

# Arquitetura de microsserviços orientada a eventos

# Apresentações

# Adolfo Sabino



Telegram: @adolfosabino

Twitter: @azedoumarmita

- BCC pela UFPR
- PHP, Java, C e Python
- Atuando como agilista
- Músico amador

**olist**

# Luiz de Prá



Em tudo: @luizdepra

Blog: luizdepra.dev

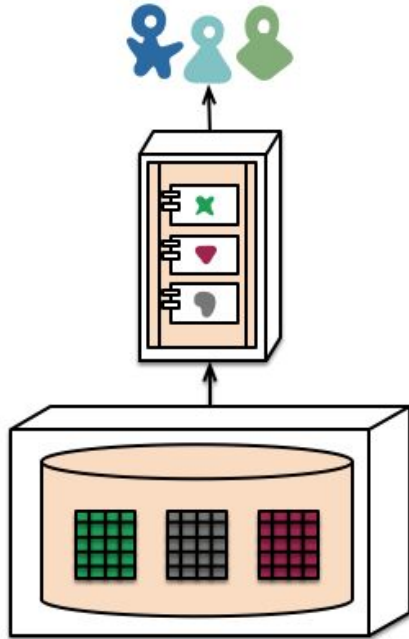
- BCC pela UFPR
- C, C++, C#, Python e Rust
- Dev back-end
- Gamedev

**olist**

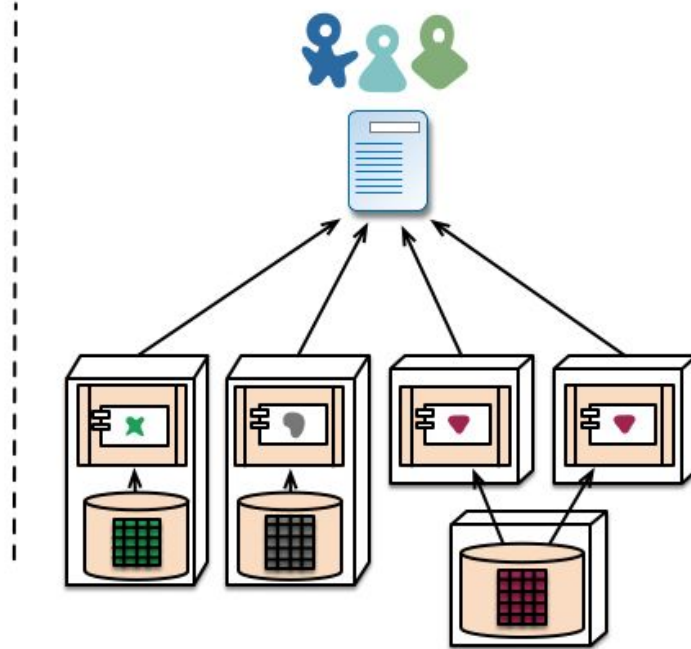
# Microserviços

olist

# Como definir um microserviço?



monolith - single database



microservices - application databases

# Por que microserviços?

# Modularidade

- Módulos com responsabilidades mais simples e claras
- Menor quantidade de código por componente
- Modificações pontuais no sistema e seu deploy tem impacto menor (trocar a roda com o carro andando)
- Mantém fácil a entrega contínua, mesmo com o sistema crescendo



# Resiliência

*"É quanto você aguenta apanhar e continua seguindo em frente"*

- BALBOA, Rocky

- Isolamento de falhas e contenção de danos.
  - *Single Point of Failure*

# Escalabilidade

- A possibilidade de escalar pontos específicos do sistema.
  - Escalabilidade seletiva - Somente algumas partes
  - Escalabilidade condicional - Somente em algumas circunstância
  - Escalabilidade on-demand - Sempre que precisar
- Permite uma gestão automatizada de recursos computacionais.

# Trade-offs - ou pontos negativos

- Gestão do sistema é mais complexa.
- Monitoramento é mais complexo.
- Problemas com fluxos de dados e consistência de informações.
- Mais difícil de prever impactos que uma alteração terá em outras partes do sistema.

