Kafka for Java Developers

Sep-2022

Nash Tech.

THE APACHE SOFTWARE FOUNDATION

kafka™

A distributed streaming platform

Agenda

Understand:

- What is event streaming?
- What is Apache Kafka?
- Why Kafka?
- Delivery guarantees & Throughput

Kafka:

- Architecture and Components

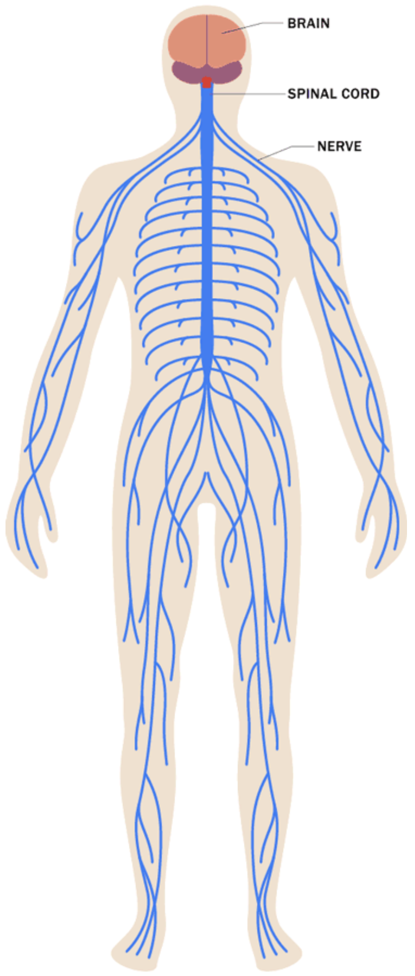
Advanced:

- Kafka Connect
- Kafka Stream
- Security
- Kafka Pros-Cons

What is event streaming?

What is event streaming?

Event streaming is the digital equivalent of the human body's central nervous system.



What is event streaming?

Technically speaking, event streaming is the practice of capturing data.

- *Sensors*
- *Mobile devices*
- *Cloud services*
- *Applications*

What is Apache Kafka?

What is Apache Kafka?

APACHE kafka™

A distributed streaming platform

- An open-source distributed event streaming platform.

- Has three key capabilities:

  1. To **publish** (write) and **subscribe to** (read) streams of events.

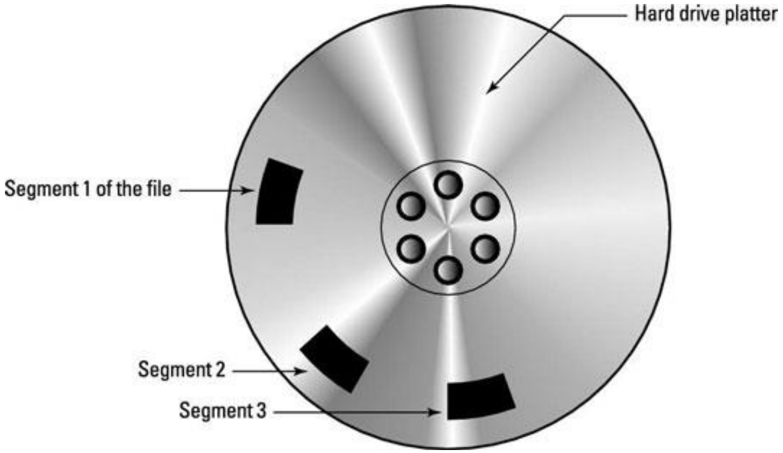
  1. To **store** streams of events durably and reliably for as long as you want.

  1. To **process** streams of events as they occur or retrospectively.

Why Kafka?

Why Kafka?

1. Fast
2. Scalable
3. Reliable
4. Higher throughput
5. Replication characteristics



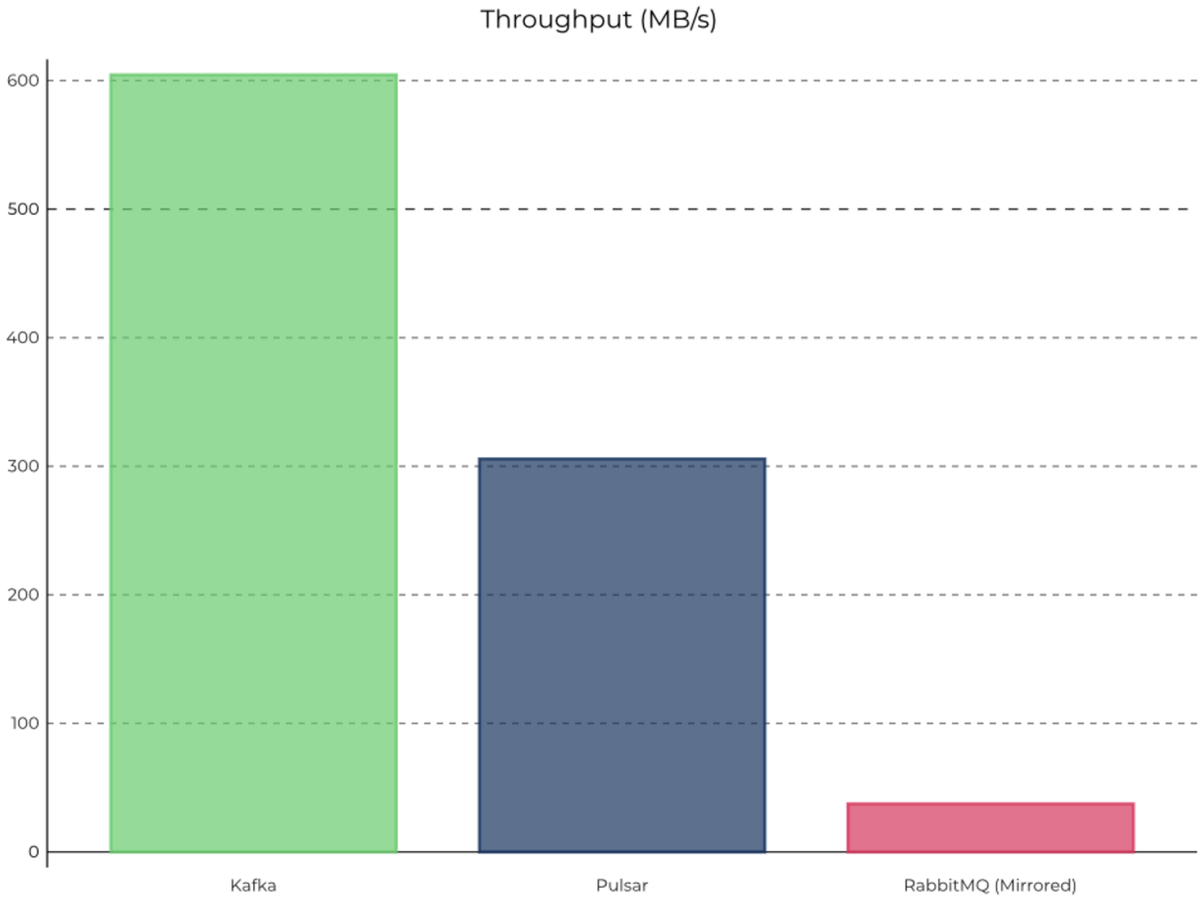
The most important reason Kafka is popular is Kafka's exceptional performance.

