DIGITAL INNOVATION ONE

Apache Kafka

TQI

Klérisson Paixão
Engenheiro de Software

José Parreira Filho
Engenheiro de Software

Mais sobre nós

Klérisson:

- PhD (2019), MSc (2009), BSc (2006)
- Experiência:
  - Finanças, Telecom, Imobiliário ...
  - Internacional: Índia e França
  - PUC (Data Science) e UNIUBE

klerissonpaixao

José:

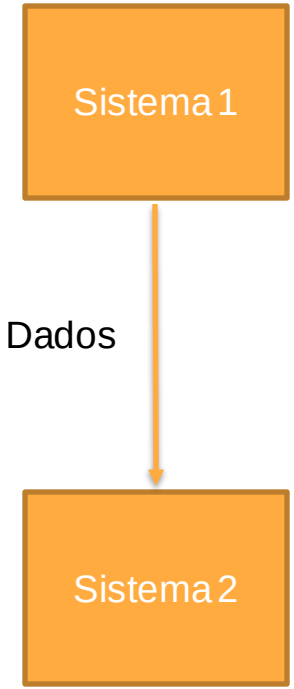
- Bacharel (2007)
- Experiência:
  - Finanças, Telecom, Sade...
  - Linguagens: Java e Kotlin

DIGITAL
INNOVATION
ONE

Kafka

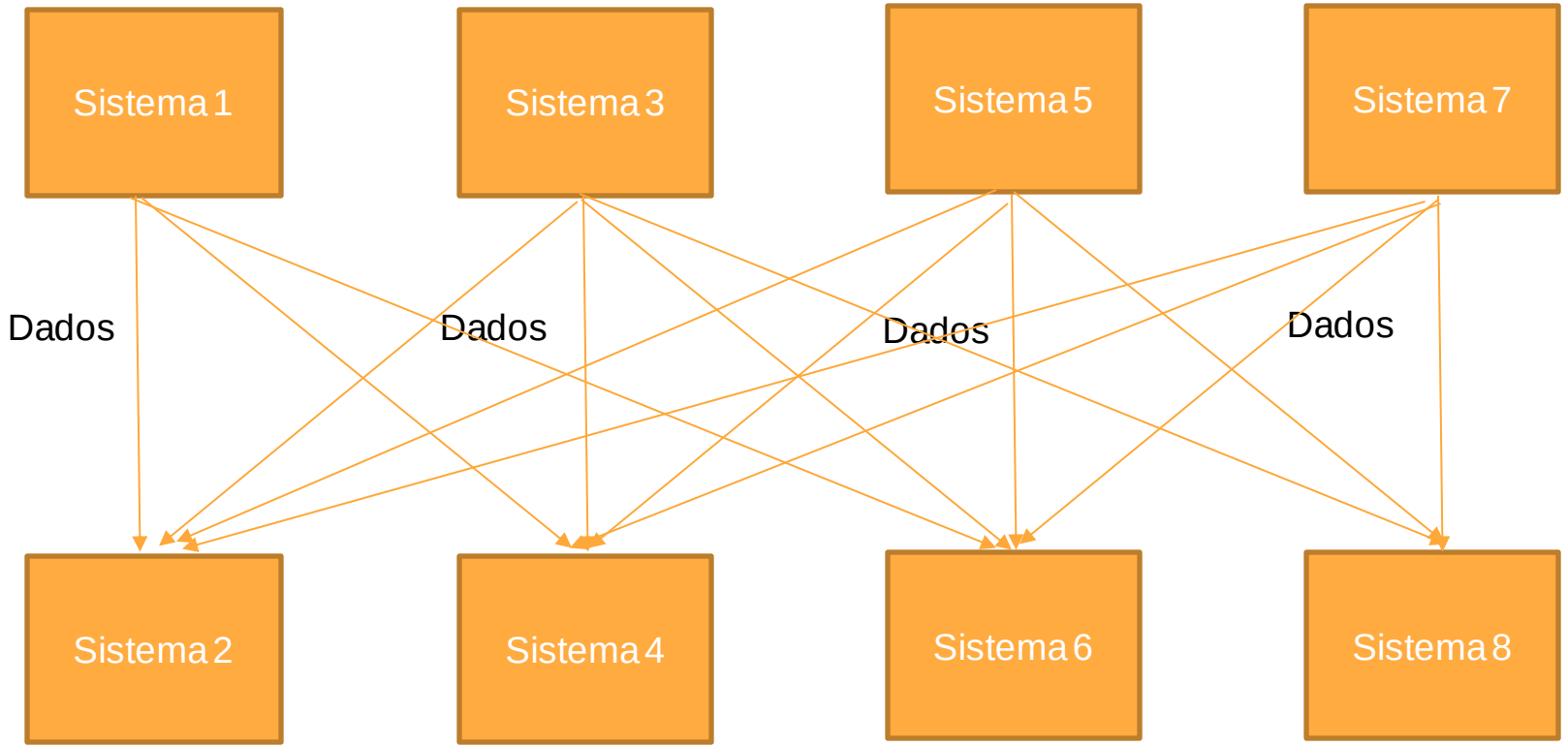
Em ~5 minutos

Como sistemas funcionam



DIGITAL INNOVATION ONE

Como sistemas funcionam



Problemas corportativos com essa arquitetura

- Com "apenas" 4 sistemas fonte e 4 sistemas alvo você precisaria desenvolver 16 integrações!

- Cada integração apresenta dificuldades (diversão) 😜

1. Protocolo: Como os dados serão transportados (TCP, HTTP, REST, FTP, JDBC ...)
2. Formato: Como os dados serão interpretados (Binário, CSV, JSON, Avro ...)
3. Esquema e evolução: Como os dados estão formatados e poderam mudar

- Cada sistema fonte terá aumentos de conexões, sobrecarga...

Problemas corportativos com essa gravidade

- Com "a… você preci…

- Cad…

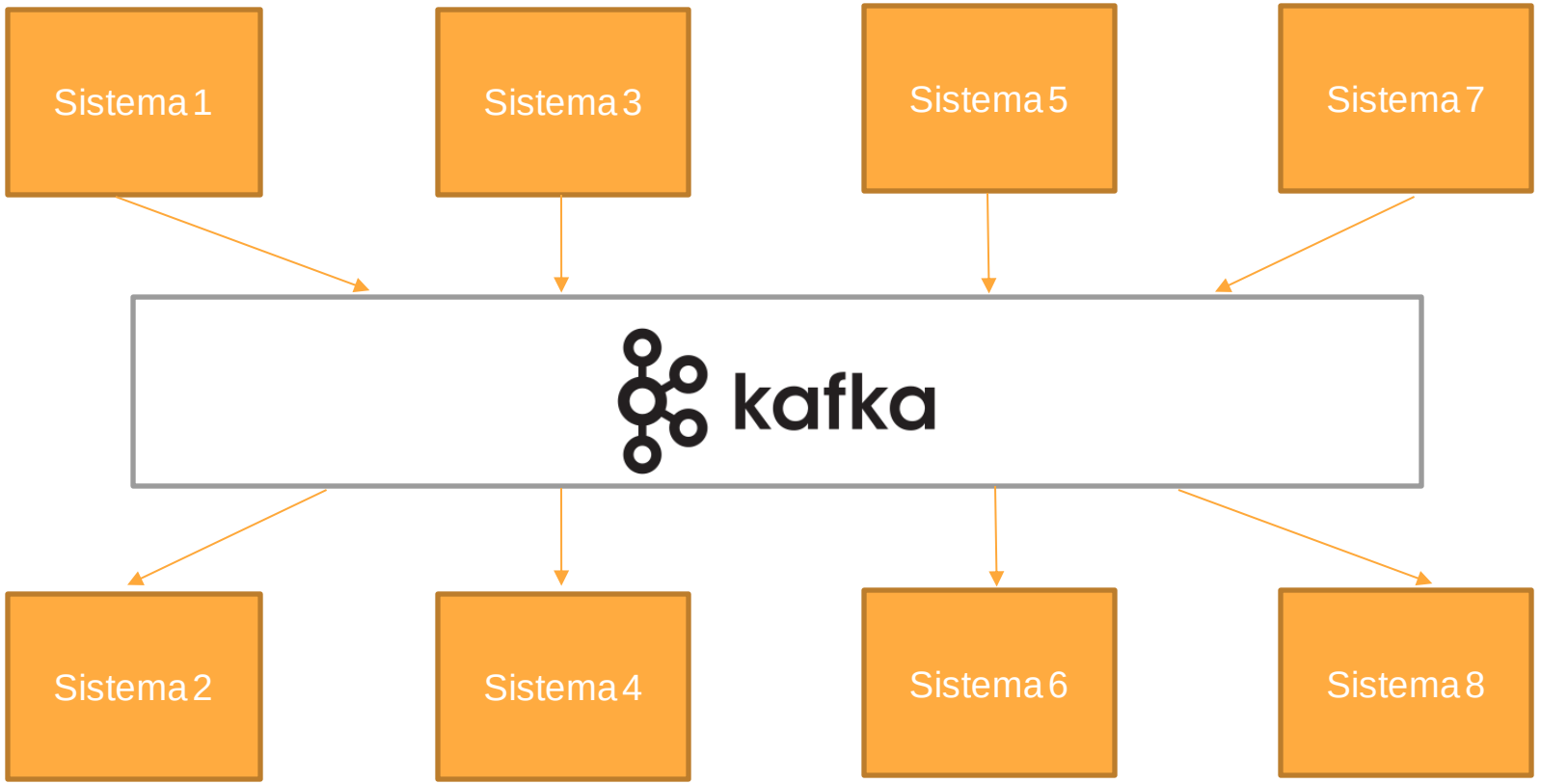
1. Proto… REST, FTP, JDBC ...)
2. Formato: Co… , CSV, JSON, Avro ...)
3. Esquema e evolução… formatados e poderam mudar