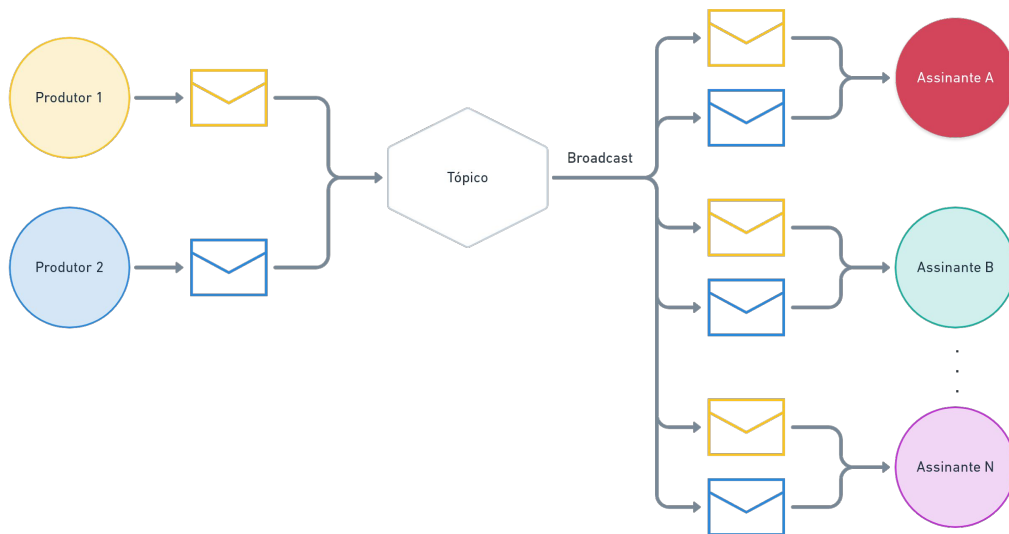
# Arquitetura baseada em eventos

# Arquitetura event-driven

- Alterações geram notificações (eventos)
  - Gerador de eventos
  - Canal de transmissão
  - Consumidores
- Feedback não é necessário ou esperado (async)
- Componentes do sistema desacoplados

# PubSub

- Publicadores
- Tópicos
- Assinantes



# Como fazemos no olist

olist

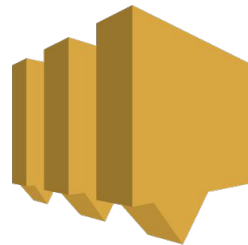
# Componentes Arquiteturais

- Mensageria
- Notificadores
- Consumidores

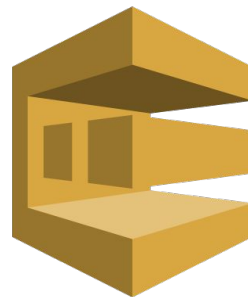


# Mensageria: tópicos & filas

- Tópico:
  - Recebe mensagens de publicador (publishers)
  - Repete a mensagem para todos os inscritos (subscribers)
- Fila
  - Recebe e armazena mensagens
  - Disponibiliza as mensagens para consumo
- Dead Queue

