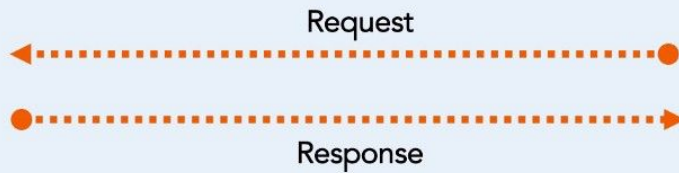


Request/Response



Server

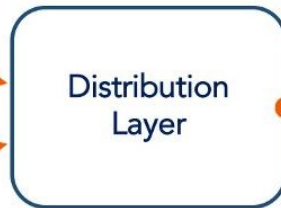


Client

Evented



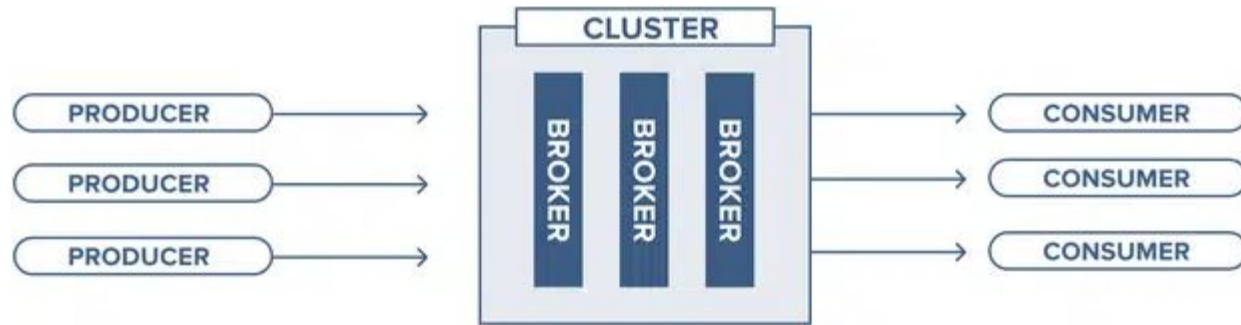
Generators



Long-Lived
Connections



