

# KAFKA MONITORING AND SCHEMA REGISTRY

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

# Kafka Monitoring and Schema Registry

## Learning Objective

- 1 Monitoring metrics
- 2 Kafka schema registry
- 3 Architecture and Components
- 4 Working: Kafka Schema Registry

# Kafka Monitoring and Schema Registry

**Monitoring and Its important**



Monitoring means to gather statistics about any applications, server or any tool which is running and suggest if it is in good state or not

# Kafka Monitoring and Schema Registry

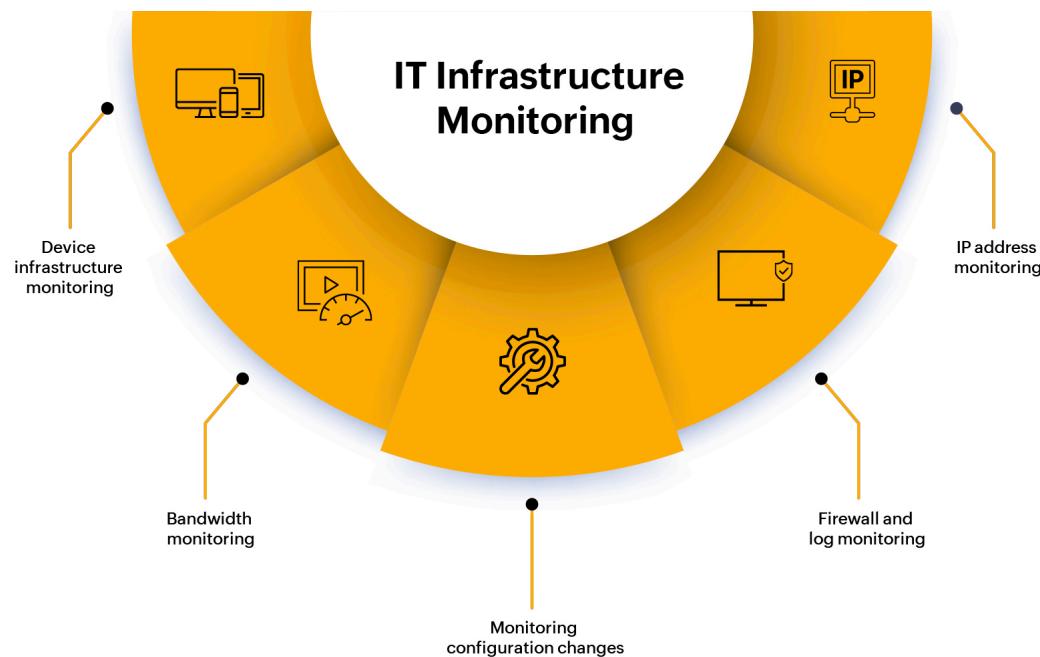
## Why Monitoring?



- 1 Monitoring helps you know more about your applications
- 2 It also helps you to know about any data breach and any attacks on your server
- 3 Helps you to optimize your applications
- 4 Helps to do capacity planning

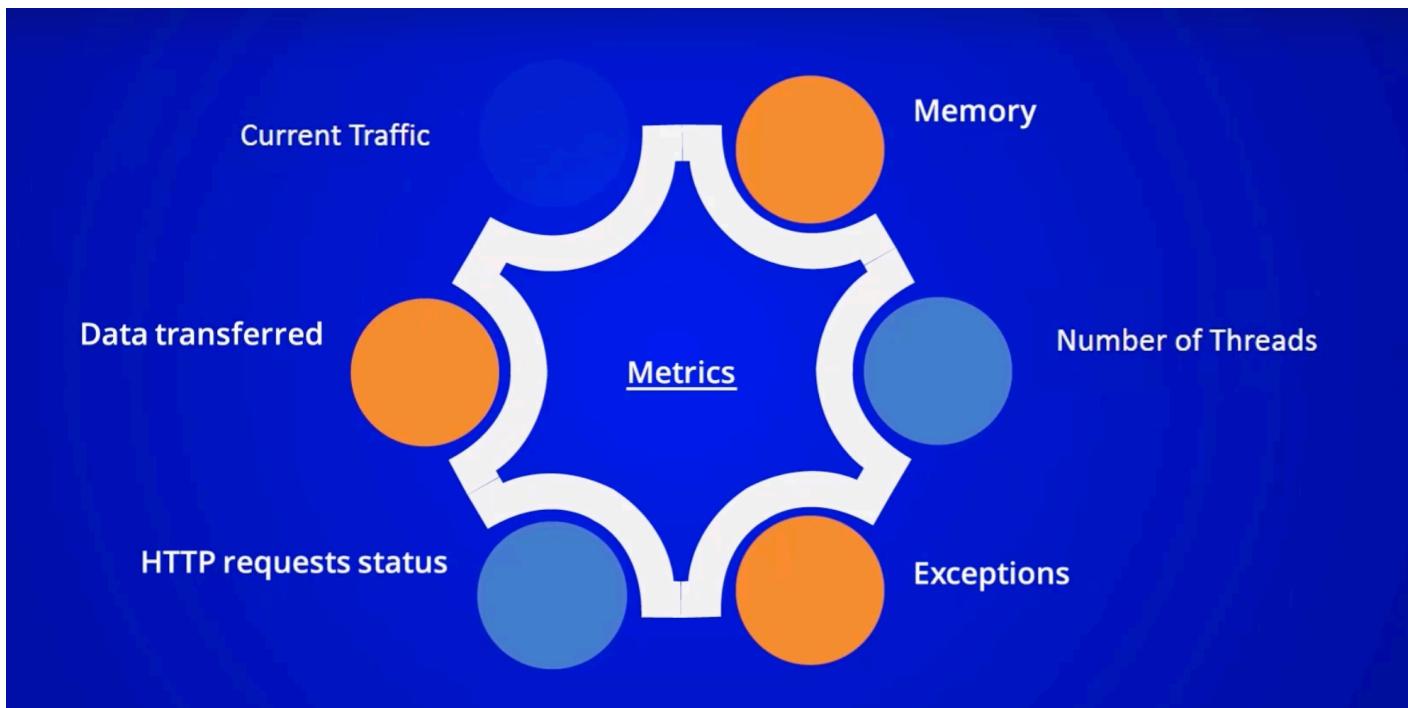
# Kafka Monitoring and Schema Registry

## Infrastructure Monitoring



# Kafka Monitoring and Schema Registry

## Application Monitoring



# Kafka Monitoring and Schema Registry

## About Kafka Monitoring?

- 1 When you want to know how producer and consumer is working and their status
- 2 Most of managed Kafka providers already provide tools to monitor Kafka performance
- 3 We have many open sources and paid tools to monitor Kafka



