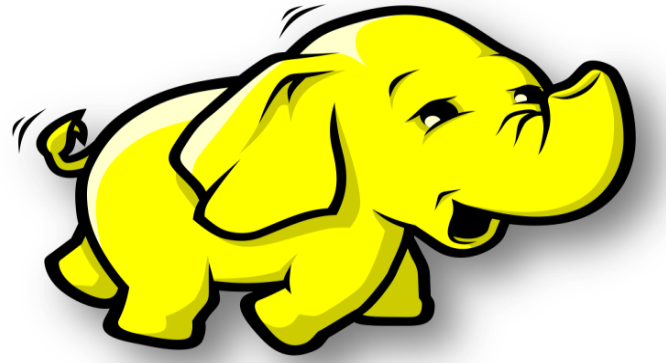
Chapter 2:

Data Ingestion Tools
Kafka

Lebanese University

Faculty of Information 1

Dr. Hussein Hazimeh



Data Ingestion Tools

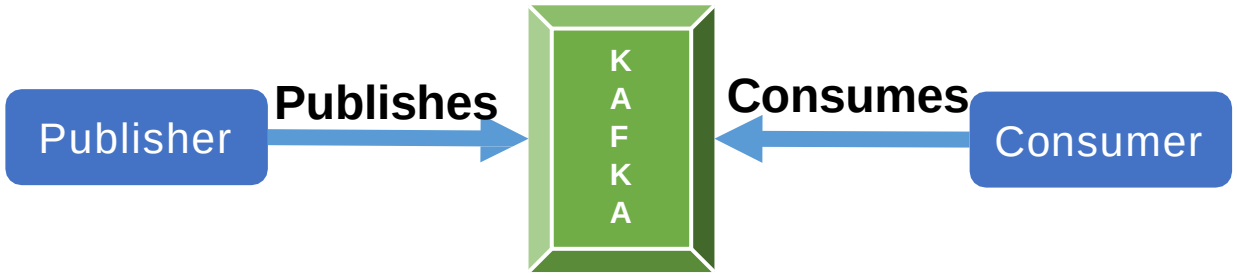
Apache Kafka

Apache Kafka – Overview

- **Challenge of Real-time data collection**
  - High speed of data
  - Requires high capacity CPU
  - Requires huge memory

Apache Kafka – Overview

- Kafka is a **high-throughput**, **distributed**, fault tolerant, and **replicated** messaging system that was first developed at LinkedIn.
- It is a distributed **publish-subscribe** messaging system



- Kafka is specifically designed and optimized **persistent real-time streams**.

Apache Kafka – Overview

- Kafka **routes** the volume of messages to **multiple consumers**
- Kafka provides **seamless integration** between information of producers and consumers
- It is designed to scale **horizontally** on **commodity software** to support hundreds of thousands of messages per second.
- Kafka is suitable for both **offline** and **online** message consumption.

Apache Kafka – Overview

- Kafka messages are persisted on the **disk** and replicated within the cluster to prevent data loss
- Kafka is built on top of the **ZooKeeper** synchronization service.
- It integrates very well with **Apache Storm** and **Apache Spark** for real-time streaming data analysis

