KAFKA ADMINISTRATION

Đơn vị: Công ty CP Giáo dục và Công nghệ QNET

QNET JOINT STOCK COMPANY

Address: 14th Floor, VTC Online Tower
18 Tam Trinh Street. Hoang Mai District
Hanoi, Vietnam

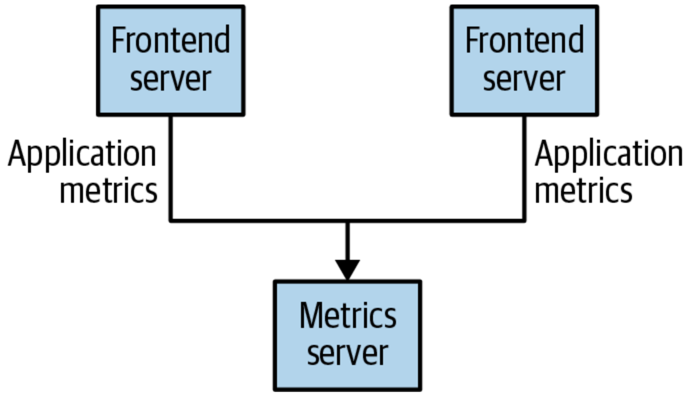
Quality Network for Education and Technology

NỘI DUNG

kafka

KAFKA: Publish/Subscribe Messaging

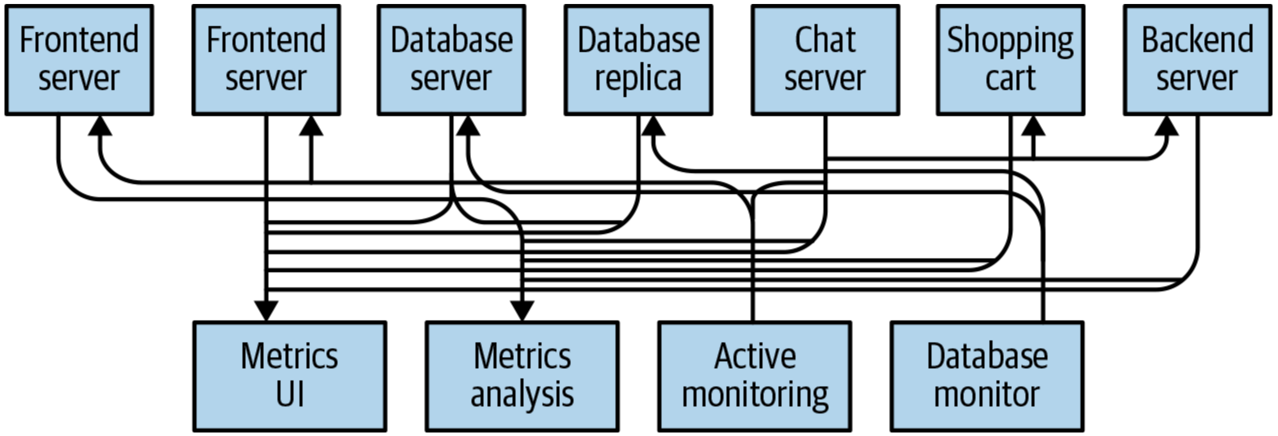
Single, direct metrics publisher



Simple publish/subscribe system with a simple message queue
Or interprocess communication channel:
- Push metrics from frontend server to metrics server to display on its dashboard