# Kafka Monitoring and Schema Registry

## Metric sources



Category	Description
Application metrics	These are the metrics you get from Kafka itself, from the JMX interface.
Logs	Another type of monitoring data that comes from Kafka itself. Because it is some form of text or structured data, and not just a number, it requires a little more processing.
Infrastructure metrics	These metrics come from systems that you have in front of Kafka but are still within the request path and under your control. An example is a load balancer.
Synthetic clients	This is data from tools that are external to your Kafka deployment, just like a client, but are under your direct control and are typically not performing the same work as your clients. An external monitor like Kafka Monitor falls in this category.
Client metrics	These are metrics that are exposed by the Kafka clients that connect to your cluster.

# Kafka Monitoring and Schema Registry

## Access Kafka metrics



All of the metrics exposed by Kafka can be accessed via the Java Management Extension (JMX) interface

To access from external monitoring system:

- Use a collection agent which may be a separate process that runs on the system and connects to the JMX interface
  - E.g: Nagios XI check\_jmx plugin or jmxtrans
- Use JMX agent that runs directly in the Kafka process to access metrics via an HTTP connection
  - E.g: Jolokia or MX4j

JMX information:

- Broker sets the configured JMX port in the broker information (stored in Zookeeper)
- /brokers/ids/znode contains hostname and jmx\_port keys

# Kafka Monitoring and Schema Registry

## Diagnosing Cluster Problem



When it comes to problems with a Kafka cluster, there are three major categories:

- Single-broker problems:
  - Easiest to detect, diagnose and respond to
  - To detect this types of problem, make sure you are monitoring the availability of the individual servers, status of storage devices, utilizing the operating system...
  - It is the cause of almost an imbalance in the load of the Kafka Cluster => client can't access balance data
- Overloaded clusters:
  - Easy to detect
  - If the cluster is balanced and many of the brokers are showing elevated latency for requests or a low request handler pool idle ratio, you are reaching the limits of your brokers to server traffic for this cluster
- Controller problems:
  - Problems with the controller of Kafka cluster are much more difficult to diagnose and often fall into the category of bugs in Kafka itself
  - This issues broker metadata out of sync, offline replicas when the brokers appear to be find and topic control actions like creation not happening property

# Kafka Monitoring and Schema Registry

## Cluster level problems

- Cluster problems usually fall into one of two categories:
  - Unbalanced load
  - Resource exhaustion
- To detect unbalanced load, you will need several metrics from the brokers in the cluster”

Broker	Partitions	Leaders	Messages in	Bytes in	Bytes out
1	100	50	13130 msg/s	3.56 MBps All topics messages in rate	9.45 MBps All topics bytes in rate
2	101	49	12842 msg/s	3.66 MBps All topics bytes in rate	9.25 MBps All topics bytes out rate
3	100	50	13086 msg/s	3.29 MBps To detect resource exhaustion, you will need several metrics from OS level, including:	9.82 MBps

- CPU utilization
- Inbound network throughput
- Outbound network throughput
- Disk average wait time
- Disk percent utilization
- In a perfectly balanced cluster, the numbers will be even across all broker in the cluster



# Kafka Monitoring and Schema Registry

Cluster level problems



Broker	Partitions	Leaders	Messages in	Bytes in	Bytes out
1	100	50	13130 msg/s	3.56 MBps	9.45 MBps
2	101	49	12842 msg/s	3.66 MBps	9.25 MBps
3	100	50	13086 msg/s	3.23 MBps	9.82 MBps

# Kafka Monitoring and Schema Registry

## Host level problems



- If the performance problem with Kafka is not present in the entire cluster and can be isolated to one or two brokers, it's time to examine that server and see what makes it different from the rest of the cluster. These types of problem fall into several general categories:
  - Hardware failures:
    - Bad bit of memory
    - CPU failure
    - Disk failure
  - Networking
  - Conflicts with another process
  - Local configuration differences

# Kafka Monitoring and Schema Registry

## Kafka Metrics



- 1 Broker-level Metrics
- 2 Topic level Metrics
- 3 Partition-level Metrics

# Kafka Monitoring and Schema Registry

## Kafka Broker Metrics

- Under-Replicated Partitions
- Active Controller Count
- Request handler idle ratio
- All topics bytes in
- All topics bytes out
- Partition count
- Leader count
- Offline partitions
- Request metrics

Count of the number of partitions where broker is leader replica but follower replicas are lagging

- If Under-replicated partitions metrics value = 0
  - No issue
- If value not 0 then
  - Might be the issue



# Kafka Monitoring and Schema Registry

## Kafka Broker Metrics

Under-Replicated Partitions
Active Controller Count
Request handler idle ratio
All topics bytes in
All topics bytes out
Partition count
Leader count
Offline partitions
Request metrics

Issue

Many broker of the cluster reporting unchanging number of under-replicated partitions

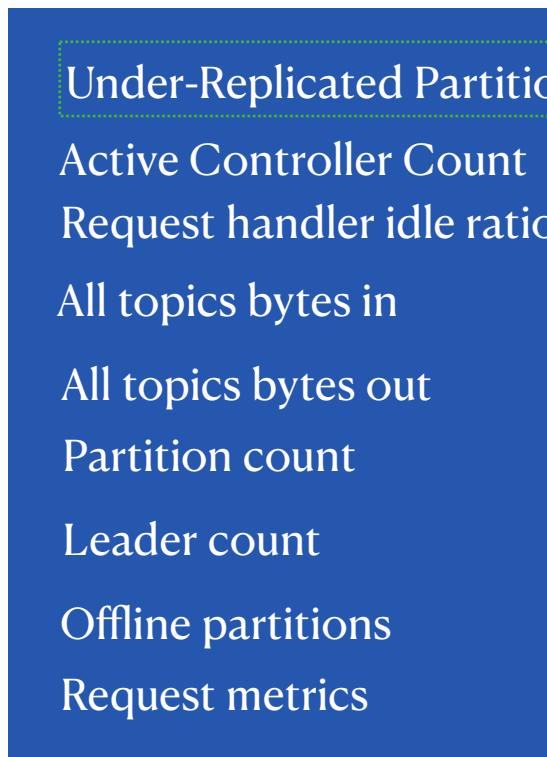
Possible

One of the brokers in the cluster is off-line

- Often a hardware failure
- May be an OS or Java issue

# Kafka Monitoring and Schema Registry

## Kafka Broker Metrics



Issue

Possible

Number of under-replicated partitions is fluctuating

or

not fluctuation but no brokers are off-line

- Indicates a performance issue in the cluster
- Need to determine if the problem is specific to a single broker or to the entire cluster
- If its on single broker (host-level/broker level issue):
  - That particular broker is the problem where other brokers are having a problem replicating messages from that one
  - If multiple brokers have under-replicated partitions
    - Potential cluster problem