Apache Kafka – Overview

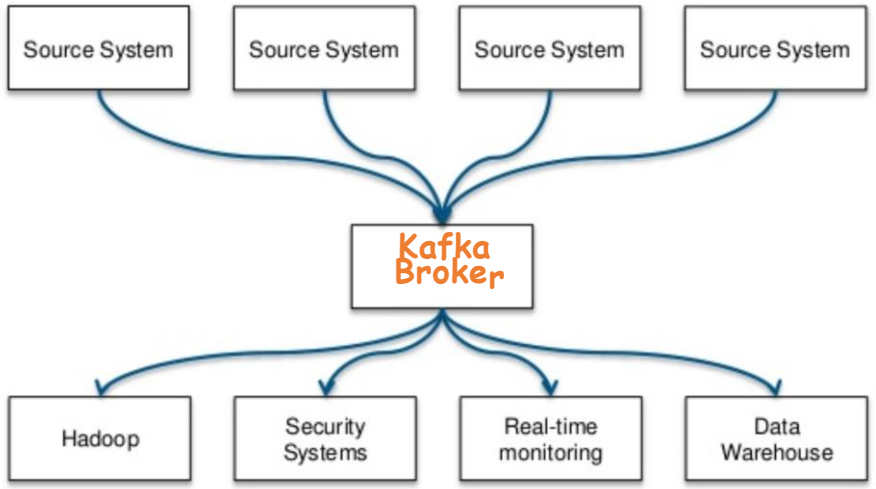
- **Characteristics**
  - **High Throughput**: Kafka is designed to work on **commodity hardware** and to support **millions of messages** per second.
  - **Real-time**: Message produced by the producer threads should be immediately visible to consumer threads.
  - **Persistent Messaging**: Kafka is designed with **O(1) disk structures** that provide **constant-time performance** even with very large stored messages.
  - **Reliability:** Kafka is **partitioned**, **replicated** and **fault tolerant**.

Apache Kafka – Overview

- **Characteristics**