- Cada sistema fonte terá aumentos de conexões, sobrecarga...

Como resolver (mitigar) esses problemas?

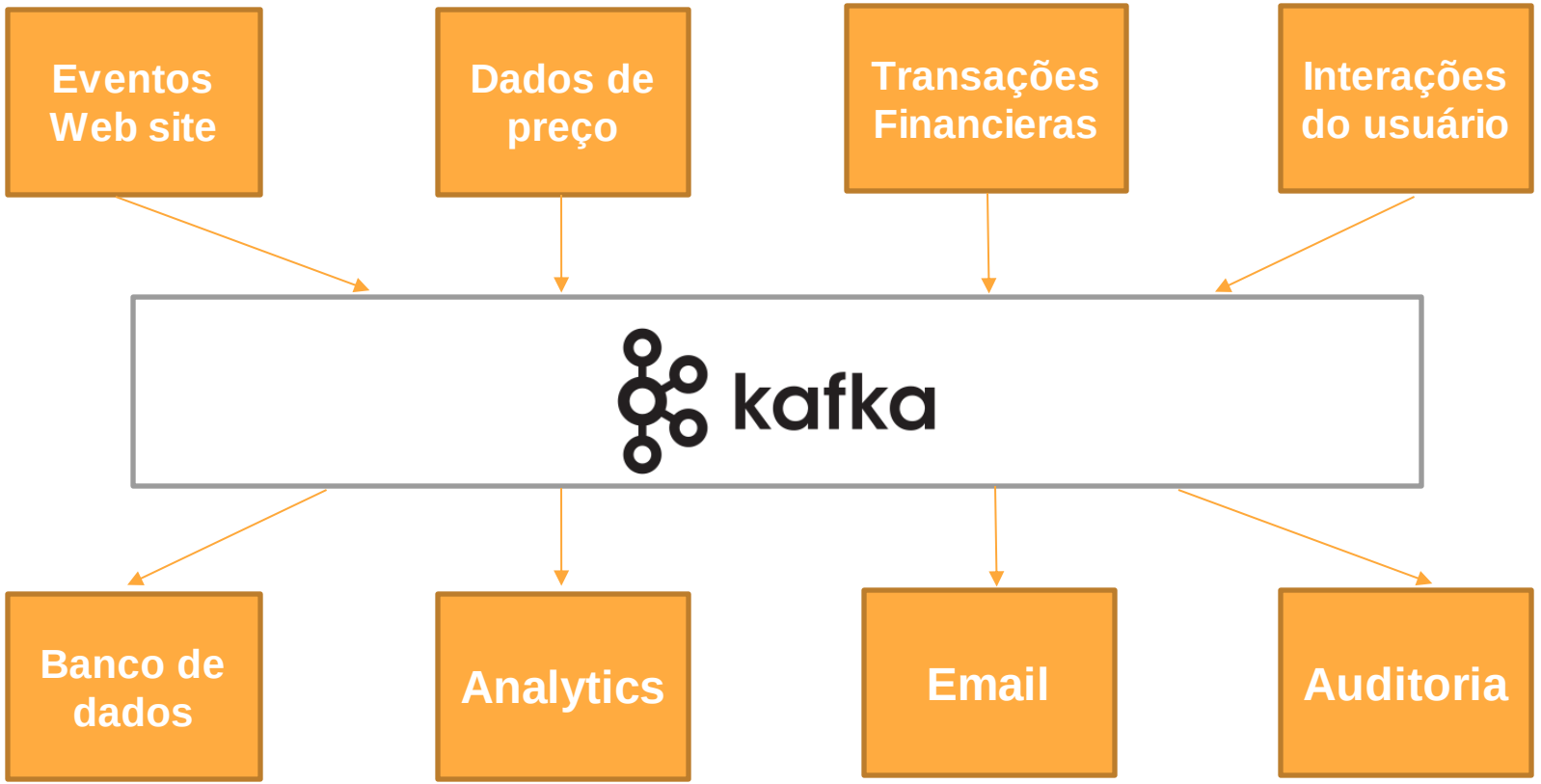
DIGITAL INNOVATION ONE

Desacoplando dados e sistemas



DIGITAL INNOVATION ONE

Desacoplando dados e sistemas



Por que Apache Kafka

- Criado pelo LinkedIn, agora Open Source (Apache)
- Arquitetura distribuída, resiliente, tolerante a falhas
- Escalabilidade horizontal:
    - Suporta centenas de "brokers"
    - Suporta milhões de mensagens por segundo

- Alto desempenho: tempo real (latência menor que 10 ms)
- 35% da Fortune 500 utilizam Kafka

Usos comuns do Kafka

- Sistemas de mensageria
- Rastreamento de atividades (eventos)
- Processamento de fluxos de dados (Stream)
- Desacoplamento de sistemas dependentes
- Integração com ecosistemas de Big Data (Spark, Flink, Storm, Hadoop, …)

DIGITAL INNOVATION ONE

Usos comuns do Kafka

- **Netflix** utiliza Kafka para ofertar recomendações em tempo real
- **Uber** utiliza Kafka para juntar dados de usuários, viagens e motoristas para prever demanda e sugerir preços
- **LinkedIn** utiliza Kafka para prevenir "spam", coletar interações de usuários.

Kafka é somente usado como mecanismo de transporte!