# Kafka Monitoring and Schema Registry

## Kafka Broker Metrics

Under-Replicated Partitions	Issue
Active Controller Count	
Request handler idle ratio	
All topics bytes in	
All topics bytes out	
Partition count	
Leader count	Possible
Offline partitions	
Request metrics	

Number of under-replicated partitions is fluctuating  
or  
not fluctuation but no brokers are off-line

- Indicates a performance issue in the cluster
- Need to determine if the problem is specific to a single broker or to the entire cluster
- If its on single broker (host-level/broker level issue):
  - That particular broker is the problem where other brokers are having a problem replicating messages from that one
- If multiple brokers have under-replicated partitions
  - Potential cluster problem

# Kafka Monitoring and Schema Registry

## Kafka Broker Metrics

Indicates whether the broker is currently the controller for the cluster

- o: Broker is not the controller
- 1: Broker is the controller

- Under-Replicated Partitions
- Active Controller Count
- Request handler idle ratio
- All topics bytes in
- All topics bytes out
- Partition count
- Leader count
- Offline partitions
- Request metrics

- If Under-replicated partitions metrics value = o
  - No issue
- If value not o then
  - Might be the issue

- Problem indication:
  - More than 1 broker indicate that they are controller
  - No broker indicate that it is the controller

- Solution:
  - Restart all those broker threads
  - Debug further to identify why controller threads are not working properly

# Kafka Monitoring and Schema Registry

## Kafka Broker Metrics

- Under-Replicated Partitions
- Active Controller Count
- Request handler idle ratio
- All topics bytes in
- All topics bytes out
- Partition count
- Leader count
- Offline partitions
- Request metrics

Indicates percentage of time the request handlers are not in use

Lower value: Broker is more loaded

- Problem indication:
  - Idle ratio < 20%
- Possible Reasons:
  - Not enough threads in the pool
  - Threads are doing unnecessary work for each request
- Solution:
  - Set the number of request handler threads = number of processors in the system
  - Make sure that all producer and consumer clients supports the 0.10 message format

# Kafka Monitoring and Schema Registry

## Kafka Broker Metrics

How much message traffic your brokers are receiving from producing clients

- Under-Replicated Partitions
- Active Controller Count
- Request handler idle ratio
- All topics bytes in
- All topics bytes out
- Partition count
- Leader count
- Offline partitions
- Request metrics

- Attributes:
  - EventType - Unit of measurement for all the attributes
  - RateUnit - Time periods unit for the rate
  - OneMinuteRate - An average over the previous 1 minute
  - FiveMinuteRate - An average over the previous 5 minutes
  - FifteenMinuteRate - An average over the previous 15 minutes
  - MeanRate - An average since the broker was started

# Kafka Monitoring and Schema Registry

## Kafka Broker Metrics

Under-Replicated Partitions  
Active Controller Count  
Request handler idle ratio  
All topics bytes in  
**All topics bytes out**  
Partition count  
Leader count  
Offline partitions  
Request metrics

Show the rate at which consumers are reading messages out



- Important to observe the metrics trend over time period

# Kafka Monitoring and Schema Registry

## Kafka Broker Metrics

- Under-Replicated Partitions
- Active Controller Count
- Request handler idle ratio
- All topics bytes in
- All topics bytes out
- All topics messages in
- Partition count
- Leader count
- Offline partitions
- Request metrics

Shows: The number of individual messages (regardless of their size, produced per second)

- Useful as a growth metric as a different measure of producer traffic
- Can also be used in conjunction with the bytes in rate to determine an average message size
- Will help to identify imbalance in the brokers as well that will alert you to maintenance work that may be required

# Kafka Monitoring and Schema Registry

## Kafka Broker Metrics

Under-Replicated Partitions  
Active Controller Count  
Request handler idle ratio  
All topics bytes in  
All topics bytes out  
All topics messages in  
**Partition count**  
Leader count  
Offline partitions  
Request metrics

Total number of partitions assigned to that broker

- Includes every replica the broker has inclusive of both leader and follower for that partition

# Kafka Monitoring and Schema Registry

## Kafka Broker Metrics

Under-Replicated Partitions  
Active Controller Count  
Request handler idle ratio  
All topics bytes in  
All topics bytes out  
All topics messages in  
Partition count  
**Leader count**  
Offline partitions  
Request metrics

Shows the number of partitions for which the broker is currently the leader

Should be even across the brokers in the cluster

Problem indication: Metric will show fewer leaders, or often zero

Solution: Perform preferred replica election in order to do leadership rebalancing in the cluster

In a balanced cluster [replication factor of 2], all brokers should be leaders for approximately 50% of their partitions

# Kafka Monitoring and Schema Registry

## Kafka Broker Metrics

Under-Replicated Partitions  
Active Controller Count  
Request handler idle ratio  
All topics bytes in  
All topics bytes out  
All topics messages in  
Partition count  
Leader count  
Offline partitions  
Request metrics

This measurement is only provided by the broker that is the controller for the cluster (all other brokers will report 0)

In a production Kafka cluster, an offline partition may be impacting the producer clients and will result into:

- Lossing messages
- Causing back-pressure in the application

This is most often a ‘site down’ type of problem and shall be addressed immediately

# Kafka Monitoring and Schema Registry

## Kafka Broker Metrics

How each of the requests performs

Under-Replicated Partitions  
Active Controller Count  
Request handler idle ratio  
All topics bytes in  
All topics bytes out  
All topics messages in  
Partition count  
Leader count  
Offline partitions  
Request metrics