Delivery Guarantees & Throughput

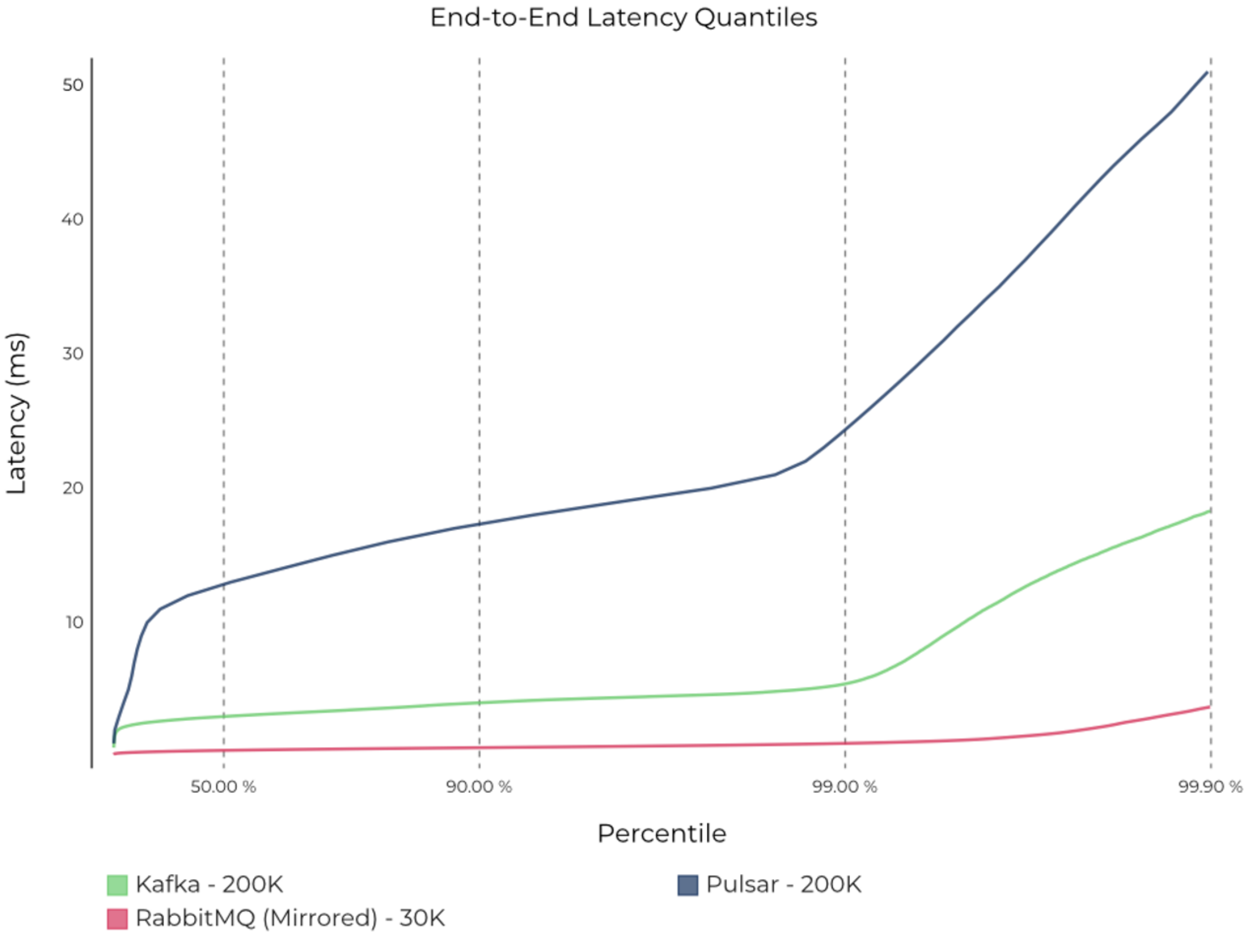
Delivery Guarantees

- ***At most once*** which can lead to messages being lost, but they cannot be redelivered or duplicated
- ***At least once*** ensures messages are never lost but may be duplicated
- ***Exactly once*** guarantees each message processing (not delivery) happens once and only once

Throughput

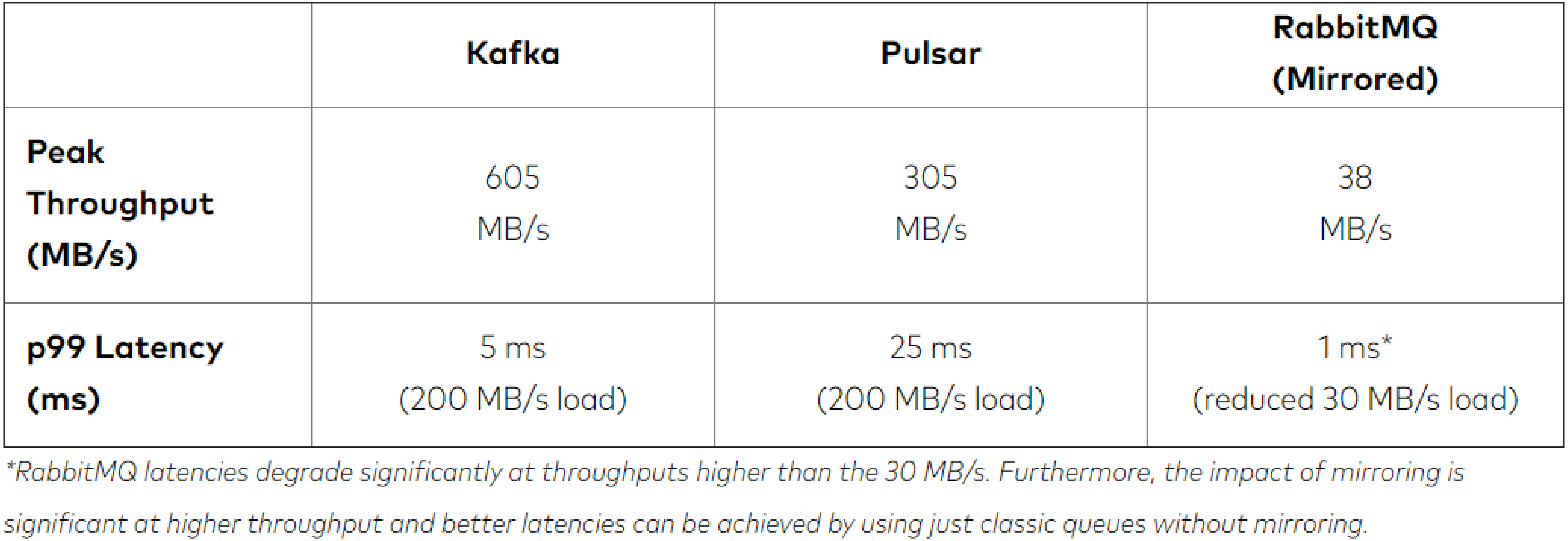


Throughput (MB/s)
600
500
400
300
200
100
0
Kafka
Pulsar
RabbitMQ (Mirrored)



End-to-End Latency Quantiles
Latency (ms)
50
40
30
20
10
50.00 %
90.00 %
99.00 %
99.90 %
Percentile
🟩 Kafka - 200K
🟦 Pulsar - 200K
🟥 RabbitMQ (Mirrored) - 30K

