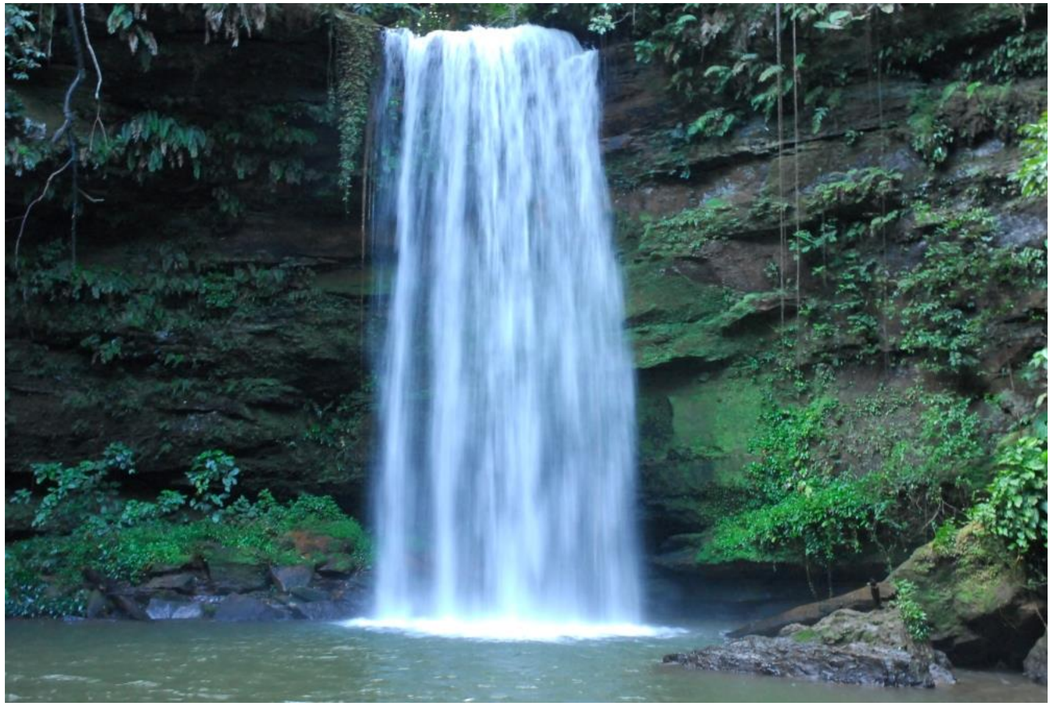
Streaming

Streaming

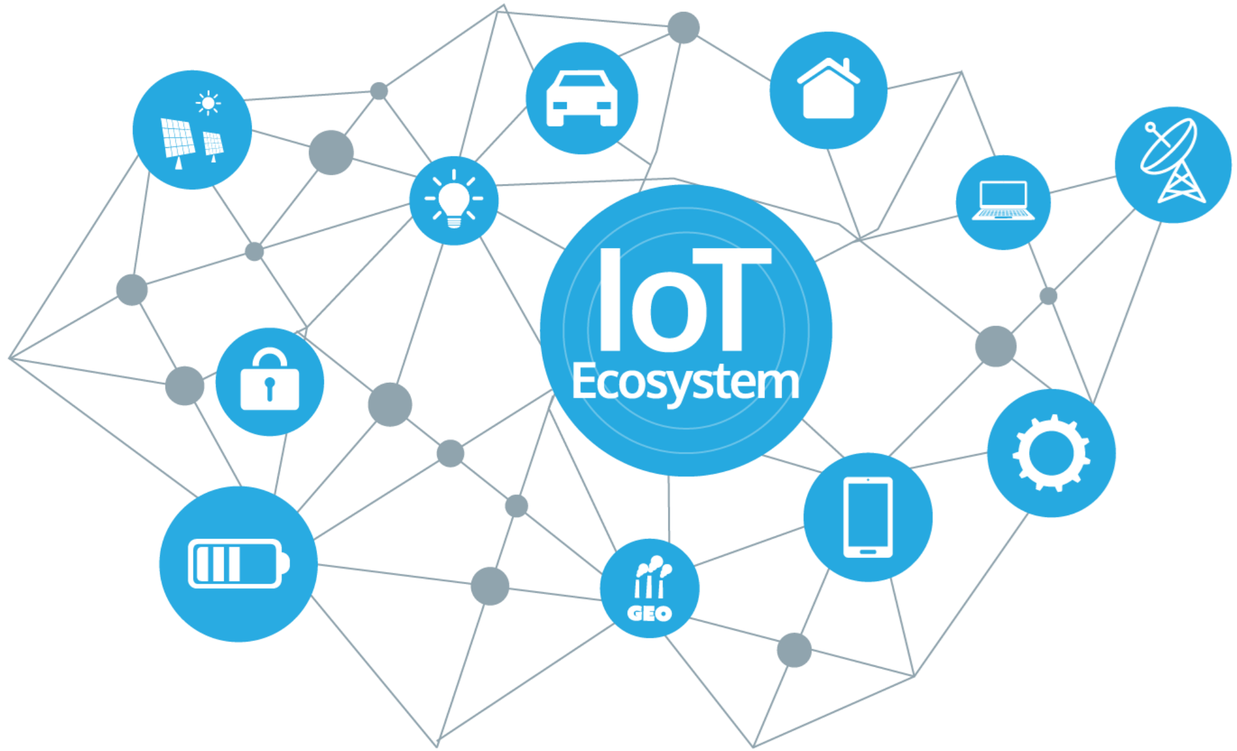
- Fluxo contínuo (contínuo ≠ constante).