DIGITAL INNOVATION ONE

Kafka

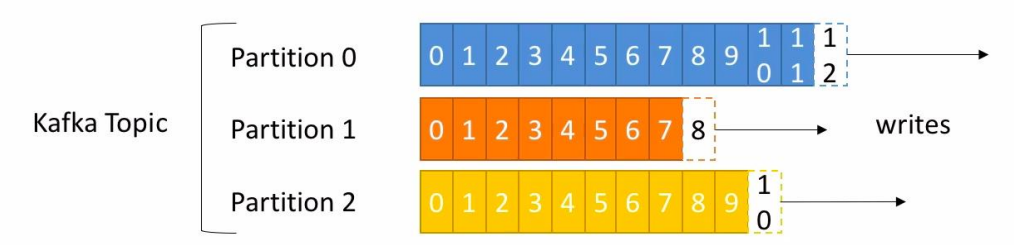
Teoria e fundamentos

Tópicos, Partições e "Offsets"

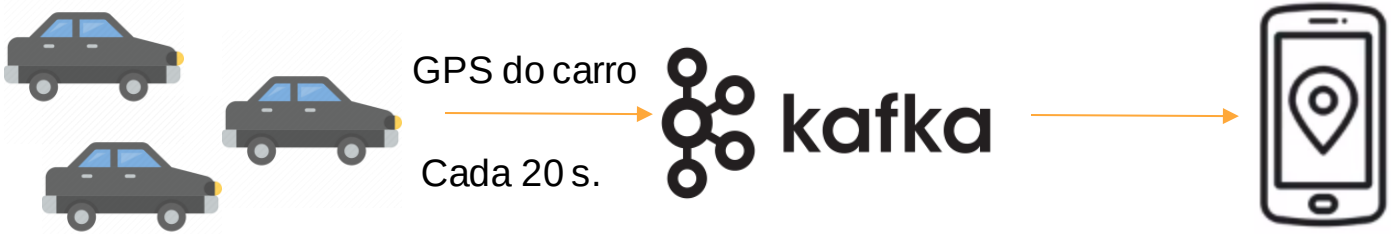
- **Tópicos**: fluxo específico de dados (stream)
  - Basta lembrar de uma tabela em um banco de dados
  - Você pode ter quantos tópicos quiser
  - Cada tópico precisa de um nome

- **Tópicos** são divididos em **Partições**
  - Partições são ordenadas
  - Cada mensagem de uma partição tem um ID (**Offset**)

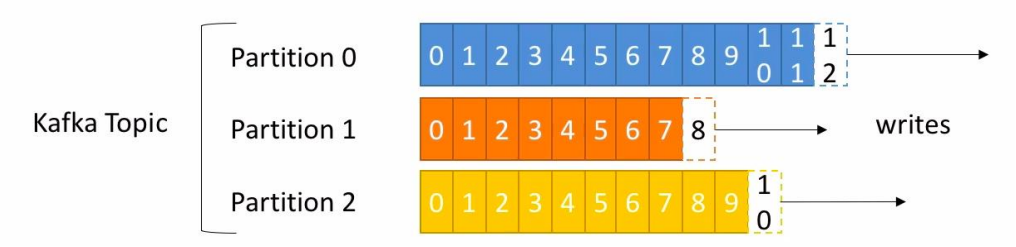
Tópicos, Partições e "Offsets"



Ex.: uber_gps

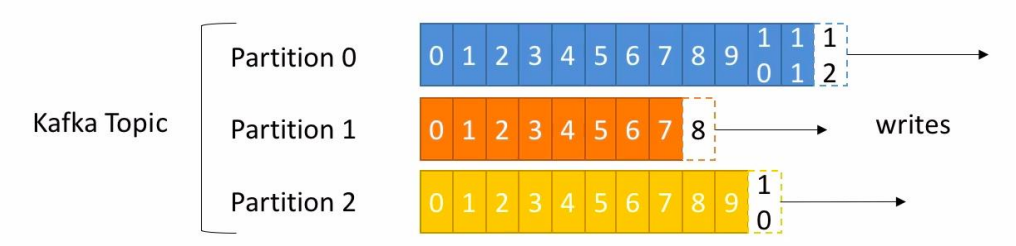


Tópicos, Partições e "Offsets"



- Offset é específico para cada partição
  - Ex.: offset 3 da partição 0 não representa o mesmo dado do offeset 3 da partição 1
- Ordenação dos offsets é somente garantida na partição
  - Não entre partições!
- Os dados são mantidos por tempo limitado no Kafka
  - Mas os offsets são mantidos

Tópicos, Partições e "Offsets"



- Dados escritos não podem ser mudados
  - Imutabilidade

- Os dados são armazenados randomicamente nas partições
  - Exceto se uma chave for fornecida

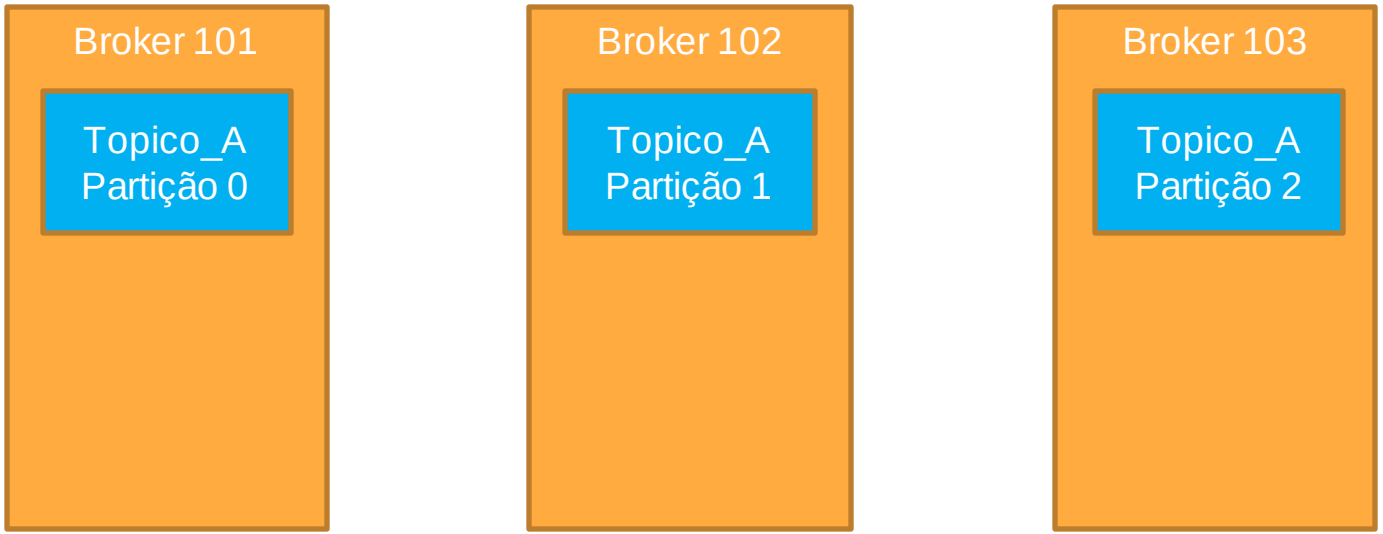
"Brokers"

Brokers = Servidores, Máquinas

- Um "cluster" Kafka é composto por múltiplos brokers
- Cada broker é identificado por um ID (numeral inteiro, ~~meu_serv~~)
- Cada broker contém certas partições de um tópico
- Conexão ao broker = conexão ao cluster
- Número mágico de brokers inicial = 3