  - **Distributed**: Kafka supports **messages partitioning** over Kafka servers and **distributing consumption** over a cluster of consumer machines.
  - **Multi Client Support**: Kafka supports **easy integration** of clients from different platforms such as Java, .Net, PHP, Ruby, and Python.
  - **Scalability**: Kafka messaging system **scales** easily without down time.
  - **Durability**: It uses **distributed commit log**

Apache Kafka – Overview

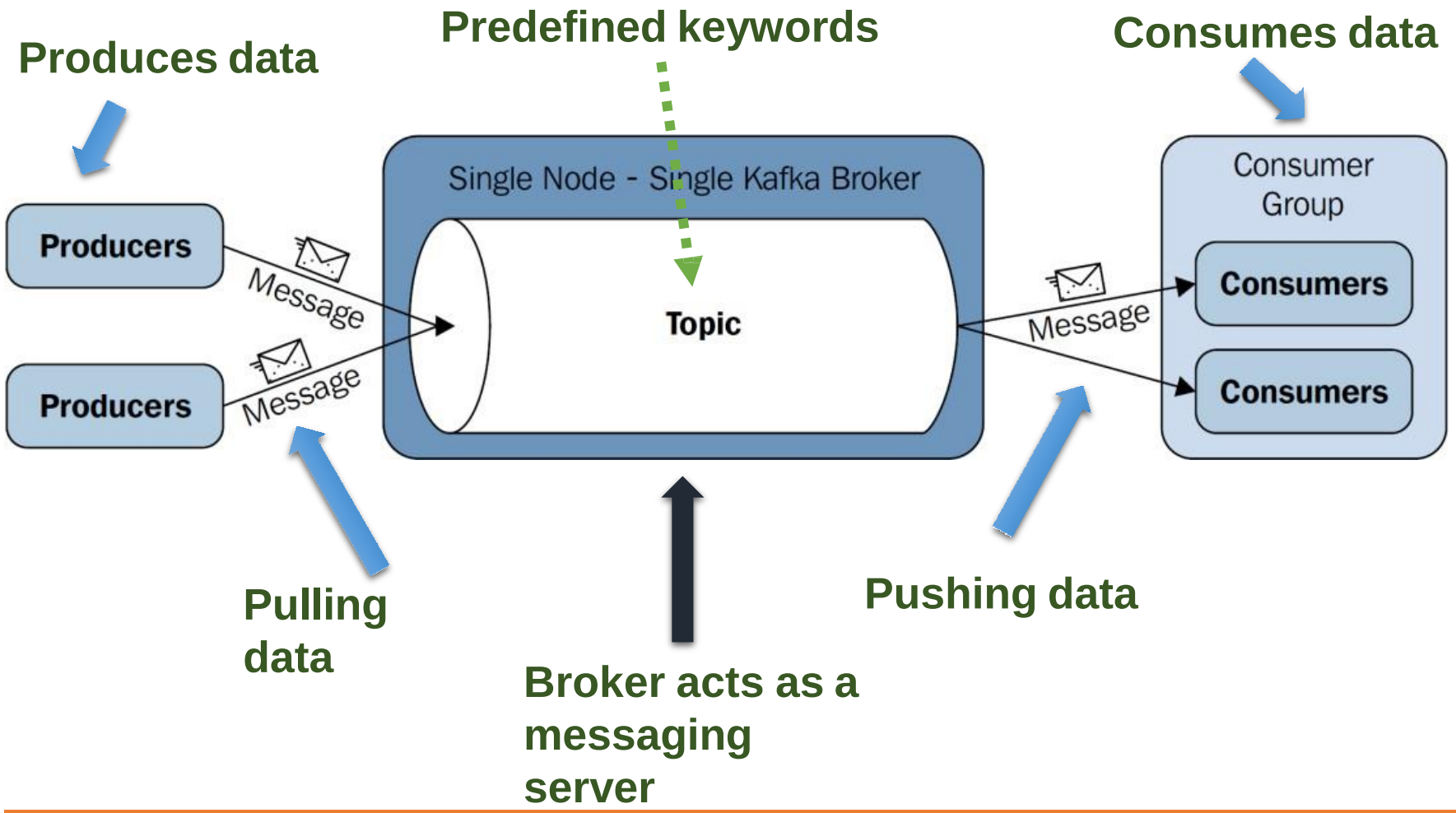
- **Why Kafka?**
  - It **unifies** offline and online processing by providing a mechanism for parallel load in Hadoop systems.