- ApiVersions
- CreateTopics
- FetchConsumer
- GroupCoordinator
- JoinGroup
- LeaveGroup

For each of these requests, there are eight Metrics provided, providing insight into each of the phases of the request Processing

# Kafka Monitoring and Schema Registry

## Kafka Topic Specific Metrics

Bytes in rate

Bytes out rate

Failed fetch rate

Failed produce rate

Message in rate

Fetch request rate

Produce request rate

Measurements are very similar to the broker metrics  
Only different is that you need to provide topic name to fetch topic specific metrics



# Kafka Monitoring and Schema Registry

## Kafka Partition Specific Metrics

Partition

Less useful metrics

Quite numerous as hundreds of topics can easily be thousands of partitions

The amount of data (in bytes) that is currently being retained on disk for the partition

Log

The number of log - segment files on disk for the partition. This may be useful along with the partition size for resource tracking

Log end

Highest offsets for messages in that partition

Log start

Lowest offsets for messages in that partition

A discrepancy between the size of two Partitions for the same topic can indicate a problem where the messages are not evenly distributed across the key that is being used

# Kafka Monitoring and Schema Registry

## Logging and clients monitoring

### Logger

kafka.controller

- INFO level
- To provide only cluster controller message logs
- Only one broker will be writing to this logger
- Log information:
  - Topic creation and

kafka.server.ClientQuotaMan

- INFO level
- To provide log for messages related to produce and consume quota activities

# Kafka Monitoring and Schema Registry

## Logging and clients monitoring

### Logger

kafka.log.LogCleaner  
kafka.log.Cleaner

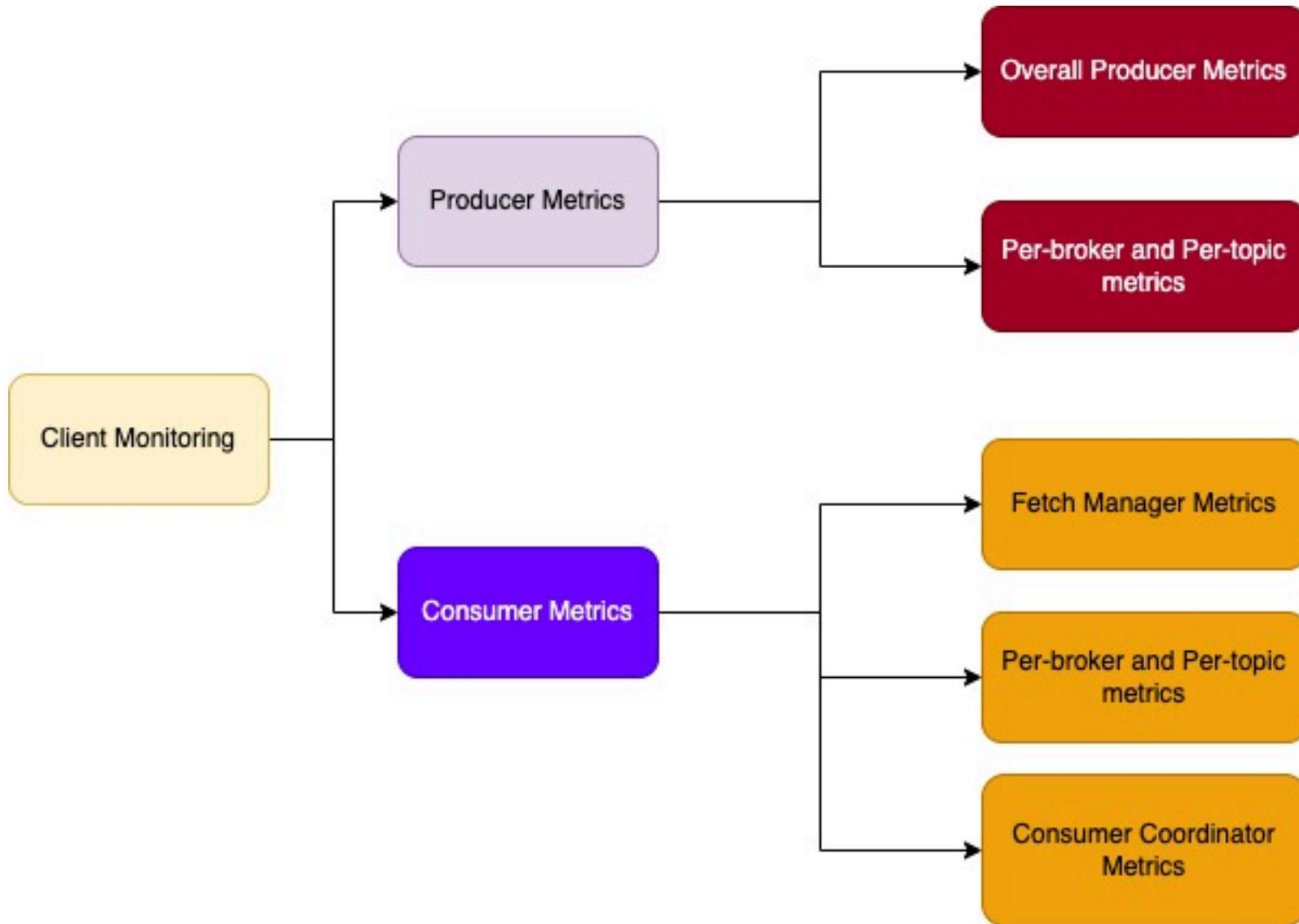
- DEBUG level
- Information about the status of log compaction threads

kafka.request.logger

- DEBUG or TRACE levels
- Logs information about every requests sent to the broker
- Log information at DEBUG level:
  - Connection end points
  - Request timings
  - Summary information

# Kafka Monitoring and Schema Registry

## Logging and clients monitoring



# Kafka Monitoring and Schema Registry

## Producer Metrics - Overall Producer Metrics

Metrics	Description
record-error-rate	Ideal case zero, If its > 0, the producer is dropping messages it is trying to send to the Kafka brokers
record-retry-rate	Track retries
request-latency-avg	Average amount of time a produce request takes to reach to the brokers Request latency will increase if produce requests are getting slower
outgoing-byte-rate	Messages in absolute size [bytes per second]
record-send-rate	Number of messages produced per second
request-rate	Number of produce requests per second, sent to the brokers
request-size-avg	Average size of the produce requests[in bytes]
batch-size-avg	Average size of a single message batch [in bytes]