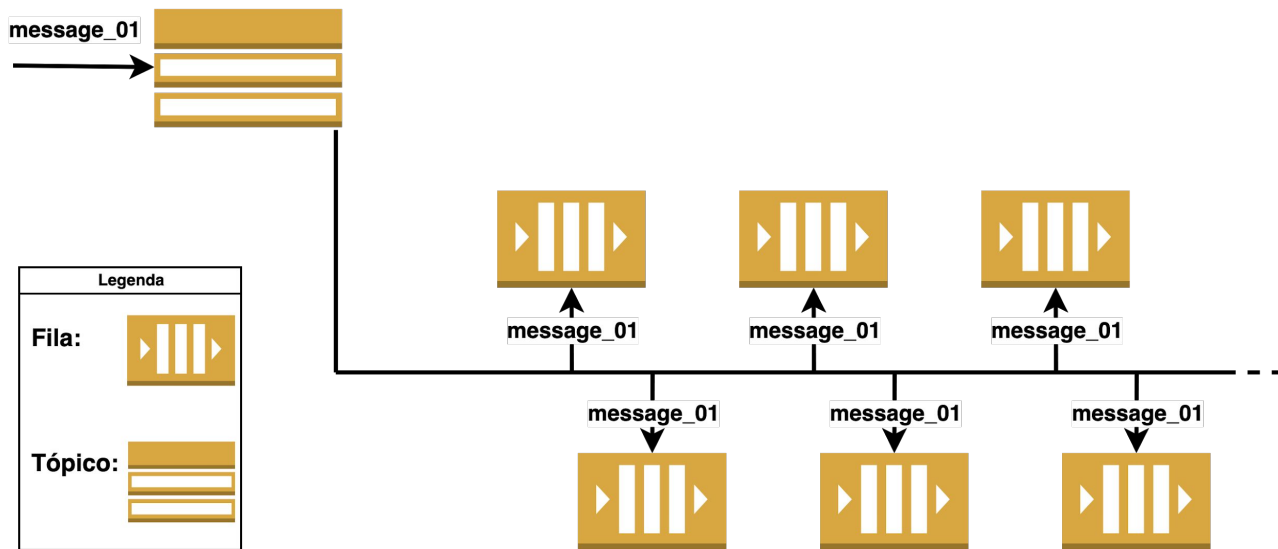


AWS SNS



AWS SQS

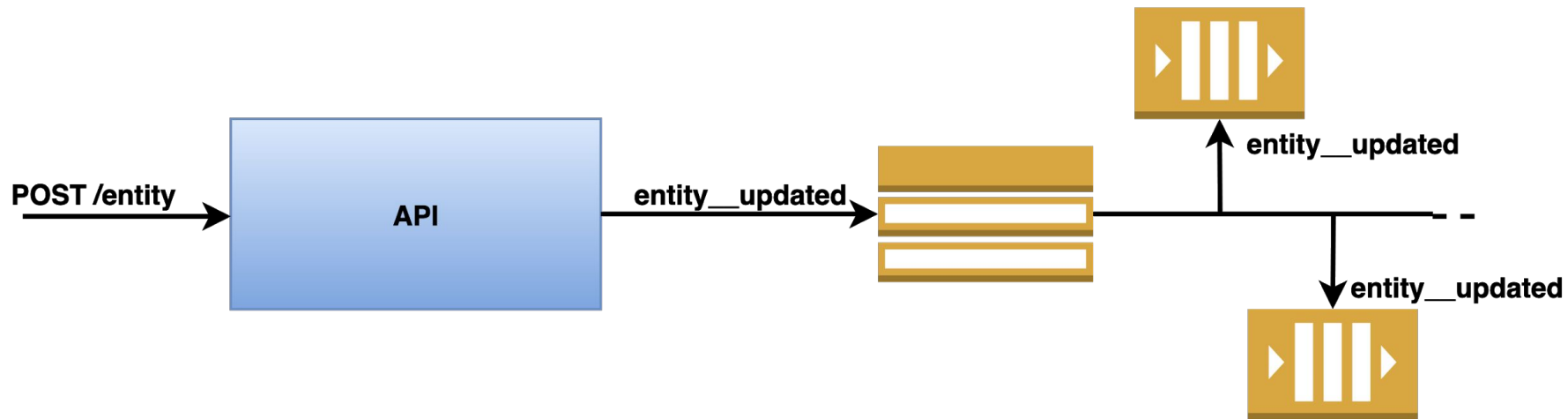
# Eventos globais com garantia de entrega de mensagens