  - Kafka **decouples** data pipeline
  - It is able to **partition** real-time consumption over a cluster of machine

Apache Kafka – Basic Concepts

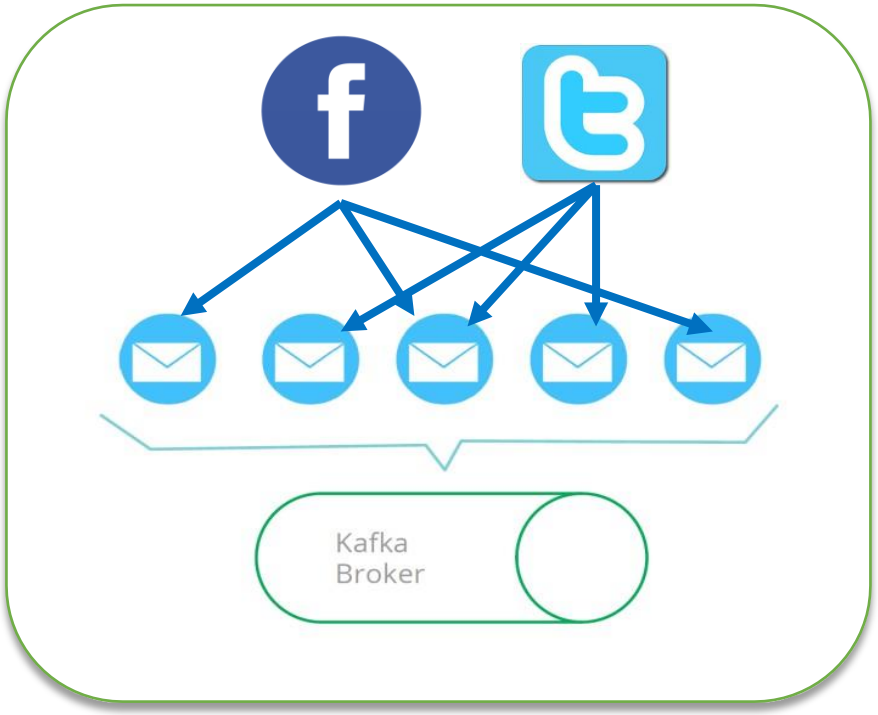
- **Basic Kafka Structural Blueprint**



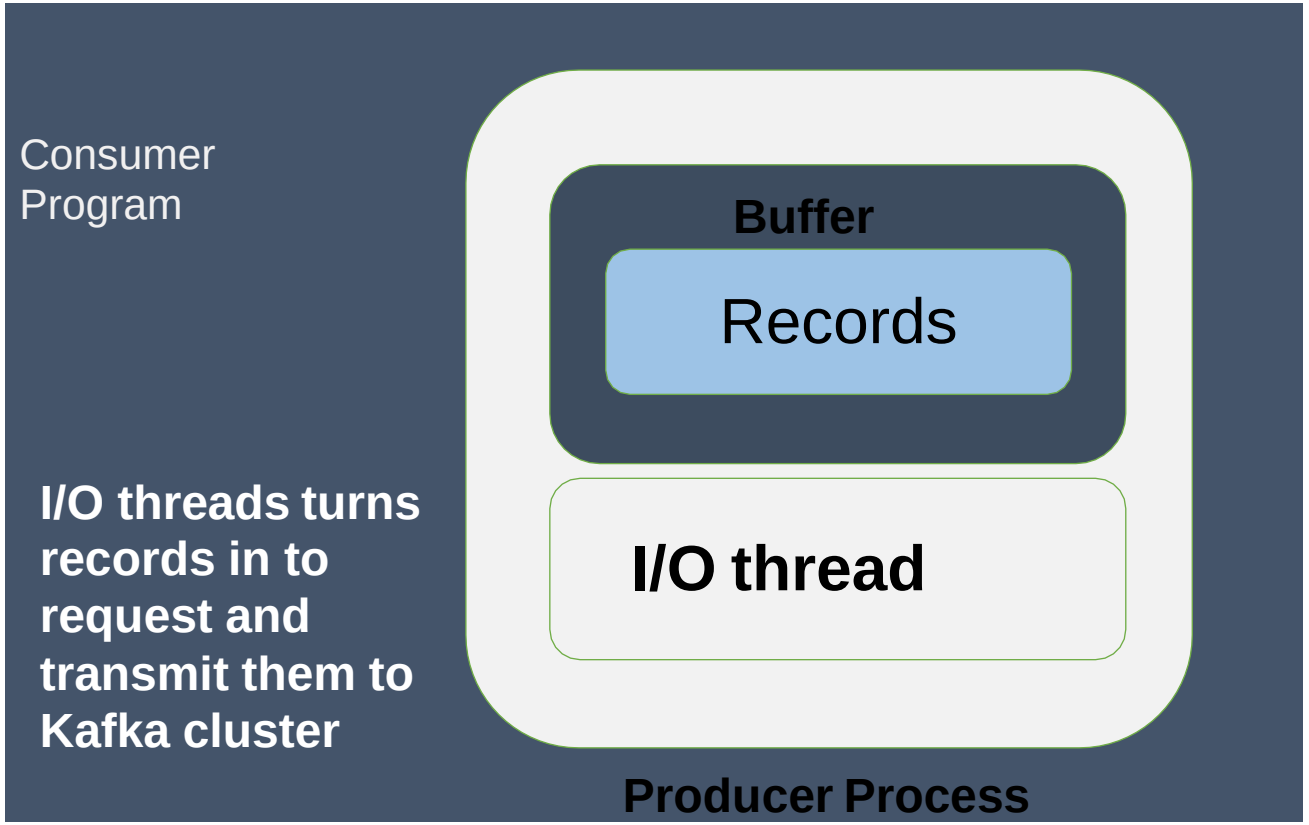
Intelligencia

Apache Kafka – Basic Concepts

- Basic Elements
  - Producer: A process/client that publishes records to the Kafka cluster

