



Arquitetura Orientada a Eventos

Aluno: Ronaldo Crispim

**Sistemas Distribuídos
Prof: Adriano Antunes Prates**



INTRODUÇÃO

A arquitetura orientada a eventos (EDA) é um padrão de arquitetura moderno criado com base em microserviços. Nessa arquitetura os serviços publicam, consomem ou encaminham eventos. Diferente das arquiteturas tradicionais orientadas a solicitações, a EDA promove um acoplamento fraco entre os serviços produtores e consumidores.



Algumas das Principais características do EDA...

Distribuída

Assíncrona

Escalável

Alto Desempenho

Eventos



Antes de tudo, o que é um evento?

Um evento é o registro imutável sobre algo que aconteceu em um sistema, como uma mudança de estado ou uma atualização, ele pode ser usado para acionar reações por outros componentes da aplicação sem os conhecimentos necessários uns dos outros.

Por exemplo: Pedro fez login em um sistema, isso é um evento que traduz uma mudança de estado da interface de login para o painel principal de uma aplicação/sistema.

Para a identificação de um evento, há algumas características a serem observadas:

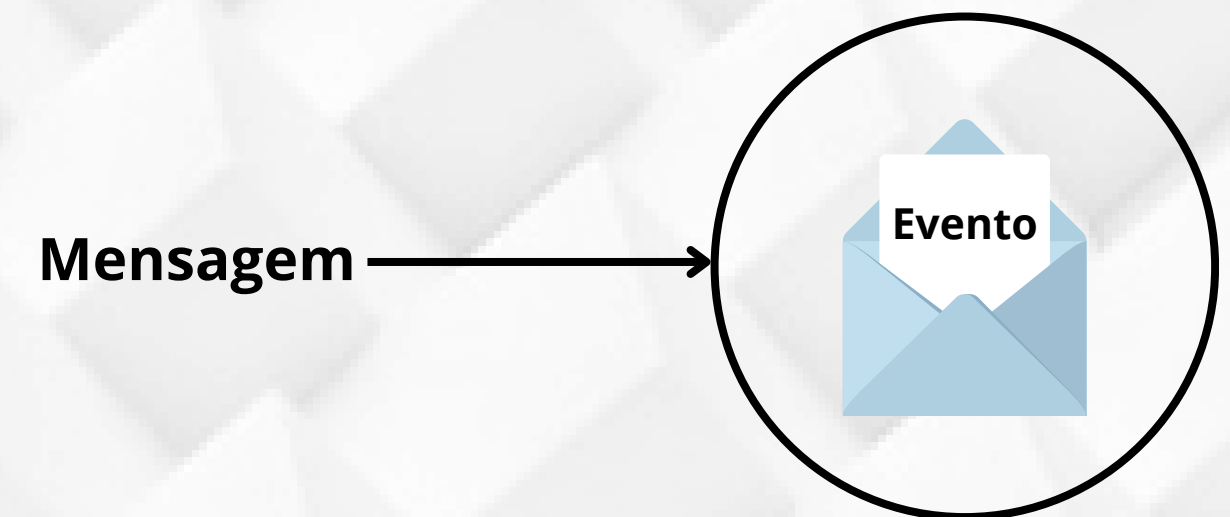
Registro de acontecimentos:

Imutável:

Acoplamento fraco:

Desacoplamento:

Pode conter dados:



Não confundir evento com mensagem!



Algumas das Principais características do EDA...