# Kafka Monitoring and Schema Registry

## Producer Metrics - Overall Producer Metrics

Metrics	Description
record-size-avg	Average size of a single record [in bytes]
records-per-request-avg	Average number of messages that are in a single produce request
record-queuetime-avg	Average amount of waiting time (in milliseconds) for a single message [Difference between time the application has sends and it has been produced to Kafka]

# Kafka Monitoring and Schema Registry

## Producer Metrics - Overall Producer Metrics

Metrics	Description
record-size-avg	Average size of a single record [in bytes]
records-per-request-avg	Average number of messages that are in a single produce request
record-queuetime-avg	Average amount of waiting time (in milliseconds) for a single message [Difference between time the application has sends and it has been produced to Kafka]

# Kafka Monitoring and Schema Registry

## Producer Metrics - Per-broker and per-topic Metrics

Per-broker Metrics	Description
request-latency-avg	Mostly stable Can still show a problem with connections to
Per-partition Metrics	
record-send-rate	Can be used to isolate dropped messages to a specific topic
record-error-rate	
byte-rate	Provides the overall messages rate in bytes per second for the topic.

# Kafka Monitoring and Schema Registry

## Consumer Metrics



Metrics provided by the consumer are useful to look at but not useful for setting up alerts on.

Fetch manager metrics

Per-broker and per-topic metrics

Consumer coordinator metrics

Metrics	Description
fetch-latency-avg	How long fetch requests to the brokers take (depends on producer metrics <code>fetch.min.bytes</code> and <code>fetch.max.wait.ms</code> )
bytes-consumed-rate	Message traffic consumed by this client instance in bytes per second
records-consumed-rate	Message traffic consumed by this client instance in messages per second
fetch-rate	Number of fetch requests per second that the consumer is performing
fetch-size-avg	Average size of those fetch requests in bytes

# Kafka Monitoring and Schema Registry

## Consumer Metrics

Fetch manager metrics

Per-broker and per-topic metrics

Consumer coordinator metrics

Per-Broker Metrics	Description
incoming-byte-rate	Break down the consumed message metrics provided by the fetch manager into per-broker bytes per second
Incoming-request-rate	Break down the consumed message metrics provided by the fetch manager into per-broker requests per second
Per-Topic Metrics	
bytes-consumed-rate	Absolute size in bytes consumed per second for the specific topic
records-consumed-rate	Absolute size in messages consumed per second for the
fetch-sizeavg	Average size of each fetch request for the topic in bytes.

# Kafka Monitoring and Schema Registry

## Quotas

To limit permitted amount of traffic (in bytes/second) from a specific client-ID

- 1 Configurable for both producer and consumer clients
- 2 Broker configuration allows to set default value of quotas for all clients.
- 3 When client exceeds its quota:  
Broker slows down the client by holding the response back to the client for enough time to keep the client under the quota.
- 4 Quota can be overridden for specific client.

# Kafka Monitoring and Schema Registry

## Lag Monitoring



Most important metrics to monitor for Kafka consumers.

Recap of Lag

Difference between the last message produced and the last message processed by the consumer [It is specific to a partition and is measured in number of messages]

Preferred method

- Have an external process to:
  - Watch states of the partition on the broker
  - Track offset of the most recent produced message
  - Watch state of the consumer
  - Track offset of the last committed offset from a consumer group
- Can use Burrow tool