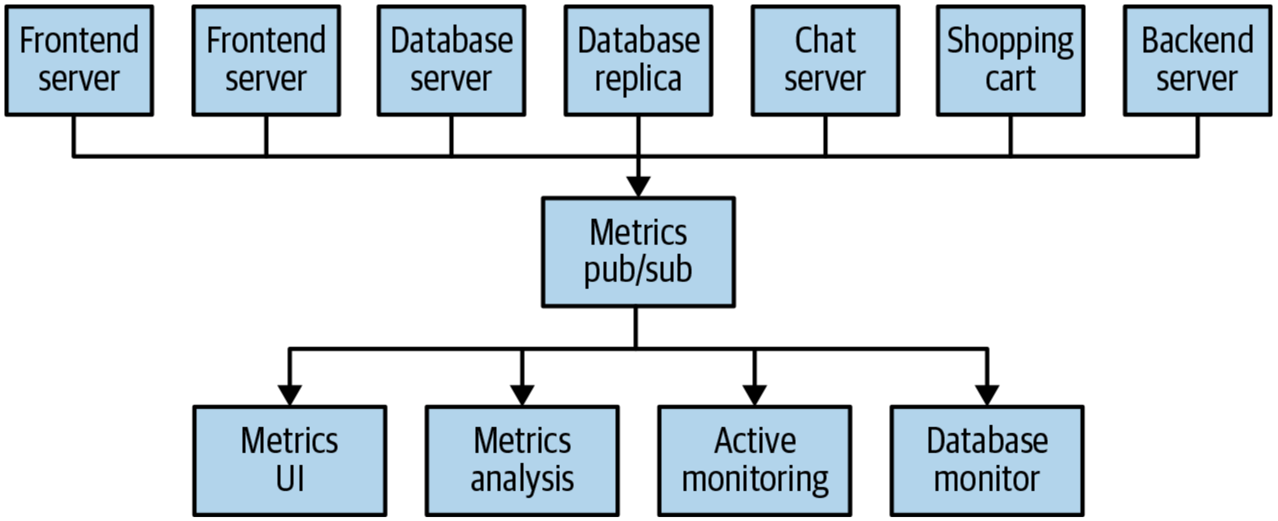
Introduction

Many metrics publishers, using direct connection

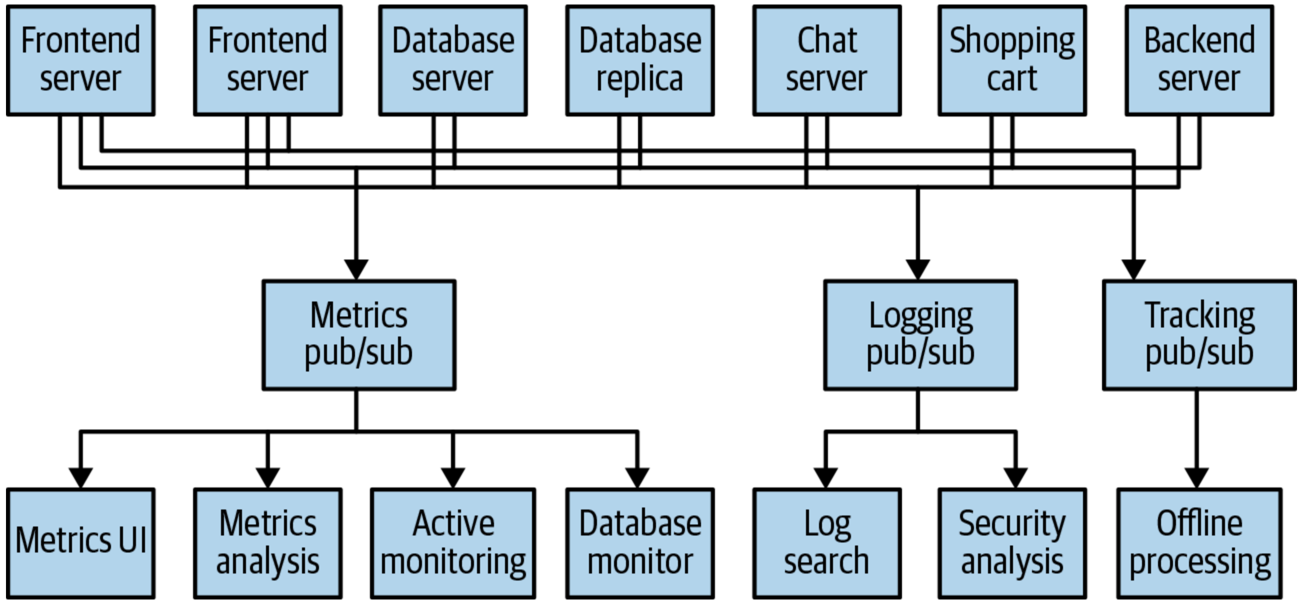


Introduction

Metric publish/subscribe system



Multiple publish/subscribe system



KAFKA

- Apache Kafka is a distributed publish/subscribe messaging system which allows publishing data, which will grow as per your business

- Apache Kafka was developed to solve LinkedIn data pipeline problem Kafka was created at LinkedIn
    - It was designed to provide a high-performance messaging system that can handle user activity and system metrics in real time
    - Release in 2010 as an Github open source project