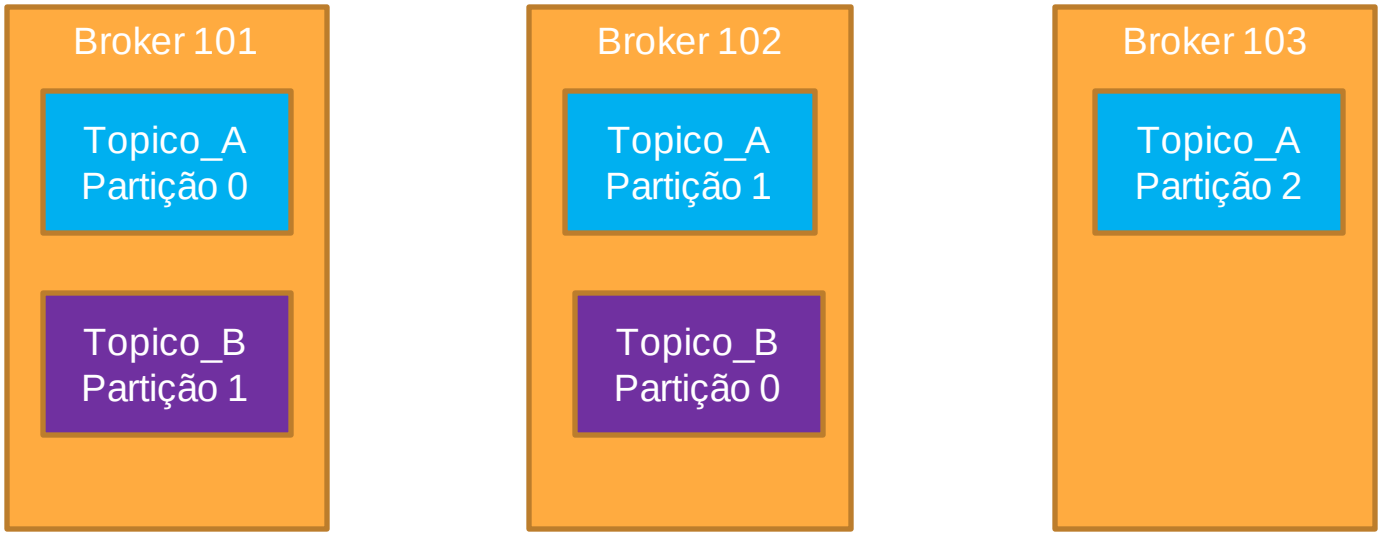
Brokers e Tópicos

Ex.: **topico_A** com 3 partições



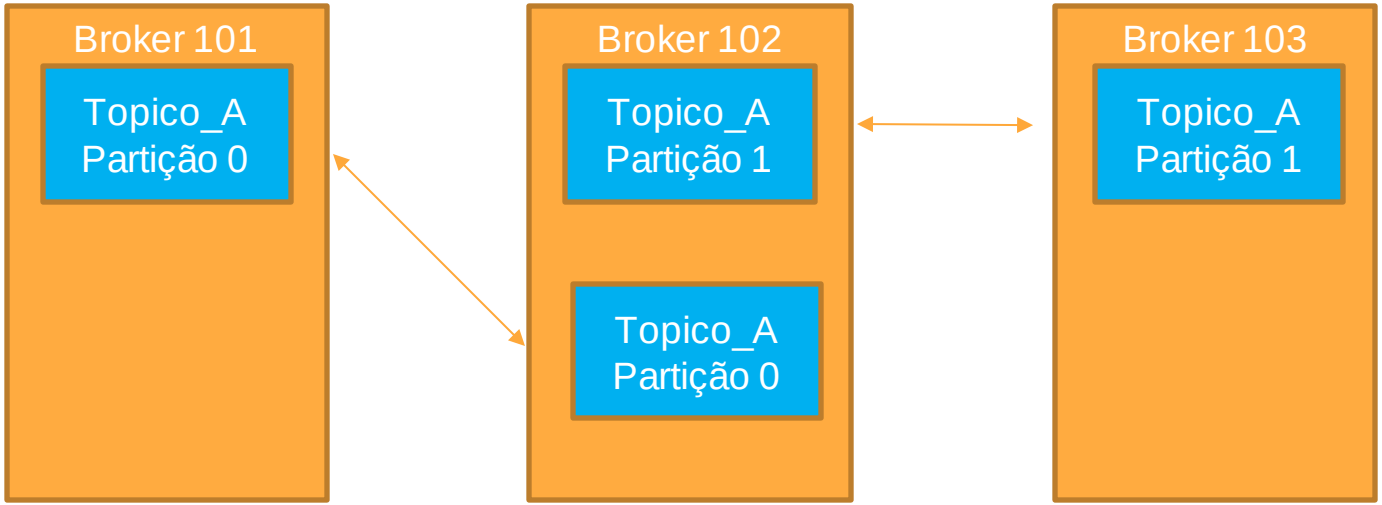
Brokers e Tópicos

Ex.: **topico_A** com 3 partições
topico_B com 2 partições



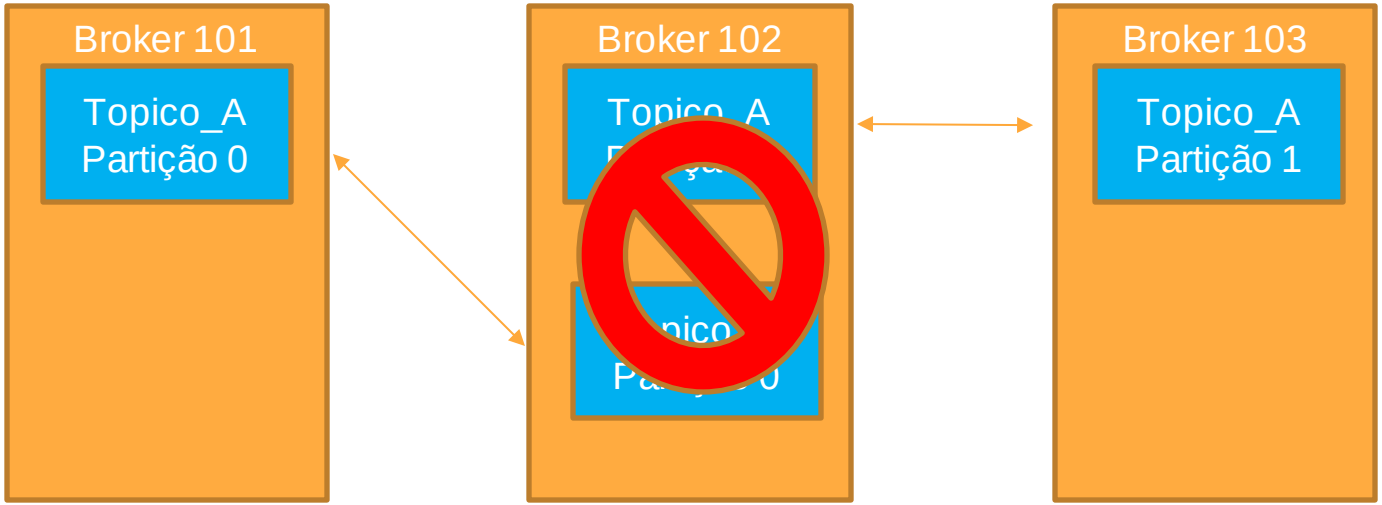
Replicação de tópicos

- Tópicos precisam ter fator de replicação > 1
- Número mágico = 3
- Ex.: topico_A com 2 partições e replicação = 2



Replicação de tópicos

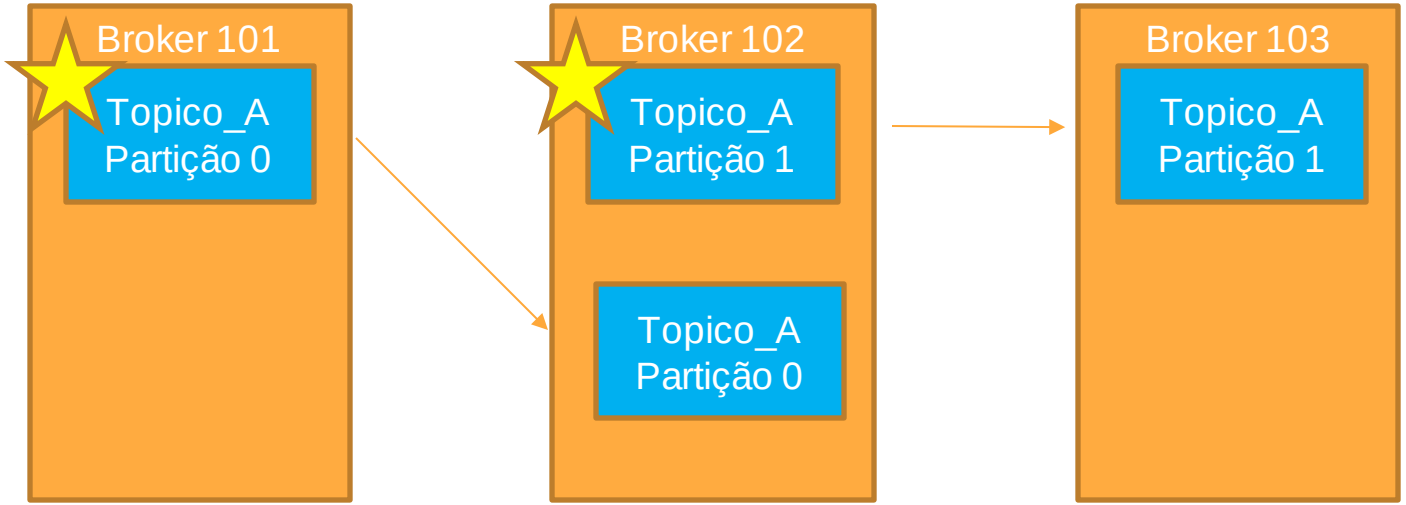
- Broker 102... "PLOFT"
- Resultado: ainda temos os dados!