Throughput

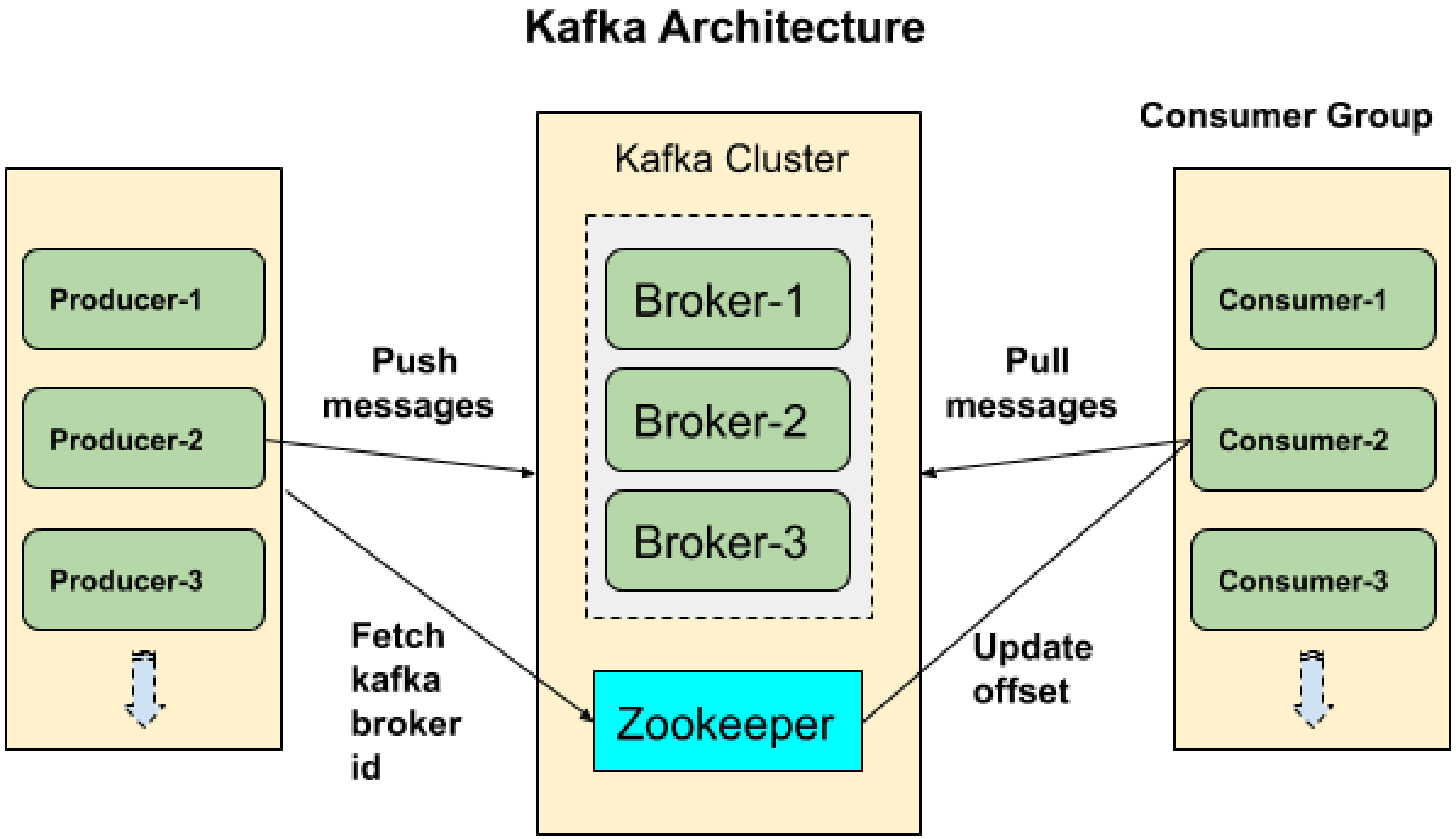


	Kafka	Pulsar	RabbitMQ (Mirrored)
Peak Throughput (MB/s)	605 MB/s	305 MB/s	38 MB/s
p99 Latency (ms)	5 ms (200 MB/s load)	25 ms (200 MB/s load)	1 ms* (reduced 30 MB/s load)

**RabbitMQ latencies degrade significantly at throughputs higher than the 30 MB/s. Furthermore, the impact of mirroring is significant at higher throughput and better latencies can be achieved by using just classic queues without mirroring.*