# Notificadores / Publicadores

- API
- Webhook
- Scheduled Jobs

# APIs

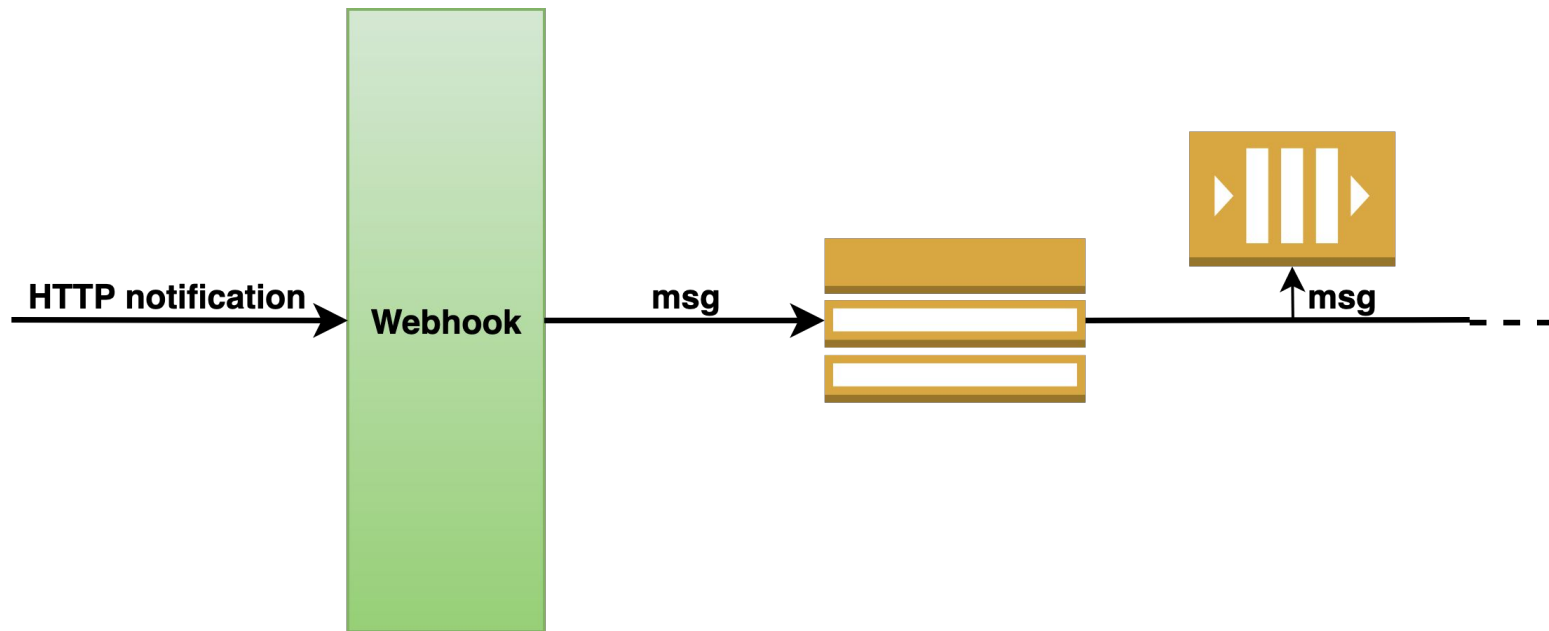


# Webhooks

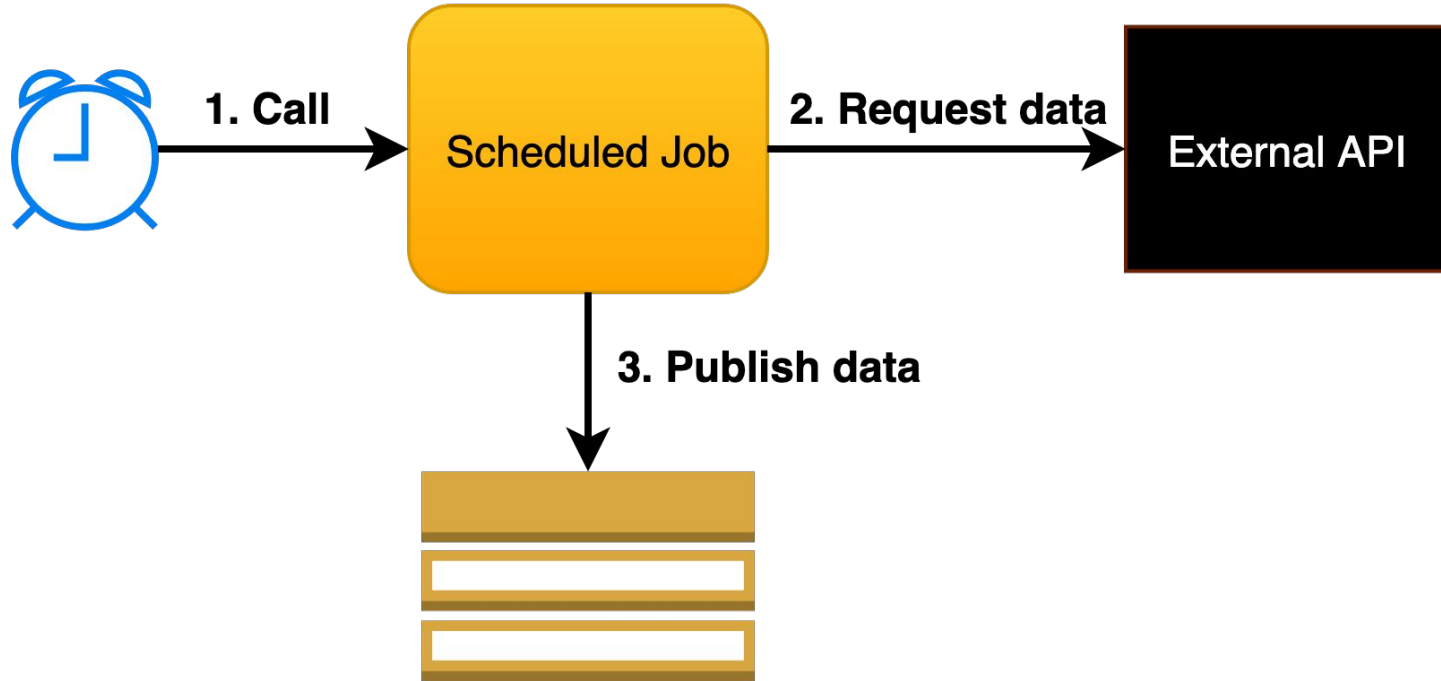


**olist**

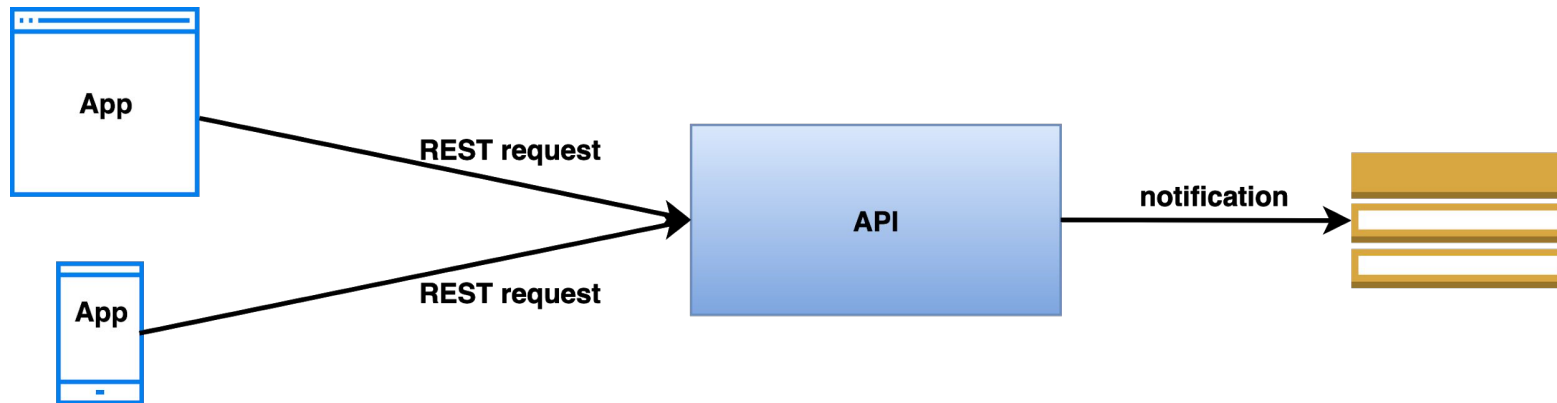
# Webhooks



# Scheduled Jobs



# Apps

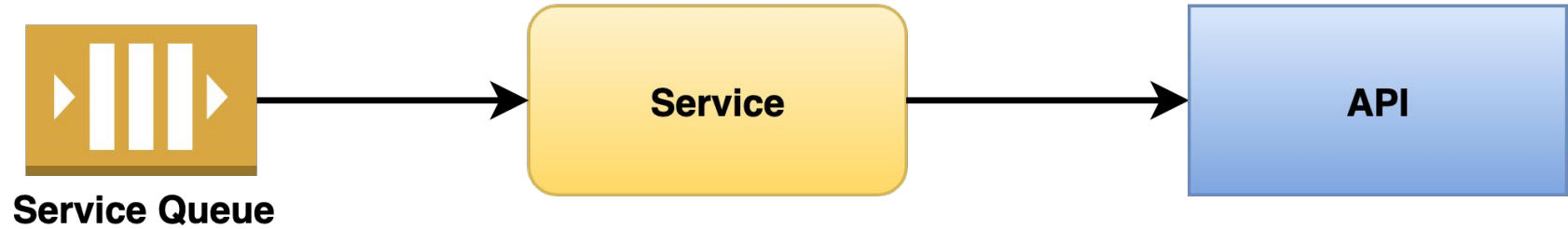