Streaming de dados

- Fluxo contínuo de dados.

Spotify®



Streaming de dados: Exemplos

- Sensores (IoT)
- Tráfego de rede
- Registros de call center
- Tendências em redes sociais
- Serviços de áudio e vídeo
- Análise de log
- Estatísticas de sites web

Tipos de streaming de dados

Dados de texto: web, log

Dados relacionais: tabelas, transações

Dados semi-estruturados: XML, json

Dados em grafo: redes sociais

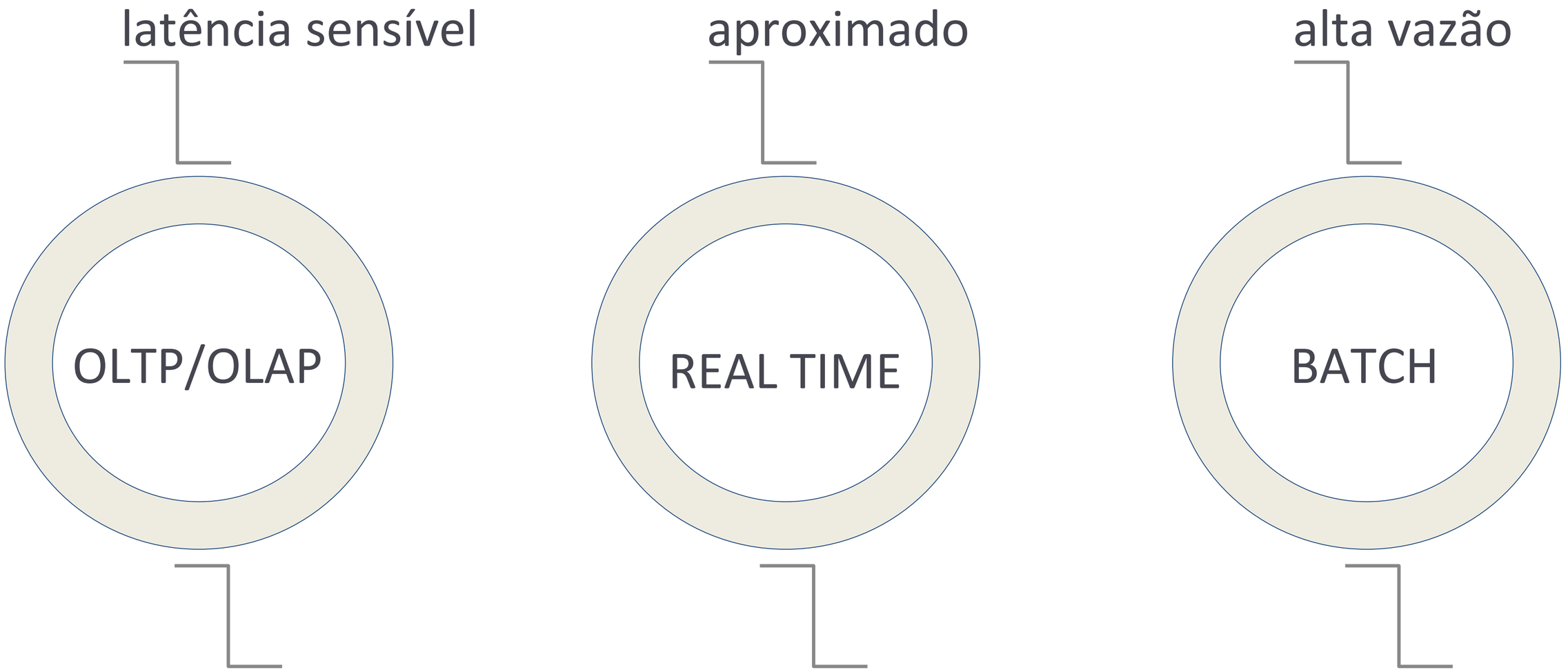
Dados de mobilidade: coordenadas geográficas x tempo