Architecture and Components

Cluster Architecture


Kafka Architecture

Kafka Cluster

Broker-1

Broker-2

Broker-3

Zookeeper

Producer-1

Producer-2

Producer-3

Push messages

Fetch kafka broker id

Consumer Group

Consumer-1

Consumer-2

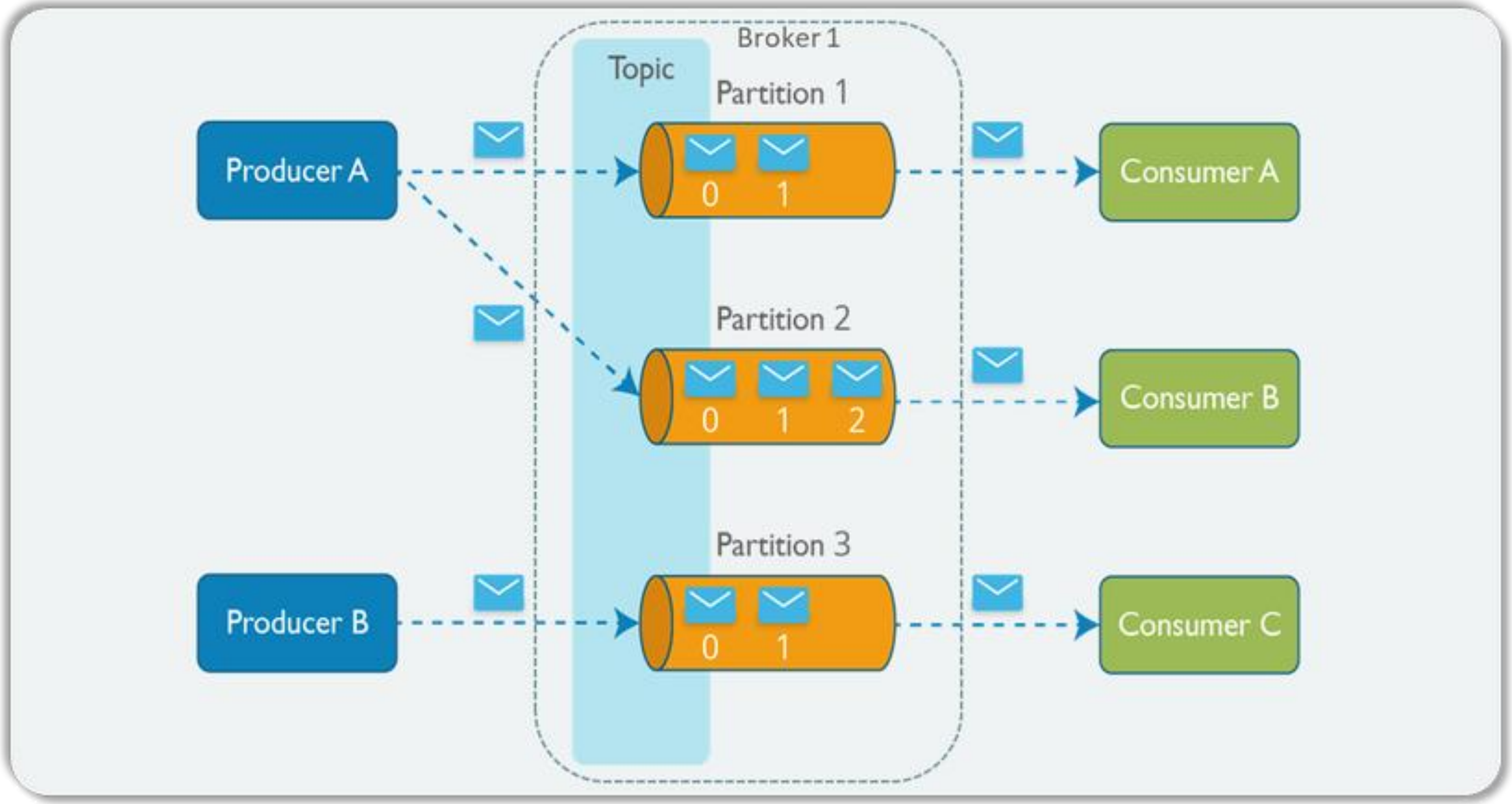
Consumer-3

Pull messages

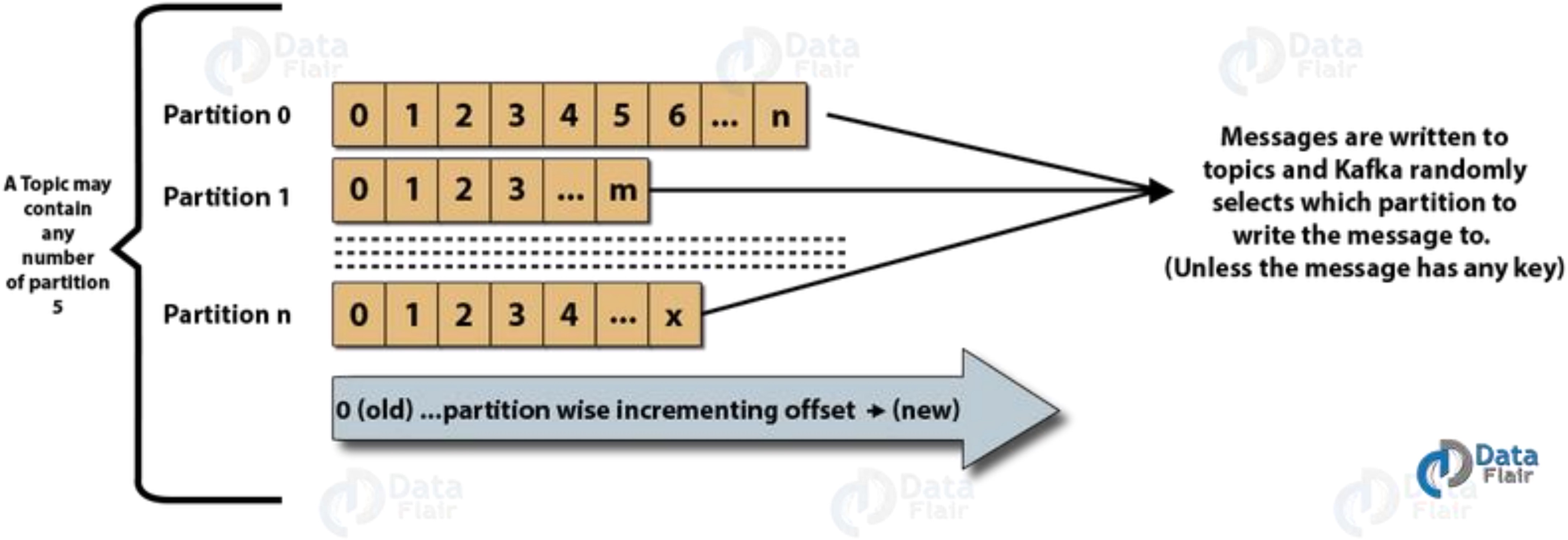
Update offset

PT

Broker


Broker 1
Topic
Partition 1
Producer A
0
1
Consumer A
Partition 2
0
1
2
Consumer B
Partition 3
Producer B
0
1
Consumer C

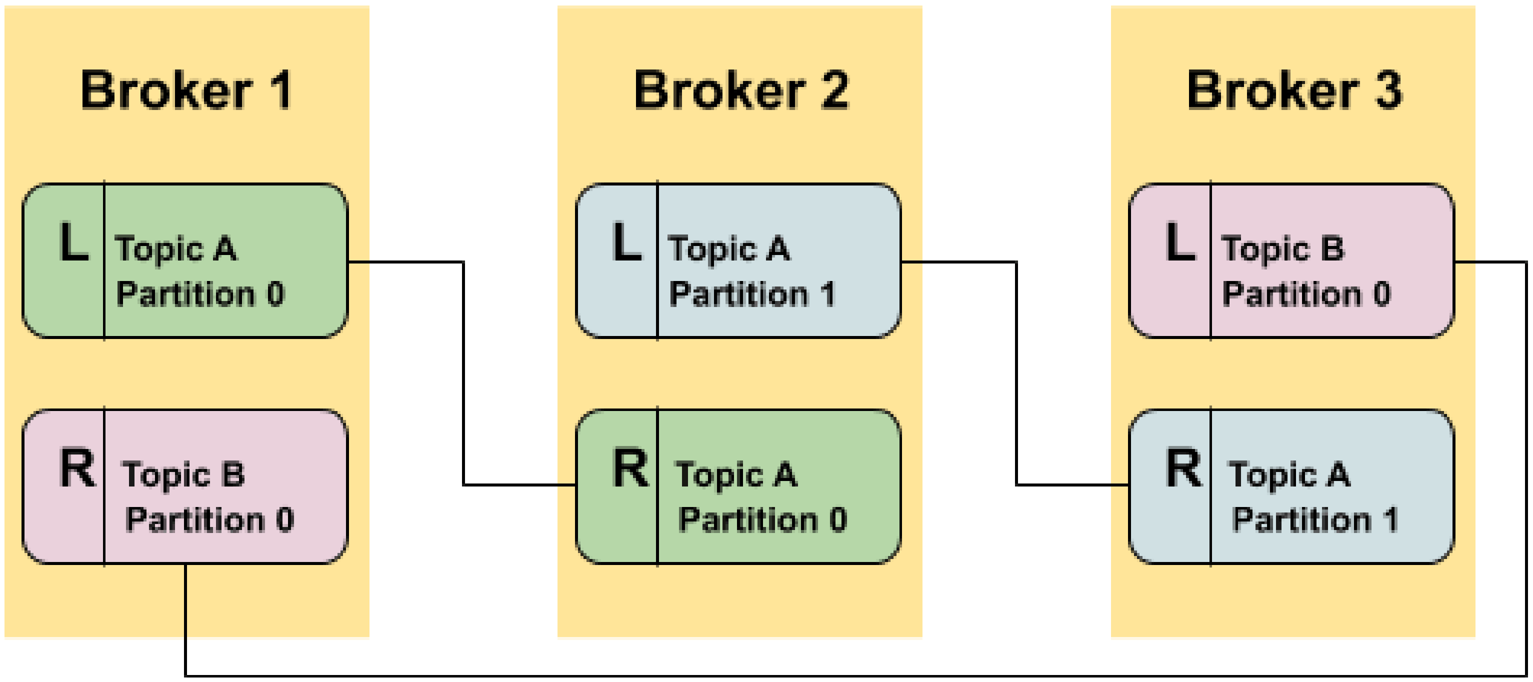
Topic Partition



A Topic may contain any number of partition 5
Partition 0
0 1 2 3 4 5 6 ... n
Partition 1
0 1 2 3 ... m
Partition n
0 1 2 3 4 ... x
0 (old) ...partition wise incrementing offset → (new)
Messages are written to topics and Kafka randomly selects which partition to write the message to. (Unless the message has any key)

Topic replication factor

Topic Replication Factor

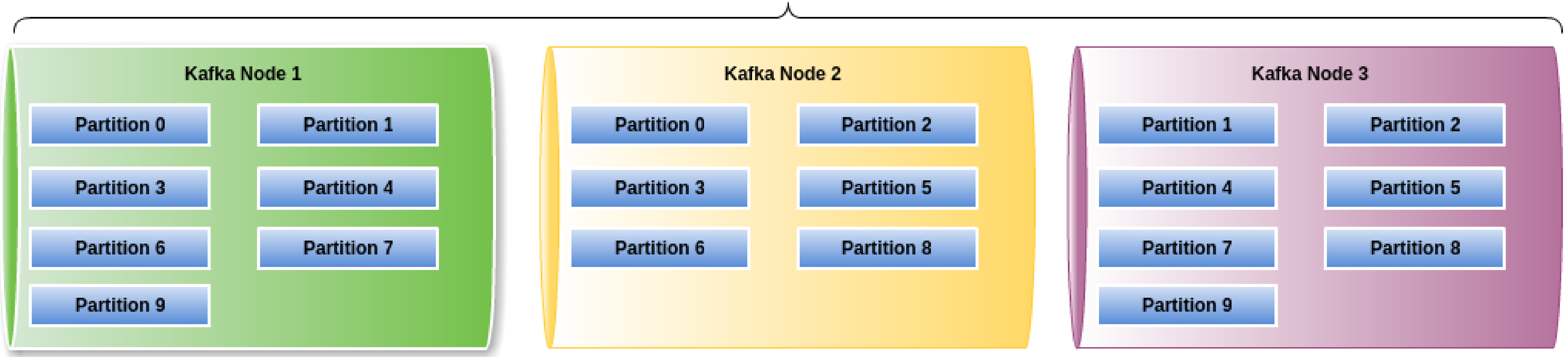


PT

L - Leader **R** - Replica

Topic configuration:

replication-factor=2, **partition**=10



Kafka Node 1

Partition 0 | Partition 1
Partition 3 | Partition 4
Partition 6 | Partition 7
Partition 9