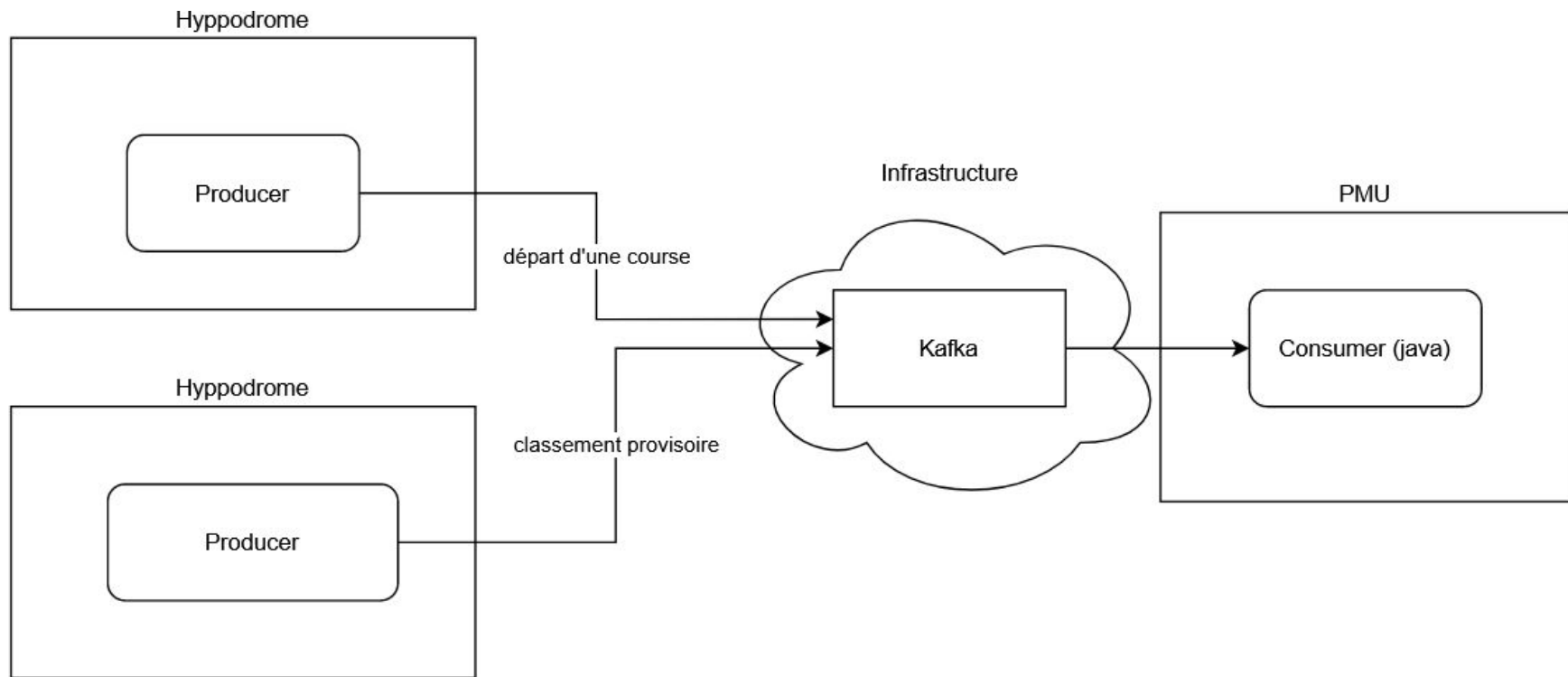
Consumers

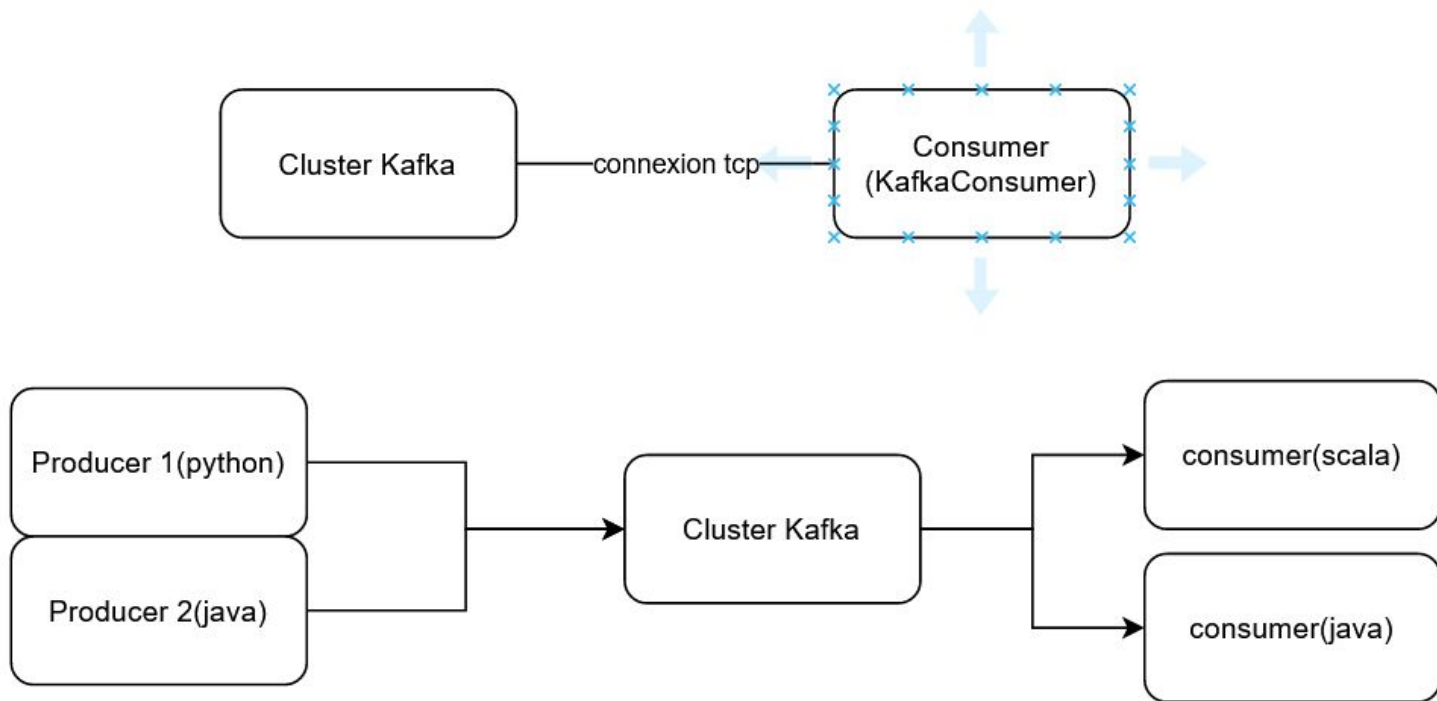


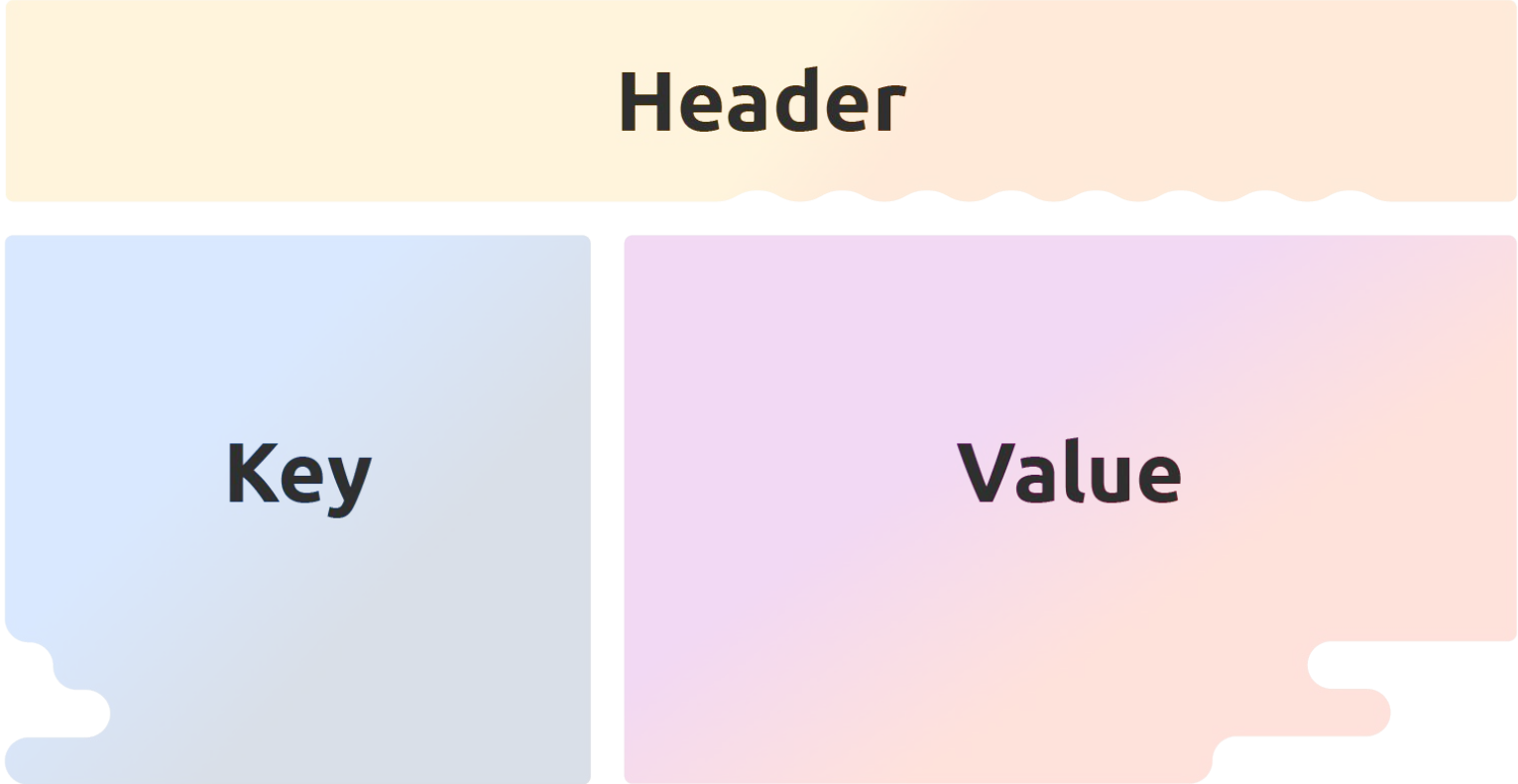
Producer : système / application qui publie un événement (message) vers Kafka