Features of Kafka

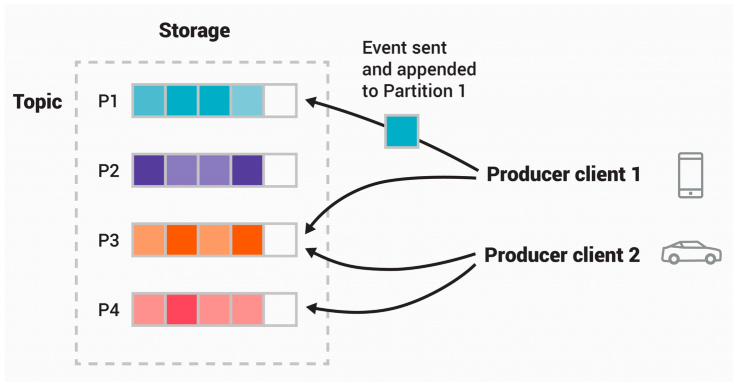
- High throughput
    - Support for millions of messages per second
- Data loss
    - Ensures no data loss
    - Provides compression & Security
- Durability
    - Provides support to persisting messages on disk
- Scalability
    - Highly scalable distributed systems with no downtime
- Stream Processing
    - Kafka can be used along with streaming framework: Spark, Flink, Storm…
- Replication
    - Messages can be replicated across clusters, which supports multiple subscribers

KAFKA USECASE

- Messaging
- Website Activity Tracking
- Metrics
- Log Aggregation
- Stream Processing
- Event Sourcing
- Commit Log

MAIN CONCEPTS AND TERMINOLOGY

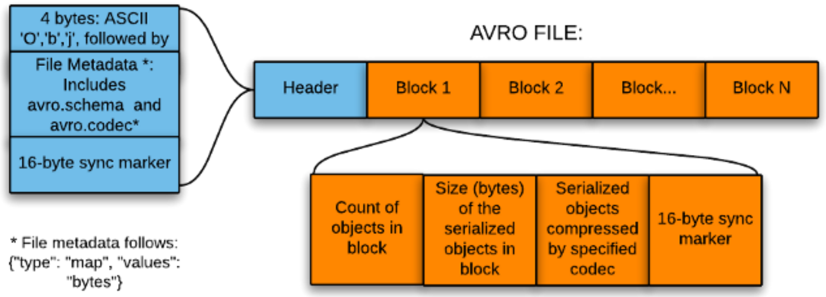
Message (Event)



- An event records the fact that 'something happened' in the world or in your business
- An event is also called record or message
- When you read and write data to Kafka, you do this in the form of events
- An event has:
  - Key
  - Value
  - Timestamp
  - Optional metadata headers
- Example:
  - Event key: Alice
  - Event Value: "Made a payment of $200 to Bob"
  - Event timestamp: "Jun, 25,2020 at 2:06 pm"

MAIN CONCEPTS AND TERMINOLOGY

Schema



AVRO FILE:
4 bytes: ASCII 'O','b','j', followed by
File Metadata *: Includes avro.schema and avro.codec*
16-byte sync marker
Header
Block 1
Block 2
Block...
Block N
Count of objects in block
Size (bytes) of the serialized objects in block
Serialized objects compressed by specified codec
16-byte sync marker
* File metadata follows: {"type": "map", "values": "bytes"}

- JSON
- XML
- Avro
- Protobuf

MAIN CONCEPTS AND TERMINOLOGY

Topics

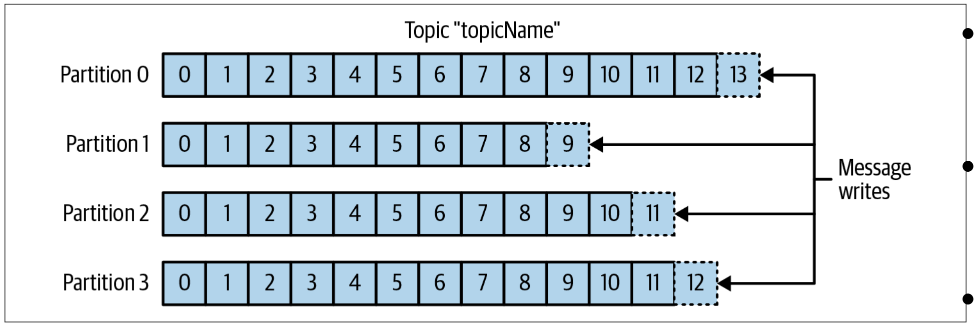


Figure 1-5. Representation of a topic with multiple partitions

- Messages are organized and durably stored in *topics*.
- Topics is similar to a folder in a file system and the events are the files in that folder
- Topics in Kafka are always multi-producers and multi-subcriber
- Messages in a topic can be read as often as needed and are not deleted after consumption unlike traditional messaging systems