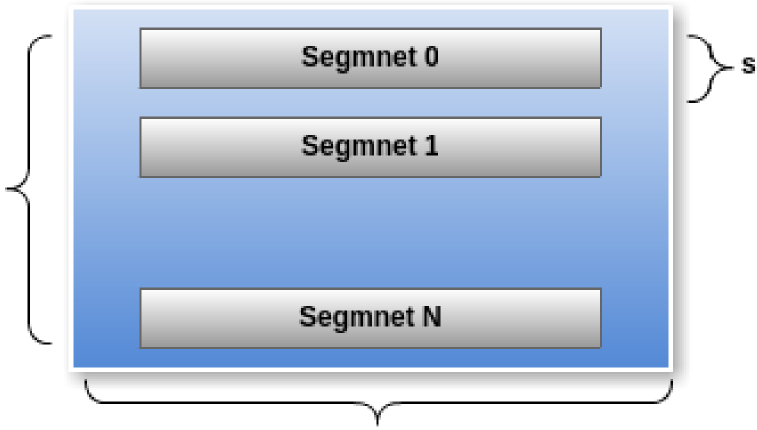
Kafka Node 2

Partition 0 | Partition 2
Partition 3 | Partition 5
Partition 6 | Partition 8

Kafka Node 3

Partition 1 | Partition 2
Partition 4 | Partition 5
Partition 7 | Partition 8
Partition 9

Topic partition configuration:



Segmnet 0

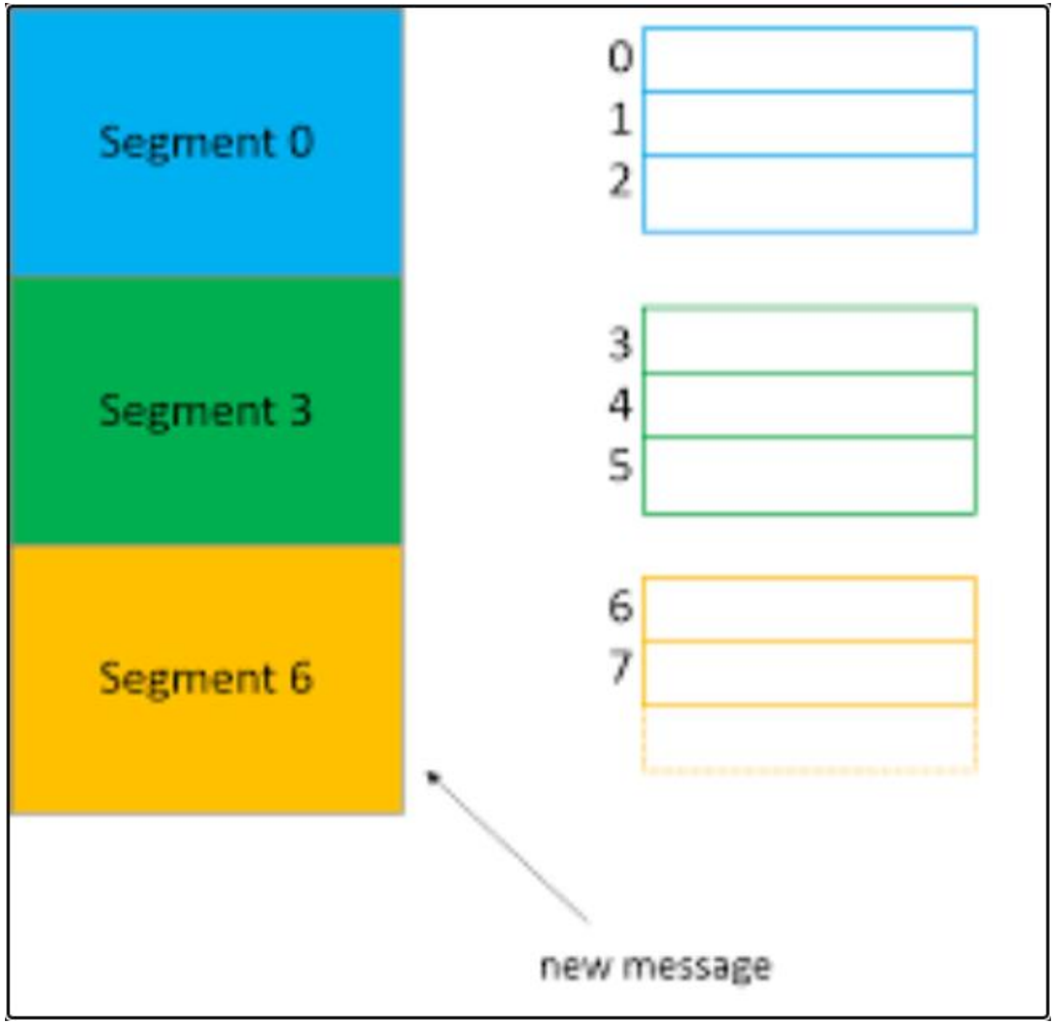
Segmnet 1

Segmnet N

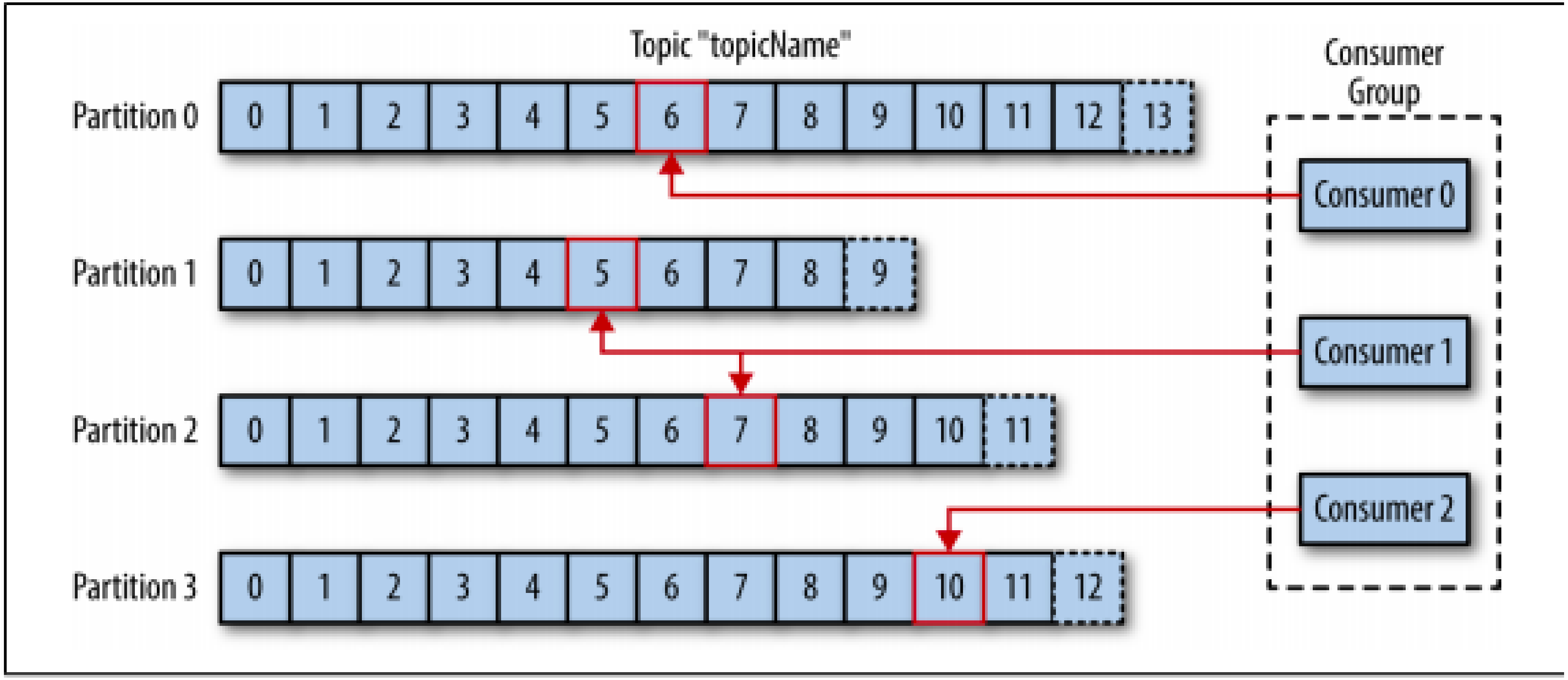
segment.bytes=105000000 (100Mb)