Distribuída

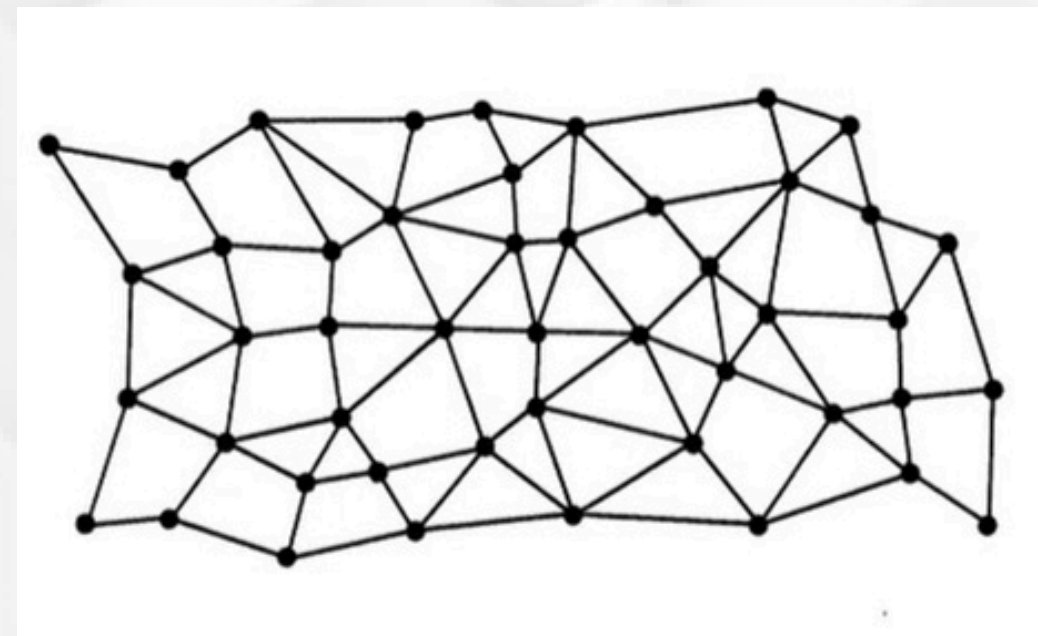
Assíncrona

Escalável

Alto Desempenho

Eventos ✓

A distribuição e assincronia do **EDA** se deve por ela usar a arquitetura Pub/Sub que distribui as mensagens por meio de produtores, consumidores e canais de eventos, como um broker, o que proporciona a comunicação assíncrona e o desacoplamento.