MAIN CONCEPTS AND TERMINOLOGY

Partitions

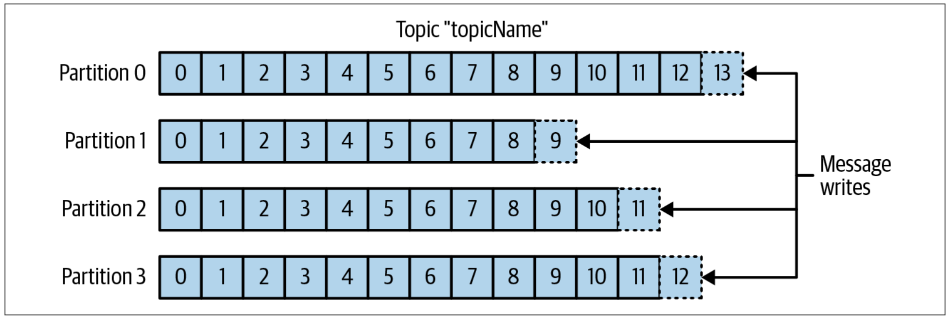
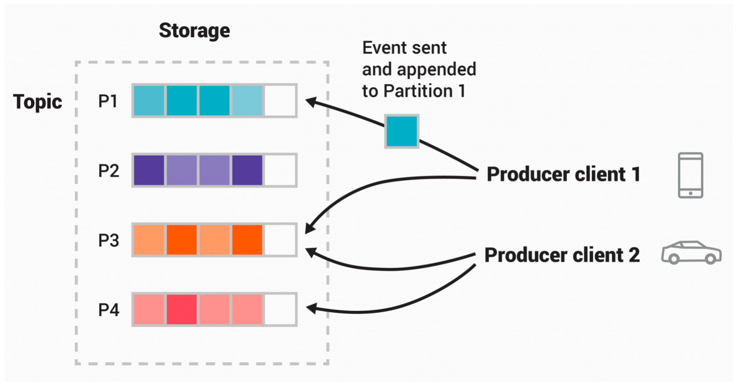


Figure 1-5. Representation of a topic with multiple partitions

- Topics are partitioned, meaning a topic is spread over a number of "bucket" located on different Kafka brokers.
- When a new event is published to a topic, it is actually appended to one of the topic's partitions.
- Events with the same event key are written to the same partition

MAIN CONCEPTS AND TERMINOLOGY

Producers and Consumers



- Producers are those client applications that publish (write) events to Kafka
- Consumers are those that subscribe to (read and process) these events.

MAIN CONCEPTS AND TERMINOLOGY

Producers and Consumers

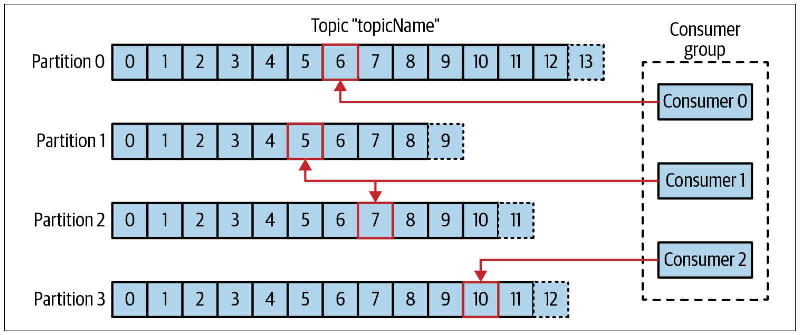
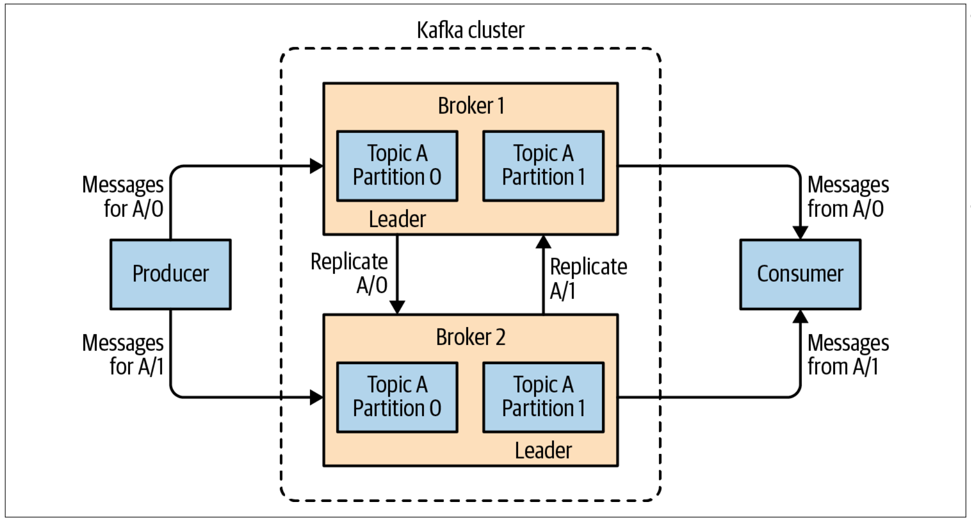