retention.bytes=1074000000 (1Gb)

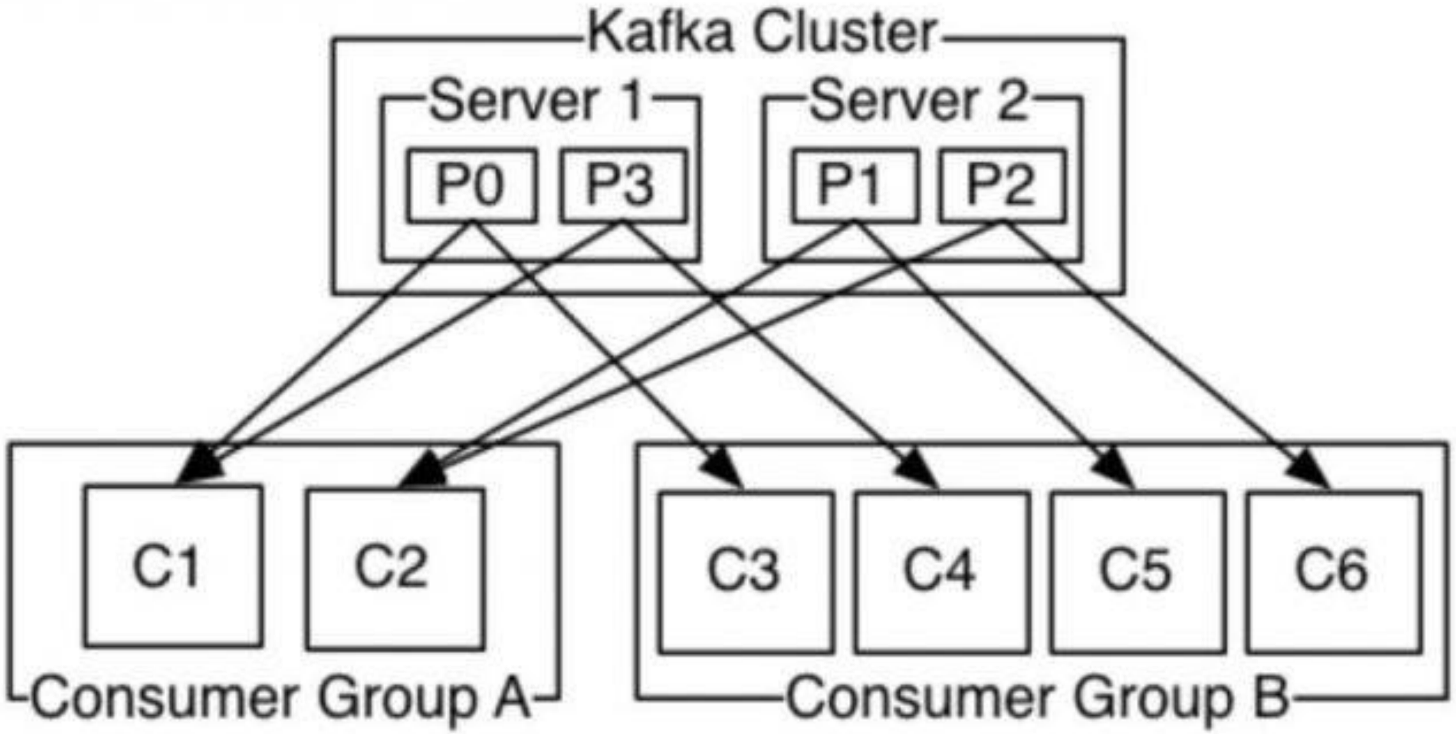
delete.retention.ms=6000 (6s)
do log compation every 6 sec