Etc.

O que é tempo real?

Milissegundos, segundos, minutos?

O que é Tempo Real?



O que é Tempo Real?

REAL TIME TRENDS

Emerging break out trends in Twitter (in the form #hashtags)

REAL TIME CONVERSATIONS

Real time sports conversations related with a topic (recent goal or touchdown)

REAL TIME RECOMMENDATIONS

Real time product recommendations based on your behavior & profile

REAL TIME SEARCH

Real time search of tweets with a budget < 200 ms

Fonte: Real-Time Analytics with Apache Storm - https://www.udacity.com/course/ud381

Problemas em streaming

1. Como **obter** os dados a partir de várias fontes em um cluster em tempo real?
2. Como **processar** esses dados?

Apache Kafka

Apache Kafka - o que é

- *O que é o Apache Kafka?*
- *“Apache Kafka é uma plataforma distribuída de mensagens e streaming”.*

Apache Kafka

- Você **produz** uma mensagem.
- Essa mensagem é **anexada** em um tópico.
- Você então **consome** essa mensagem.

Por que usar?

- *“Se você quer mover e transformar um grande volume de dados em tempo real entre diferentes sistemas, então Apache Kafka pode ser exatamente o que você precisa”.*

Apache Kafka

- Sistema de mensagens
  - Distribuído
  - Com alta vazão (*throughput*)
    - De geração (publicação) e leitura (sub-inscrição)
- Principais casos de uso:
  - Agregação de log
  - Processamento em tempo real
  - Monitoramento

Apache Kafka

- Originalmente desenvolvido pelo LinkedIn.
- Implementado em scala/Java.
- *Producers* & *Consumers.*
- Mensagens são associadas a tópicos, os quais representam um stream específico.
  - Logs web
  - Dados de sensores
- *Consumers* se inscrevem em um ou mais tópicos.