Figure 1-6. A consumer group reading from a topic

- Producers are those client applications that publish (write) events to Kafka
- Consumers are those that subscribe to (read and process) these events.

MAIN CONCEPTS AND TERMINOLOGY

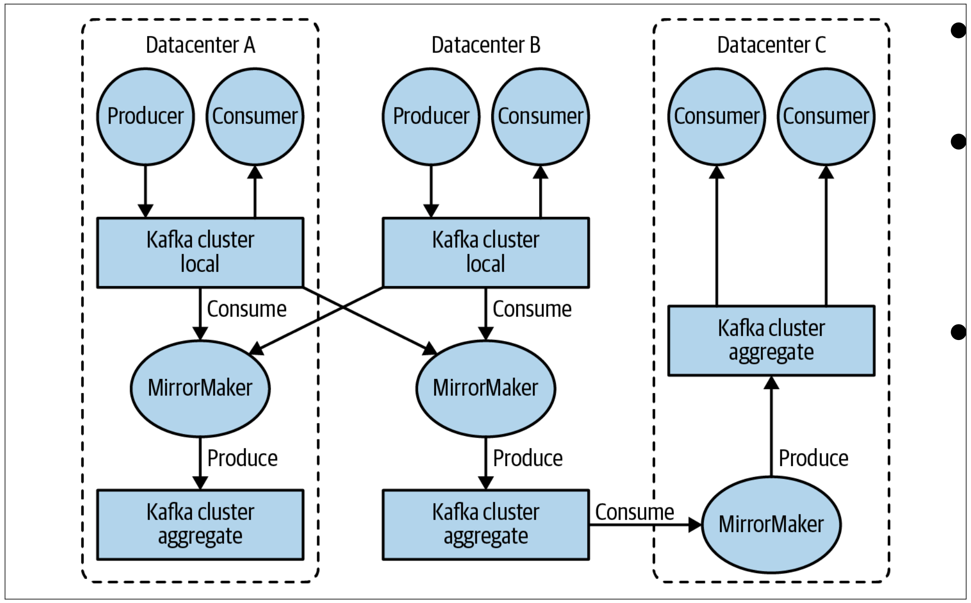
Brokers and Clusters



- A single Kafka server is called a **broker**
- Kafka brokers are designed to operate as part of a **cluster**

MAIN CONCEPTS AND TERMINOLOGY

Multiple Clusters



- Segregation of types of data
- Isolation for security requirements
- Multiple datacenters (DR)

