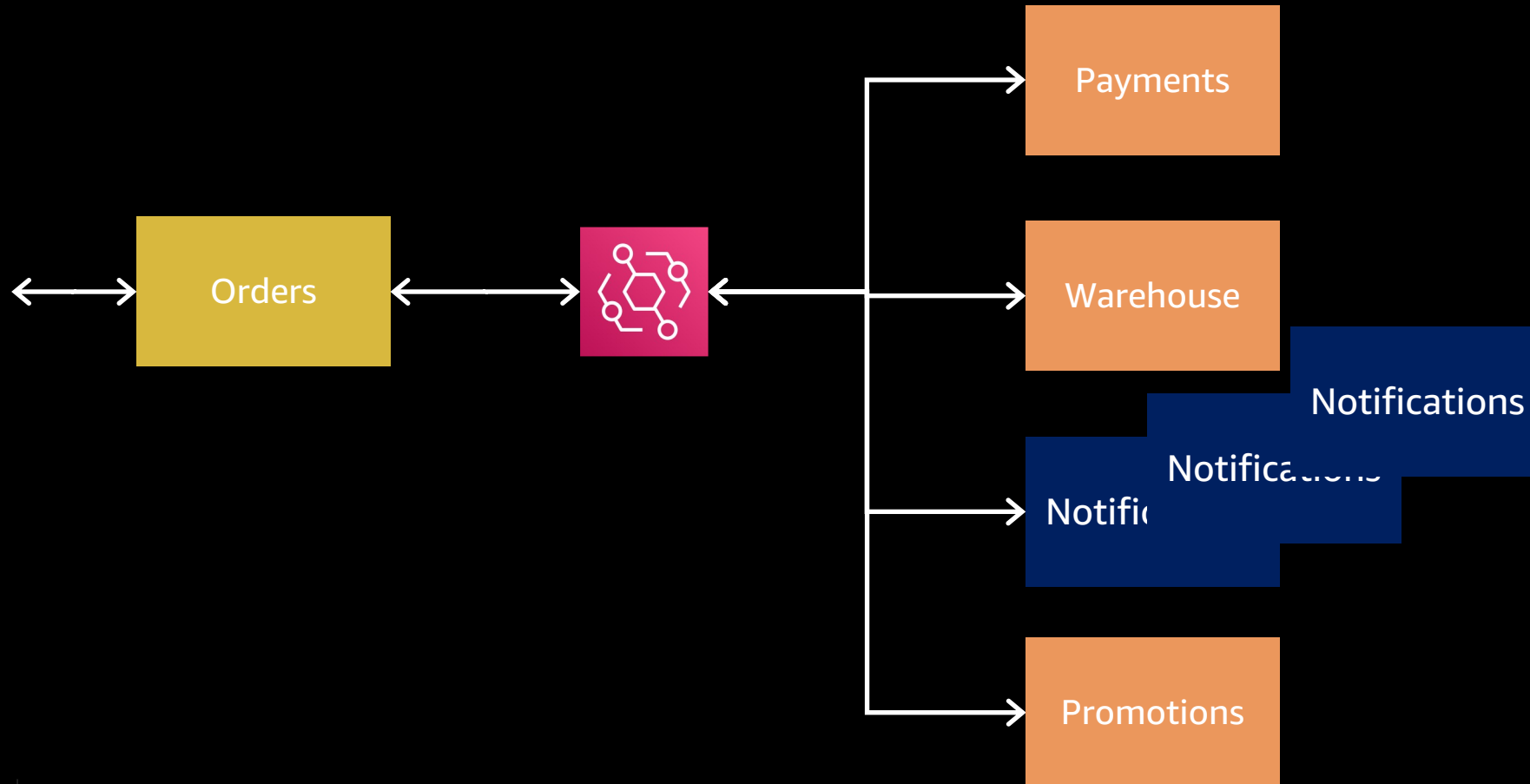
# Exemplo de Desacoplamento



# Sample Decoupling Architecture

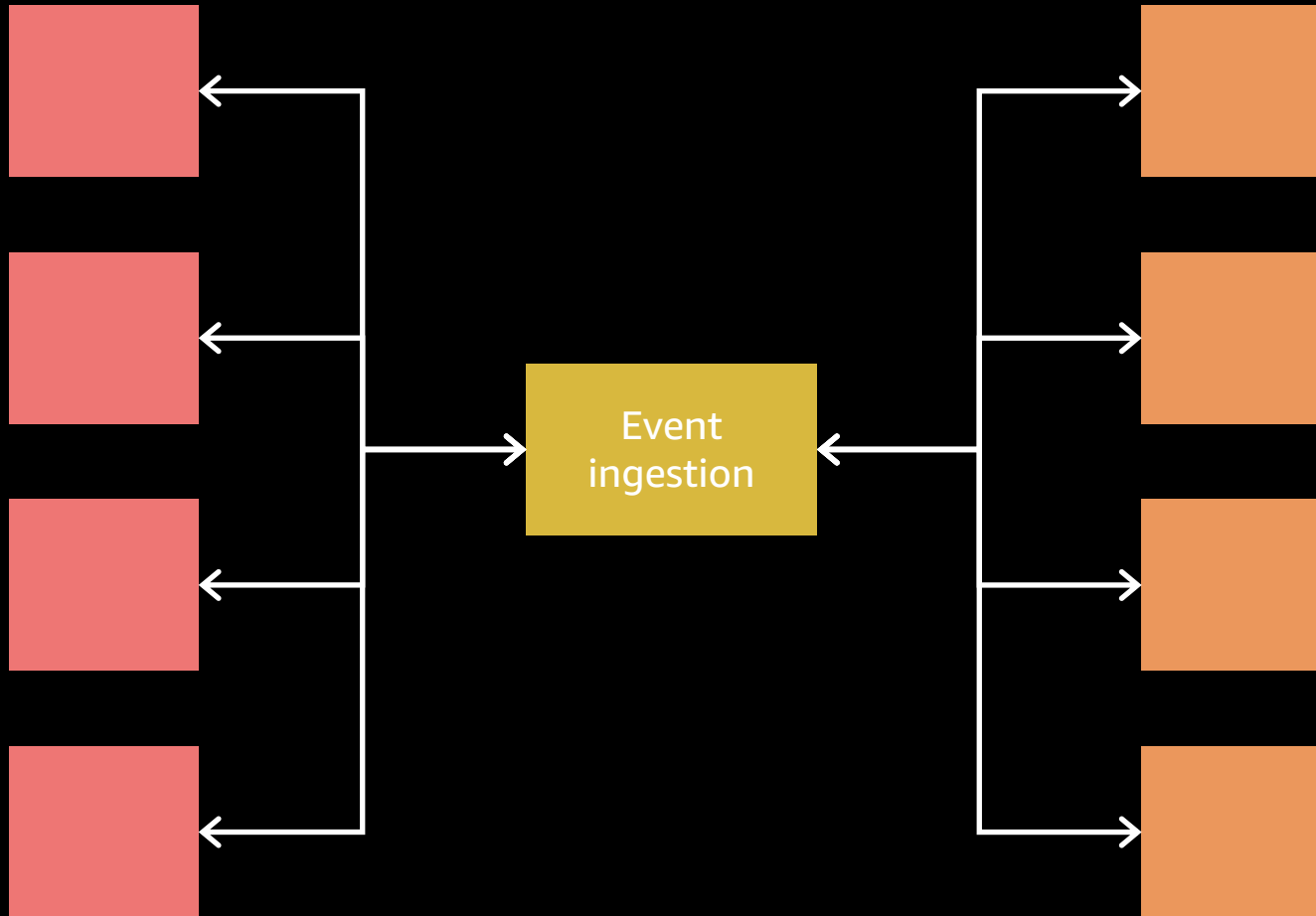


# Sample Decoupling Architecture



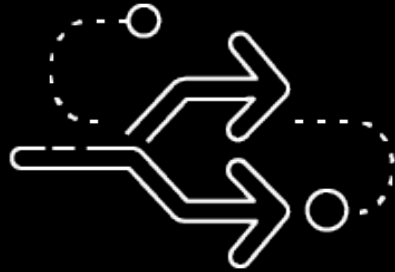
O serviço notifications escala, sem precisar escalar os demais serviços

# Arquiteturas desacopladas orientadas em eventos permitem aplicações resilientes e desenvolvimento rápido de novas features



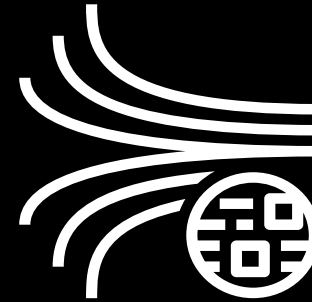


# Event ingestion



Event router

Filtra e encaminha eventos e  
“empurra” eventos para os  
consumidores subscritores



Event stream

Fluxo contínuo de eventos  
dos quais os consumidores  
podem puxar eventos

# O que mais vem por aí?



🍿 Próximo Ep #3 - Orquestração de Eventos com Step Functions! 🎬

📅 29/03/2023

🕒 17h30



Nosso [GitHub](#) está recheado de materiais adicionais!



YouTube

Inscreva-se em nosso [canal](#) do YouTube! Ative o sininho 🔔 das notificações e receba um alerta assim que algum novo conteúdo for publicado!



Siga também nossa [página](#) no LinkedIn!



in

/peterson-larentis



in

/jnagase