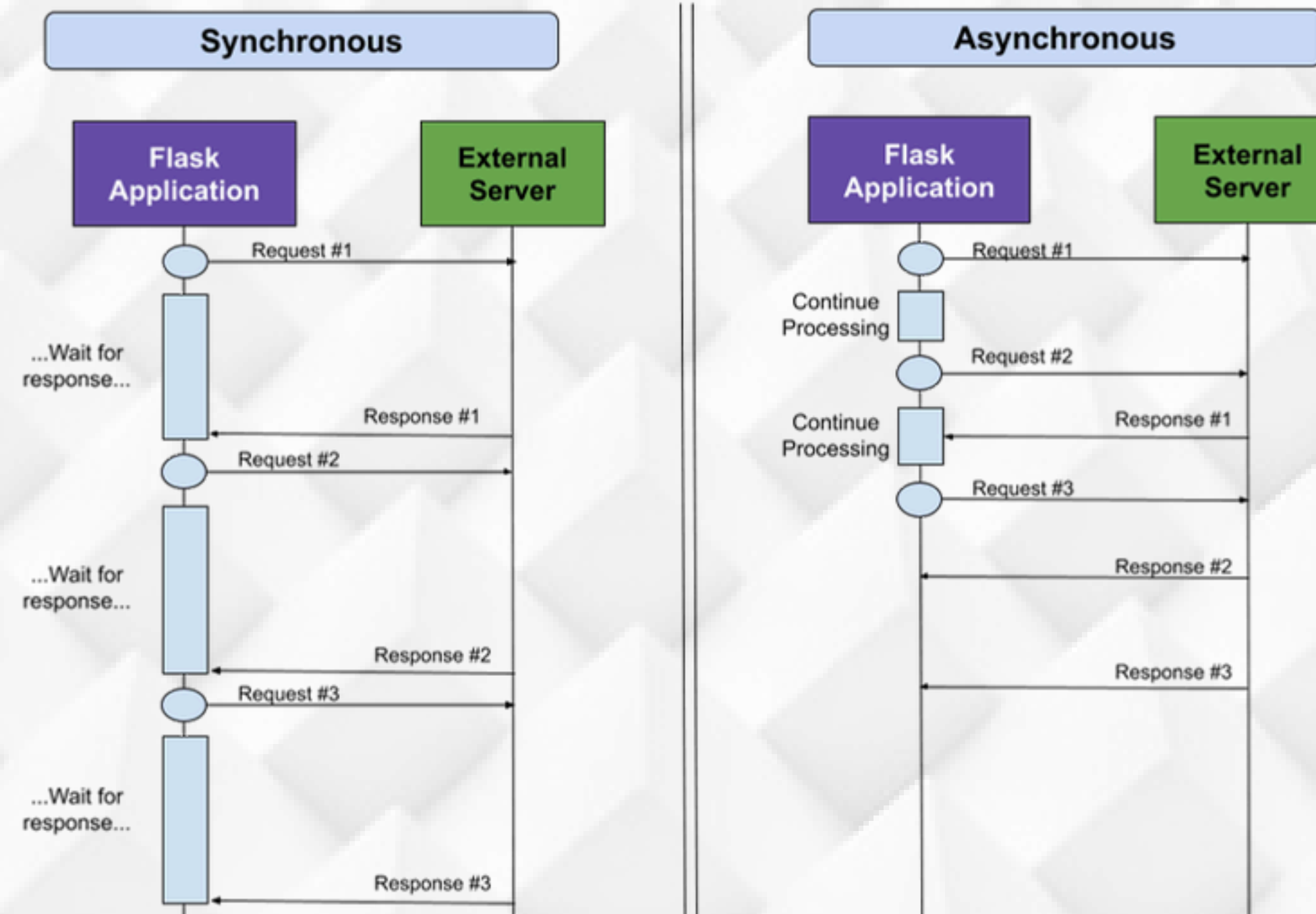




Publisher e Subscriber

Em sistemas distribuídos baseados em nuvem, normalmente, os componentes do sistema precisam fornecer informações a outros componentes à medida que os eventos acontecem.

Mensagens assíncronas são uma maneira eficiente de separar os remetentes dos consumidores e evitar o bloqueio do remetente até receber uma resposta.