Replicação de tópicos

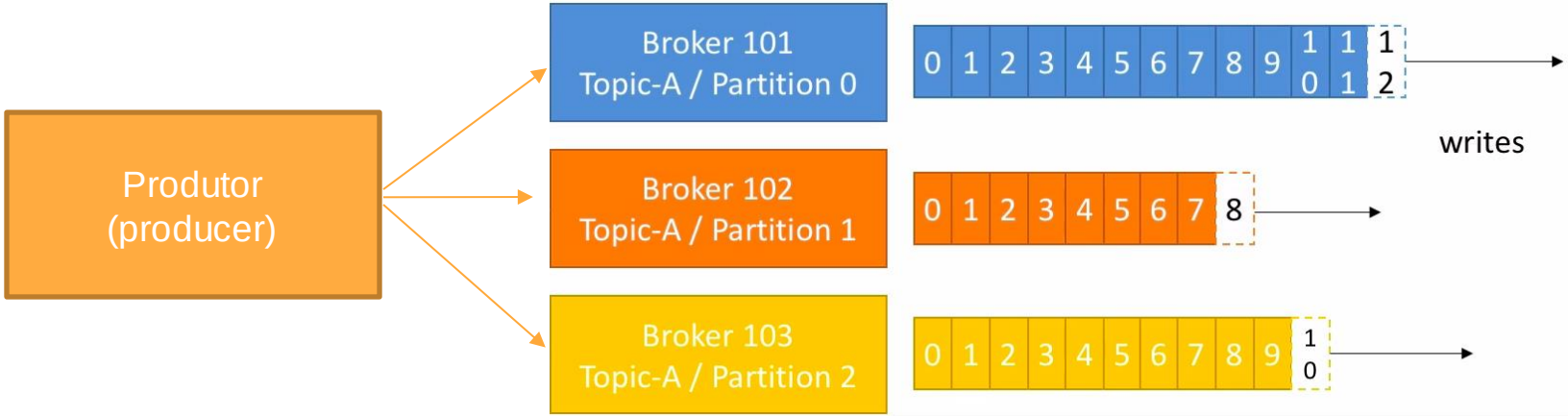
Líder de uma partição:

- Apenas um broker pode ser o líder de uma partição
- Somente esse líder pode processar os dados dessa partição
- Os outros brokers servem para sincronização

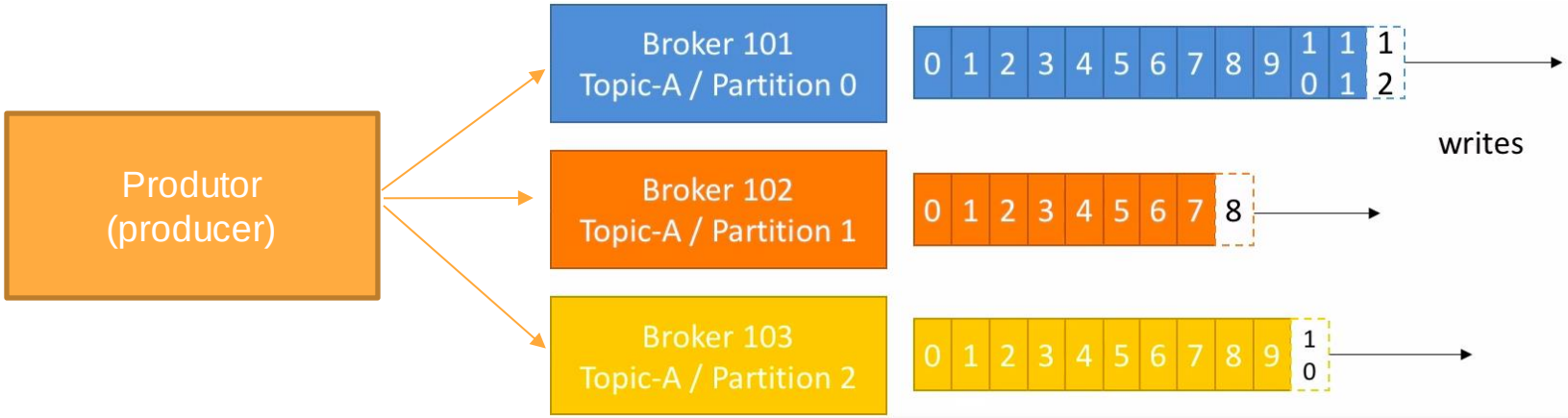


Produtores

- Escrevem dados nos tópicos (nas partições)
- Automaticamente sabem em broker e partição deve escrever
- Em caso de falha de broker, produtores se recuperam



Produtores



Produtores podem receber confirmação da escrita do dado:

1. acks=0: Produtor não espera confirmação alguma
2. acks=1: Produtor vai esperar o líder confirmar
3. acks=all: Confirmação de líder e réplicas

Produtores e mensagens com ID (keys)

Produtores podem enviar uma chave (ID) com os dados

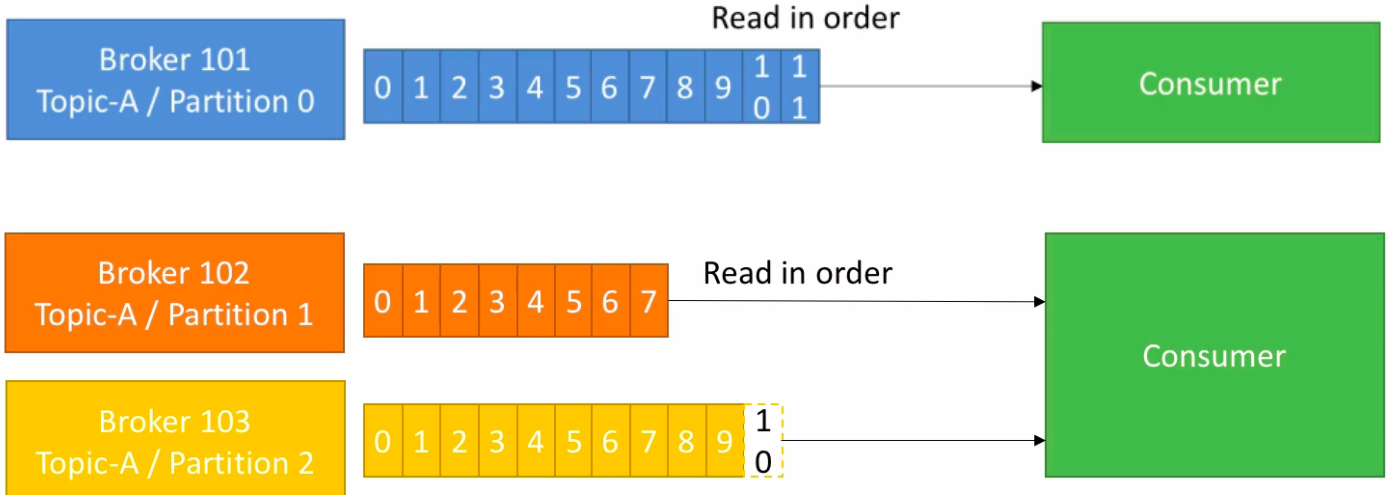
- Essa chave pode ser qualquer caracter (string, número, …)
- Se a chave for nula, os dados serão enviados em ordem para os brokers: 101, 102, 103... (round robin)
- Se uma chave for enviada, então todas as mensagens daquela chave sempre irão para a mesma partição
- Logo, uma chave caso queira manter uma ordem