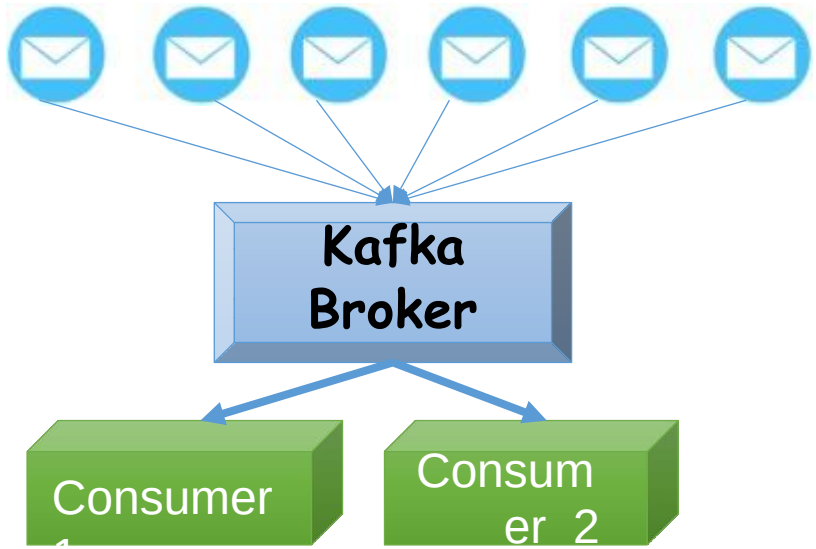


Producer Example



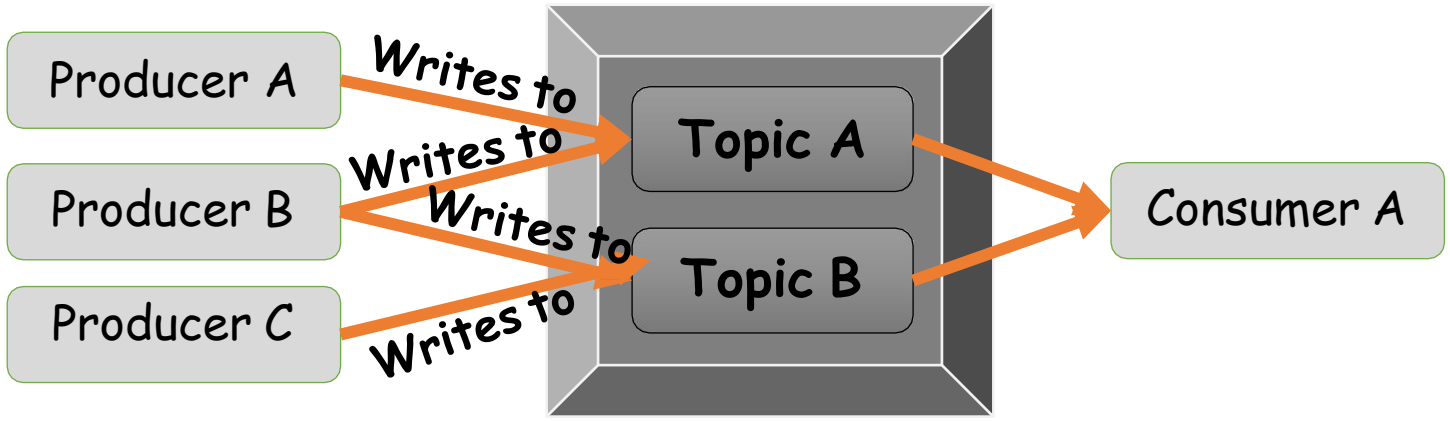
Apache Kafka – Basic Concepts

- Basic Elements
  - Consumer
    - A consumer is Kafka process/client that subscribes to topics and consumes records from a cluster
    - It maintains TCP connections to the necessary brokers to fetch data