Segment 0
Segment 3
Segment 6
0
1
2
3
4
5
6
7
new message

Consumer Group



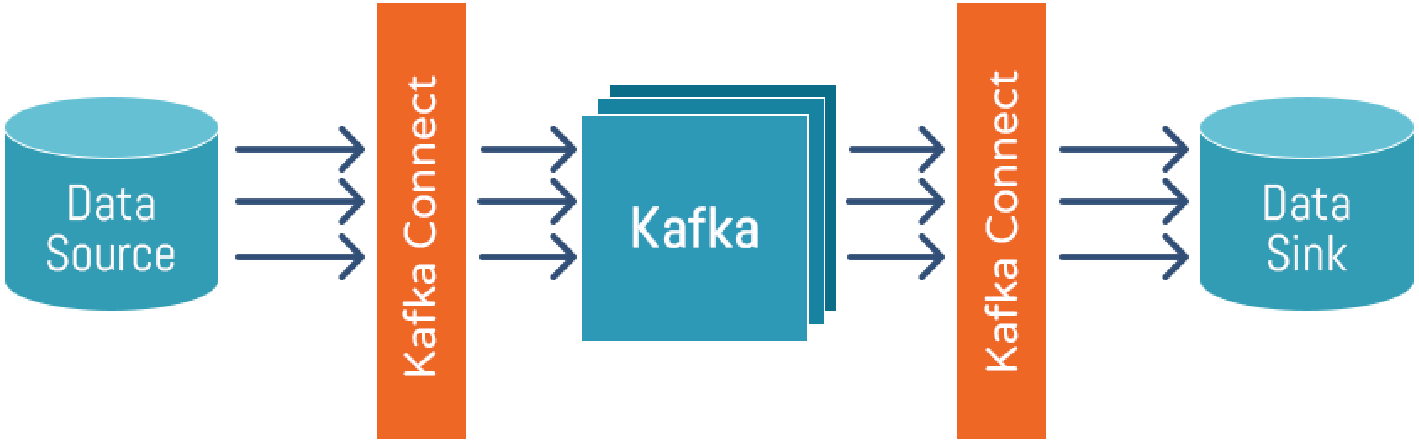
Consumer Group



Kafka Connect

Kafka Connect

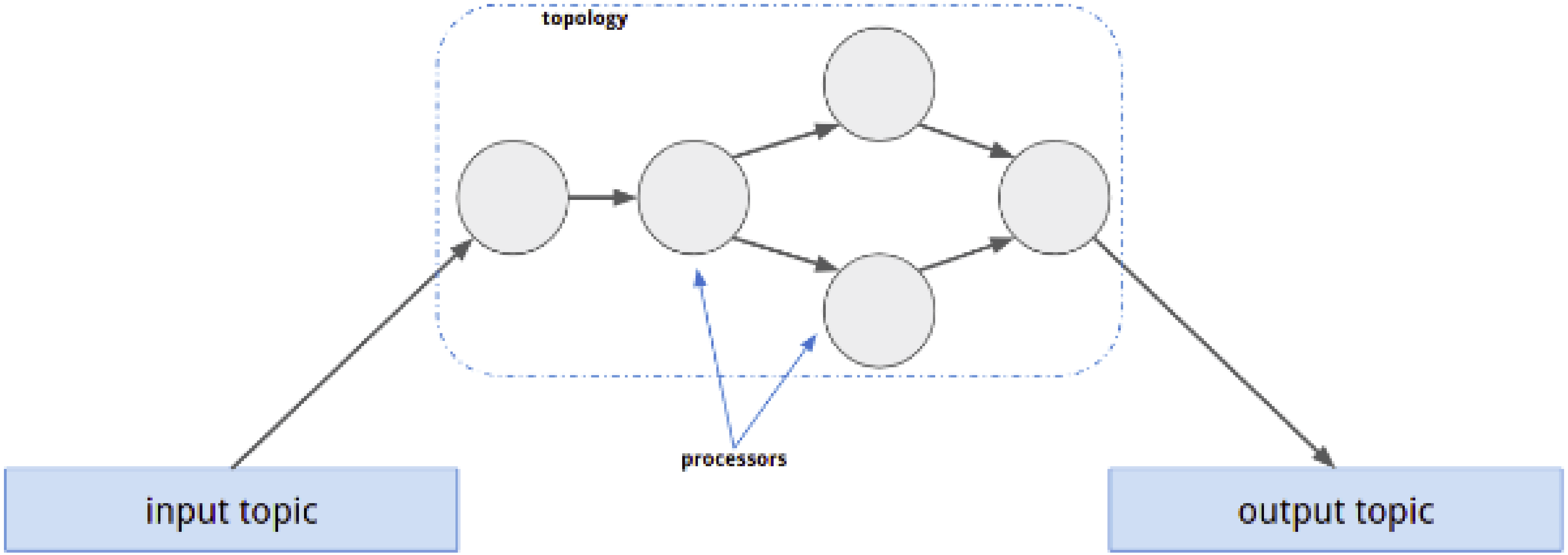
- A tool for scalably and reliably streaming data between Apache Kafka and other systems.
- Move large collections of data into and out of Kafka



Kafka Stream

Kafka Stream

- A client library for building applications and microservices, where the input and output data are stored in Kafka clusters.



Stream DSL

- The Kafka Streams DSL (Domain Specific Language) is built on top of the Streams Processor API.

- **KStream**: *is an abstraction of a **record stream.***

- **KTable**: *is an abstraction of a **changelog stream.***

- **GlobalKTable**: *is an abstraction of a **changelog stream**.*

Security

Security

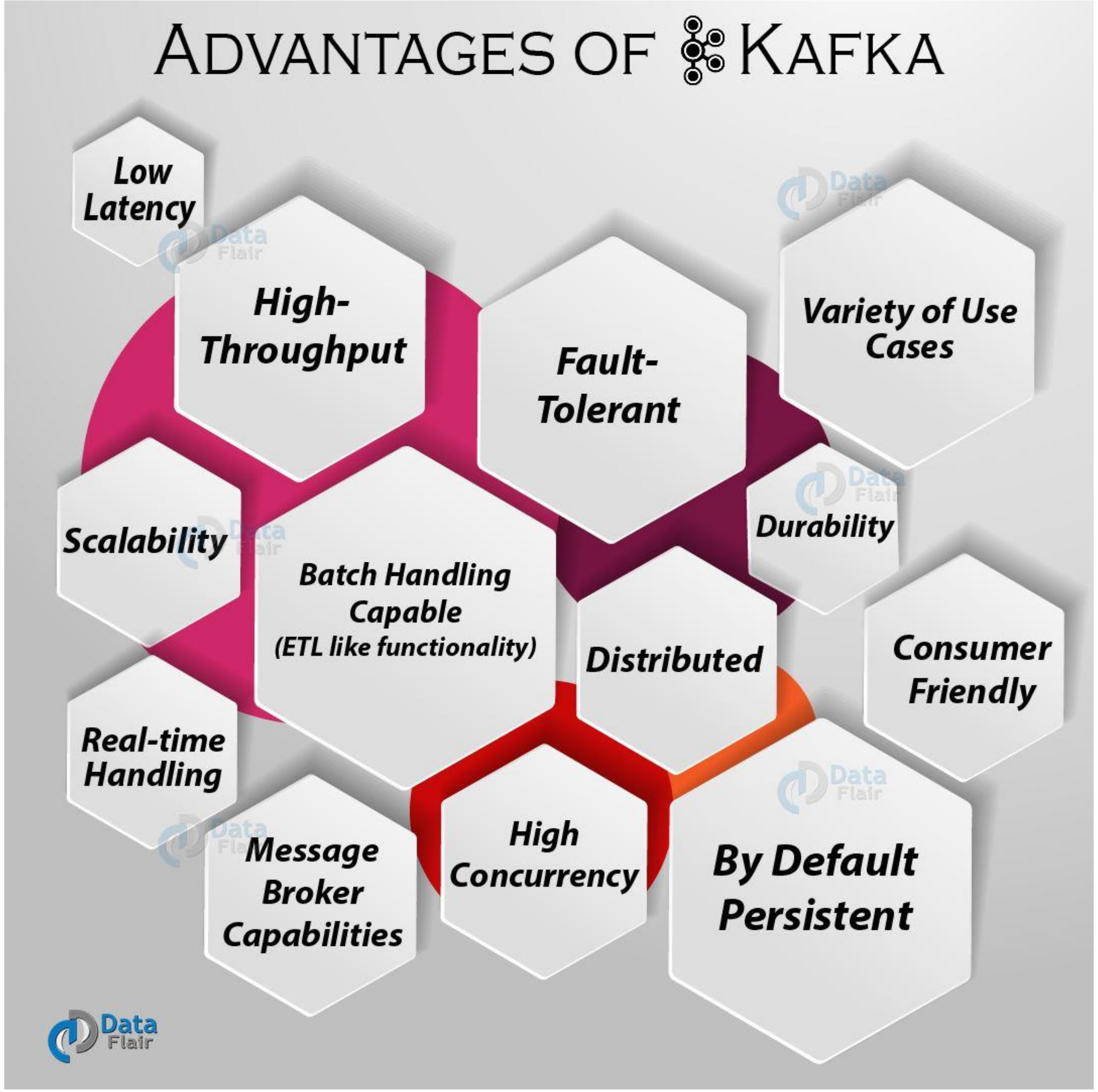
- Kafka supports the following SASL mechanisms:
  SASL/GSSAPI (Kerberos)
  SASL/PLAIN
  SASL/SCRAM-SHA-256 and SASL/SCRAM-SHA-512
  SASL/OAUTHBEARER

- Authentication of connections from brokers to ZooKeeper

- Encryption of data transferred between brokers and clients, between brokers.