Consumidores

- Consumidores leêm dados dos tópicos
- Sabem de qual broker precisam ler os dados
- Em caso de falha, consumidores conseguem se recuperar
- Dados são lidos em ordem em cada partição (Não entre partições)

DIGITAL INNOVATION ONE

Consumidores



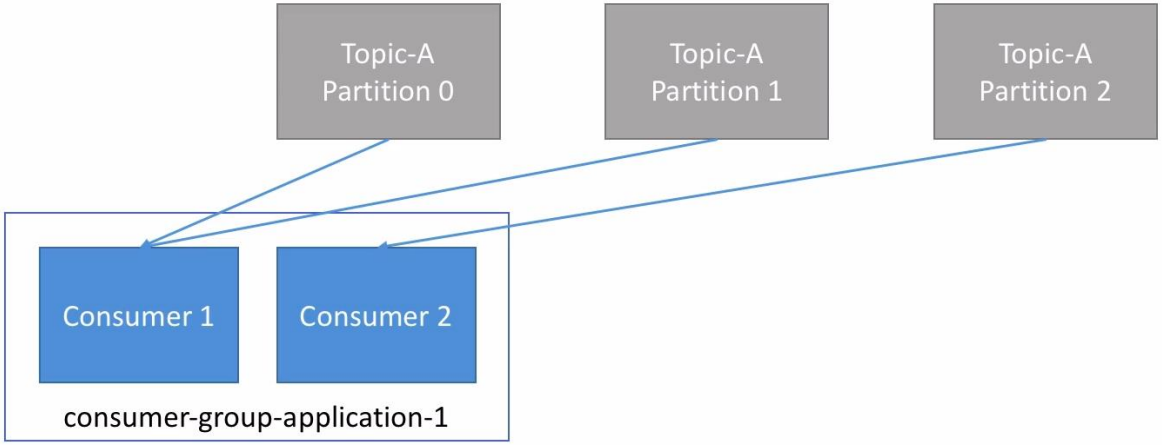
DIGITAL INNOVATION ONE

Grupos de Consumidores

- Consumidores leêm dados dos tópicos em grupos de consumidores
- Cada consumidor de um grupo lê dados exclusivamente de uma única partição
- Se tivermos mais consumidores que partições, serão inativos

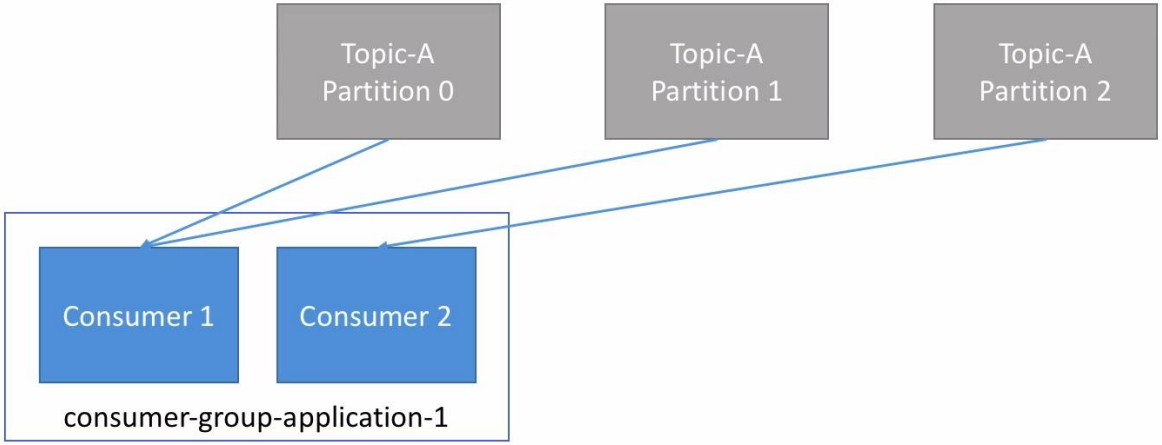
Grupos de Consumidores

- Consumidores leêm dados dos tópicos em grupos de consumidores
- Cada consumidor de um grupo lê dados exclusivamente de uma única partição
- Se tivermos mais consumidores que partições, serão inativos



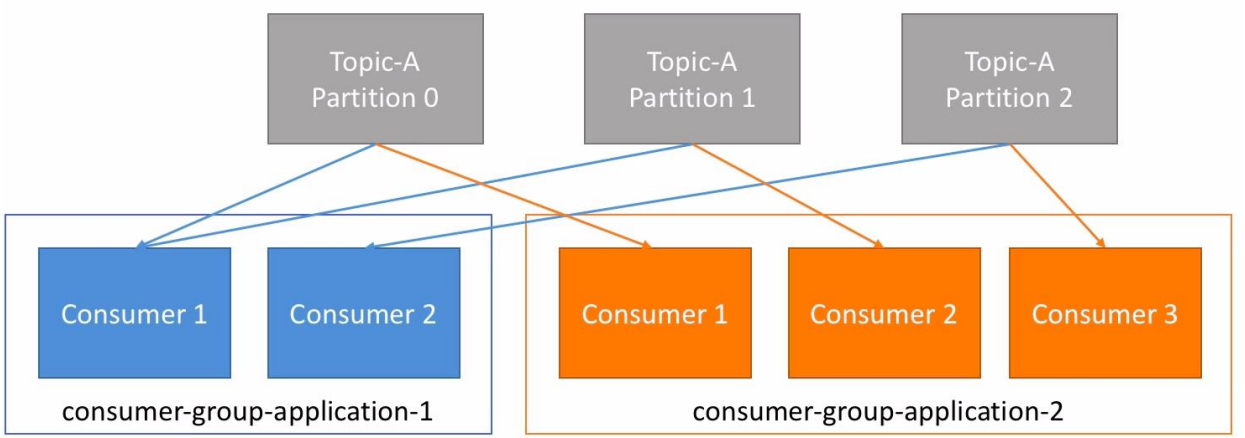
Grupos de Consumidores

- Consumidores leêm dados dos tópicos em grupos de consumidores
- Cada consumidor de um grupo lê dados exclusivamente de uma única partição
- Se tivermos mais consumidores que partições, serão inativos



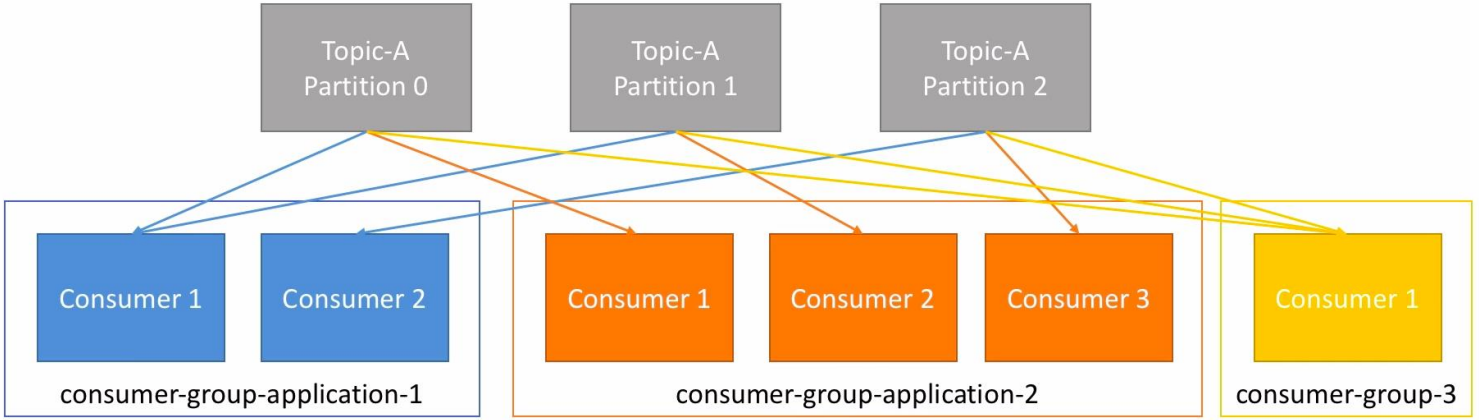
Grupos de Consumidores

- Consumidores leêm dados dos tópicos em grupos de consumidores
- Cada consumidor de um grupo lê dados exclusivamente de uma única partição
- Se tivermos mais consumidores que partições, serão inativos