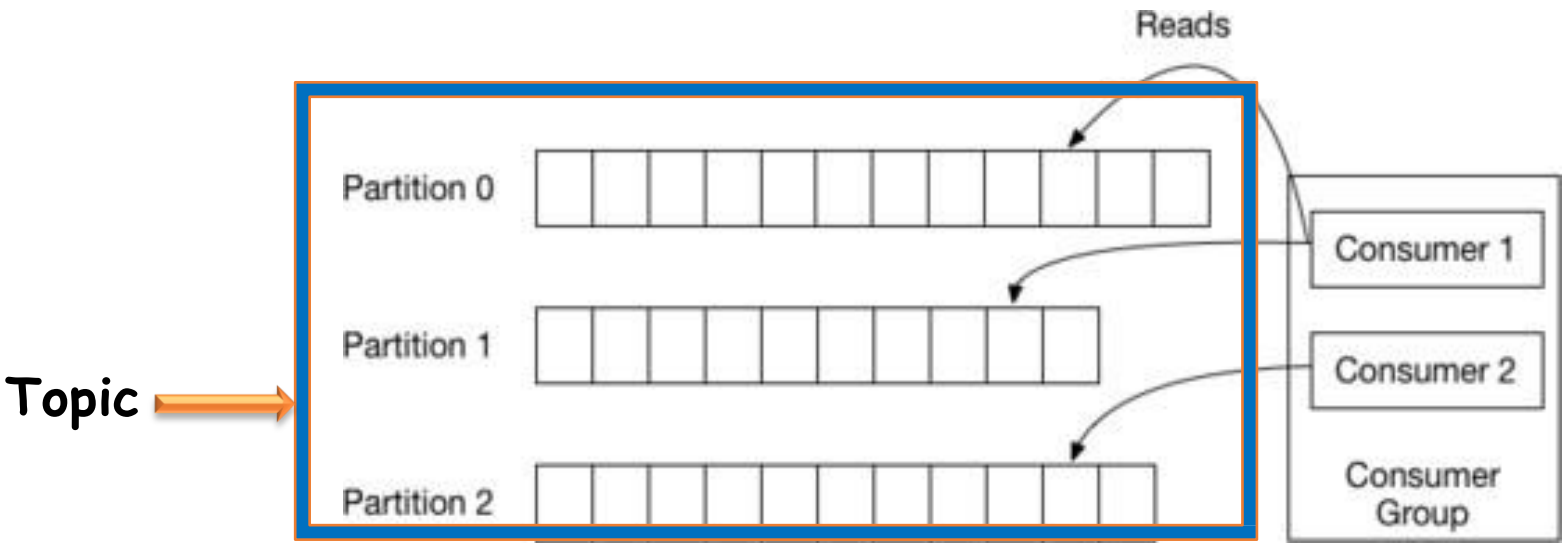
Apache Kafka – Basic Concepts

- Basic Elements
  - Topic
    - A topic is a category or feed name to which records are published
    - The Kafka cluster **stores** topics
    - Once a data is published to a topic, it can't be changed/updated.



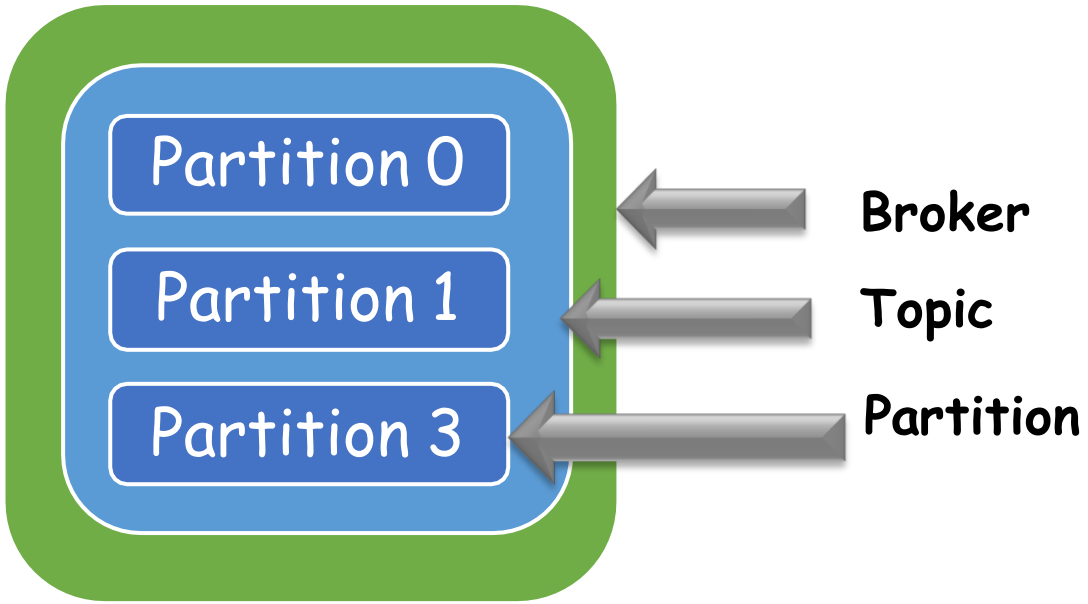
Apache Kafka – Basic Concepts

- Basic Elements
  - Consumer Group
    - A set of consumers sharing a c**ommon group identifier**
    - Each consumer within the group reads from a **unique partition** and the group as a whole consumes all messages from the entire topic