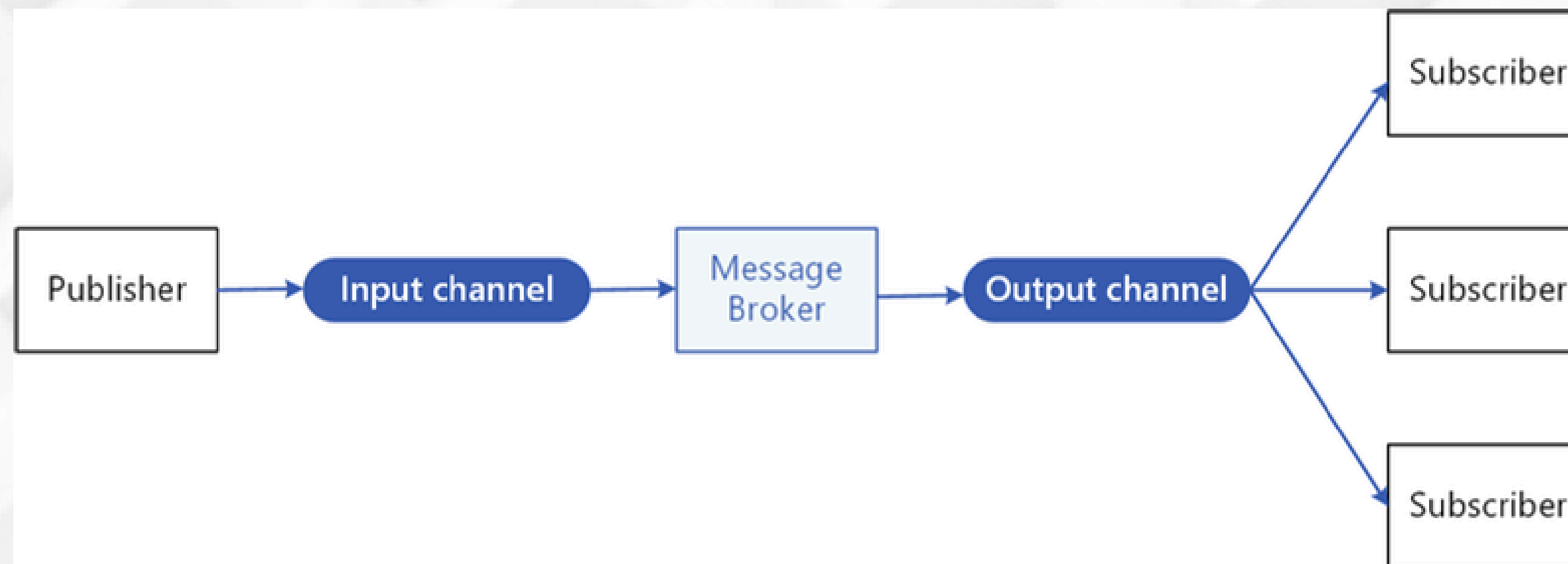




Publisher e Subscriber

Esse padrão utiliza um canal de entrada(input) para eventos que o publicador quer publicar como mensagens para seus assinantes, e um canal de saída(output) que é por onde o Broker ou Mensageiro vai mandar as mensagens para cada um dos assinantes.

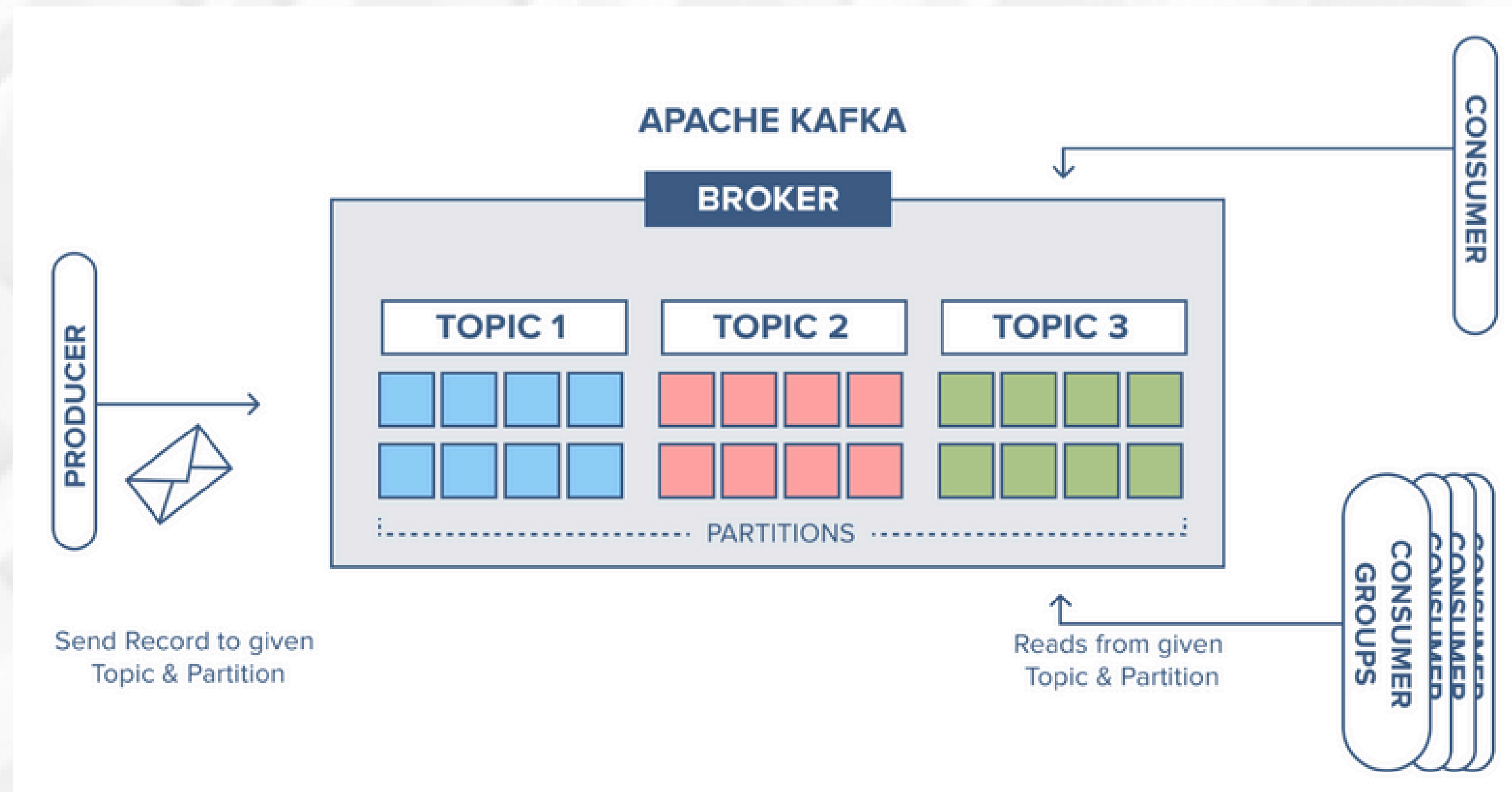
Por haver diversos canais de saída sendo eles para cada um dos assinantes faz essa ser a arquitetura adotada na EDA quando se trata de assincronia.