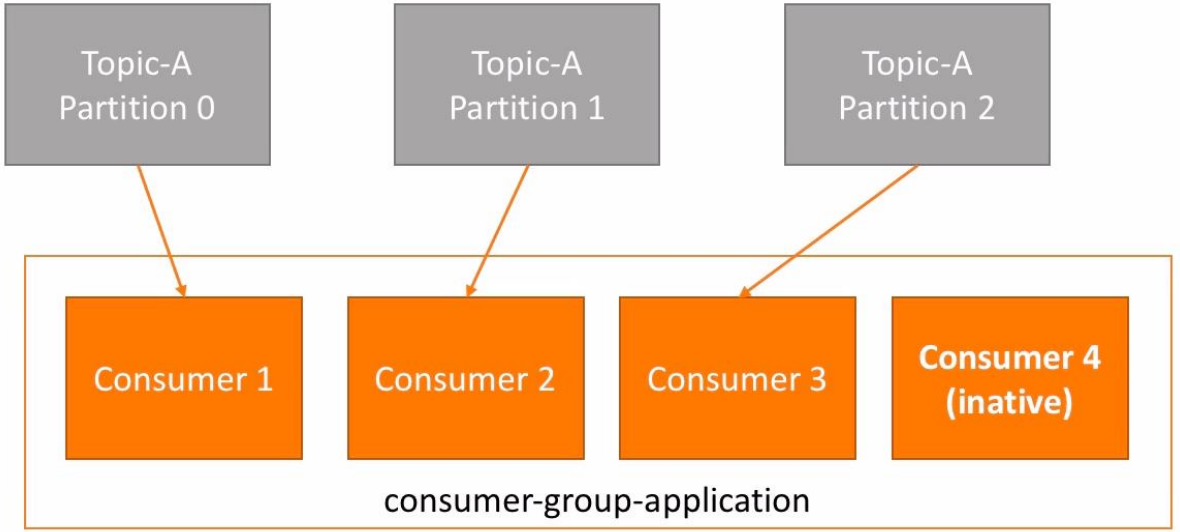
Grupos de Consumidores

- Consumidores leêm dados dos tópicos em grupos de consumidores
- Cada consumidor de um grupo lê dados exclusivamente de uma única partição
- Se tivermos mais consumidores que partições, serão inativos



Grupos de Consumidores

- Se tivermos mais consumidores que partições, serão inativos
- Logo, grande número de partições, grande números de consumidores

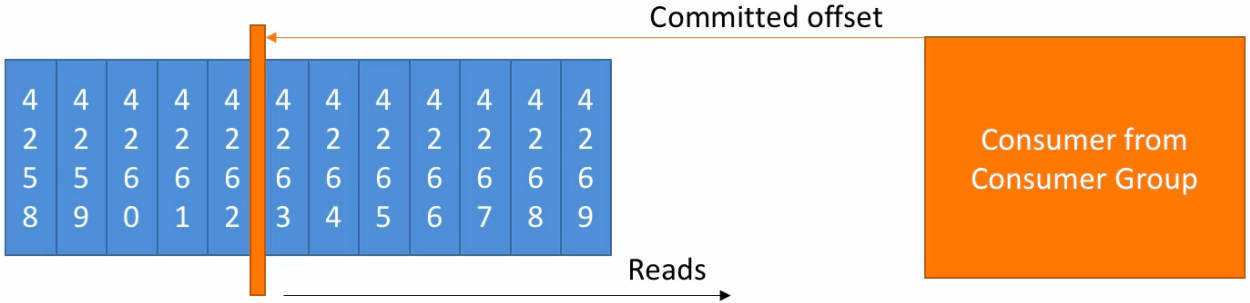


Offsets de consumidores

- Kafka armazena os "offsets" que cada grupo consumidor leu
- Offsets são registrados em um tópico do Kafka (__consumer_offsets)
- Em caso de morte (PLOFT) de uma consumidor, será possível recomeçar a ler os dados de onde parou

Offsets de consumidores

- Kafka armazena os "offsets" que cada grupo consumidor leu
- Offsets são registrados em um tópico do Kafka (__consumer_offsets)
- Em caso de morte (PLOFT) de uma consumidor, será possível recomeçar a ler os dados de onde parou



Registrando Offsets

- 3 formas de registrar os offsets:

1. "At most once": offsets são registrados (commited) assim que que a mensagem (dados) são recebidos.
2. "At least once": offsets são registrados após o processamento. Ou seja, seu sistema pode ler duas vezes a mesma mensagem.

3. "Exactly once": Apenas de Kafka para Kafka

DIGITAL INNOVATION ONE

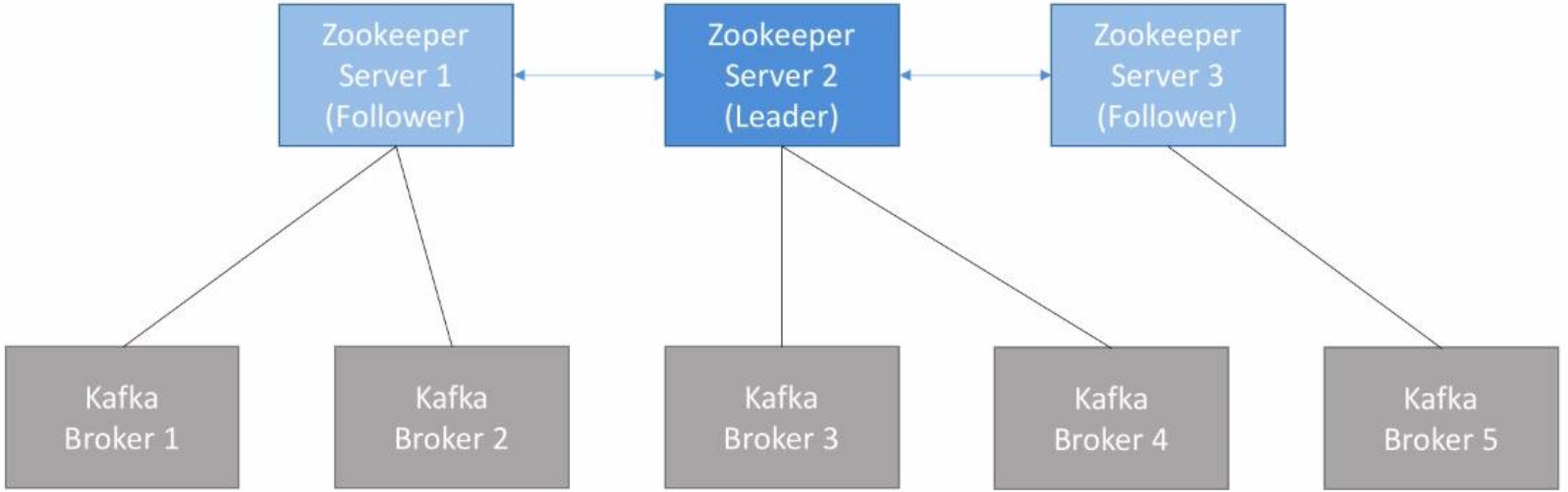
Zookeeper

- Gerencia brokers (mantém a lista)
- Auxilia na eleição de partições líderes
- Envia notificações para o Kakfa em caso de mudanças: novos tópicos, PLOFTs (morte de brokers), novo broker, tópicos apagado, etc)
- Não temos Kafka sem Zookeeper. Depois da versão 0.10 do Kafka não mais armazena offsets