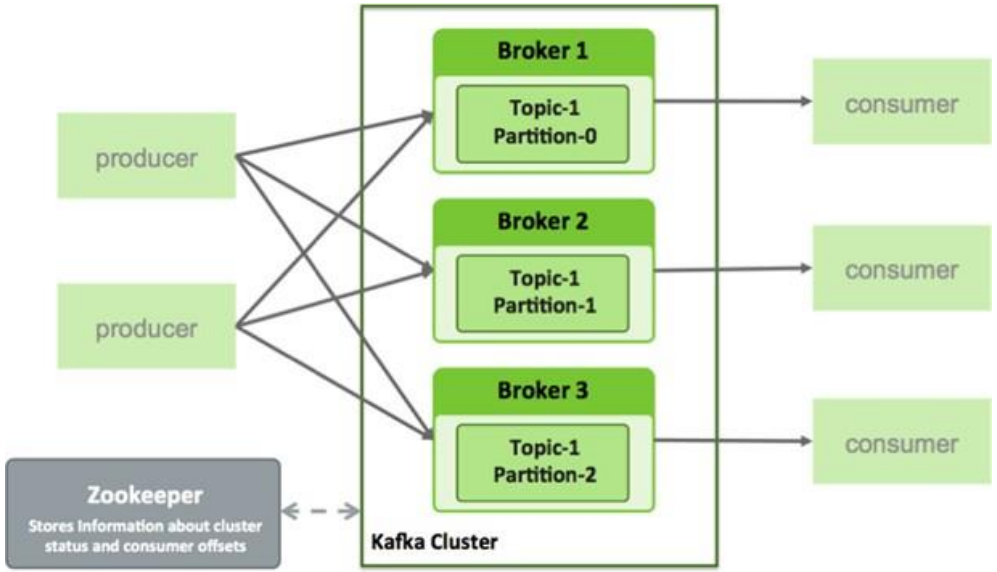
Apache Kafka – Basic Concepts

- Basic Elements
  - Broker
    - A broker is a Kafka server (physical or virtual machine)