Brokers comumente utilizados

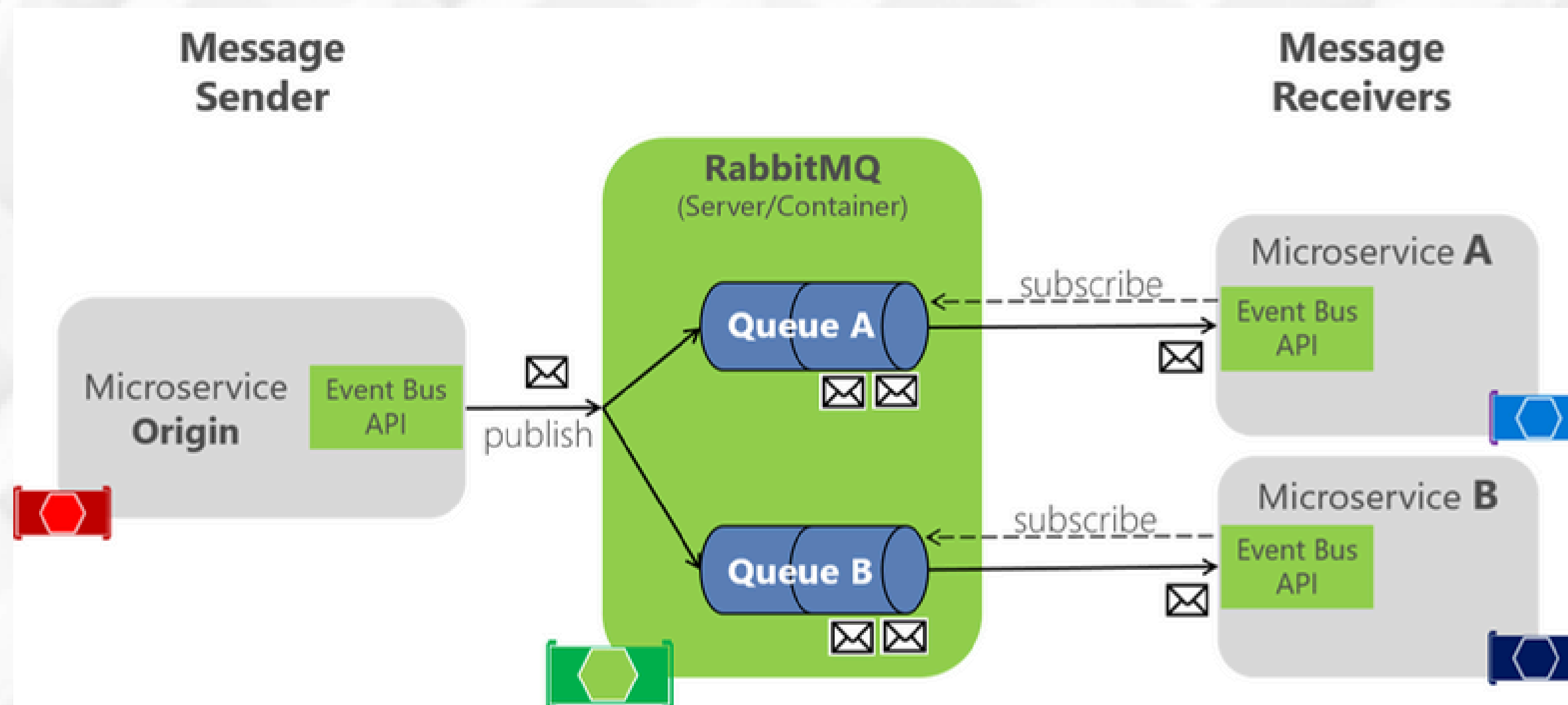
O Kafka utilizado quando o desenvolvedor deseja trabalhar com o armazenamento das mensagens por um longo período de tempo e ter a capacidade de ter um relatório de cada mensagem de evento.