Consumer: système / application qui lie et process envoyé par kafka







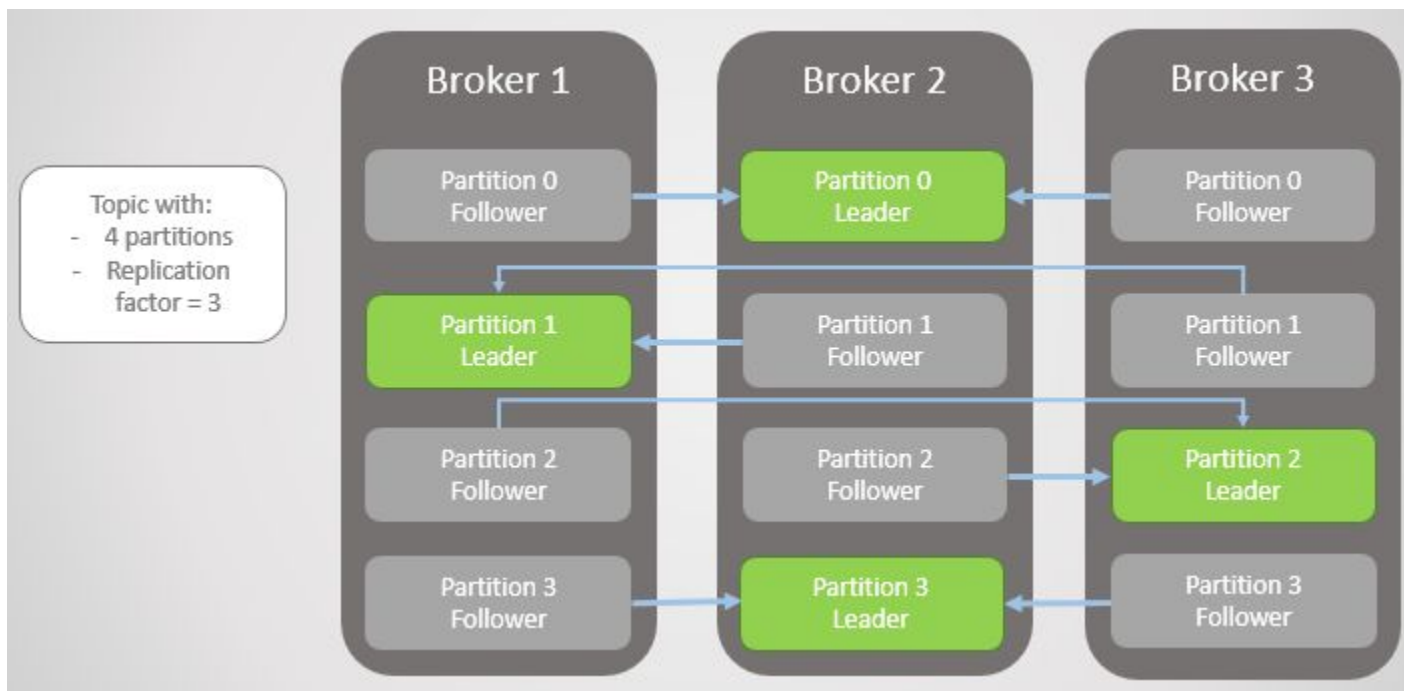


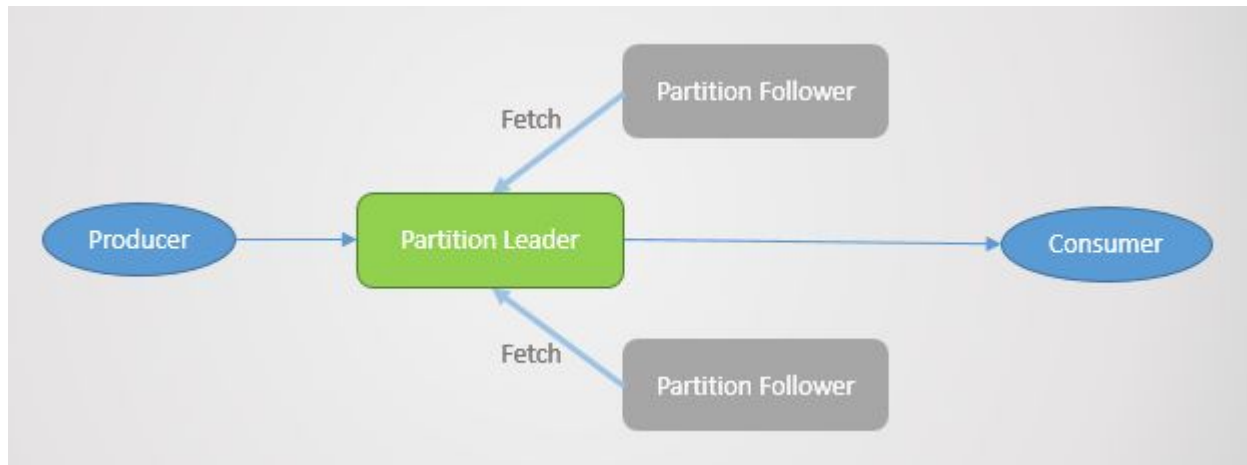
The diagram illustrates a data structure layout. At the top is a light orange rectangular box labeled 'Header'. Below it are two side-by-side boxes: a light blue box on the left labeled 'Key' and a light pink box on the right labeled 'Value'. The 'Header' box has a wavy bottom edge. The 'Key' box has a notch on its left side and a protrusion on its bottom-left corner. The 'Value' box has a protrusion on its right side and a notch on its bottom-right corner.