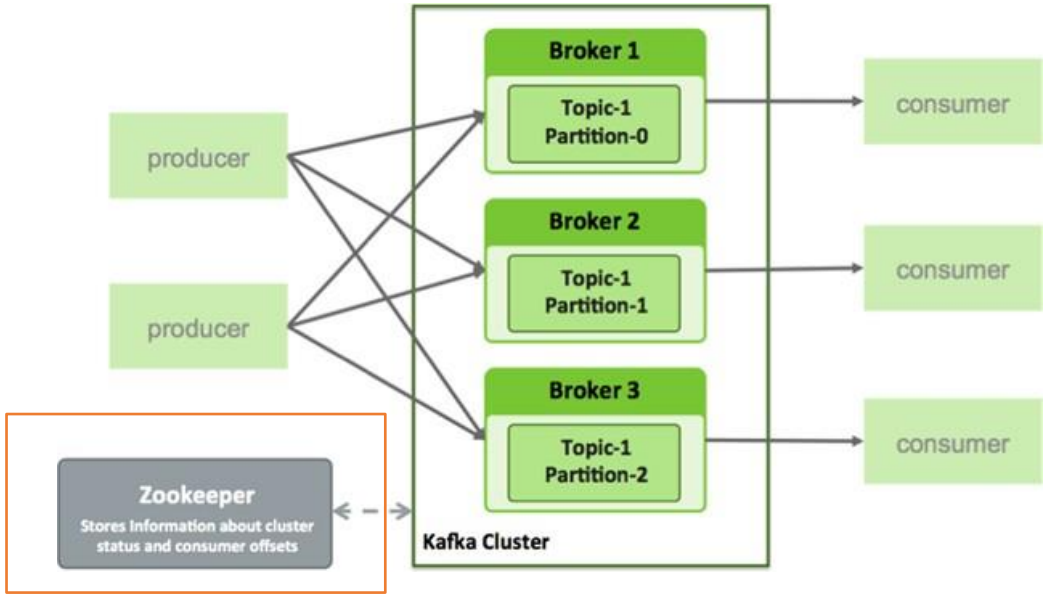
Apache Kafka – Basic Concepts

- Basic Elements
  - Kafka Cluster
    - A Kafka cluster consists multiple brokers
    - Each broker has an integer identification number
    - Each broker can contain multiple partitions of same topic
    - A producer or consumer can connect to any broker and in turn gets connected to the entire cluster



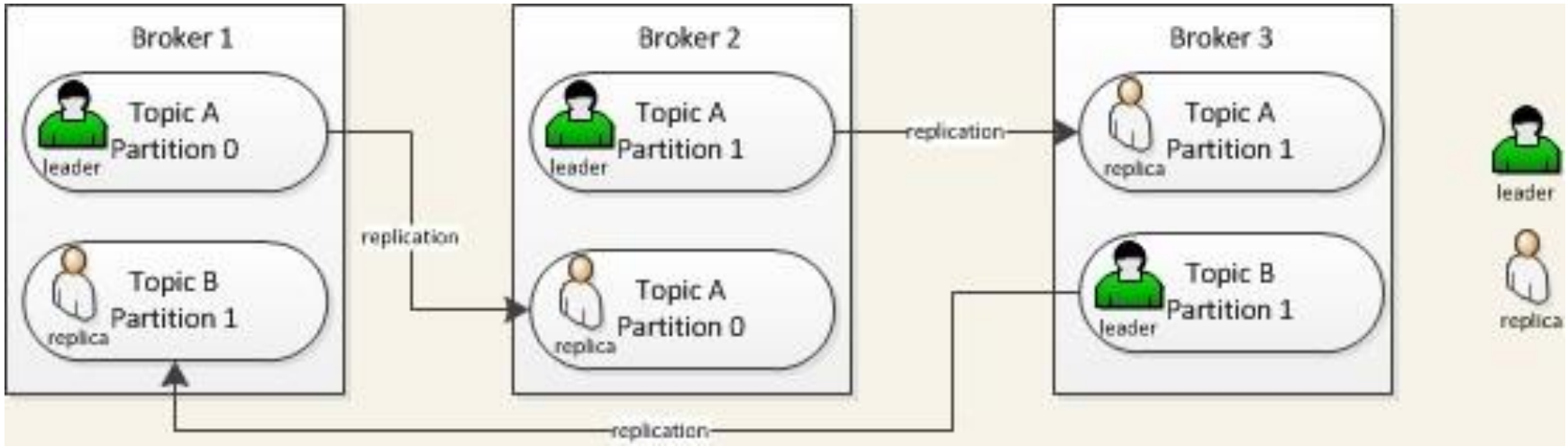
Apache Kafka – Basic Concepts

- Basic Elements
  - Zookeeper
    - It manages Kafka brokers
    - It helps in **maintaining** the cluster membership
    - It also manages **topic configurations** like number of partitions a topic has, leader of the partitions for a topic



Apache Kafka – Basic Concepts

- Basic Elements
  - Partition Replication
    - Replication happens in the **partition level**
    - Kafka partitions are replicated within Kafka cluster
    - A broker can hold one ore more partitions that belong to **the same** or **different** topics
    - A partition can be **leader** or **follower**



Intelligencia

Apache Kafka – Basic Concepts

- Basic Elements
  - Leader
    - A leader is the primary replica of a partition.
    - Leader handles all reads and writes of records for the partition
    - All writes and reads to a topic go through the leader
    - The leader coordinates updating replicas with new data
    - If a leader fails, a replica takes over as the new leader
  - Follower
    - The secondary replica of a partition