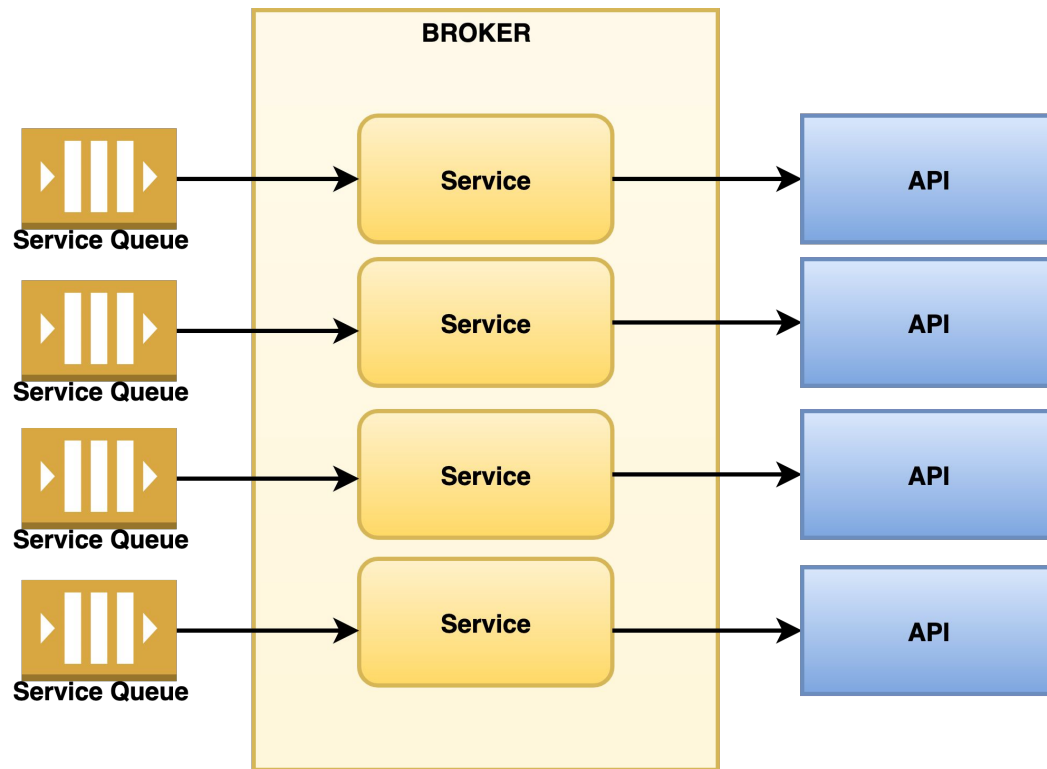
# Consumidores

- Serviços
  - Dequeueuer
  - Broker

# Dequeuer



# Broker



olist

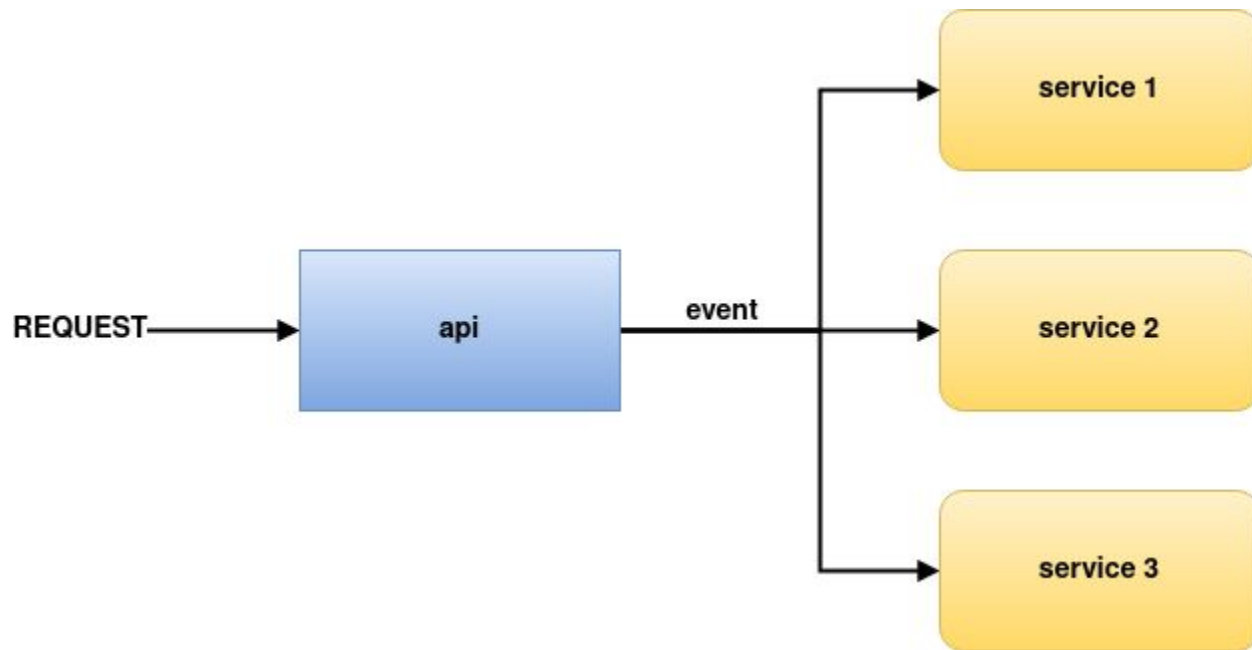
# Funções $\lambda$

- Podem assumir o papel de:
  - APIs, um endpoint por função
  - Serviços



# Cenários

# Um pub e vários subs



# Processamento de um evento gerando várias alterações

- É uma boa prática? Por que?
- E se uma request falhar no meio do processamento?

# Reprocessamento de DeadQueues

- Enviar mensagens da deadqueue para a fila quente.
- Dúvidas:
  - Por que temos mensagens em deadqueue? Bug? Instabilidade?
  - Todas as mensagens vão ser processadas?
  - E se algo voltar para a deadqueue?
  - A ordem do processamento das mensagens podem ser um problema? Event Sourcing?
  - Mensagens com conteúdos antigos podem ser um problema?