Header

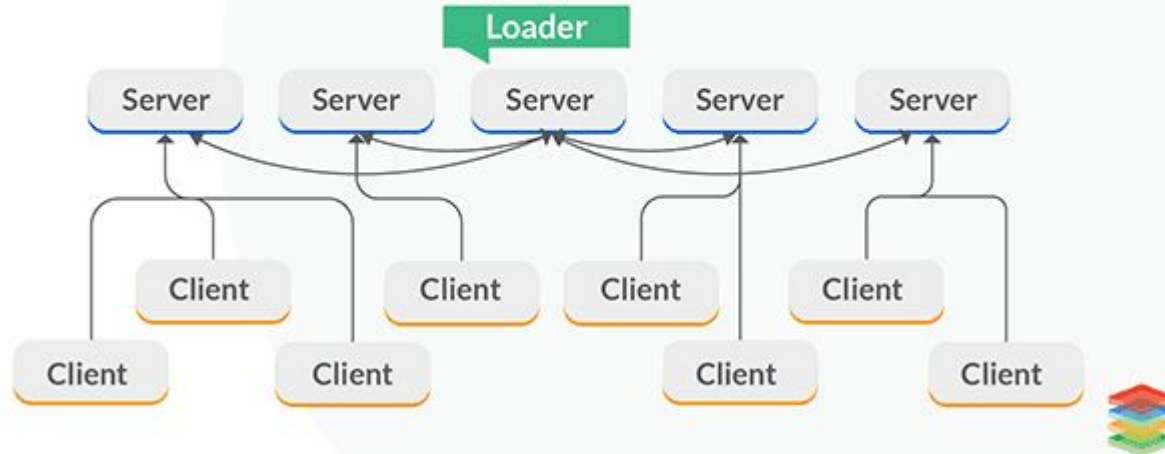
Key

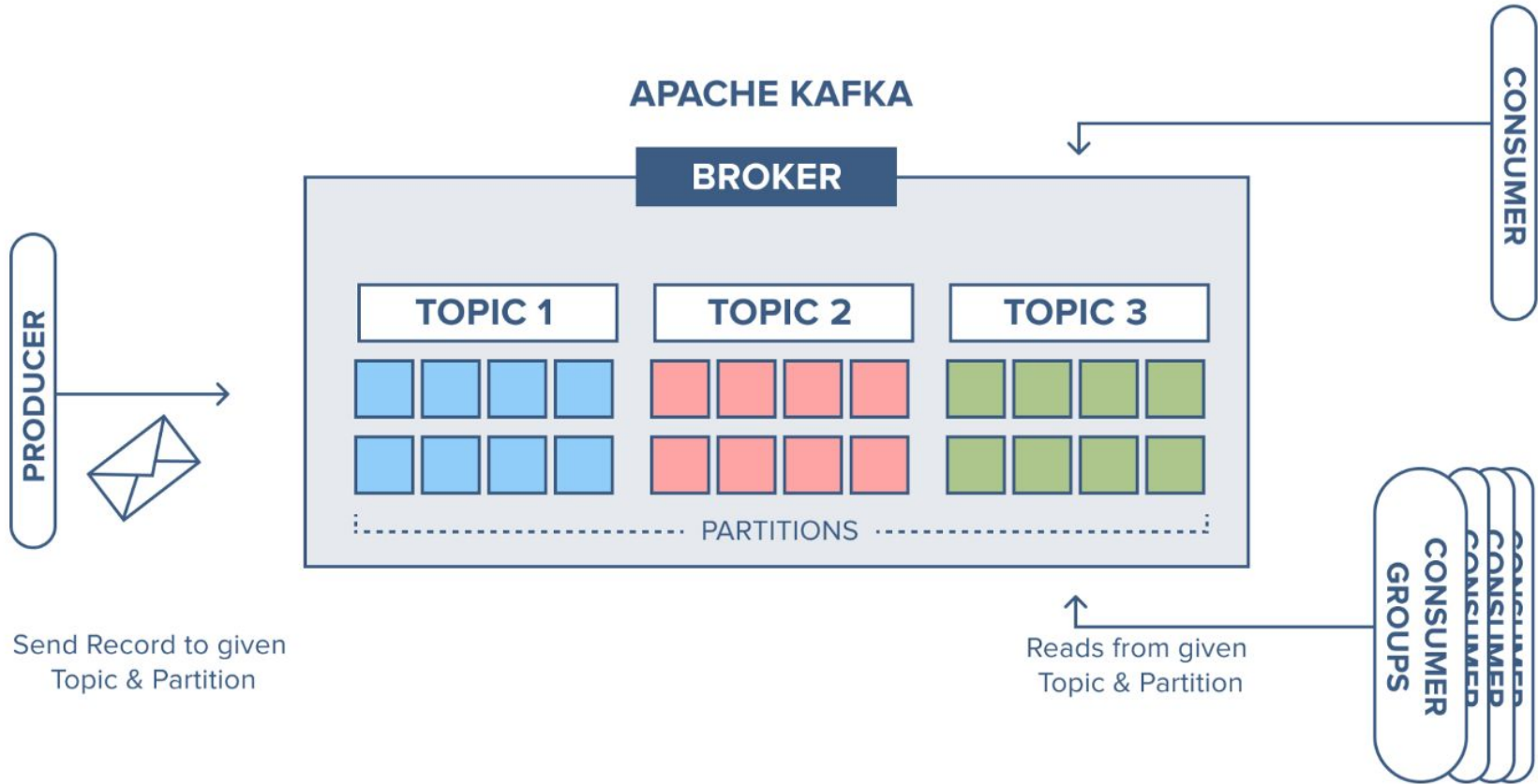
Value

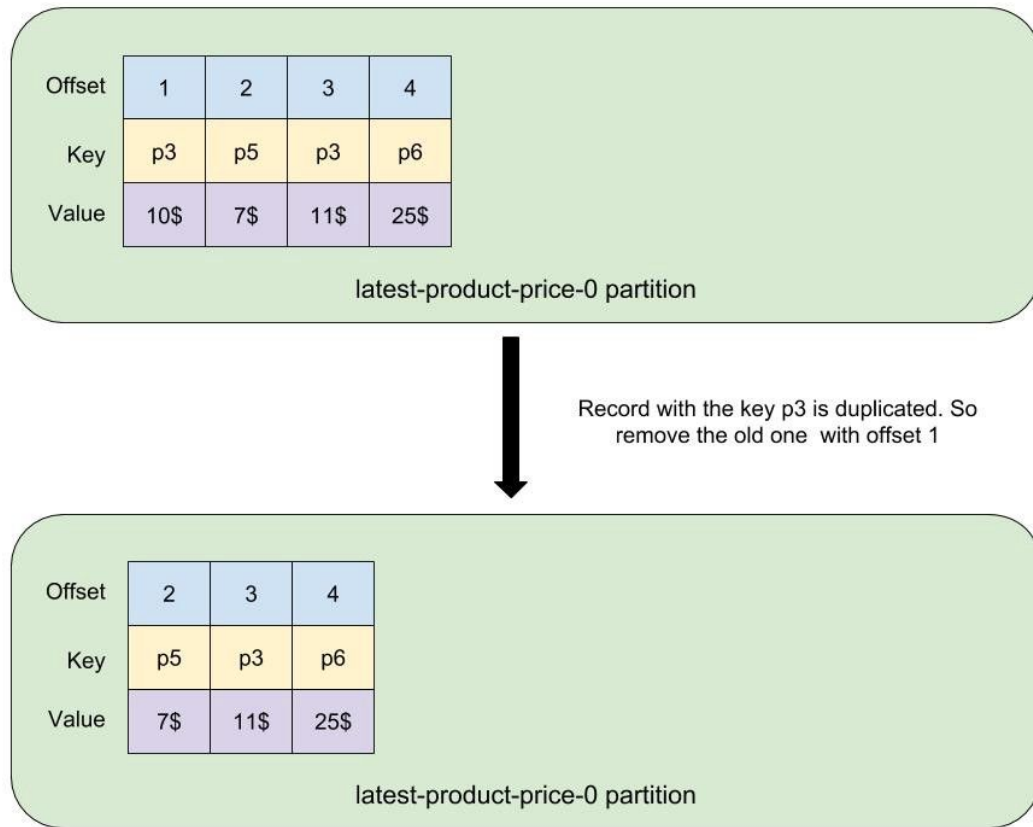




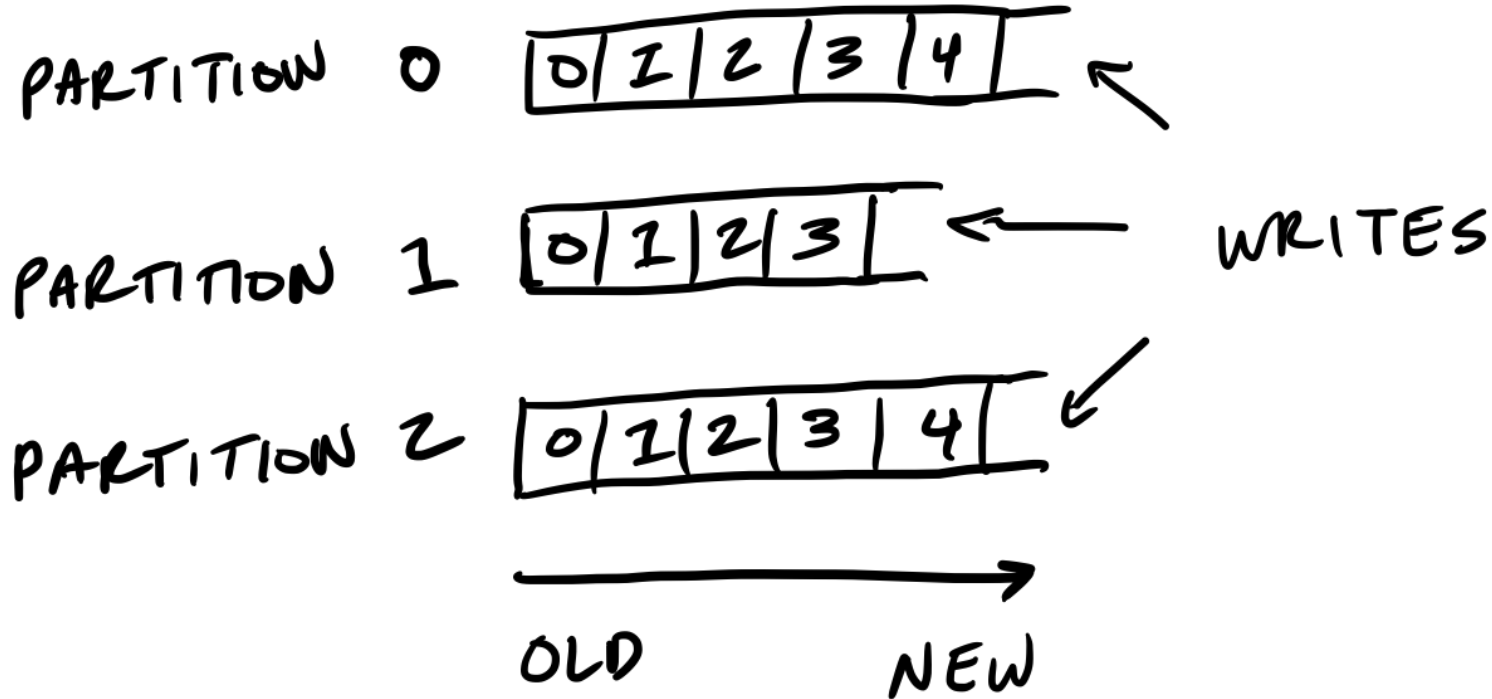
Zookeeper Architecture



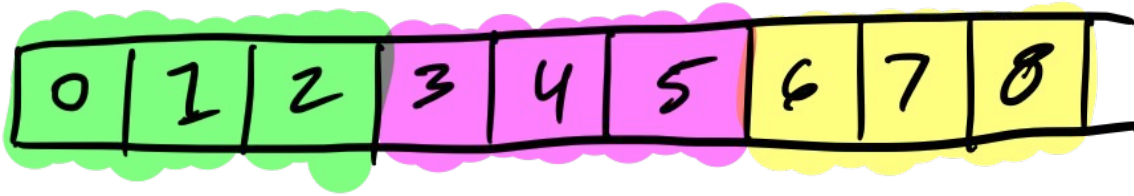




TOPIC