Brokers comumente utilizados.

RabbitMQ é mais utilizado pela sua capacidade de rotear mensagens de forma eficiente e garantir a entrega correta dos eventos, mesmo em cenários de falhas.





Algumas das Principais características do EDA...

Distribuída ✓

Assíncrona ✓

Escalável

Alto Desempenho

Eventos ✓

Com os subsistemas funcionando de forma assíncrona e operando de forma independente suas determinadas funções, esses subsistemas podem falhar sem afetar o sistema como um todo, dessa maneira a EDA consegue ser altamente escalável.

Por que ela é de alto desempenho?

- Processamento em Tempo Real
- Utilização Otimizada de Recursos
- Velocidade Aumentada
- Tolerância a Falhas



Diferenças entre a EDA (AO a Eventos) e a SOA (AO a Microserviços)

Comunicação:

EDA

Assíncrona (Pub/Sub)

SOA

Síncrona (Requisição/
Resposta)

Acoplamento:

Desacoplado (publicador
e assinante não
precisam se conhecer)

Acoplado (o publicador
precisa conhecer o
serviço chamado)

Tolerância a Falhas:

Maior resiliência; se
um assinante falhar, o
publicador não é
afetado

Uma falha em um
serviço pode impactar
diretamente seus
dependentes

Escalabilidade:

Alta escalabilidade,
pois os componentes
podem ser escalados
independentemente

Limitada por dependências
síncronas



Não é por que há diferença entre as arquiteturas que não podemos utilizá-las em conjunto.

Uma arquitetura de microsserviços pode utilizar uma comunicação orientada a eventos para melhorar a resiliência e o desacoplamento entre os serviços.

Microsserviços definem a estrutura: Como a aplicação é dividida em componentes independentes.

EDA define a comunicação: Como os componentes conversam de forma assíncrona e reativa.

Exemplo: Em um e-commerce, o serviço de "Pedidos" pode publicar um evento "PedidoCriado". Os serviços de "Pagamento", "Estoque" e "Notificação" podem então assinar este evento e executar suas respectivas lógicas de forma independente e paralela.



Bom, mas quando utilizar essa arquitetura?

Distribuída ✓

- Quando vários subsistemas devem processar os mesmos eventos.

Assíncrona ✓

- Quando o processamento em tempo real com o atraso mínimo de tempo é essencial.

Escalável ✓

- Quando necessário alto volume e alta velocidade de dados, como com IoT.

Alto Desempenho ✓

Eventos ✓

- Quando precisa desacoplar produtores e consumidores para metas independentes de escalabilidade e confiabilidade.



https://www.sap.com/brazil/products/technology_platform/what-is-event-driven-architecture.html

<https://aws.amazon.com/pt/what-is/eda/>

<https://learn.microsoft.com/pt-br/azure/architecture/guide/architecture-styles/event-driven>

<https://learn.microsoft.com/pt-br/azure/architecture/patterns/publisher-subscriber>

<https://www.youtube.com/watch?v=78VJCs-8DQ>

Fontes:



Obrigado !