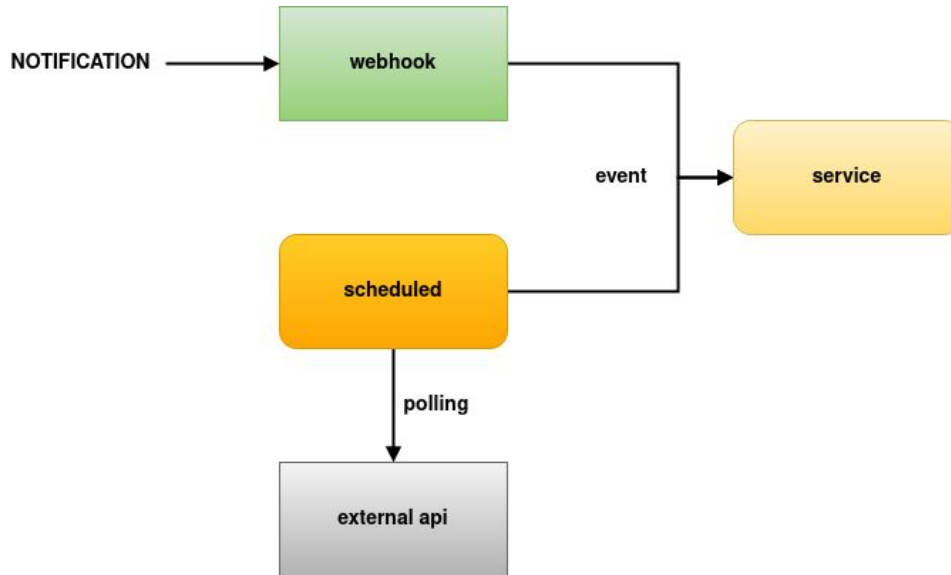


# Replicação de dados entre APIs

- Quando é necessário?
  - Cruzamento de APIs
  - Em interfaces ou em serviços?
  - Resposta rápida
- Problema: sincronia dos dados entre APIs

# Dependência com entidades externas

- Webhooks & Scheduled Jobs (polling)
- Garantir que nada foi perdido



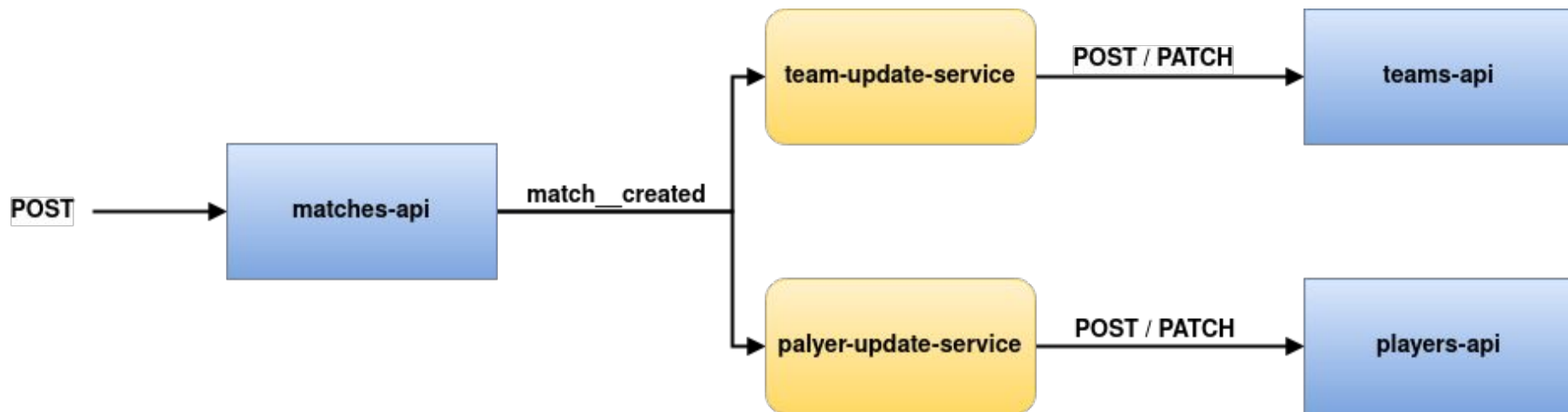
**Show me the code!**

# Campeonato Paranaense de Futebol

Repositório: <https://github.com/olist/techtalk-arquitetura>

- Componentes
  - API de Partidas (matches-api)
  - API de Times (teams-api)
  - API de Jogadores (players-api)
  - Serviços de Atualização
- Entidades
  - Match
  - Team
  - Player

# Arquitetura



olist

**Obrigado!**

**olist**

**Tá quase acabando!**

**olist**

**O que você achou do evento?**  
**Responda à pesquisa:**

**<http://bit.ly/38HUwkL>**



**olist**