PARTITION



 SEGMENT 0

 SEGMENT 3

 SEGMENT 6

WRITE COMES IN NOW
ACTIVE SEGMENT (6) IS FULL
CREATE NEW SEGMENT (9)
SET AS THE ACTIVE SEGMENT

Kafka Log Compaction Process

Before Compaction

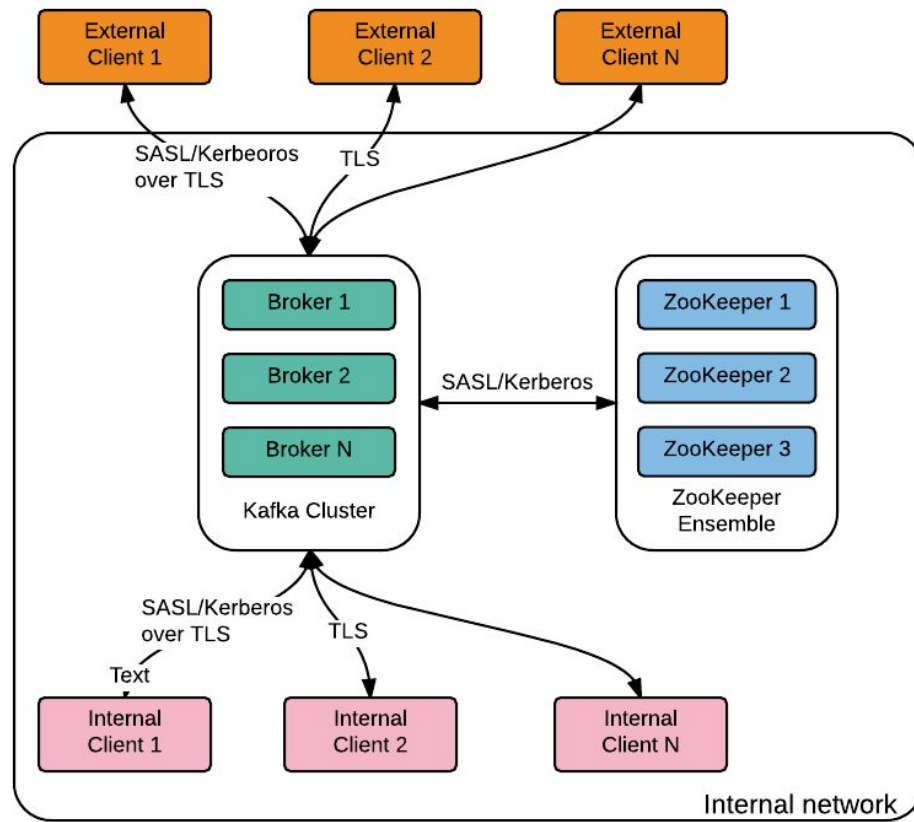
| | | | | | | | | | | | | |
|--------|----|----|----|----|----|----|----|----|----|----|-----|-----|
| Offset | 13 | 17 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| Keys | K1 | K5 | K2 | K7 | K8 | K4 | K1 | K1 | K1 | K9 | K8 | K2 |
| Values | V5 | V2 | V7 | V1 | V4 | V6 | V1 | V2 | V9 | V6 | V22 | V25 |