Kafka: conceitos

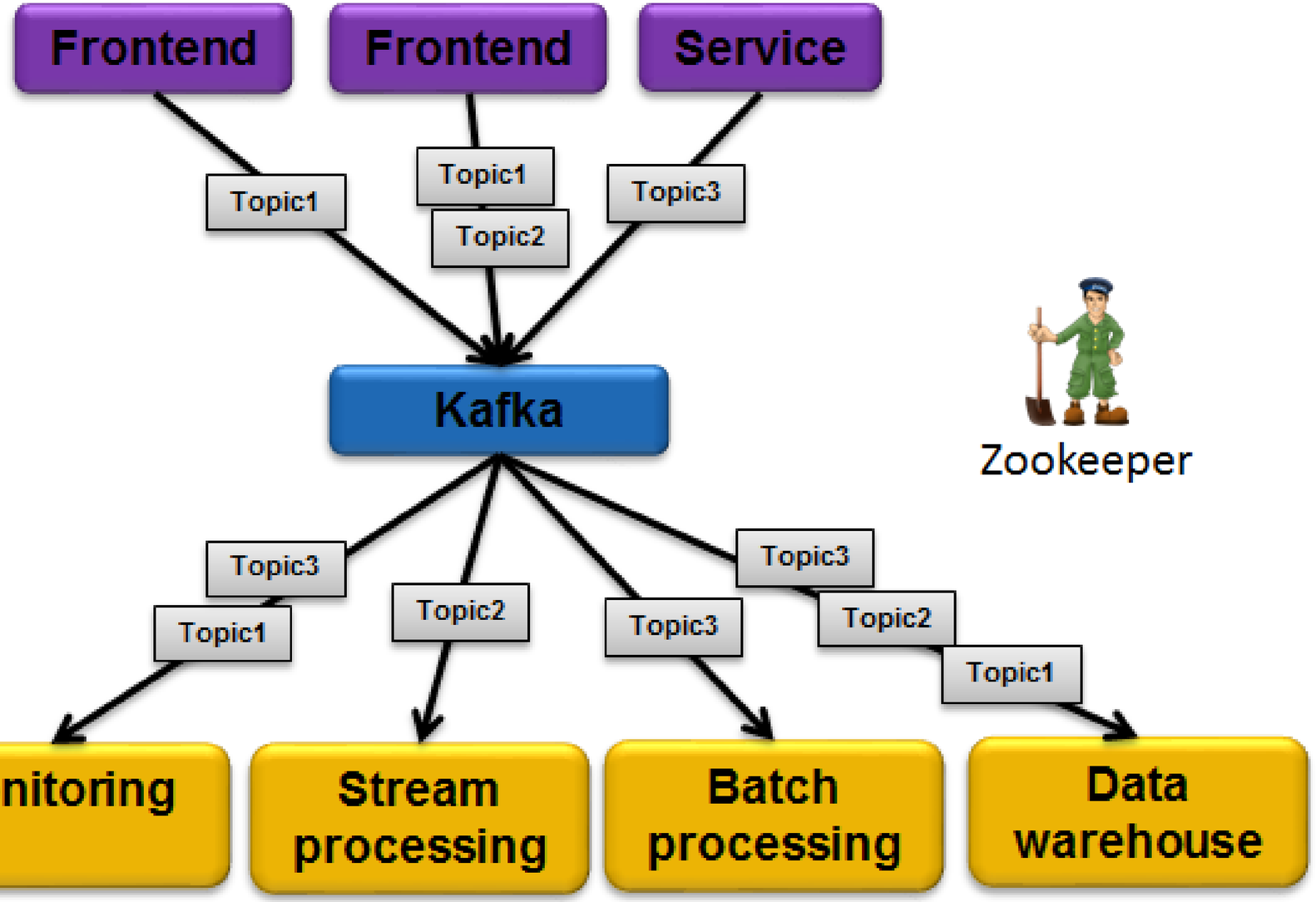
Producers

Broker

Consumers

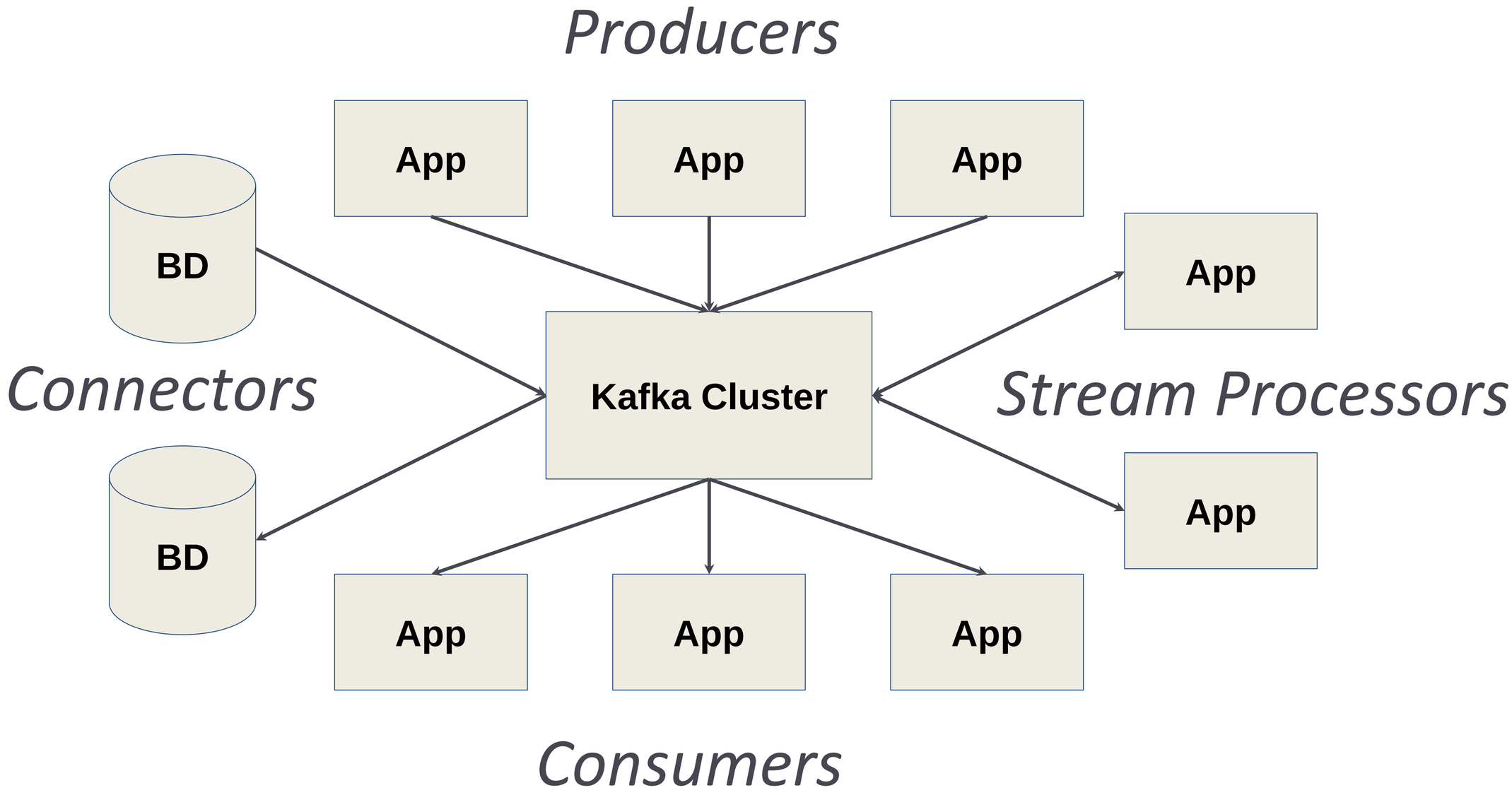
Push

Pull



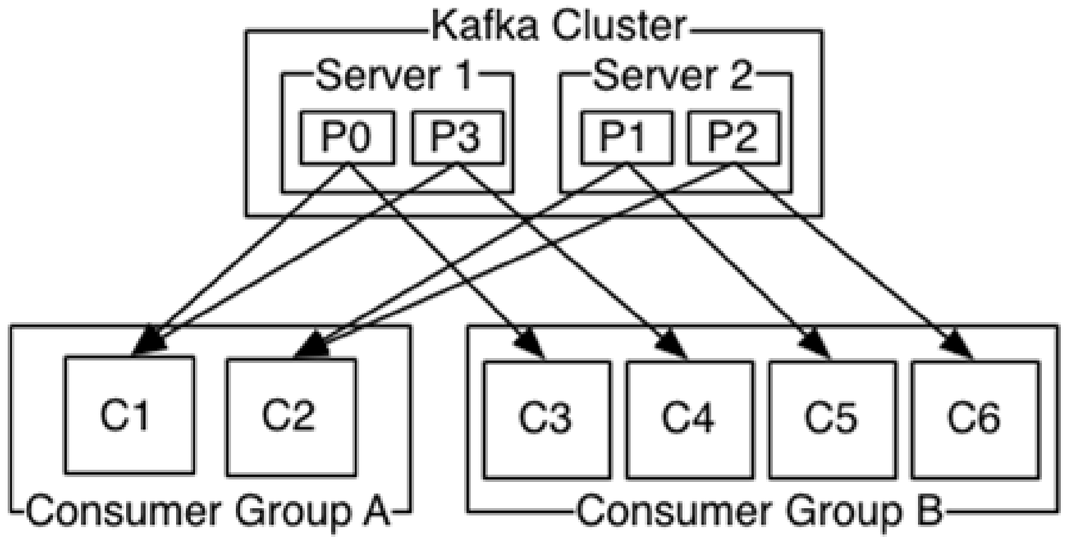
Zookeeper

Kafka: arquitetura



Kafka: escalabilidade

- Kafka pode ser distribuído entre muitos processos em vários servidores.
- *Consumers* também podem ser distribuídos.
- Tolerante a falhas.



Fonte: https://kafka.apache.org/intro.html

Kafka: pontos a considerar

- Simples sistema de mensagens, não de processamento.
- Não vive sem o **Zookeeper**, o qual pode se tornar um gargalo quando o número de tópicos/partições é muito grande (>>10000).
- Não otimizado para latências de milissegundos.