# Kafka Monitoring and Schema Registry

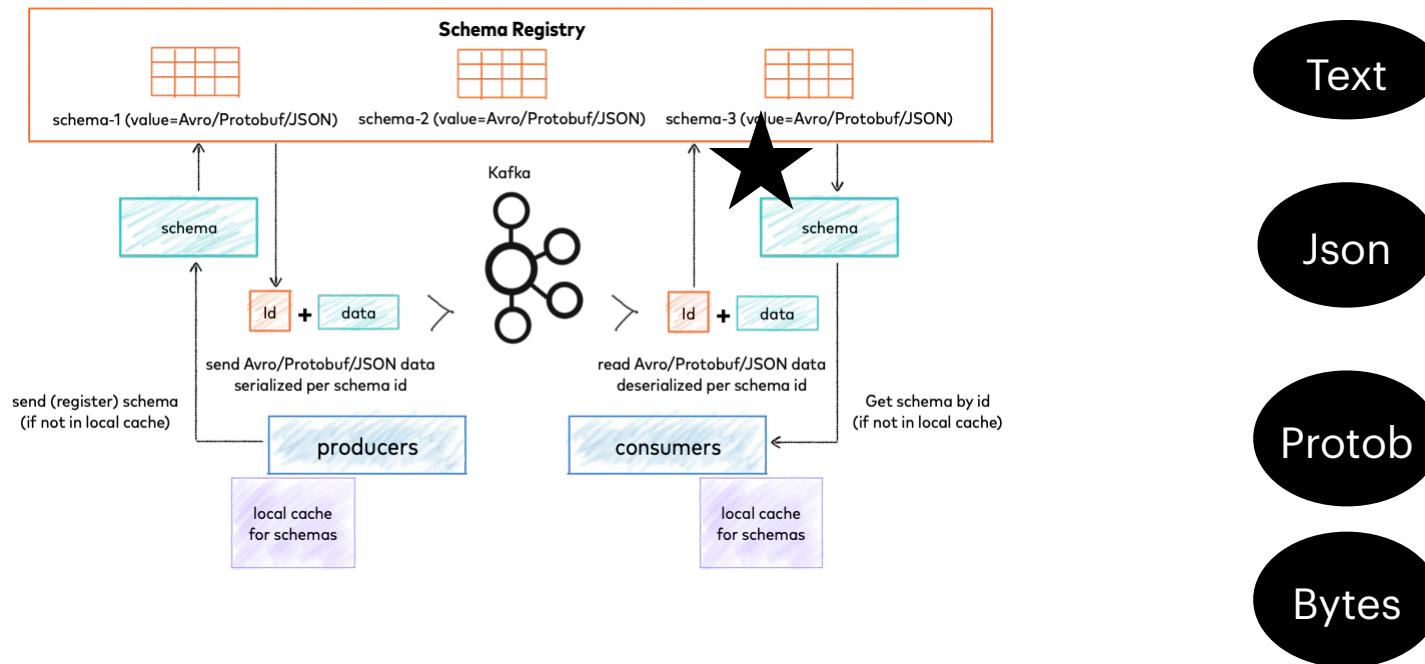
**DEMO: Kafka Offset Monitoring Tools**

# **DISCUSSION**

# Kafka Schema Registry

## Introduction

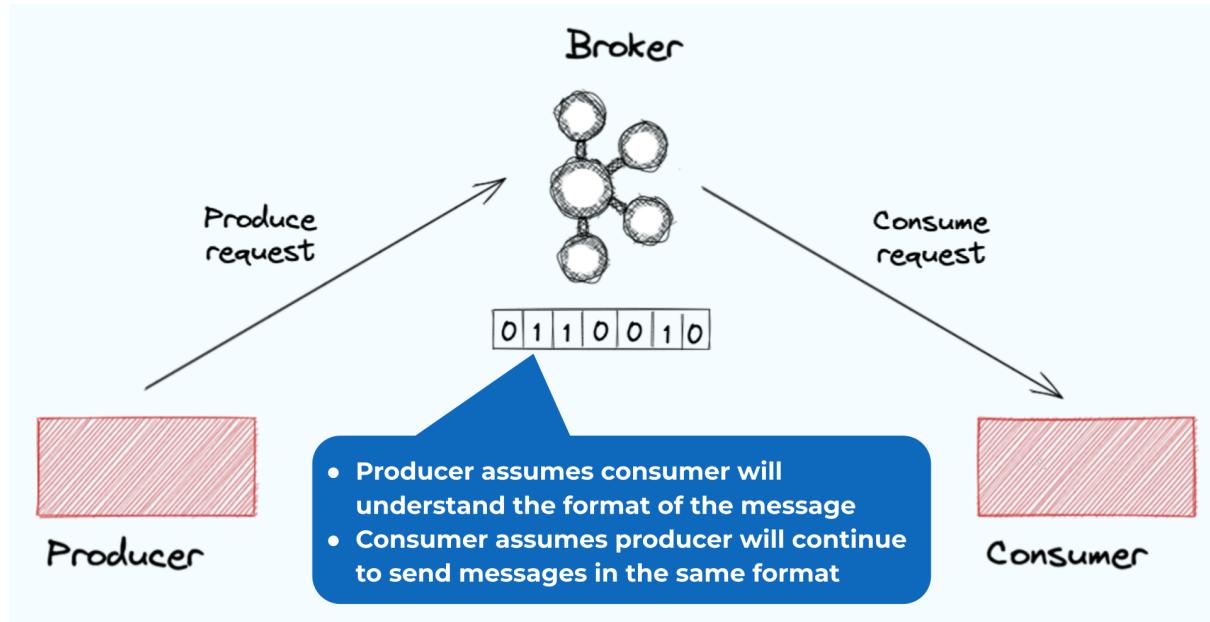
Kafka schema registry is the service which manages schemas and their versions