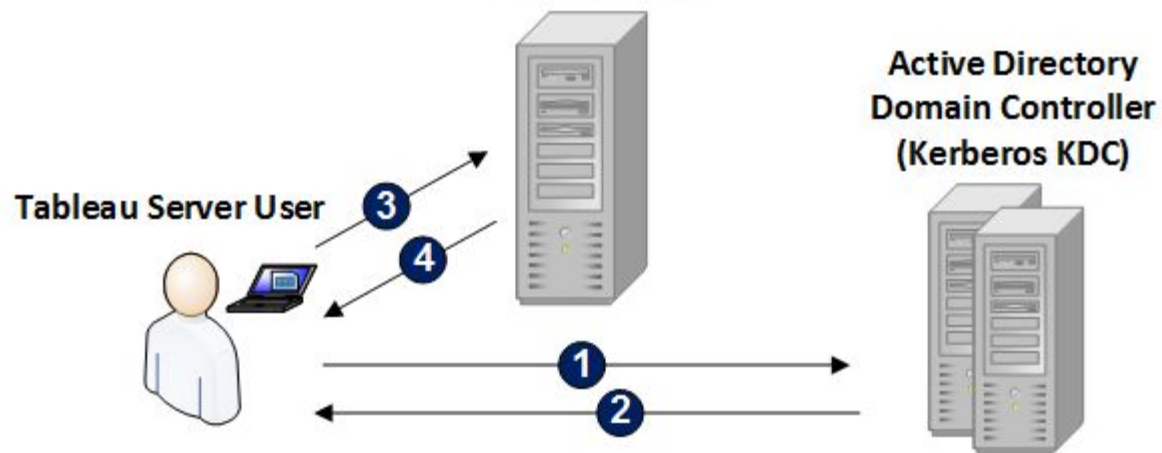
Cleaning

Only keeps latest version of key. Older duplicates not needed.

| | | | | | | | |
|--------|----|----|----|----|----|-----|-----|
| Offset | 17 | 20 | 22 | 25 | 26 | 27 | 28 |
| Keys | K5 | K7 | K4 | K1 | K9 | K8 | K2 |
| Values | V2 | V1 | V6 | V9 | V6 | V22 | V25 |