- Authorization of read / write operations by clients

Kafka Pros-Cons



Advantages of Kafka

Low Latency

High-Throughput

Fault-Tolerant

Variety of Use Cases

Scalability

Batch Handling Capable (ETL like functionality)

Durability

Distributed

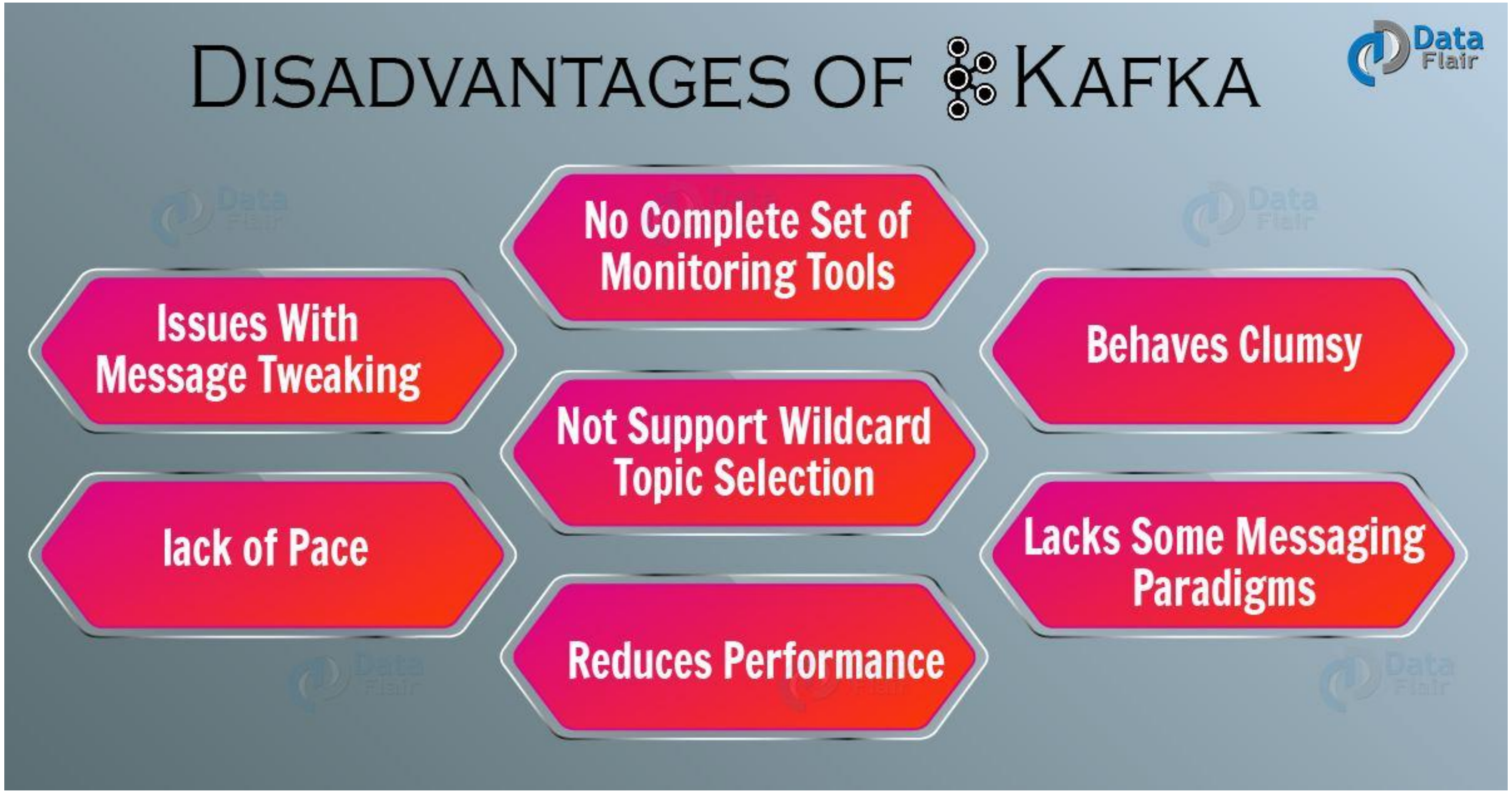
Consumer Friendly

Real-time Handling

Message Broker Capabilities

High Concurrency

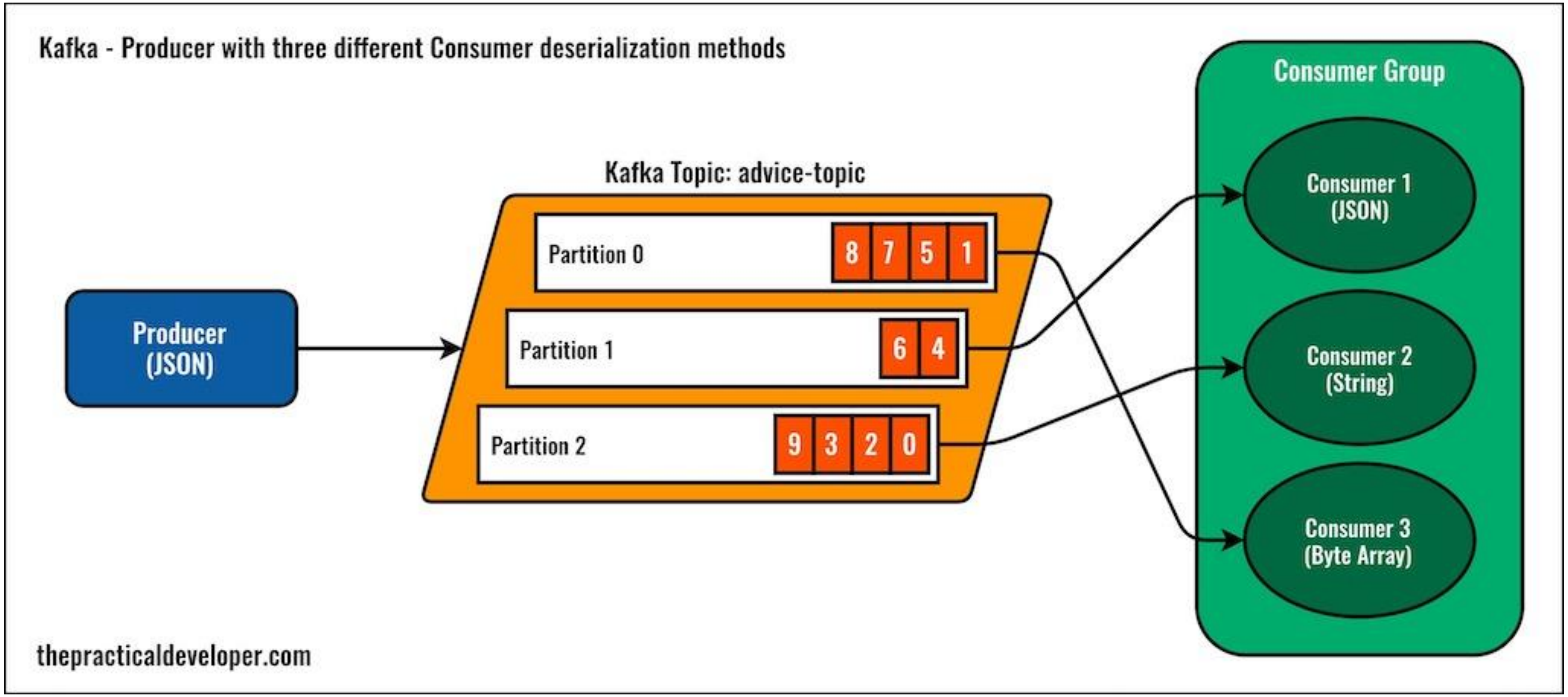
By Default Persistent



Disadvantages of Kafka

- Issues With Message Tweaking
- lack of Pace
- No Complete Set of Monitoring Tools
- Not Support Wildcard Topic Selection
- Reduces Performance
- Behaves Clumsy
- Lacks Some Messaging Paradigms

Demo


Kafka - Producer with three different Consumer deserialization methods
Producer (JSON)
Kafka Topic: advice-topic
Partition 0
8
7
5
1
Partition 1
6
4
Partition 2
9
3
2
0
Consumer Group
Consumer 1 (JSON)
Consumer 2 (String)
Consumer 3 (Byte Array)
thepracticaldeveloper.com

References

- *https://kafka.apache.org/documentation*
- *https://kafka.apache.org/20/documentation/streams/developer-guide/dsl-api.html*
- *https://docs.confluent.io/platform/current/ksqldb/index.html*
- *https://dzone.com/articles/what-is-kafka#:~:text=Why%20Kafka%3F,due%20to%20volume%20and%20responsiveness*

Thanks