DIGITAL INNOVATION ONE

Zookeeper

APACHE ZooKeeper™

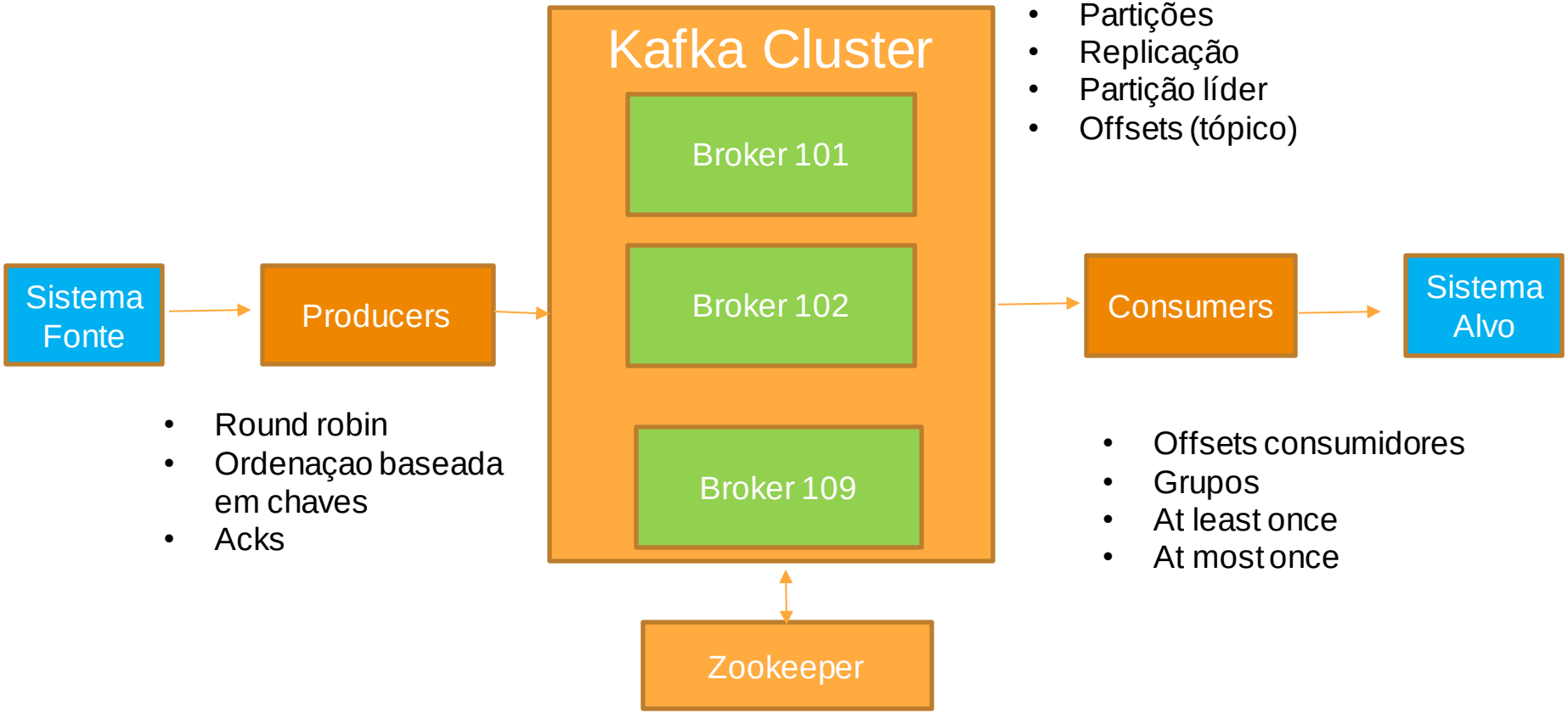


DIGITAL INNOVATION ONE

Garantias do Kafka

- Mensagens são acrescentadas no tópico-partição na ordem que são enviadas
- Consumidores leêm mensagens na ordem armazenada no tópico-partição
- Dado um fator de replicação N, produtores e consumidores toleram até N – 1 brokers morrerem
- Enquanto o número de partições for constante para um tópico, a mensagem com chave será sempre enviada para a mesma partição

Resumo


Kafka Cluster
Broker 101
Broker 102
Broker 109
Zookeeper
Sistema Fonte
Producers
Consumers
Sistema Alvo
Tópicos
Partições
Replicação
Partição líder
Offsets (tópico)
Round robin
Ordenaçao baseada em chaves
Acks
Offsets consumidores
Grupos
At least once
At most once

DIGITAL INNOVATION ONE

Kafka

Aplicação real

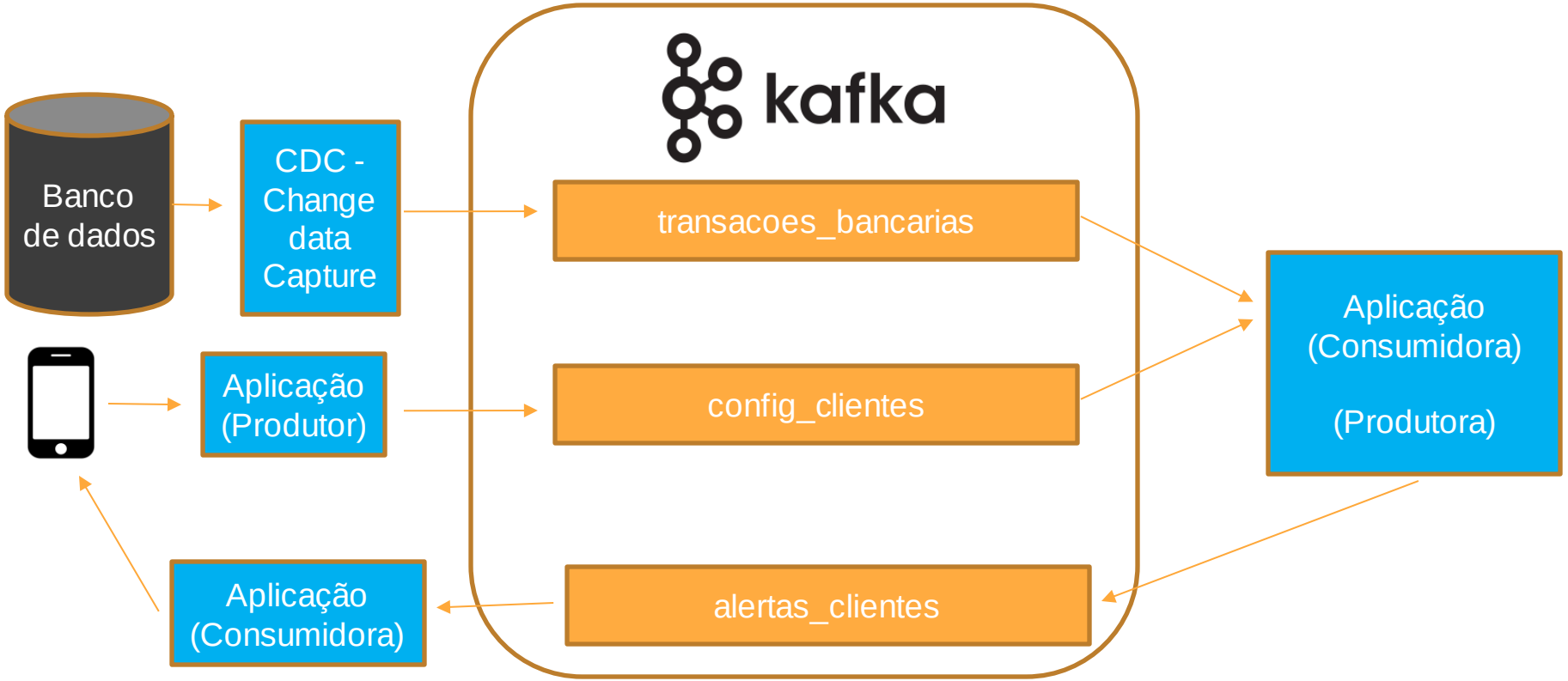
Sistema Bancário

- "MyBank" gostaria de alertar seus clientes em tempo real sobre a ocorrência de transações financeiras de alto valor.

- As transações existem em um banco de dados SQL
- Os limiares (o que é alto para cada cliente) precisa ser definido pelo cliente

Então, como fariamos isso com Kafka?

Sistema Bancário



Dúvidas?

DIGITAL INNOVATION ONE

Kafka

Mão na massa