# Kafka Schema Registry

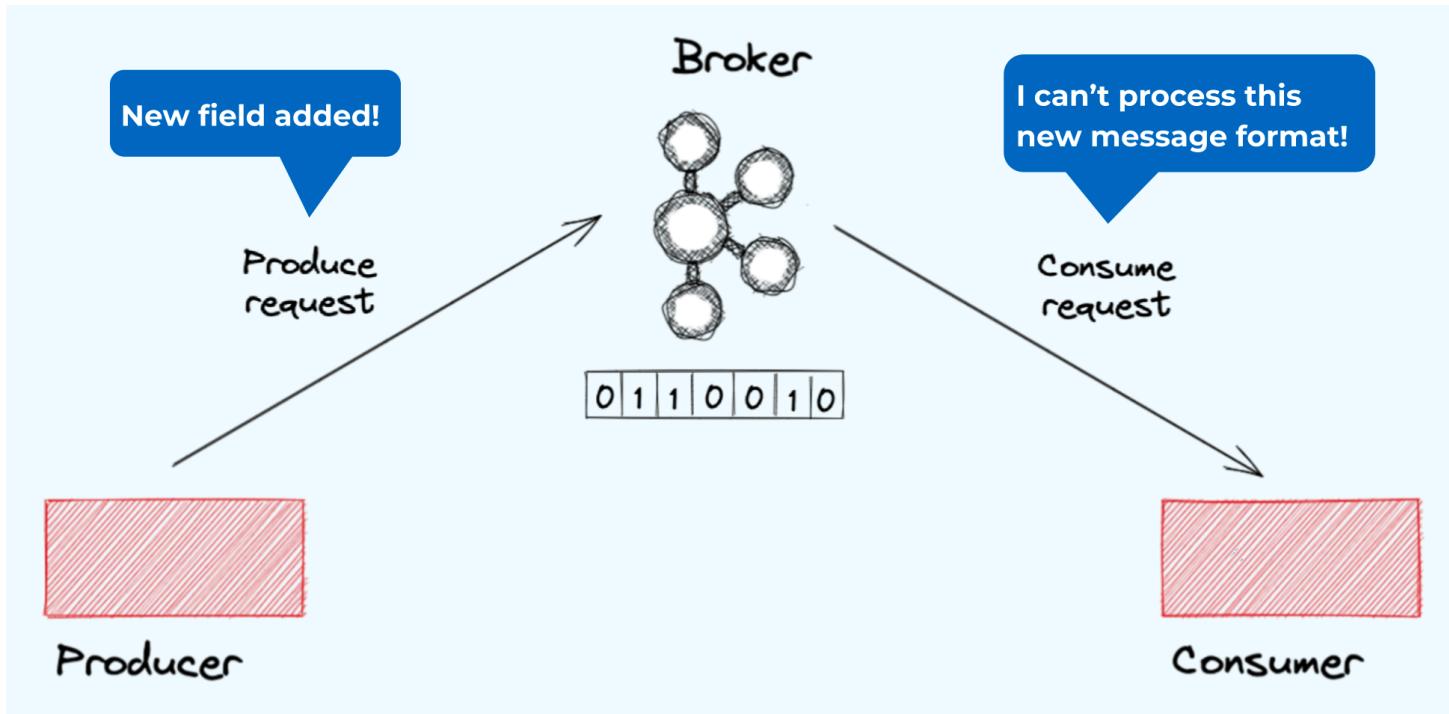
## Introduction

Implied contract between Kafka applications



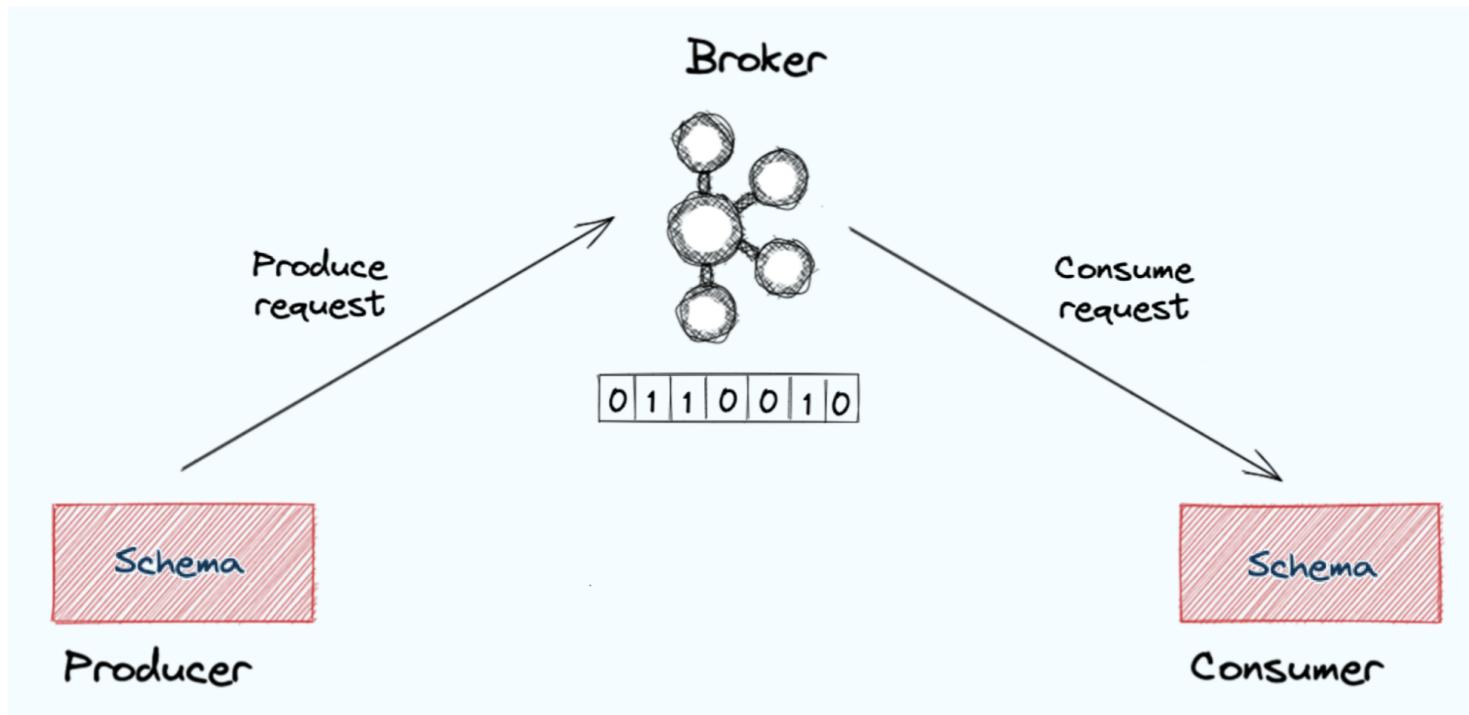
# Kafka Schema Registry

## Introduction



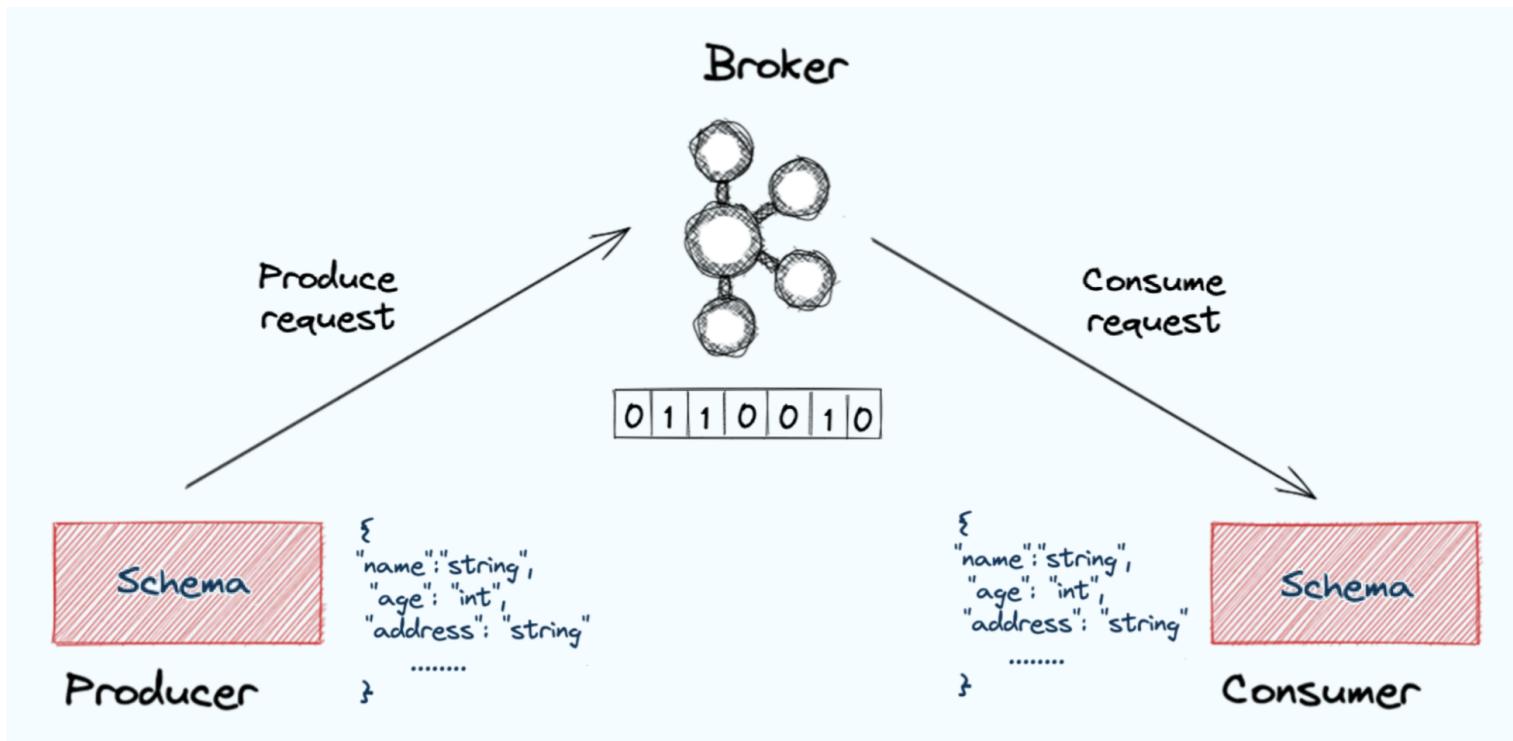
# Kafka Schema Registry

The Schema is a contract



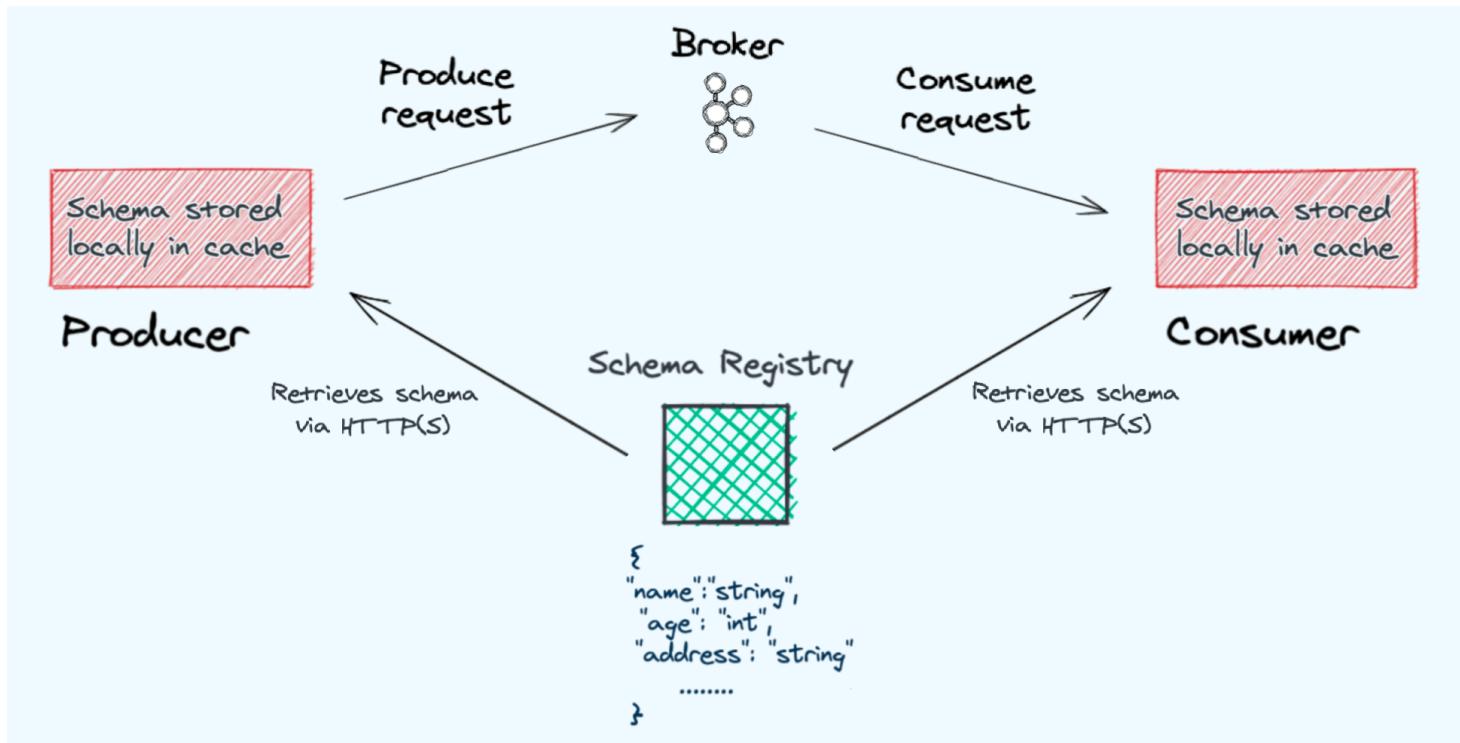
# Kafka Schema Registry

Schema establishes the format of message



# Kafka Schema Registry

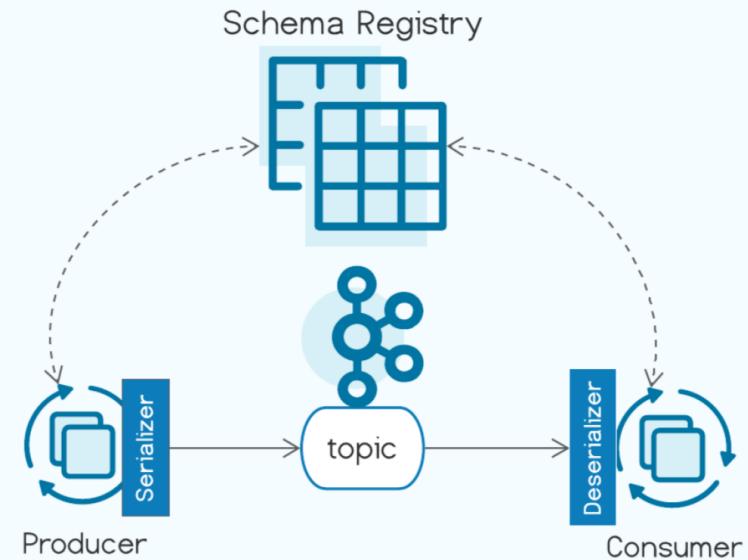
Schema Registry is a contract arbitrator



# Kafka Schema Registry

## Confluent Schema Registry

- **Versioned** schema repository
- **Safe** schema evolution
- **Resilient** data pipelines
- **Enhanced** data integrity
- **Reduce** storage and computation
- **Discover** your data
- **Cost-efficient** ecosystem



# **Kafka Schema Registry**

**Confluent Schema Registry: Demo Kafka Schame Registry with Kafka**

# **Conclusion**



Quality Network for Education and Technology

XIN CHÂN THÀNH CẢM ƠN!