KAFKA API

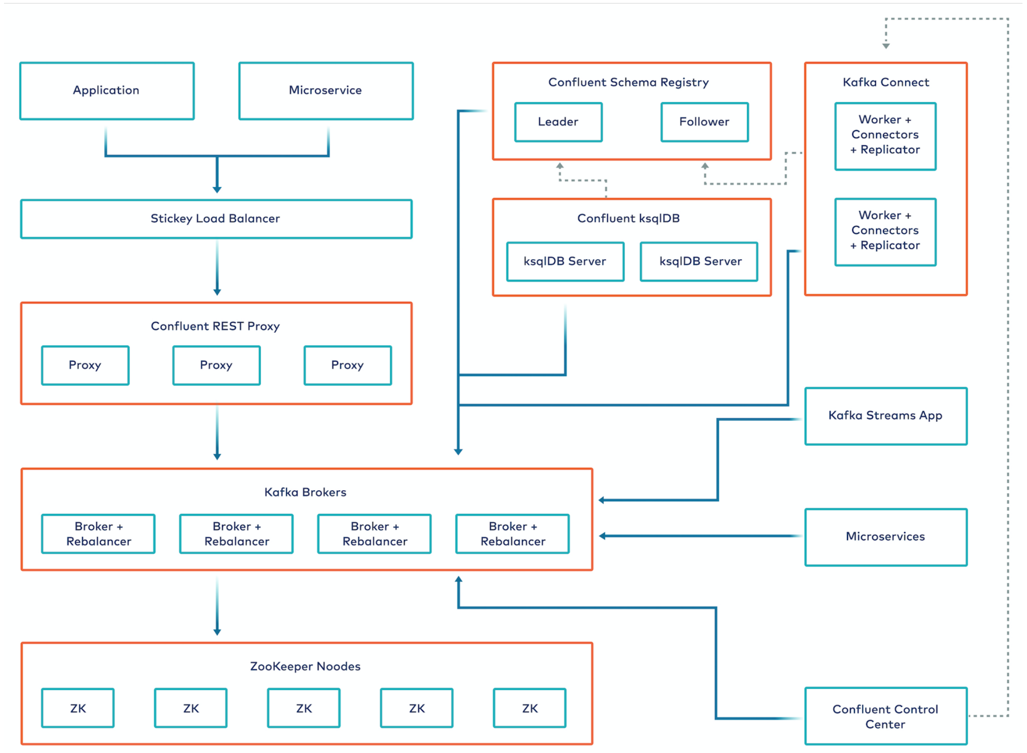
- Admin API: to manage and inspect topics, brokers and other Kafka objects
- Producer API: to publish a stram of events to one or more Kafka topics
- Consumer API: to subscribe one or more topics and to process the stream of events
- Kafka Stream API: to implement stream processing applications and microservers. It provides higher-level functions to process event streams, including transformations, stateful operations like aggregations and joins, windowing....
- Kafka Connect API : to build and run reusable data import/export connectors that consume or producer stream of events

Why Kafka ?

- Multiple Producers
- Multiple Consumers
- Disk-bases retention
- Scalable
- High Performance
- Platform Features

Confluent Platform

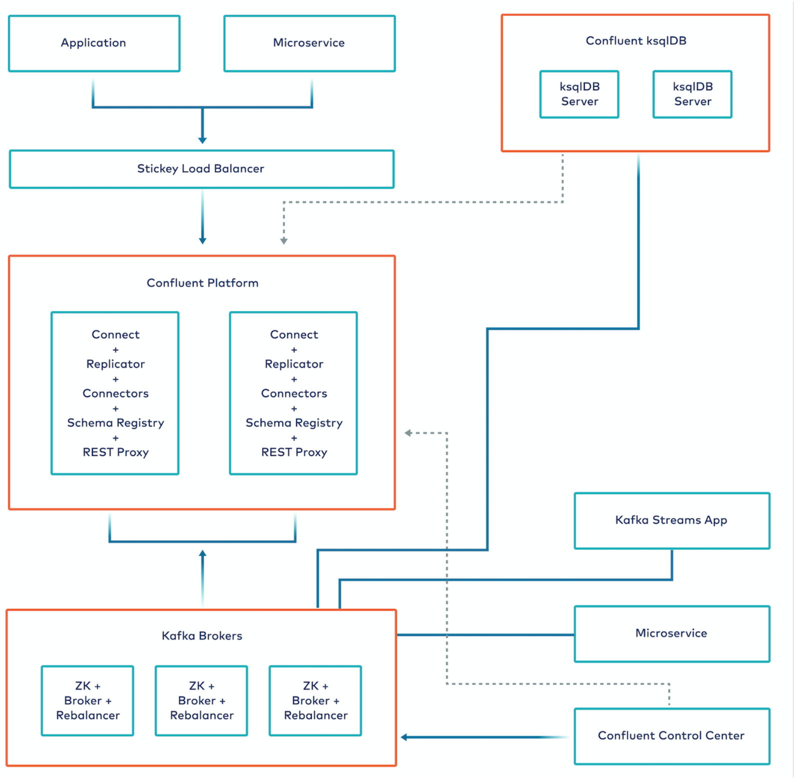
Large Cluster Reference Architecture



Quality Network for Education and Technology

Confluent Platform

Small Cluster Reference Architecture



CONCLUSION

QNET®

Quality Network for Education and Technology

XIN CHÂN THÀNH CẢM ƠN!