After Compaction



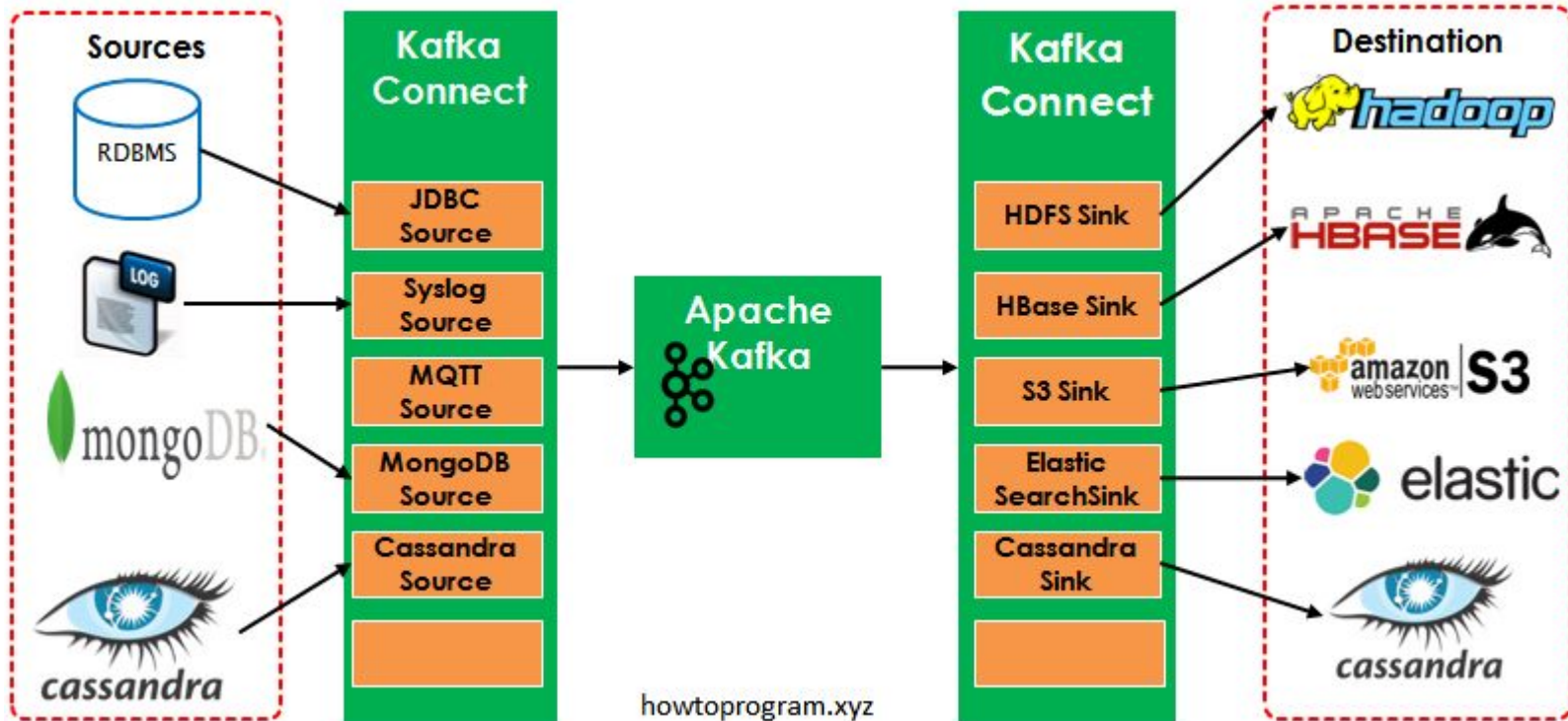


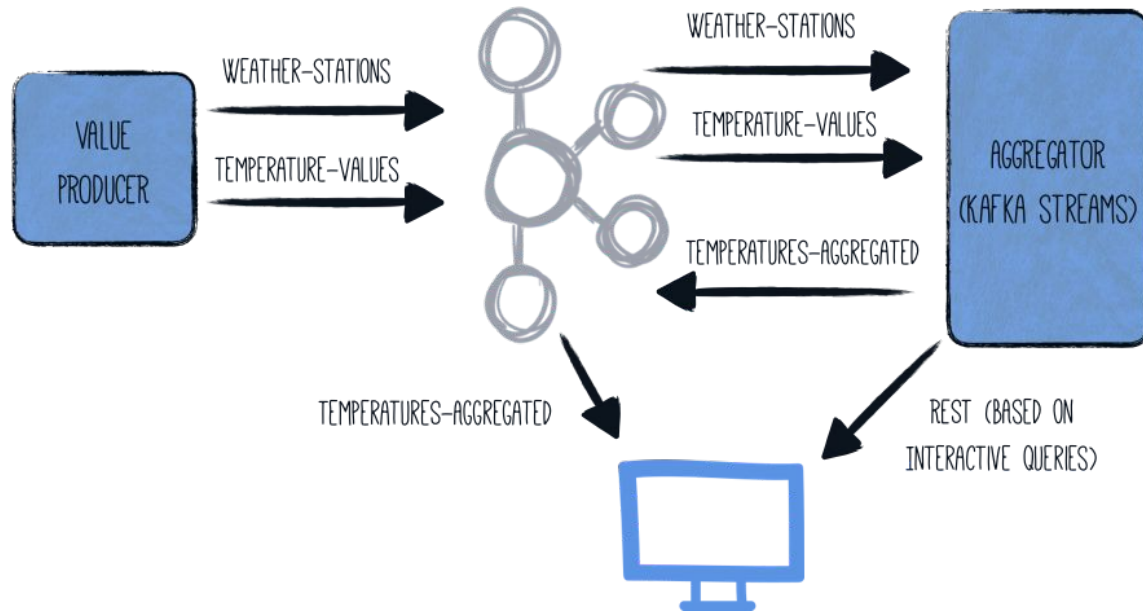
1. AKHQ
2. Kowl
3. Kafdrop
4. UI for Apache Kafka
5. Lenses
6. CMAK
7. Confluent CC
8. Conduktor

The screenshot displays the AKHQ (Apache Kafka HQ) web interface. The left sidebar contains navigation links for Clusters (local), Nodes, Topics, Live Tail, Consumer Groups, ACLS, Schema Registry, and Connects. The main panel is titled 'Topic: stream-map' and shows the 'Data' tab. It features a table of message data with columns for Key, Date, Partition, Offset, Headers, and Schema. The table shows five rows of data, each with a 'Value: 1' button. Below the table, there are buttons for 'Live Tail' and 'Produce to topic'.

| Key | Date | Partition | Offset | Headers | Schema |
|-----|----------------------|-----------|--------|---------|---|
| 1 | Mar 9, 2020, 9:15 PM | 0 | 0 | 0 | Value: 1 |
| | | | | | { "id": 1, "name": "WaWa", "breed": "ABYSSINIAN" } |
| 5 | Mar 9, 2020, 9:15 PM | 0 | 1 | 0 | Value: 1 |
| | | | | | { "id": 5, "name": "Forrest", "breed": "ORIENTAL" } |
| 7 | Mar 9, 2020, 9:15 PM | 0 | 2 | 0 | Value: 1 |
| | | | | | { "id": 7, "name": "Desmond", "breed": "RAGDOLL" } |
| 8 | Mar 9, 2020, 9:15 PM | 0 | 3 | 0 | Value: 1 |

| | AKHQ | Kowl | Kafdrop | UI for Apache Kafka | Lenses | CMAK | Confluent CC | Conductor |
|--------------------------------------|---------|-------|---------|---------------------|------------------|------|--------------|-----------|
| Multi-Cluster Management | Yes | No | No | Yes | Yes | Yes | Yes | No |
| Message Browsing | Yes | Yes | Yes | Yes | Yes | No | Yes | Yes |
| Protobuf Support | Partial | Yes | Yes | Yes | No | No | Yes | Yes |
| Avro Support | Yes | Yes | Yes | Yes | Yes | No | Yes | Yes |
| Live Tailing | Yes | Yes | No | Roadmap | No | No | Yes | Yes |
| Dynamic Topic Configuration | No | No | No | Yes | No | Yes | Yes | Yes |
| Authentication | Yes | Paid | No | Yes | No | No | Yes | No |
| Authorization | Yes | Paid | No | Yes | No | No | Yes | No |
| Partition Increase | No | No | No | Yes | No | Yes | Yes | Yes |
| Replica Change | No | No | No | Yes | No | Yes | Yes | Yes |
| Amazon MSK IAM support | No | Yes | No | Yes | No | No | No | Yes |
| Kafka Connect management | Yes | No | No | Yes | Separate Service | No | Yes | Yes |
| UI Quality | Bad | Great | Average | Good | Great | Good | Good | Good |
| Schema Registry | Yes | No | No | Yes | No | No | Yes | Yes |
| KSQL Integration | No | No | No | Yes | No | No | Yes | Yes |
| Kafka Streams Topologies | No | No | No | Roadmap | No | No | Yes | No |
| Read-Only Mode | Yes | No | No | Yes | No | No | Yes | Yes |
| JMX Metrics Visualization and charts | No | No | No | Roadmap | No | No | Yes | Yes |





```
1 # specify the source_topic and destination_topic to the agent
2 @app.agent(source_topic, sink=[destination_topic])
3 async def hello(messages_stream):
4     async for records in messages_stream.take(5, within=10):
5         # do something
6         yield Test(msg='This message is from the AGENT')
```

ksqlDB

The database purpose-built for
stream processing applications.

Utiliser une syntaxe
proche du SQL pour
gérer ses application
de streamining

